



**Universidade de Brasília**

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

# Utilização de técnicas de aprendizagem de máquina nos pagamentos de cobertura do Proagro

Urias Cruz da Cunha

Dissertação apresentada como requisito parcial para conclusão do  
Mestrado Profissional em Computação Aplicada

Orientador  
Prof. Dr. Alexandre Zaghetto

Brasília  
2019

Ficha catalográfica elaborada automaticamente,  
com os dados fornecidos pelo(a) autor(a)

Cu Cruz da Cunha, Urias  
Utilização de técnicas de aprendizagem de máquina nos  
pagamentos de cobertura do Proagro / Urias Cruz da Cunha;  
orientador Alexandre Zaghetto. -- Brasília, 2019.  
141 p.

Monografia (Especialização - Mestrado Profissional em  
Computação Aplicada) -- Universidade de Brasília, 2019.

1. Seguro Agrícola. 2. Proagro. 3. Aprendizagem de  
Máquina. 4. Classificador. 5. Irregularidade. I. Zaghetto,  
Alexandre, orient. II. Título.



**Universidade de Brasília**

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

**Utilização de técnicas de aprendizagem de máquina  
nos pagamentos de cobertura do Proagro**

Urias Cruz da Cunha

Dissertação apresentada como requisito parcial para conclusão do  
Mestrado Profissional em Computação Aplicada

Prof. Dr. Alexandre Zaghetto (Orientador)  
CIC/UnB

Prof. Dr. Marcelo Ladeira    Prof.<sup>a</sup> Dr.<sup>a</sup> Priscila Tiemi Maeda Saito  
CIC/UnB    UTFPR

Prof.<sup>a</sup> Dr.<sup>a</sup> Aletéia Patrícia Favacho de Araújo  
Coordenadora do Programa de Pós-graduação em Computação Aplicada

Brasília, 25 de julho de 2019

# Dedicatória

Dedico este trabalho a Deus, o Eterno, pois somente por sua imensurável graça pude lograr êxito, e aos meus familiares, em especial minha esposa Aline, pelo apoio em todos os momentos.

# Agradecimentos

Ao professor Alexandre Zaghetto pela orientação e dedicação dispensada. Seus comentários sempre pertinentes e suas contribuições foram essenciais para a produção desta pesquisa.

Aos meus colegas do Banco Central que apoiaram de diversas formas esse empreendimento.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

# Resumo

O seguro agrícola é um instrumento efetivo de proteção de investimentos de produtores rurais contra as perdas decorrentes dos perigos naturais do campo. Como forma de oferecer um seguro para proteção do capital investido no empreendimento agrícola, o governo brasileiro criou o Programa de Garantia da Atividade Agropecuária (Proagro). Desde sua criação, coube ao Banco Central do Brasil a administração e supervisão do programa, enquanto às instituições financeiras coube a sua operação. Para tornar as ações da supervisão mais efetivas, foi estabelecida uma área de monitoramento cujo objetivo é gerar alertas para a supervisão trazendo possíveis casos de irregularidades. A fim de otimizar o uso dos dados adotados na geração de sinalizações, foi proposta a construção de um classificador, por meio do emprego de algoritmos e técnicas de aprendizagem de máquina, capaz de distinguir as comunicações de ocorrência de perda procedentes das improcedentes. Utilizando o modelo de referência CRISP-DM, foram executadas atividades das etapas que vão desde a preparação dos dados até a modelagem. Na modelagem, foram aplicados os algoritmos *Support Vector Machine*, *Naive Bayes*, Redes Neurais Artificiais e *Random Forest*. Ao final da modelagem, obteve-se, com o algoritmo *Random Forest*, um modelo vencedor com precisão média de 0,550 nos dados de teste. A partir da análise da curva precisão-sensibilidade, verifica-se que o modelo vencedor deve alcançar uma precisão de 80% para uma sensibilidade de 30%. Além do modelo classificatório, como resultado do trabalho teve-se a criação de *scripts* de captura automática de dados dos sistemas Agritempo e Sisdagro, o que deverá proporcionar ganhos de quantidade, tempestividade e qualidade nas análises das comunicações de ocorrência de perda.

**Palavras-chave:** BCB, seguro agrícola, Proagro, aprendizagem de máquina, classificador, *Random Forest*

# Abstract

Crop insurance is an effective mechanism which aims to protect farmers investments against loss resulting from natural hazards. As a way of providing insurance for protection for capital invested in agricultural ventures, the Brazilian government created the Program of Guarantee of the Agricultural Activity (PROAGRO). Since its creation, it has been the duty of the Central Bank of Brazil to administer and oversee the program whereas the operation has been delegated to the financial institutions authorized to work with the program. In order to make the oversight actions more effective, a monitoring division was established whose goal is to generate alerts to the oversight team with likely irregular insurance claims. To optimize the use of data applied to the generation of alerts, we proposed the implementation of a classifier using machine learning techniques. The classifier should be capable of distinguishing normal and irregular insurance claims. Following the CRISP-DM reference model, we carried out a series of activities since data preparation until modeling. In the modeling phase, we applied Support Vector Machine, Naive Bayes, Artificial Neural Networks, and Random Forest algorithms. At the end of the modeling phase, we obtained a model generated by Random Forest which achieved an average precision of 0.550 on the test dataset. With this model, it is possible to achieve a precision of 80% while keeping the sensitivity at 30%. In addition to this, this research produced scripts that will allow automatic data collection of Agritempo and Sisdagro systems, which will provide gains in volume, timeliness, and quality for the insurance claims analyses.

**Keywords:** BCB, crop insurance, Proagro, machine learning, classifier, Random Forest

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Problema de Pesquisa . . . . .	1
1.2	Justificativa . . . . .	3
1.3	Objetivos . . . . .	4
1.3.1	Objetivos específicos . . . . .	5
1.4	Estrutura do documento . . . . .	5
<b>2</b>	<b>Revisão bibliográfica</b>	<b>6</b>
2.1	Fundamentação teórica . . . . .	8
2.1.1	CRISP-DM . . . . .	10
2.1.2	Seleção e extração de características e redução de dimensionalidade . .	11
2.1.3	Balanceamento de classes . . . . .	16
2.1.4	Algoritmos de Classificação . . . . .	21
2.1.5	Métricas de Desempenho . . . . .	33
2.2	Trabalhos correlatos . . . . .	39
<b>3</b>	<b>Método proposto</b>	<b>46</b>
3.1	Entendimento do negócio . . . . .	48
3.2	Compreensão dos dados . . . . .	49
3.2.1	Sicor . . . . .	50
3.2.2	Agritempo . . . . .	54
3.2.3	Sisdagro . . . . .	58
3.3	Preparação dos dados . . . . .	60
3.3.1	Construção de novos atributos . . . . .	60
3.3.2	Imputação e agregação de valores . . . . .	62
3.3.3	Descarte de atributos e instâncias . . . . .	63
3.3.4	Padronização dos dados . . . . .	66
3.3.5	Transformação dos dados . . . . .	67
3.3.6	Redução de dimensionalidade e de multicolinearidade . . . . .	69



3.3.7	Balanceamento de classes . . . . .	69
3.4	Modelagem . . . . .	70
<b>4</b>	<b>Resultados</b>	<b>74</b>
4.1	Resultados Validação Cruzada . . . . .	74
4.1.1	Abordagem monolítica . . . . .	74
4.1.2	Abordagem hierárquica . . . . .	76
4.2	Resultados Dados de Teste . . . . .	79
4.2.1	Abordagem monolítica . . . . .	79
4.2.2	Abordagem hierárquica . . . . .	82
4.2.3	Comparação melhores modelos das abordagens monolítica e hierárquica	86
<b>5</b>	<b>Conclusão</b>	<b>90</b>
	<b>Referências</b>	<b>92</b>
	<b>Apêndice</b>	<b>100</b>
<b>A</b>	<b>Tabelas</b>	<b>101</b>
<b>B</b>	<b>Figuras</b>	<b>118</b>

# Lista de Figuras

1.1	Fluxograma funcional dos processos de contratação de crédito rural e seguro agrícola e de solicitação de pagamento de cobertura. . . . .	2
2.1	Um <i>framework</i> para seleção de características. . . . .	13
2.2	Gráfico de dispersão das variáveis $X_1$ e $X_2$ exibindo as coordenadas projetadas $p_{1i}$ e $p_{2i}$ sobre os eixos principais $P_1$ e $P_2$ . . . . .	14
2.3	Gráfico de dispersão com os pontos projetados nos novos eixos $P_1$ e $P_2$ . . .	15
2.4	Descrição simplificada das etapas de geração de uma instância sintética utilizando a técnica SMOTE . . . . .	18
2.5	Descrição simplificada das etapas de remoção de instâncias da classe majoritária utilizando a técnica <i>Tomek Links</i> . . . . .	20
2.6	Representação gráfica do hiperplano, da margem máxima e dos vetores de suporte definidos a partir do SVM com margem rígida. . . . .	23
2.7	Representação gráfica do hiperplano, da margem máxima e dos vetores de suporte definidos a partir do SVM com margem suave. . . . .	24
2.8	Representação gráfica de uma rede MLP com $L$ camadas e com $n_h$ neurônios em cada camada oculta $l$ . . . . .	27
2.9	Exemplo de uma árvore de decisão gerada com o algoritmo CART após a poda . . . . .	31
2.10	Exemplo de um gráfico Curva ROC com os desempenhos de três classificadores . . . . .	37
2.11	Exemplo de um gráfico curva precisão-sensibilidade com os desempenhos de dois classificadores . . . . .	38
3.1	Diagrama com as fases do modelo CRISP-DM executadas nesta pesquisa. .	46
3.2	Distribuição das COPs por tipo de evento . . . . .	53
3.3	Distribuição geográfica das estações físicas (pontos vermelhos) e virtuais (pontos verdes) . . . . .	55
3.4	Gleba antes e depois da redução do número de vértices com o algoritmo <i>Ramer</i> . . . . .	57

3.5	Excesso e déficit hídricos ao longo do ciclo da cultivar . . . . .	58
3.6	Estimativa de impacto na produtividade ao longo do ciclo da cultivar . . .	59
3.7	Mapa de Clima do Brasil em escala 1:5.000.000 representando as diferentes zonas climáticas do país. . . . .	61
3.8	Distribuição de classes das COPs por tipo de evento . . . . .	66
3.9	Alterações na distribuição de probabilidade da variável <i>cli_temp_med</i> de- pois de aplicadas as transformações de potência e quantílicas. . . . .	68
3.10	Fluxo do processo de classificação na abordagem monolítica . . . . .	71
3.11	Fluxo do processo de classificação na abordagem hierárquica . . . . .	71
4.1	Curvas ROC e precisão-sensibilidade para a métrica $F_1$ -score . . . . .	81
4.2	Curvas ROC e precisão-sensibilidade para a métrica acurácia . . . . .	81
4.3	Curvas ROC e precisão-sensibilidade para a métrica precisão média . . . .	82
4.4	Curvas ROC e precisão-sensibilidade para a métrica área sob ROCAUC . .	82
4.5	Curvas ROC e precisão-sensibilidade dos modelos do evento seca que obti- veram os melhores desempenho na métrica precisão média . . . . .	85
4.6	Curvas ROC e precisão-sensibilidade dos modelos do evento chuva excessiva que obtiveram os melhores desempenho na métrica precisão média . . . . .	85
4.7	Curvas ROC e precisão-sensibilidade dos modelos do evento geada que ob- tiveram os melhores desempenho na métrica precisão média . . . . .	86
4.8	Curvas ROC e precisão-sensibilidade dos resultados das abordagens mo- nolítica e hierárquica recalculados para a métrica precisão média. . . . .	88
4.9	Curva precisão-sensibilidade com ponto identificando a sensibilidade má- xima de 30% para uma precisão de 80%. . . . .	89
B.1	Curvas ROC e precisão-sensibilidade dos modelos do evento seca que obti- veram os melhores desempenho na métrica $F_1$ -score . . . . .	118
B.2	Curvas ROC e precisão-sensibilidade dos modelos do evento seca que obti- veram os melhores desempenho na métrica acurácia . . . . .	119
B.3	Curvas ROC e precisão-sensibilidade dos modelos do evento seca que obti- veram os melhores desempenho na métrica área sob ROCAUC . . . . .	119
B.4	Curvas ROC e precisão-sensibilidade dos modelos do evento chuva excessiva que obtiveram os melhores desempenho na métrica $F_1$ -score . . . . .	120
B.5	Curvas ROC e precisão-sensibilidade dos modelos do evento chuva excessiva que obtiveram os melhores desempenho na métrica acurácia . . . . .	120
B.6	Curvas ROC e precisão-sensibilidade dos modelos do evento chuva excessiva que obtiveram os melhores desempenho na métrica área sob ROCAUC . .	121

B.7	Curvas ROC e precisão-sensibilidade dos modelos do evento geadá que obtiveram os melhores desempenho na métrica $F_1$ -score . . . . .	121
B.8	Curvas ROC e precisão-sensibilidade dos modelos do evento geadá que obtiveram os melhores desempenho na métrica acurácia . . . . .	122
B.9	Curvas ROC e precisão-sensibilidade dos modelos do evento geadá que obtiveram os melhores desempenho na métrica área sob ROCAUC . . . . .	122
B.10	Curvas ROC e precisão-sensibilidade dos resultados das abordagens monolítica e hierárquica recalculados para a métrica $F_1$ -score. . . . .	123
B.11	Curvas ROC e precisão-sensibilidade dos resultados das abordagens monolítica e hierárquica recalculados para a métrica acurácia. . . . .	123
B.12	Curvas ROC e precisão-sensibilidade dos resultados das abordagens monolítica e hierárquica recalculados para a métrica ROCAUC. . . . .	124

# Lista de Tabelas

2.1	Exemplo de uma matriz de confusão com suas quatro categorias. . . . .	34
3.1	Variáveis do empreendimento . . . . .	51
3.2	Variáveis da COP . . . . .	51
3.3	Variáveis do relatório de comprovação de perdas . . . . .	52
3.4	Associação das cores às suas respectivas zonas climáticas . . . . .	62
3.5	Conjunto de variáveis finais . . . . .	65
3.6	Conjuntos de dados gerados no pré-processamento. . . . .	70
4.1	Melhores desempenhos por algoritmo e métrica obtidos com os modelos monolíticos. . . . .	75
4.2	Intervalos de confiança das médias calculadas na validação cruzada para a abordagem monolítica . . . . .	75
4.3	Melhores desempenhos por algoritmo e métrica obtidos com modelos treinados com COPs do evento seca. . . . .	76
4.4	Intervalos de confiança das médias das métricas obtidas com os modelos do evento seca . . . . .	76
4.5	Melhores desempenhos por algoritmo e métrica obtidos com modelos treinados com COPs do evento chuva excessiva. . . . .	77
4.6	Intervalos de confiança das médias das métricas obtidas com os modelos do evento chuva excessiva . . . . .	77
4.7	Melhores desempenhos por algoritmo e métrica obtidos com modelos treinados com COPs do evento geadas. . . . .	77
4.8	Intervalos de confiança das médias das métricas obtidas com os modelos do evento geadas . . . . .	78
4.9	Desempenho dos modelos monolíticos estimado a partir dos dados de teste.	80
4.10	Matriz de confusão para os modelos monolíticos com melhor resultado na métrica $F_1$ -score. . . . .	80
4.11	Desempenho dos modelos hierárquicos para o evento seca estimados a partir dos dados de teste. . . . .	83

4.12	Desempenho dos modelos hierárquicos para o evento chuva excessiva estimados a partir dos dados de teste. . . . .	83
4.13	Desempenho dos modelos hierárquicos para o evento geada estimados a partir dos dados de teste. . . . .	83
4.14	Matriz de confusão para os modelos hierárquicos com melhor resultado na métrica $F_1$ -score construídos com base no evento seca. . . . .	84
4.15	Matriz de confusão para os modelos hierárquicos com melhor resultado na métrica $F_1$ -score construídos com base no evento chuva excessiva. . . . .	84
4.16	Matriz de confusão para os modelos hierárquicos com melhor resultado na métrica $F_1$ -score construídos com base no evento geada. . . . .	84
4.17	Desempenho dos modelos das abordagens hierárquica e monolítica calculado sobre o mesmo conjunto dados de teste. . . . .	87
4.18	Matriz de confusão para os melhores modelos das abordagens hierárquica e monolítica com relação à métrica $F_1$ -score. . . . .	87
A.1	Hiperparâmetros e seus valores . . . . .	101
A.2	Arranjos dos hiperparâmetros dos modelos monolíticos e suas respectivas métricas calculadas na validação cruzada. . . . .	102
A.3	Arranjos dos hiperparâmetros dos modelos hierárquicos treinados com COPs do evento seca e suas respectivas métricas calculadas na validação cruzada. . . . .	103
A.4	Arranjos dos hiperparâmetros dos modelos hierárquicos treinados com COPs do evento chuva excessiva e suas respectivas métricas calculadas na validação cruzada. . . . .	105
A.5	Arranjos dos hiperparâmetros dos modelos hierárquicos treinados com COPs do evento geada e suas respectivas métricas calculadas na validação cruzada. . . . .	107
A.6	Arranjos dos hiperparâmetros dos modelos monolíticos e suas respectivas métricas calculadas sobre os dados de teste. . . . .	109
A.7	Arranjos dos hiperparâmetros dos modelos hierárquicos treinados com COPs do evento seca e suas respectivas métricas calculadas nos dados de teste. . . . .	111
A.8	Arranjos dos hiperparâmetros dos modelos hierárquicos treinados com COPs do evento chuva excessiva e suas respectivas métricas calculadas nos dados de teste. . . . .	113
A.9	Arranjos dos hiperparâmetros dos modelos hierárquicos treinados com COPs do evento geada e suas respectivas métricas calculadas nos dados de teste. . . . .	115

A.10 Métricas calculadas com os melhores modelos das abordagens monolítica e hierárquica sobre os dados de teste contendo os eventos seca, chuva excessiva e geada. . . . .	117
---	-----

# Lista de Abreviaturas e Siglas

**Agritempo** Sistema de Monitoramento Agrometeorológico.

**BCB** Banco Central do Brasil.

**CART** *Classification And Regression Tree.*

**CMN** Conselho Monetário Nacional.

**CNN** *Condensed Nearest Neighbor Rule.*

**COP** Comunicação de Ocorrência de Perda.

**CRISP-DM** *Cross-Industry Standard Process for Data Mining.*

**FA** *Factor Analysis.*

**FCIC** *Federal Crop Insurance Corporation.*

**FFNN** *Feedforward Neural Network.*

**FIPA** *Farm Income Protection Act.*

**KNN** *K-Nearest Neighbors.*

**LDA** *Linear Discriminant Analysis.*

**MCR** Manual de Crédito Rural.

**MLP** *Multilayer Perceptron.*

**NB** *Naive Bayes.*

**OLS** *Ordinary Least Squares.*

**PCA** *Principal Component Analysis.*



**PNN** *Potential Nearest Neighbor.*

**PPP** parceria público-privada.

**Proagro** Programa de Garantia da Atividade Agropecuária.

**RBF** *Radial Basis Function.*

**RF** *Random Forest.*

**RMA** *Risk Management Agency.*

**RNAs** Redes Neurais Artificias.

**Sicor** Sistema de Operações do Crédito Rural e do Proagro.

**Sisdagro** Sistema de Suporte à Decisão na Agropecuária.

**SMOTE** *Synthetic Minority Oversampling Technique.*

**SRA** *Standard Reinsurance Agreement.*

**SVM** *Support Vector Machine.*

# Capítulo 1

## Introdução

Este capítulo apresenta os principais motivos para a realização deste projeto de pesquisa. No primeiro momento, o problema proposto é explicitado juntamente com sua justificativa, na qual são apresentados os motivos. Na sequência, são apresentados os objetivos da pesquisa e a estrutura do documento.

### 1.1 Problema de Pesquisa

Empreendimentos agrícolas estão sujeitos a adversidades climáticas como inundação, seca e geada. Uma das formas de mitigar os riscos de prejuízos gerados por essas adversidades se dá por meio da obtenção de seguros agrícolas. O seguro agrícola fornece uma meio efetivo de proteger os investimentos dos produtores rurais contra as perdas decorrentes dos perigos naturais do campo. No entanto, as incertezas associadas aos riscos climáticos e questões intrínsecas à segurabilidade, como seleção adversa e risco moral, fazem com que os valores para contratação de seguros agrícolas integralmente privados tornem-se mais elevados, levando à necessidade da participação do Estado na oferta de seguros para o setor agrícola [1].

Em razão da necessidade de prover o setor agrícola nacional com uma proteção para seus empreendimentos de risco, o governo brasileiro criou o Programa de Garantia da Atividade Agropecuária (Proagro), por meio da Lei 5.969, de 11 de dezembro de 1973. Implantado em 1975, o Programa surgiu com o objetivo de desonerar o produtor rural do cumprimento de obrigações financeiras originadas da contratação de crédito rural, para os casos em que houvesse perda de receita decorrente de eventualidades que comprometessem a lavoura, tais como pragas, doenças, estiagens prolongadas e chuvas excessivas[2, 3].

Diversos foram os fatores motivadores da criação do Proagro, dentre os quais podemos destacar: (i) inexistência de mecanismo de proteção contra perdas dos investimentos aplicados na produção agropecuária que levavam à descapitalização e ao endividamento

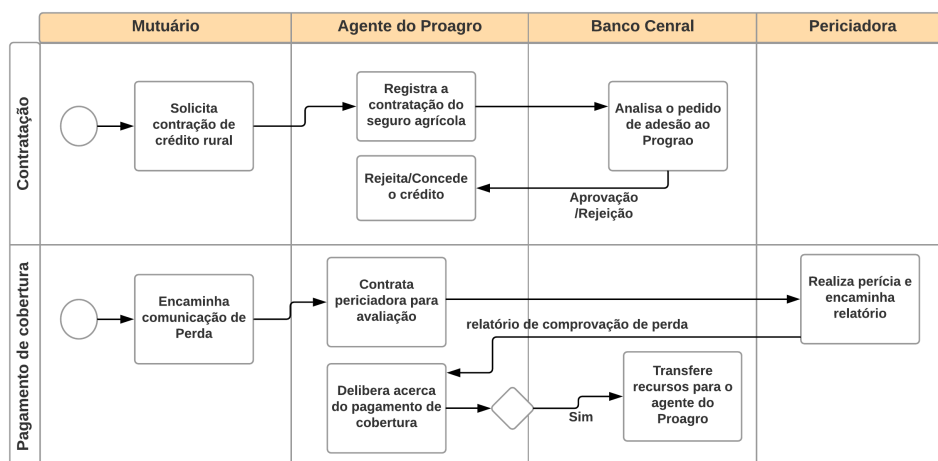


Figura 1.1: Fluxograma funcional dos processos de contratação de crédito rural e seguro agrícola e de solicitação de pagamento de cobertura.

do produtor rural; (ii) insucesso na implantação de seguro rural próprio para a proteção do agricultor que porventura sofresse prejuízos decorrentes de fenômenos naturais; e (iii) existência de modelos adotados por outros países em que o governo tanto pode conceder o crédito rural quanto pode assumir as despesas causadas pela perda da produção [2]. Ao ser criado, o Programa estabeleceu que as despesas advindas do pagamento das coberturas – indenizações – devem ser custeadas com recursos provenientes da União e da contribuição paga pelo produtor rural – o adicional –, bem como das receitas obtidas com a aplicação desses recursos<sup>1</sup>.

A Lei nº 8.171, de 17 de janeiro de 1991, ao dispor sobre a política agrícola, determinou que cabe ao Banco Central do Brasil (BCB) a administração do Proagro, devendo o BCB exercê-la em conformidade com as normas, os critérios e as condições definidas pelo Conselho Monetário Nacional (CMN) [4]. A operação do Proagro, por sua vez, coube às instituições financeiras autorizadas a atuar em crédito rural – denominadas aqui agentes financeiros –, de modo que recai sobre elas a competência para: contratar as operações de custeio; formalizar a adesão do mutuário ao Programa; efetuar a cobrança do adicional; realizar as análises dos processos; decidir sobre os pedidos de cobertura, ficando, nesse caso, o agente financeiro que enquadrou a operação no programa responsável pela comprovação dos prejuízos [3]; encaminhar os recursos à Comissão Especial de Recursos – CER; pagar e registrar as despesas<sup>1</sup>. O fluxograma funcional apresentado na Figura 1.1 descreve as principais atividades executadas nos processos de contratação de crédito rural e seguro agrícola e de solicitação de pagamento de cobertura, bem como a atuação de seus atores nas atividades de cada processo.

<sup>1</sup>Ministério da Agricultura, Pecuária e Abastecimento. Proagro. Disponível em: <http://www.agricultura.gov.br/assuntos/riscos-seguro/risco-agropecuario/proagro>

Observa-se, dos papéis e das responsabilidades descritos acima, que a atuação do BCB é limitada, visto que, embora administre os recursos do Programa, é dos agentes financeiros a responsabilidade de verificar, por meio de laudos periciais, a veracidade e a lisura das comunicações de ocorrência de perda (COPs) e das solicitações de cobertura emitidas pelos mutuários – produtores rurais que aderiram ao Programa. Ficam, então, evidenciados os riscos aos quais a administração do BCB está exposta, pois os controles capazes de prevenir possíveis falhas e fraudes no pagamento de coberturas são exercidos em quase sua totalidade pelos agentes financeiros, cumprindo, portanto, ao BCB a missão de avaliar, por meio de supervisão, a eficácia e efetividade desses controles.

Ao passo que o processo de supervisão das ações dos agentes financeiros tem sua eficácia limitada, notadamente em virtude das adversidades impostas pelos aspectos logísticos e regulamentares do Programa, a área de monitoramento das operações do Proagro do BCB busca na implantação de ações de monitoramento um meio de garantir maior eficiência e eficácia ao processo de identificação e prevenção de falhas e fraudes nos pagamentos de cobertura.

Nesse contexto, consideramos que as técnicas de aprendizagem de máquina podem alavancar os resultados do processo de monitoramento das operações do Proagro. Como forma de alavancar os resultados, propõe-se a análise dos pagamentos de cobertura do Proagro de forma contínua e com suporte de ferramentas tecnológicas. A utilização dos dados e das informações gerados a partir da operacionalização do Programa – especialmente aqueles que tratam de solicitações de pagamento de cobertura e das respectivas decisões quanto ao deferimento – em conjunto com técnicas avançadas de aprendizagem de máquina, tal como Redes Neurais Artificiais (RNAs)[5], pode levar à criação de modelos classificatórios capazes de identificar fraudes ou falhas nos pagamentos de coberturas.

O uso de técnicas de aprendizagem de máquina requer do analista de dados conhecimento dos aspectos do negócio e das técnicas de mineração de dados envolvidas pois o profissional deve ser capaz de selecionar a tecnologia e a técnica mais adequada para solução do problema em questão, e de empregá-las de forma a obter resultados satisfatórios. Este projeto objetiva o estudo e a aplicação de técnicas de aprendizagem de máquina para prevenção de fraudes e identificação e correção de falhas e deficiências no processo de pagamento de coberturas no âmbito do Proagro.

## 1.2 Justificativa

Como parte do processo de melhoria e alinhamento às melhores práticas, foi criada uma divisão de monitoramento das operações do Proagro. As ações de monitoramento têm

como objetivo gerar sinalizações de eventos que indiquem possíveis irregularidades evoluindo, entre outras coisas, a comprovação de perdas.

Diante desse quadro, a análise automatizada das comunicações de perda do Proagro surge como ferramenta cujo objetivo é apoiar as ações de monitoramento. A operacionalização do Programa produz grande volume de dados estruturados que, se adequadamente utilizados, poderão permitir a identificação e notificação tempestivas das fraquezas e das lacunas dos controles do Programa. Esses dados, que vão desde o tipo de empreendimento (e.g.: cesta: irrigado; produto: milho; zoneado, etc.) até detalhes do evento que suscitou a comunicação de perda (e.g.: motivo: chuva excessiva; data do ocorrido; etc.), são obtidos e armazenados por meio do Sistema de Operações do Crédito Rural e do Proagro (Sicor) [6, 7].

Nos anos agrícolas de 2013-2014 a 2015-2016<sup>2</sup>, cerca de 1,3 milhão de empreendimentos foram enquadrados nas modalidades de seguro Proagro Tradicional e Proagro Mais<sup>3</sup>, o que demonstra o expressivo volume de transações geradas na operacionalização do Proagro. Esses enquadramentos implicaram, aproximadamente, 133 mil comunicações de perdas, que resultaram em torno de 105 mil coberturas deferidas e o consequente pagamento da ordem de R\$ 2 bilhões [6].

Nessa conjuntura, vislumbra-se, com a aplicação de técnicas de aprendizagem de máquina, um meio de otimizar o uso dos dados disponíveis. A partir da classificação das comunicações de perda em dois grupos (procedentes e improcedentes), espera-se poder direcionar os esforços de análise dos analistas para os casos classificados com improcedentes com alta probabilidade, aumentando, assim, a eficiência das ações de monitoramento e, ao mesmo tempo, tornando o processo de identificação e de prevenção de fraudes e irregularidades mais tempestivo e preciso.

## 1.3 Objetivos

O objetivo deste projeto é construir um modelo classificatório capaz de identificar comunicações de perdas indevidas emitidas pelos produtores rurais que possam gerar despesas com o pagamento de coberturas.

Para a consecução desse objetivo, foram analisadas as bases de dados do Sicor, do Sistema de Suporte à Decisão na Agropecuária (Sisdagro), do Sistema de Monitoramento Agrometeorológico (Agritempo) e, eventualmente, de outras bases disponíveis que pudessem auxiliar na classificação das comunicações de perda. Foram, também, estuda-

---

<sup>2</sup>De acordo com o Relatório Circunstanciado 2015-2016, a apuração dos dados para o ano agrícola 2015-2016 estava em andamento para efeito de cobertura de perda.

<sup>3</sup>Seguro público destinado a atender aos pequenos produtores vinculados ao Programa Nacional de Fortalecimento da Agricultura Familiar (PRONAF).

das, selecionadas e aplicadas técnicas relacionadas a seleção e extração de características, transformação de características, balanceamento de classes, combinação de classificadores, etc., que, em conjunto ou isoladamente, pudessem aumentar o desempenho do modelo classificatório final.

Espera-se empregar o resultado deste trabalho no processo de monitoramento de pagamento de COPs do BCB, a fim de permitir melhor direcionamento dos esforços aplicados na investigação de fraudes e na detecção de falhas de controle envolvendo o Proagro.

### 1.3.1 Objetivos específicos

São estes os objetivos específicos desta pesquisa:

- Criar *scripts* para coleta automática dos dados do Sicor, do Sisdagro e do Agritempo;
- Construir um modelo classificatório que identifique COPs com fraude ou erro.

## 1.4 Estrutura do documento

O presente trabalho está estruturado da seguinte forma: o Capítulo 1 consiste na presente Introdução, a qual traz consigo a justificativa, os objetivos e a metodologia aplicada; o Capítulo 2 contém a revisão da bibliografia, em que estão descritos o modelo *Cross-Industry Standard Process for Data Mining* (CRISP-DM), as principais técnicas associadas ao aprendizado de máquina, como balanceamento dos dados, seleção e extração de características, algoritmos de classificação, etc., e, por último, os trabalhos correlatos.

No Capítulo 3 está descrito o método proposto, cujo conteúdo apresenta a sequência de passos necessários para consecução dos objetivos. O Capítulo 4 apresenta os detalhes da execução dos passos definidos no Capítulo 3 e os resultados obtidos ao final de cada passo. Por fim, o Capítulo 5 traz as conclusões obtidas e as possibilidades de trabalhos futuros identificadas.

# Capítulo 2

## Revisão bibliográfica

Qualquer que seja o setor econômico de uma país, sempre haverá riscos inerentes às suas atividades. O setor de agricultura, no entanto, sofre não só com o risco de mercado, mas também com riscos relacionados a eventos naturais, que são, muitas vezes, imprevisíveis e sem método efetivo de mitigação de seus impactos. Eventos, tais como pestes, doenças e variações climáticas adversas, podem impactar negativamente o setor, afetando os custos de produção e desacelerando seu crescimento. Esses eventos causam prejuízos financeiros não só aos produtores rurais, mas também aos seus credores em decorrência do inadimplimento, resultando na diminuição da disponibilidade de recursos financeiros destinados a empreendimentos agrícolas. A fim de mitigar tais riscos, produtores rurais recorrem a instrumentos de proteção contra perdas financeiras, como o seguro agrícola [8, 9]

Em muitas partes do mundo, o seguro agrícola tem sido um dos programas de gerenciamento de risco e manutenção da estabilidade para produtores rurais mais bem-sucedidos [8]. Países como o Canadá e os Estados Unidos têm adotado o seguro agrícola como forma de fomento à produção agrícola e de proteção de renda dos produtores rurais [1, 10]. Entretanto, dado os desafios inerentes à “segurabilidade” na agricultura, como riscos climáticos sistêmicos, e à segurabilidade de modo geral, como seleção adversa e risco moral, o gerenciamento dos riscos de perda na agricultura mostra-se difícil tanto para o setor privado quanto para público.

Para lidar com tais desafios e garantir um programa de seguro agrícola sustentável, muitos países recorrem a parcerias público-privadas (PPP), especialmente aqueles que possuem mercados emergentes, onde não há recursos financeiros adequados para lidar com casos extremos de perda originados a partir de graves desastres naturais [1]. A abordagem PPP pode variar significativamente entre países, resultando em diferentes formas de compartilhamento de risco. Em um extremo dessa abordagem, o seguro agrícola é ofertado unicamente por companhias do setor privado, de modo que o prêmio, cujo valor é calculado com base na perda esperada, e os demais custos de emissão da apólice não

são subsidiados. No outro extremo, o governo realiza a maior parte das funções de um programa de seguro, como definição de taxas, pagamento de subsídio para o prêmio e para os gastos com administração e carregamento, venda de apólices, fornecimento de resseguro etc.

Para fins de ilustração dos diferentes programas de seguro agrícola existentes no mundo, descrevemos abaixo as principais características dos modelos adotados nos Estados Unidos e no Canadá.

Em 1938, os Estados Unidos estabeleceram o Programa de Seguro Agrícola Federal Americano - *The U.S. Federal Crop Insurance Program* (FCIP)[11]. Desde então, diversas mudanças foram realizadas, como o estabelecimento de parceiras público-privadas e a criação da Agência de Gestão de Risco (RMA, do inglês *Risk Management Agency*). É por meio dos acordos de resseguro padrão (SRA, do inglês *Standard Reinsurance Agreement*) que se define o relacionamento entre a *Federal Crop Insurance Corporation* (FCIC) e as companhias de seguro autorizadas a vender apólices de seguro agrícola com prêmios subsidiados. É também por meio desses acordos que se estabelecem as condições nas quais a FCIC fornece subsídios e estabelece acordos de compartilhamento de riscos com as companhias de seguros.

Cerca de 62% do prêmio total é subsidiado pelo governo, o qual também assume integralmente os custos de venda das apólices. Diferentemente de outros tipos de seguro, as seguradoras aprovadas que vendem o seguro agrícola não podem recusar cobertura a um produtor elegível. Por conta disso e como forma de encorajamento do envolvimento do setor privado, os seguros são ressegurados pelo Departamento de Agricultura dos Estados Unidos. Além do mais, no compartilhamento dos riscos com o governo federal, as seguradoras têm a liberdade de ressegurar suas apólices no mercado privado [1].

Já no Canadá, o gerenciamento de riscos e seguros do setor agrícola é realizado por meio de quatro programas: *AgriInsurance*, *AgriStability*, *AgriInvest* e *AgriRecovery* [12]. O *AgriInsurance*, programa que trata do pagamento de indenizações decorrentes de perdas na produção, foi criado em 1959 por meio do *Crop Insurance Act* e legislado sob o *Farm Income Protection Act* (FIPA).

A operacionalização dos planos de seguro do *AgriInsurance* fica sob a responsabilidade das províncias, as quais a exercem por meio de corporações governamentais de seguro[13]. Já o governo federal atua na supervisão, no fornecimento de subsídio para os custos de administração e prêmio e no financiamento do déficit. Os governos das províncias e o governo federal juntos assumem 60% dos custos do prêmio, enquanto os 40% restantes são pagos pelos próprios produtores. Por fim, 100% dos custos de emissão do seguro são subsidiados pelos governos das províncias e pelo governo federal na razão de 40/60, respectivamente.



De acordo com Rejesus *et al.*[14], seguros agrícolas são mais propensos à fraude quando estão vinculados a um programa em que há subsídios ofertados com recursos públicos, semelhantemente ao que ocorre no Brasil, Canadá e Estados Unidos. Os pagamentos de indenização indevidos fazem aumentar o custo dos programas de seguro agrícola, diminuindo, ao mesmo tempo, sua efetividade. Isso ocorre porque, ao se aumentar o valor pago em indenizações, cresce também o valor do prêmio a ser pago na contratação do seguro para que seja mantido o equilíbrio financeiro. Uma vez que os prêmios são subsidiados e que existe um limite orçamentário para gastos com pagamento de subsídio, a elevação do valor do prêmio fará com que um menor número de produtores tenha acesso ao seguro, eventualmente impedindo que empreendimentos agrícolas legítimos sejam atendidos.

Como ferramenta de auxílio ao processo de detecção de fraudes, técnicas de mineração de dados ou, mais especificamente, técnicas de aprendizagem de máquina, têm sido aplicadas em diversos ramos de negócio, inclusive em seguros agrícolas [15]. Seu emprego tem ocorrido em áreas como *marketing*, controle de fraudes em cartões de crédito, controle de fraude em seguros, análise de crédito e controle de fraudes em serviços de telefonia [14, 16]. Companhias de seguro de automóveis e de saúde têm utilizado substancialmente técnicas de mineração de dados, como análise de *outliers*, na detecção de solicitações de indenização atípicas ou fraudulentas. A utilização de mineração de dados na detecção de solicitações atípicas tem ocorrido de maneira rotineira nessas companhias. Em geral, busca-se identificar por meio da mineração de dados comportamentos que sugerem a presença de fraude nesse tipo de operação.

Na próxima seção, serão apresentados os principais conceitos envolvendo aprendizagem de máquina e técnicas de mineração de dados. O conteúdo aborda, ainda, o modelo de referência CRISP-DM, algumas técnicas comuns aplicadas no pré-processamento dos dados e alguns algoritmos de classificação dentre os mais conhecidos.

## 2.1 Fundamentação teórica

A aprendizagem constitui-se de um fenômeno multifacetado [17]. O processo de aprendizagem inclui a obtenção de um novo conhecimento declarativo, o desenvolvimento de habilidades motoras e cognitivas por meio de instrução ou prática, a organização de novos conhecimentos, representações efetivas e a descoberta de novos fatos e teorias por meio da observação e experimentação. Desde o início da era da computação, pesquisadores têm se esforçado para implementar tais capacidades nos computadores. Solucionar esse problema tem sido um dos mais desafiadores e fascinantes objetivos da inteligência artificial. O estudo e a modelagem computacional de processos de aprendizagem constituem o principal assunto da aprendizagem de máquina.

Atualmente, sistemas de computadores não podem de fato aprender a executar uma tarefa por meio de exemplos ou por analogia a uma tarefa previamente executada [17]. Tampouco são capazes de evoluírem significativamente com base nos erros cometidos, ou mesmo adquirirem novas habilidades por meio da observação vicariante, isto é, aprendizagem de um comportamento por meio da observação do comportamento de um outro organismo. A pesquisa de aprendizagem de máquina busca fornecer a possibilidade de instruir computadores a aprenderem por meio de erros e de observação. A rápida expansão de aplicações e disponibilidade de computadores torna esta possibilidade ainda mais atraente e desejável.

Os sistemas de aprendizagem de máquina podem ser classificados em três diferentes dimensões [18]. A primeira dimensão corresponde às *estratégias de aprendizado subjacentes* utilizadas, que podem ser enumeradas como aprendizado por programação, aprendizado por memorização, aprendizado por instrução, aprendizado por analogia, aprendizado por exemplos e aprendizado por observação e descobertas. A segunda dimensão corresponde aos *tipos de conhecimento adquiridos*, tais como parâmetros em expressões algébricas, árvores de decisão, gramáticas formais, regras de produção, expressões lógicas, redes e grafos, esquemas, programas de computadores, taxinomia e múltiplas representações. A última dimensão consiste no *domínio da aplicação*, que pode envolver domínios relacionados a agricultura, educação, processamento de linguagens naturais, predição de sequência, etc.

De modo geral, as tarefas de aprendizagem podem ser agrupadas em quatro diferentes categorias [19]: classificação, agrupamento, associação e regressão. Na classificação, o aprendizado é realizado por meio da apresentação de um conjunto de exemplos já classificados, a partir dos quais espera-se aprender uma maneira de classificar novos exemplos. Na associação, busca-se identificar qualquer associação entre as características, e não somente aquelas que predizem uma classe em particular. No agrupamento, busca-se agrupar conjuntos de exemplos que possuem características semelhantes. Na regressão, o resultado a ser predito não é uma classe discreta, mas sim um valor numérico (contínuo).

No tocante aos problemas de aprendizagem, podemos classificá-los em quatro diferentes categorias: supervisionada, não supervisionada, semi-supervisionada e por reforço [5]. Problemas de aprendizagem supervisionada envolvem aplicações nas quais os dados de treinamento são formados pelos vetores de entrada juntamente com o vetor da variável alvo correspondente e, em geral, estão relacionados a problemas de classificação e regressão. Já na aprendizagem não supervisionada, a variável alvo não está presente nos dados de treinamento, sendo o agrupamento o objetivo mais comumente encontrado. Nos problemas de aprendizagem semi-supervisionada, os dados de treinamento contêm exemplos com e sem a variável de interesse. O processo de aprendizagem semi-supervisionada pode

ser visto com uma combinação de classificação e agrupamento, em que as observações acompanhadas da variável de interesse são utilizadas para classificar as observações que não possuem a variável, aumentando, assim, o número de observações classificadas disponíveis para o treinamento do classificador final [19]. Por fim, problemas de aprendizagem por reforço estão relacionados a descoberta das ações mais adequadas a serem tomadas em uma dada situação a fim de maximar a recompensa [5].

Nesta pesquisa, será adotado o aprendizado supervisionado envolvendo tarefas de classificação, sendo sua adoção possível em virtude das observações contidas no conjunto de dados de treinamento estarem identificadas com as respectivas classes. Nas próximas subseções serão apresentados os modelos de referência para mineração de dados, as técnicas de seleção e extração de características e os algoritmos de classificação associados à aprendizagem supervisionada que serão empregados nesta pesquisa.

### 2.1.1 CRISP-DM

A mineração de dados é um processo criativo que requer diferentes habilidades e conhecimentos [20]. O sucesso ou fracasso de um projeto de mineração de dados está fortemente relacionado com o indivíduo ou a equipe que o executa, o que compromete a repetibilidade das práticas adotadas em outros projetos. Para dirimir esses efeitos, o processo de mineração de dados requer a aplicação de uma abordagem padronizada que ajude a traduzir os problemas de negócio em tarefas de mineração, que dê sugestões acerca de transformações de dados e de técnicas de minerações apropriadas, e que forneça meios para avaliação da efetividade dos resultados e para documentação da experiência.

O projeto *CRoss-Industry Standard Process for Data Mining* – CRISP-DM – endereça parte dessas necessidades ao definir um modelo de processo que fornece um *framework* para execução de projetos de mineração, tendo como uma de suas características relevantes o fato de que sua utilização independe do setor da indústria envolvido ou da tecnologia adotada [20]. Espera-se com a adoção do CRISP-DM projetos de mineração de dados menos custosos, mais confiáveis, mais repetíveis, mais gerenciáveis e mais rápidos.

O CRISP-DM foi desenvolvido por um consórcio inicialmente composto por Daimler Chrysler, SPSS e NCR e, atualmente, é um dos modelos de referência mais conhecidos. Ele consiste de um ciclo composto por seis etapas [21, 22]:

- i. compreensão do negócio: foca no entendimento dos objetivos e requisitos a partir da perspectiva do negócio, transformando-os em um problema de mineração de dados e em um plano de para alcance dos objetivos;

- ii. compreensão dos dados: consiste na coleta inicial dos dados e na execução de atividades que levam a um melhor entendimento dos dados, à identificação de problemas de qualidade e à descoberta de *insights* com base nos dados;
- iii. preparação dos dados: consiste nas atividades de transformação dos dados brutos em dados próprios para modelagem, tais como limpeza, remoção de dados duplicados ou faltantes, atribuição de valores para os *missing values*, etc.;
- iv. modelagem: nessa etapa, diversas técnicas de modelagem são aplicadas, ao passo que seus parâmetros são ajustados para o valor ótimo;
- v. avaliação: consiste na avaliação do modelo (ou modelos) obtido na etapa anterior e na revisão das atividades que levaram ao modelo a fim de assegurar o alcance dos objetivos de negócio;
- vi. implantação: última etapa na qual o conhecimento adquirido é organizado e apresentado em um formato que possa ser utilizado pelo usuário.

Adotaremos neste trabalho o modelo de referência CRISP-DM. As atividades serão executadas e registradas seguindo as etapas do modelo de referência a fim de possibilitar a repetibilidade do processo e o registro adequado das atividades executadas.

### **2.1.2 Seleção e extração de características e redução de dimensionalidade**

Atualmente, aplicativos têm gerado grandes volumes de dados de diversos tipos, como vídeos, fotos, textos, etc. Muitas vezes, esses dados apresentam alta dimensionalidade, tornando a análise de dados um desafio. Para endereçar o problema, técnicas de seleção e extração de características (dimensões) têm sido desenvolvidas ao longo dos anos. Tais técnicas têm se mostrado efetivas no processamento de dados com alta dimensionalidade e no aumento da eficiência do aprendizado de máquina [23].

#### **Seleção de características**

A seleção de características é o processo de obtenção de um subconjunto de características extraídas do conjunto original sem que ocorra transformação dos dados durante o processo [24]. A seleção é feita com base em um determinado critério, o qual busca selecionar as características mais relevantes do conjunto de dados original para compor o novo conjunto de dados de treinamento. Ela desempenha um papel importante na redução do volume de dados de treinamento, visto que, ao final do processo, as variáveis (características)

redundantes e irrelevantes são removidas. Sua utilização ocorre na fase de preparação dos dados e tem como produto esperado um conjunto de dados contendo apenas as variáveis mais informativas e não redundantes, de modo a favorecer o aumento da acurácia da predição, a redução do tempo na aprendizagem e a simplificação dos resultados [23].

Técnicas de seleção de características têm sido utilizadas em diversos domínios, como reconhecimento de imagem, mineração de texto, detecção de intrusão, etc. De modo geral, elas podem ser categorizadas nos seguintes padrões:[23, 24]

- i. supervisionada ou não supervisionada: a depender dos dados de treinamento (classificados, não classificados ou parcialmente classificados);
- ii. modelo de filtro, *wrapper* ou *embedded*: definido de acordo com a interação com o método de aprendizagem;
- iii. critério de avaliação por correlação, distância euclidiana, consistência, dependência e medida de informação;
- iv. modelos *forward increase*, *backward deletion*, aleatório e híbrido: definidos a partir das estratégias de busca;
- v. modelo de ranqueamento (*weighting*) ou modelo de seleção de subconjunto: estabelecido de acordo com o tipo de saída.

A Figura 2.1 descreve o funcionamento de um *framework* de seleção de instâncias, apresentando suas principais etapas.

## Extração de características

A seleção de características e a extração de características são duas maneiras distintas empregadas para o alcance da redução de dimensionalidade. Enquanto a seleção de características apenas seleciona as características mais relevantes dentre as existentes sem realizar qualquer tipo de alteração ou combinação de variáveis, a extração de características pressupõe algum tipo de transformação que favoreça o reconhecimento de padrões [24].

A extração de características utiliza as características existentes para compor um espaço de características de menor dimensão. Para isso, mapeia informações úteis contidas no conjunto de dados original em um subconjunto de características, ignorando as informações redundantes e irrelevantes. De uma perspectiva matemática, isso corresponde a uma transformação de um vetor  $n$ -dimensional  $X$ , em que  $X = [x_1, x_2, \dots, x_n]^T$ , em um vetor  $m$ -dimensional  $Y$ , em que  $Y = [y_1, y_2, \dots, y_m]^T$  e  $m < n$ , por meio da aplicação de uma função de mapeamento  $f$  [25].

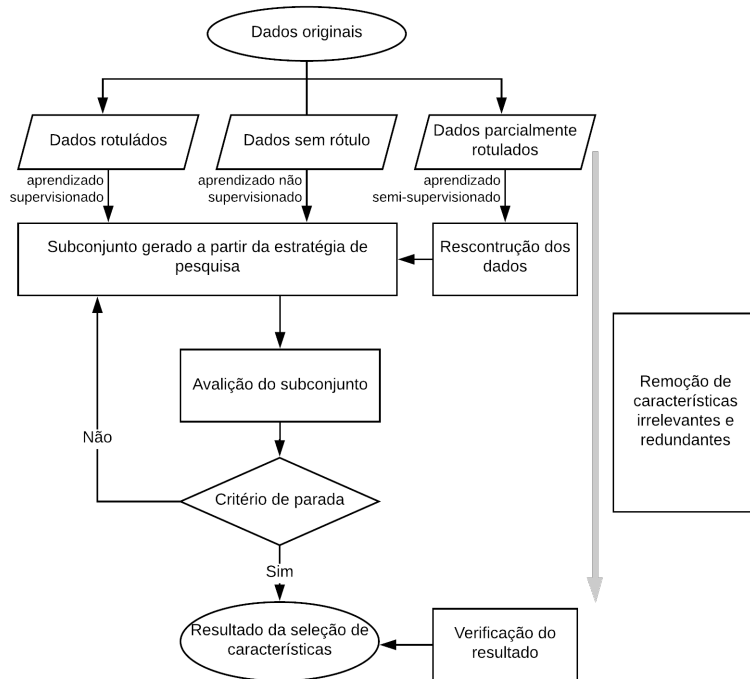


Figura 2.1: Um *framework* para seleção de características. Modificado a partir de Cai *et al.*[23].

A teoria de análise estatística é frequentemente utilizada na extração de características. Os métodos estatísticos são baseados em teorias contundentes e possuem diversos algoritmos eficientes já implementados. Entre os métodos mais comuns, podemos destacar a Análise de Componente Principal (PCA, do inglês *Principal Component Analysis*), a Análise Discriminante Linear (LDA, do inglês *Linear Discriminant Analysis*), a Análise de Fator (FA, do inglês *Factor Analysis*) e os Mínimos Quadrados Ordinários (OLS, do inglês *Ordinary Least Squares*) [25].

Na subseção a seguir, apresentaremos, brevemente, os principais aspectos do PCA, dado que o adotaremos neste estudo.

### *Análise de Componente Principal*

A PCA foi formulada primeiro por Pearson [26], o qual descreve a análise como encontrar “linhas e planos que melhor se ajustam a um sistema de pontos no espaço”. Dentre os objetivos mais comuns na aplicação da PCA, podemos destacar: simplificação, redução de dimensionalidade, detecção de *outlier* e seleção de variáveis [27].

A Análise de Componente Principal pode ser descrita como uma rotação dos eixos do sistema de coordenadas da variável original para novos eixos ortogonais chamados *eixos principais*, os quais devem apontar na direção de máxima variância [28]. Na prática,

a PCA consiste, primeiramente, da identificação dos *autovalores*  $\lambda_j$  e *autovetores*  $\vec{u}_j$  da matriz da covariância (ou da matriz correlação) extraída do conjunto de dados original. Os autovalores representam as variâncias dos dados projetados sobre os novos eixos. Já os componentes (coeficientes) dos autovetores são provenientes dos cossenos dos ângulos formados pelos eixos originais e principais. A Figura 2.2 esboça as interações entre os eixos originais e principais, os autovetores e as coordenadas originais e projetadas.

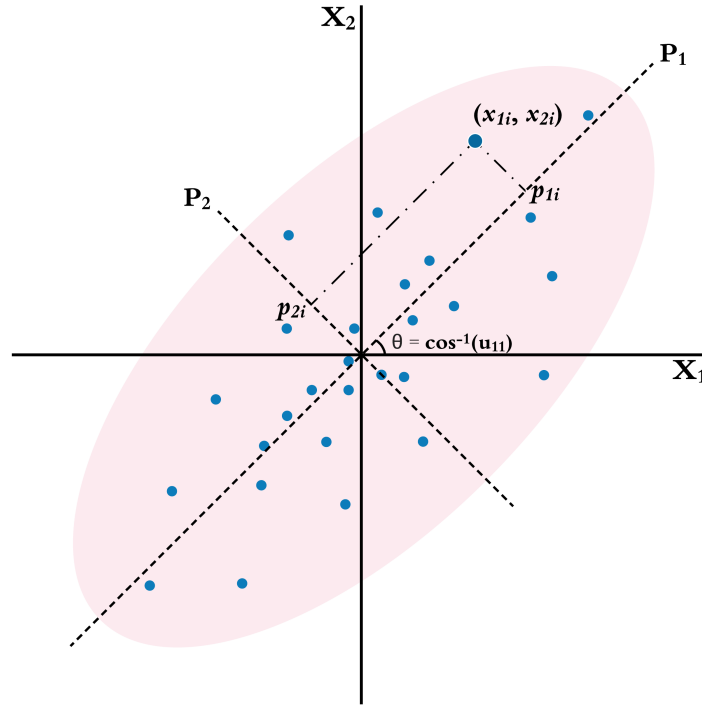


Figura 2.2: Gráfico de dispersão das variáveis  $X_1$  e  $X_2$  exibindo as coordenadas projetadas  $p_{1i}$  e  $p_{2i}$  sobre os eixos principais  $P_1$  e  $P_2$  (adaptada a partir de Campbell and Atchley [28]).

Na Figura 2.2, os pontos  $p_{1i}$  e  $p_{2i}$  representam os *scores* de componente principal da observação  $x_i = (x_{1i}, x_{2i})$ , *i.e.*, as novas coordenadas projetadas sobre os eixos principais [28]. O cosseno do ângulo formado pelo eixo original  $X_1$  e pelo primeiro eixo principal  $P_1$  (eixo de maior variância) dá origem ao primeiro coeficiente  $u_{11}$  do autovetor  $\vec{u}_1$  ao passo que o cosseno do ângulo gerado pelos eixos  $X_2$  e  $P_1$  origina o coeficiente  $u_{12}$ , resultando no autovetor  $\vec{u}_1 = (u_{11}, u_{12})$ . Da mesma forma, os cossenos dos ângulos formados pelo segundo eixo principal  $P_2$  (eixo com segunda maior variância) e os eixos originais  $X_1$  e  $X_2$  dão origem aos coeficientes  $u_{21}$  e  $u_{22}$ , resultando no autovetor  $\vec{u}_2 = (u_{21}, u_{22})$ .

Como dito anteriormente, os autovetores  $u_j$  podem ser calculados a partir das matrizes de covariância e correlação. Neste caso, o cálculo pode ser realizado meio de *decomposição espectral*, conforme descrito equação a seguir:

$$\mathbf{C} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \quad (2.1)$$

em que  $\mathbf{C}$  é a matriz de covariância,  $\mathbf{U}$  é a *matriz de autovetores* e  $\mathbf{\Lambda}$  é a *matriz diagonal* contendo os autovalores reais na diagonal principal. Visto que a matriz de covariância é simétrica, os autovetores  $u_j$  são ortogonais entre si.

Uma vez de posse da matriz de autovetores  $\mathbf{U}$ , em que  $\mathbf{U} = [\vec{u}_1 \vec{u}_2 \dots \vec{u}_d]$  e  $d$  é o número de eixos principais, para obter os *scores* dos componentes principais basta multiplicar o conjunto de dados original  $\mathbf{X}$  por  $\mathbf{U}$ , conforme equação:

$$\mathbf{P} = \mathbf{XU} \quad (2.2)$$

, em que  $\mathbf{P}$  representa o conjunto de dados transformados. A Figura 2.3 exemplifica a projeção ocorrida das variáveis  $X_1$  e  $X_2$  (Figura 2.2) nos eixos  $P_1$  e  $P_2$ .

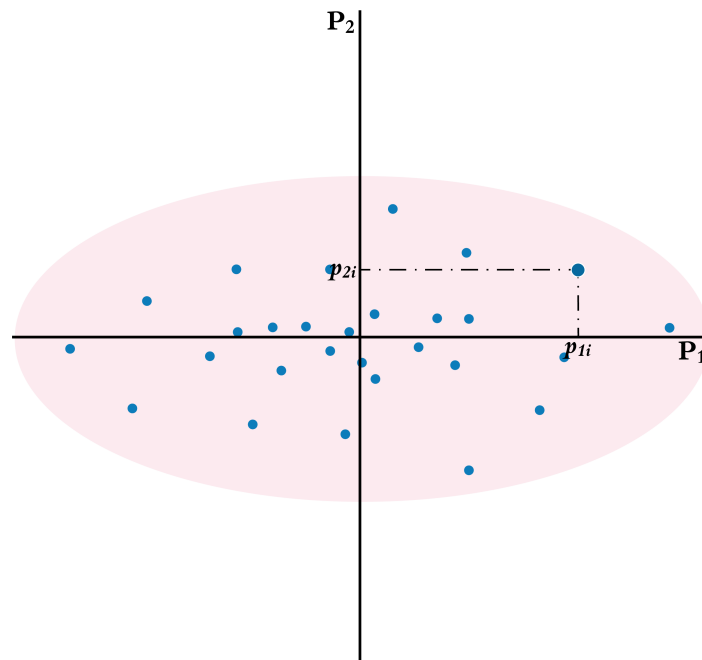


Figura 2.3: Gráfico de dispersão com os pontos projetados nos novos eixos  $P_1$  e  $P_2$ .

Para promover a redução de dimensionalidade, ao invés de multiplicarmos  $\mathbf{X}$  por  $\mathbf{U}$ , multiplicamos  $\mathbf{X}$  por  $\mathbf{U}_{(\text{redu})}$ , em que  $\mathbf{U}_{(\text{redu})} = [\vec{u}_1 \vec{u}_2 \dots \vec{u}_k]$  para  $k < d$ . Desse modo, são descartados os últimos autovetores de  $\mathbf{U}$ , os quais estão associados aos componentes principais de menor variância, e o conjunto de dados transformado será  $k$ -dimensional ao invés de  $d$ -dimensional.

Nesta pesquisa, empregaremos métodos de extração de características, especificamente a PCA, objetivando a melhoria do desempenho do classificador e a redução do custo computacional proveniente do processo de treinamento.



### 2.1.3 Balanceamento de classes

Segundo Visa [29], para um tarefa de classificação envolvendo duas classes, quando uma das classes (majoritária) supera em grande parte a outra classe (minoritária), surge então um problema de distribuição de classes desbalanceadas. Esse tipo de problema pode levar algoritmos de classificação a apresentarem baixo desempenho. O mau desempenho de classificadores gerados por meio de algoritmos de aprendizagem de máquina padrão aplicados sobre bases de treinamento em que há presença de classes desbalanceadas ocorre, principalmente, devido aos seguintes fatores: (i) adoção da acurácia como métrica de desempenho — os algoritmos padrão são conduzidos pela acurácia (minimização do erro geral) para a qual a classe minoritária contribui muito pouco —; (ii) distribuição de classe — os classificadores atuais assumem que os algoritmos operarão sobre dados extraídos da mesma distribuição apresentada nos dados de treinamento, não obstante isso raramente aconteça —; e (iii) custo do erro — os classificadores assumem que os erros provenientes das diferentes classes possuem mesmo custo.

Zhi *et al.* [30] afirmam que uma das soluções para o problema de desbalanceamento é a reamostragem. Por meio dela, o conjunto de dados desbalanceado pode tornar-se balanceado. As técnicas de reamostragem podem ser agrupadas em três categorias: *oversampling*, *undersampling* e híbrido.

Nas subseções a seguir, discorremos sobre os três grupos de técnicas aplicáveis no tratamento de conjuntos de dados desbalanceados.

#### Oversampling

Nas técnicas de *oversampling*, cria-se um superconjunto a partir do conjunto de dados replicando algumas das instâncias da classe minoritária ou criando novas instâncias provenientes de instâncias da classe minoritária [31]. Entre as técnicas de *oversampling*, temos a *random oversampling* — aumento do número de instâncias da classe rara por meio da replicação aleatória dessas instâncias — e a *Synthetic Minority Oversampling Technique* (SMOTE) — aumento do número de instâncias da classe rara por meio da geração de novas observações sintetizadas a partir de instâncias da rara contidas no conjunto de dados original.

No estudo conduzido por Batista *et al.* [32], no qual foram comparados os desempenhos de diversas técnicas de *oversampling* aplicadas sobre diferentes conjuntos de dados com diferentes proporções de desbalanceamento, os autores constataram que, para os casos em que o número de observações da classe de interesse é significativamente inferior aos das demais classes, a adoção da SMOTE combinado com técnicas de *undersampling* —

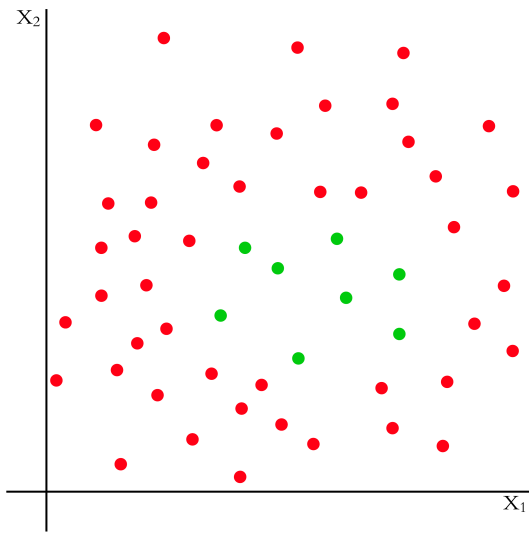
*Tomek Links* — leva a desempenhos superiores aos obtidos com a adoção da técnica *random oversampling*.

Na subseção abaixo, exporemos, sinteticamente, as principais características da SMOTE, visto que ela será aplicada neste estudo.

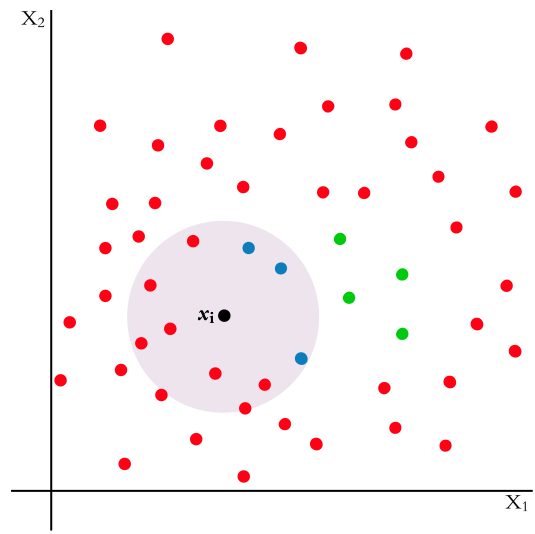
### SMOTE

A técnica SMOTE, proposta por Chawla *et al.* [33], consiste na criação de instâncias “sintéticas” da classe minoritária, ao invés da simples replicação aleatória dessas instâncias. As novas instâncias são criadas da seguinte maneira: (i) seleciona-se uma instância  $\vec{x}_i$  entre as instâncias da classe minoritária; (ii) dentre os  $k$  vizinhos mais próximos a  $\vec{x}_i$  pertencentes à mesma classe de  $\vec{x}_i$ , escolhe-se um aleatoriamente ( $\vec{x}_j$ ); (iii) calcula-se a diferença entre  $\vec{x}_i$  e  $\vec{x}_j$ , resultando em  $\vec{d}$ ; (iv) multiplica-se  $\vec{d}$  por um valor escolhido aleatoriamente entre 0 e 1; (v) soma-se  $\vec{x}_i$  a  $\vec{d}$  e o resultado será uma nova instância sintética. Esses passos devem ser repetidos até que se obtenha um novo conjunto de dados que contenha instâncias da classe minoritária na proporção desejada.

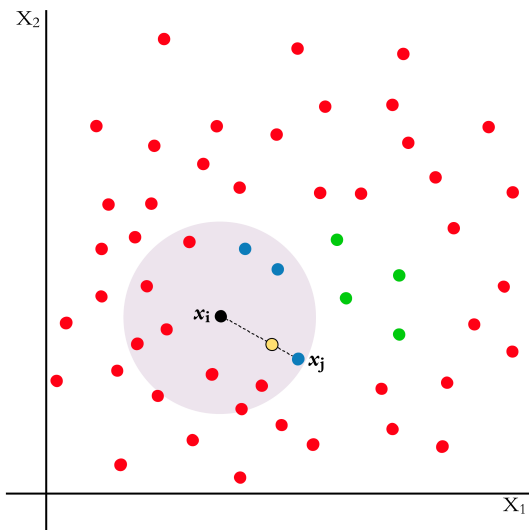
O processo descrito acima leva a seleção aleatória de um ponto no segmento de reta que é formado entre duas instâncias. A Figura 2.4 retrata o processo criação de uma nova instância a partir do seguimento de reta criado entre as instâncias  $\vec{x}_i$  e  $\vec{x}_j$ . Inicialmente, o conjunto de dados possui oito instâncias da classe minoritária. Ao final do processo, o conjunto de dados passa a possuir nove instâncias da classe minoritária. A nova instância, no entanto, é completamente diferente das demais.



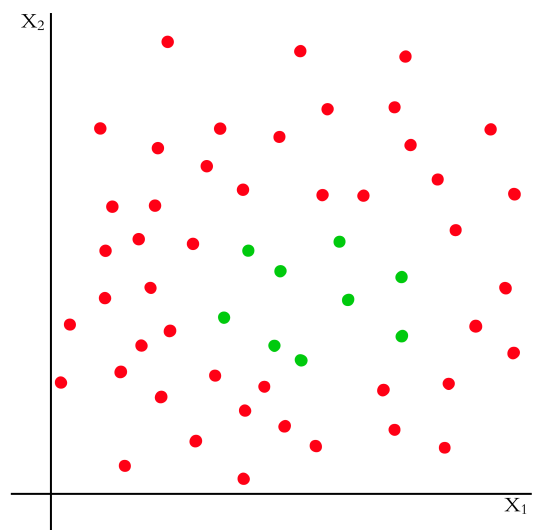
(a) Conjunto de dados inicial contendo oito instâncias da classe minoritária.



(b) Seleção da instância  $x_i$  da classe minoritária e de seus  $k$  vizinhos mais próximos ( $k = 3$ ).



(c) Seleção aleatória de um dos  $k$  vizinhos mais próximos ( $x_j$ ) e criação de instância (ponto amarelo) sobre seguimento de reta formado entre  $x_i$  e  $x_j$ .



(d) Conjunto de dados contendo nove instâncias da classe minoritária.

Figura 2.4: Descrição simplificada das etapas de geração de uma instância sintética utilizando a técnica SMOTE. Os pontos vermelhos representam as instâncias da classe majoritária e os pontos verdes correspondem às instâncias da classe minoritária.

Segundo Chawla *et al.* [33], o desempenho superior da técnica SMOTE sobre a *random oversampling* pode ser explicado pelo fato de a técnica *random oversampling* acabar por reduzir a região de decisão que resulta na classificação da classe minoritária, enquanto a SMOTE expande essa região de decisão. Isso ocorre porque, ao apenas replicar as

instâncias da classe minoritária, o algoritmo de classificação sobreajusta o modelo às instâncias replicadas, reduzindo, portanto, a região de decisão. O efeito inverso se observa com a aplicação da SMOTE, que, em função da inserção de ruído nos dados originais, impele o algoritmo de classificação a expandir a região de decisão da classe minoritária.

## Undersampling

Nas técnicas de *undersampling*, cria-se um subconjunto a partir do conjunto de dados original mediante a eliminação de algumas instâncias da classe majoritária [31]. Entre as técnicas de *undersampling*, temos a *random undersampling* — redução do número de instâncias da classe majoritária por meio de remoção aleatória dessas instâncias —, *Tomek Links* — eliminação de instâncias da classe majoritária que representam ruído ou instância de fronteira —, *Condensed Nearest Neighbor Rule* (CNN) — identificação de um subconjunto  $\hat{D}$  derivado do conjunto original  $D$  ( $\hat{D} \subseteq D$ ) que, quando utilizado no treinamento de um modelo KNN com  $k = 1$ , produza um modelo capaz de classificar corretamente as instâncias do conjunto original  $D$  —, e *One-sided Selection* — aplicação da técnica *Tomek Links* seguida da técnica CNN [32].

Conforme foi comentado acima, Batista *et al.* verificaram que a combinação da técnica SMOTE com a *Tomek Links* pode resultar em desempenhos superiores aos que seriam obtidos com a utilização de outras técnicas de balanceamento de classes ou com a não aplicação de qualquer técnica.

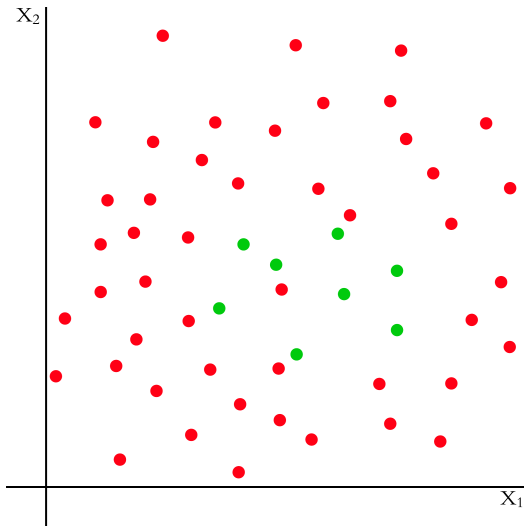
Na subseção abaixo, discutiremos, sumariamente, os principais aspectos da técnica *Tomek Links*, uma vez que ela será aplicada neste estudo.

### *Tomek Links*

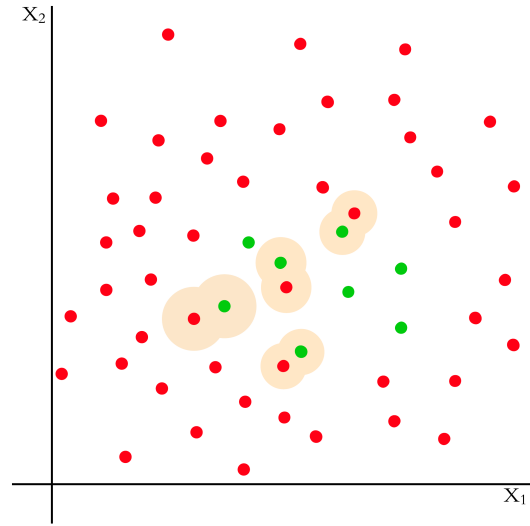
Propostos por Tomek [34], *Tomek Links* podem ser descritos como pares de instâncias de classes opostas cujo círculo de influência não contém qualquer outra instância além das instâncias que compõem o par [35]. A remoção de instâncias utilizando *Tomek Links* ocorre da seguinte maneira: seja  $\vec{x}_i$  e  $\vec{x}_j$  duas instâncias de classes opostas e  $d(\vec{x}_i, \vec{x}_j)$  a distância entre  $\vec{x}_i$  e  $\vec{x}_j$ , o par  $(\vec{x}_i, \vec{x}_j)$  é considerado um *Tomek Link* caso nenhuma outra instância  $x_n$  exista, tal que  $d(\vec{x}_n, \vec{x}_i) < d(\vec{x}_i, \vec{x}_j)$  ou  $d(\vec{x}_n, \vec{x}_j) < d(\vec{x}_i, \vec{x}_j)$ ; com base na definição anterior, identifica-se todos os pares *Tomek Links* do conjunto de dados; por último, remove-se as instâncias da classe majoritária que formam os pares *Tomek Links* [36].

Se duas instâncias formam um par *Tomek Link*, ou uma das instâncias é ruído ou ambas constituem instâncias de borda [32]. A Figura 2.5 apresenta, de forma sucinta, as etapas de remoção de instâncias que pertencem à classe majoritária e que compõem pares

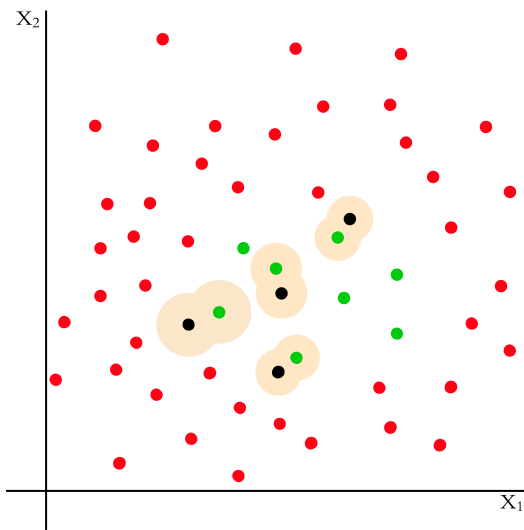
*Tomek Links*, os quais, no exemplo apresentado, são formados tanto por ruídos quanto por instâncias de borda.



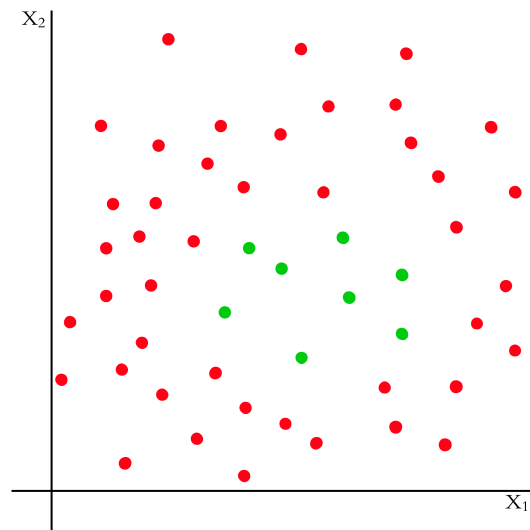
(a) Conjunto de dados inicial contendo 47 instâncias da classe majoritária.



(b) Delineamento dos círculos de influência e identificação dos pares *Tomek Links*.



(c) Remoção das instâncias pertencentes à classe majoritária que integram os pares *Tomek Links* (pontos pretos).



(d) Conjunto de dados contendo 43 instâncias da classe majoritária.

Figura 2.5: Descrição simplificada das etapas de remoção de instâncias da classe majoritária utilizando a técnica *Tomek Links*. Os pontos vermelhos representam as instâncias da classe majoritária enquanto os pontos verdes correspondem às instâncias da classe minoritária.

## Híbrido

As técnicas híbridas são técnicas que aplicam *oversampling* e *undersampling* de forma conjunta. Entre as técnicas existentes, podemos citar SMOTE+ENN, Borderline+SMOTE1, Borderline+SMOTE2, Safe-Level+SMOTE e SMOTE+Tomek. Discorreremos brevemente sobre o SMOTE+Tomek.

Proposto por Batista *et al.*, o método SMOTE+Tomek combina a técnica SMOTE com a técnica *Tomek Links*. Nesse método, aplica-se primeiro a SMOTE para geração de novas instâncias sintéticas e, na sequência, eliminam-se as instâncias contidas nos pares *Tomek Links*, mas desta vez a remoção é feita de forma indistinta, *i.e.*, eliminam-se instâncias de ambas as classes. Desse modo, durante o processo de remoção, também são removidas as instâncias sintéticas da classe minoritária que eventualmente invadam a região da classe majoritária.

Neste trabalho, será aplicado o método de rebalanceamento de classes SMOTE+Tomek, porquanto a proporção entre COPs deferidas e COPs indeferidas caracteriza um problema de classes desbalanceadas.

### 2.1.4 Algoritmos de Classificação

Nesta seção serão abordados os algoritmos de classificação a serem aplicados na pesquisa.

Quando o problema de aprendizagem tem o objetivo de construir um modelo capaz de determinar a classe de uma nova observação dentre um conjunto de classes previamente conhecidas, estamos diante de uma tarefa de classificação. Se para cada observação contida no conjunto de dados de treinamento existe a identificação da classe a qual a observação pertence, então trata-se de um problema de classificação com aprendizagem supervisionada.

Para criação de modelo de classificação, é necessária a utilização de algoritmos próprios para classificação. Dentre os algoritmos supervisionados de classificação comumente utilizados, adotaremos Máquina de Vetores de Suporte (SVM, do inglês *Support Vector Machine*), Redes Neurais Artificiais (RNAs), *Naive Bayes* (NB) e *Random Forest* (RF). A escolha do SVM e do NB decorre do fato desses terem sido os algoritmos que apresentaram maior desempenho no estudo semelhante a este conduzido por Ramos [37]. Já a escolha dos algoritmos RF e RNAs foi baseada em recentes publicações [38, 39, 40, 41] nas quais esses algoritmos mostraram desempenho proeminente.

Nas próximas subseções, versaremos acerca dos algoritmos SVM, RNAs, NB e RF, conforme explicitado acima.

## Máquina de Vetores de Suporte

A Máquina de Vetores de Suporte é um eficiente método de aprendizagem de máquina com uma sólida fundamentação teórica [42]. Considerado uma ferramenta robusta própria para classificação binária e regressão, o SVM tem alcançado excelente desempenho em diversos problemas de predição reais, como categorização de texto, predição de séries temporais, reconhecimento de padrão, processamento de imagem etc. [43].

Seu funcionamento consiste no aprendizado do hiperplano ótimo que separa duas classes no espaço de características. Em geral, o SVM apresenta alta robustez e capacidade de generalização com apenas um pequeno conjunto de observações de treinamento, ou seja, o hiperplano ótimo é definido pelo número mínimo de vetores de suporte [42].

Para um conjunto de dados contendo duas classes ( $A$  e  $B$ ), o hiperplano ótimo é aquele que separa as classes  $A$  e  $B$  com a maior margem de separação possível. Para exemplificar, considere um conjunto de dados linearmente separáveis  $(\vec{x}_i, y_i), i = 1, 2, \dots, N$ , em que  $\vec{x}_i \in R^D$  representa o vetor de características e  $y_i \in \{+1, -1\}$  representa a classe correspondente. A equação do hiperplano em um espaço vetorial de  $D$ -dimensões é dada por  $\vec{\omega} \cdot \vec{x} + b = 0$ , em que  $\vec{\omega}$  é um vetor normal ao hiperplano e  $b$  uma constante, que, juntamente com  $\vec{\omega}$ , determina a distância entre o hiperplano e a origem.

Com base nos requisitos para o hiperplano ótimo, o problema pode ser solucionado por meio de programação quadrática a partir da transformação da equação do hiperplano na seguinte função objetiva:

$$\begin{aligned} \min \Phi(\vec{\omega}) &= \frac{1}{2} \|\vec{\omega}\|^2 \\ \text{sujeito a } & y_i [(\vec{\omega} \cdot \vec{x}_i) + b] \geq 1, i = 1, 2, 3, \dots, N. \end{aligned} \tag{2.3}$$

A Figura 2.6 apresenta o funcionamento do SVM quando aplicado com margens rígidas. Nela, é possível observar que as observações estão, ou sobre a margem, ou afastadas da margem em sentido contrário ao das observações da classe oposta.

Quando estamos lidando com dados não linearmente separáveis, é preciso permitir que haja algum tipo de violação da margem ou do hiperplano pelas observações, ou seja, é preciso permitir que algumas observações “avancem” sobre a margem ou “cruzem” o hiperplano, acessando a região da classe oposta. Para tal, faz-se necessário modificar a Equação 2.3, de modo a obter a seguinte equação:

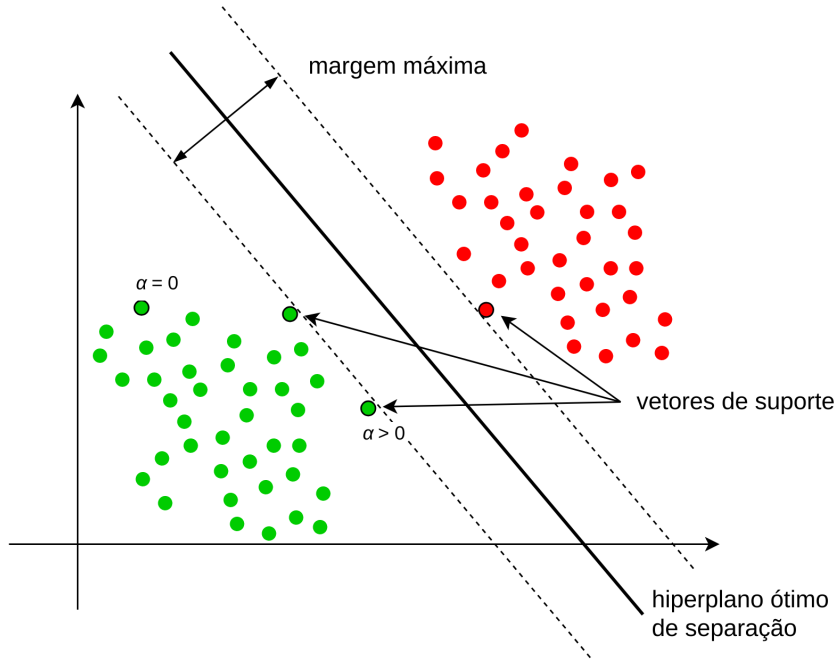


Figura 2.6: Representação gráfica do hiperplano, da margem máxima e dos vetores de suporte definidos a partir do SVM com margem rígida. Modificado a partir de Jain

$$\begin{aligned} \min \Phi(\vec{\omega}, \vec{\xi}) &= \frac{1}{2} \|\vec{\omega}\|^2 + C \left( \sum_{i=1}^N \xi_i \right) \\ \text{sujeito a } y_i [(\vec{\omega} \cdot \vec{x}_i) + b] &\geq 1 - \xi_i, i = 1, 2, 3, \dots, N \\ \xi_i &\geq 0, i = 1, 2, 3, \dots, N \end{aligned} \quad (2.4)$$

em que  $\xi_i$  representa a variável de folga da observação  $i$  ou, em outras palavras, quanto à observação  $i$  violou a margem.

Depois de introduzir os multiplicadores de Lagrange ( $\alpha$ ), a equação dual do problema pode ser reescrita da seguinte forma [44]:

$$\begin{aligned} \max \mathcal{L}(\alpha) &= \sum_{i=1}^N \alpha_i - \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (\vec{x}_i \cdot \vec{x}_j) \\ \text{sujeito a } 0 &\leq \alpha_i \leq C, i = 1, 2, 3, \dots, N \\ \sum_{i=1}^N y_i \alpha_i &= 0 \end{aligned} \quad (2.5)$$

em que  $C$  é o parâmetro que determina o custo do erro, logo, quanto maior for  $C$  menor será o total de violações permitidas.

A solução da Equação 2.5 produzirá um vetor contendo os multiplicadores de Lagrange,



$\vec{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_N\}$ , em que cada multiplicador  $\alpha_i$  está associado a uma instância do conjunto de dados de treinamento. Combinando os multiplicadores de Lagrange, as instâncias do conjunto de dados de treinamento, os valores correspondentes às classes das instâncias e a constante  $b$ , obtém-se a seguinte equação de classificação:

$$f(\vec{x}) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i (\vec{x} \cdot \vec{x}_i) + b\right) \quad (2.6)$$

em que  $\text{sign}$  é uma função que retorna “-1” caso o valor da entrada seja negativo e “+1” caso o valor seja positivo.

Os vetores de suporte serão as instâncias para as quais  $\alpha_i > 0$ , enquanto as instâncias em que  $\alpha_i = 0$  serão consideradas vetores não suporte [45]. São os vetores de suporte que definem as margens e, conseqüentemente, o hiperplano, daí o nome “máquina de vetores de suporte”.

A Figura 2.7 apresenta o SVM aplicado com margens suaves. A partir da figura, é possível observar que, ao utilizar margens suaves, algumas instâncias avançam sobre a margem e sobre o hiperplano.

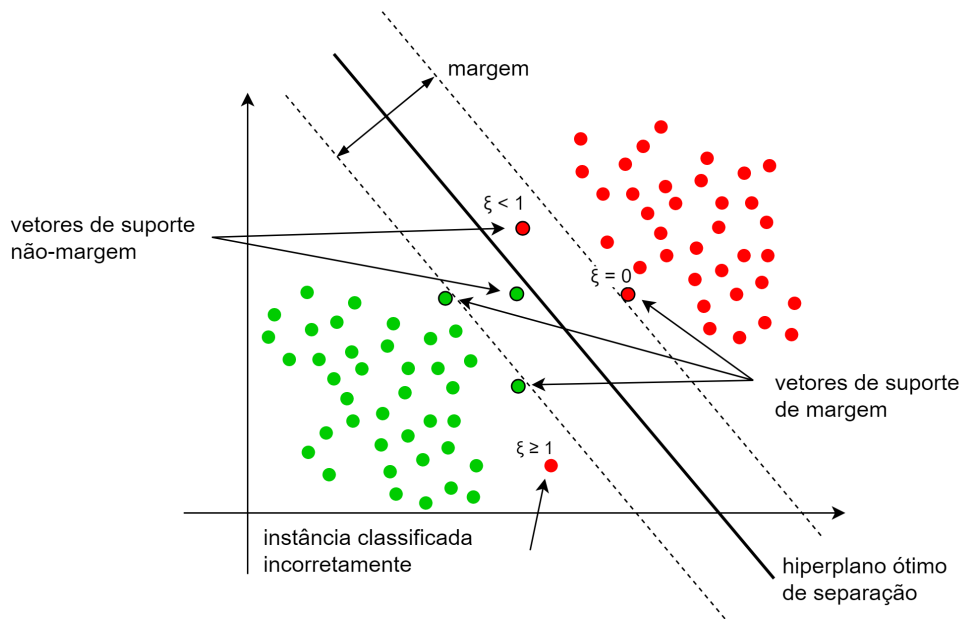


Figura 2.7: Representação gráfica do hiperplano, da margem máxima e dos vetores de suporte definidos a partir do SVM com margem suave. Modificada a partir de Nanni *et al.*[46] e Oberlin[45].

As Equações 2.3 e 2.4 determinam o funcionamento do processo de identificação dos vetores de suporte, ou seja, o processo de identificação das instâncias que definem as margens máximas do hiperplano ótimo. A Equação 2.3 implementa o SVM com margens rígidas, quando não são permitidas violações, enquanto a Equação 2.4 implementa o

SVM com margens suaves, em que são permitidas violações, sendo a quantidade total de violações determinada pelo parâmetro  $C$ .

## Redes Neurais Artificiais

Nas últimas décadas, as RNAs têm emergido como uma tecnologia prática com aplicações bem sucedidas em diferentes áreas [47]. Com sua utilização como ferramenta de classificação, as RNAs apresentam-se como uma alternativa promissora aos vários métodos convencionais de classificação [48]. As vantagens das RNAs residem nos seguintes aspectos: (i) são métodos orientados a dados (*data-driven*) auto-adaptativos, ou seja, elas podem ajustar-se aos dados sem qualquer especificação explícita da forma funcional ou da distribuição do modelo subjacente; (ii) são consideradas aproximadores universais, em outras palavras, são capazes de aproximar qualquer função com acurácia arbitrária [49]; (iii) são modelos não lineares, o que as tornam flexíveis na modelagem das complexas relações existentes em aplicações do mundo real; e (iv) são capazes de estimar a probabilidade *a posteriori* [50], a qual provê as bases para o estabelecimento das regras de classificação e para a análise estatística.

As RNAs têm sido aplicadas com sucesso em uma variedade de tarefas de classificação presentes na áreas da indústria, ciência e negócios [48]. As aplicação incluem predição de falência, reconhecimento de palavras manuscritas, reconhecimento de discurso, inspeção de produtos, detecção de falhas, diagnósticos médicos e classificação de títulos de dívida.

Existem diferentes estruturas de redes neurais tais como *multilayer perceptron* (MLP), *radial basis function* (RBF), *continuous-time dynamic*, etc. [51]. Nesta subseção, discorreremos acerca das redes MLP visto que são um dos tipos mais populares de redes neurais, sendo utilizadas em diversas aplicações.

As redes MLP são parte de uma classe geral de estruturas chamada *feedforward neural networks* (FFNNs) [51]. As FFNNs estão entre as mais simples e comuns estruturas de redes neurais, tendo como característica principal o fato da informação mover-se sempre para frente [52], em outras palavras, não há ciclos direcionados na arquitetura da rede neural [5]. Na estrutura da rede MLP, os neurônios são agrupados em diversas camadas. A primeira e a última camada são denominadas camada de entrada e camada de saída, respectivamente. As demais camadas são chamadas de camadas ocultas. Tipicamente, uma rede MLP inclui uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída [51].

Cada neurônio de uma rede MLP é composto por uma função de ativação – generalização de uma função discriminante em que o resultado da combinação linear dos parâmetros de entrada são transformados por meio de uma função não linear [47] – e cada camada é completamente conectada com a próxima, de modo que a saída da função de ativação de

um neurônio pertencente a uma camada será um dos parâmetros de entrada para cada um dos neurônios da camada seguinte [40]. A combinação de todos os neurônios resulta na criação de uma fronteira de decisão, ou seja, na definição dos limites de uma região contida no espaço vetorial em que um determinado ponto (observação) é considerado de uma classe, e não de outra [47].

Quando o tarefa de classificação envolve aplicações em que há somente duas classes a serem preditas, o classificador baseado em rede neural MLP terá na sua camada de saída apenas um neurônio[19]. Para os casos em que há mais de uma classe, duas abordagens podem ser adotadas: a primeira consiste em criar um classificador para cada classe a ser predita, em que, novamente, a camada de saída de cada classificador deverá conter um único neurônio; e a segunda abordagem consiste em criar um classificador em que o número de neurônios na camada de saída será igual ao número de classes a serem preditas.

Para exemplificar a segunda abordagem, considere  $L$  como o número do total de camadas em uma rede MLP em que cada camada é apresentada por  $l$  [51]. A primeira camada  $l_1$  é a camada de entrada, a camada  $l_L$  corresponde à camada de saída e as camadas  $l_2, l_3, \dots, l_{L-1}$  correspondem às camadas ocultas. Neste exemplo, o número de neurônios nas camadas ocultas é o mesmo para todas as camadas, o qual é definido pela variável  $n_h$ . Além disso, o vetor de características  $\vec{x}$  possui dimensão  $n_x$ ,  $\vec{x} = (x_1, x_2, \dots, x_{n_x})$ , e o vetor resposta  $\vec{y}$  (vetor contendo a predição da classe a qual instância pertence) possui dimensão  $n_y$ ,  $\vec{y} = (y_1, y_2, \dots, y_{n_y})$ . Com base nesses dados, teríamos uma rede neural semelhante à descrita na Figura 2.8. Nessa figura, o vetor  $\vec{x}$  é inserido na camada de entrada da rede e obtém-se na camada de saída o vetor  $\vec{y}$ , em que cada elemento  $y_i$  representa a probabilidade *a posteriori* da observação pertencer à classe  $i$ . É importante considerar que os resultados da camada de saída podem ser considerados estimativas de probabilidades Bayesianas quando a função de custo utilizada for o erro quadrático ou a entropia cruzada [50].

Entre as funções de ativação existentes, uma das mais comuns é a função *sigmoide*. Ela é definida pela seguinte equação [19, 53]:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.7)$$

, possuindo as seguintes propriedades:

$$\begin{aligned} \lim_{x \rightarrow -\infty} \sigma(z) &= 0 \quad \text{e} \\ \lim_{x \rightarrow +\infty} \sigma(z) &= 1 \end{aligned}$$

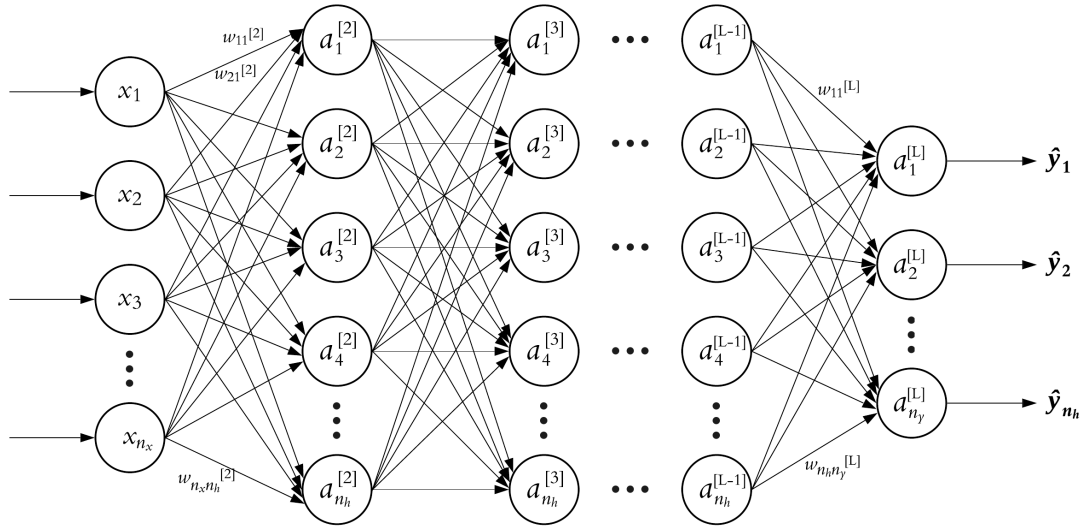


Figura 2.8: Representação gráfica de uma rede MLP com  $L$  camadas e com  $n_h$  neurônios em cada camada oculta  $l$ .

Na Equação 2.7,  $z$  é a soma ponderada dos valores de entrada adicionada ao viés (*bias*). Podemos representar  $z$  por meio da equação:

$$z_j^{[l]} = \vec{w}_j^{[l]} \cdot \vec{a}^{[l-1]} + b_j^{[l]} \quad (2.8)$$

, em que  $b_j^{[l]}$  e  $\vec{w}_j^{[l]}$  são, respectivamente, o *bias* e o vetor de pesos associados ao  $j$ -ésimo neurônio da camada  $l$ , e o vetor  $\vec{a}^{[l-1]}$  corresponde às saídas produzidas pelas funções de ativação dos neurônios da camada oculta  $l - 1$ .

As redes MLP utilizam uma técnica de aprendizagem supervisionada chamada *back propagation*, que pode ser descrita como um processo iterativo para minimização da função custo. O processo consiste em calcular as derivadas da função custo com relação aos pesos  $\vec{w}_j^{[l]}$ . A partir das derivadas parciais, determina-se o gradiente e ajustam-se os pesos a fim de minimizar a função custo [47]. O processo é repetido até que se encontre o valor mínimo local para a função custo ou uma condição de parada seja satisfeita. A técnica de ajuste dos pesos com base no gradiente é descrita como uma estratégia de gradiente descendente [19].

## Naive Bayes

*Naive Bayes* é um algoritmo de aprendizagem de máquina que combina a utilização do teorema de Bayes com a assunção de independência das variáveis preditoras condicionada a classe [27]. Os classificadores construídos a partir do *Naive Bayes* são considerados classificadores probabilísticos. Eles têm sido aplicados em diversos domínios e, apesar

de a suposição de independência ser frequentemente violada, bons resultados têm sido obtidos.

Entre as principais características de um classificador *Naive Bayes*, podemos destacar as seguintes [54]: (i) eficiência computacional — a complexidade de tempo para treinamento é linear em relação ao número de observações e atributos, e, para classificação, linear em relação ao número de atributos —; (ii) baixa variância — visto que o *Naive Bayes* não utiliza busca para otimização dos parâmetros, ele possui baixa variância, embora, em contrapartida, isso resulte em um viés elevado —; (iii) robustez contra ruído nos dados — o *Naive Bayes* sempre utiliza todos os atributos (variáveis preditoras) para predição e, portanto, é relativamente insensível à presença de ruído nas instâncias a serem classificadas —; e (iv) robustez a valores ausentes (*missing values*) — uma vez que o *Naive Bayes* utiliza todos os atributos em todas as predições, caso o valor de um atributo esteja ausente, as informações dos atributos remanescentes ainda serão utilizadas, impactando diminutamente o desempenho.

O teorema de Bayes, no qual um classificador *Naive Bayes* é baseado, pode ser descrito pela seguinte equação:

$$p(c_k|\vec{x}) = \frac{p(\vec{x}|c_k)p(c_k)}{p(\vec{x})} \quad (2.9)$$

em que  $p(c_k|\vec{x})$  é a probabilidade *a posteriori* de  $c_k$ , *i.e.*, probabilidade da classe  $c_k$  dado que observamos o vetor de características  $\vec{x}$ ,  $p(\vec{x}|c_k)$  é a probabilidade (ou densidade de probabilidade para variáveis contínuas) de  $\vec{x}$  dado a classe  $c_k$ ,  $p(c_k)$  é a probabilidade *a priori* e  $p(\vec{x})$  é a probabilidade de ocorrência do vetor  $\vec{x}$ .

Dado a assunção de que os atributos são condicionalmente independentes,  $p(\vec{x}|c_k)$  pode ser calculada da seguinte forma:

$$p(\vec{x}|c_k) = \prod_{j=1}^n p(x_j|c_k) \quad (2.10)$$

em que  $n$  representa a dimensão do vetor de características.

Visto que no momento da classificação  $\vec{x}$  é constante, temos:

$$p(c_k|\vec{x}) \propto p(\vec{x}|c_k)p(c_k) \quad (2.11)$$

logo, para determinar a classe à qual uma observação pertence, basta adotarmos a seguinte regra<sup>1</sup>:

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} p(c) \prod_{j=1}^n p(x_j|c) \quad (2.12)$$

em que  $C = \{c_1, c_2, \dots, c_m\}$ , sendo  $m$  o número de classes.

---

<sup>1</sup>[https://scikit-learn.org/stable/modules/naive\\_bayes.html](https://scikit-learn.org/stable/modules/naive_bayes.html)

O treinamento de um classificador Naive Bayes consiste em determinar os parâmetros de distribuição de cada variável preditora  $X_1, X_2, \dots, X_n$ . Os parâmetros são estimados pelo método de *máxima verossimilhança* utilizando os dados de treinamento. As probabilidades *a priori* são estimadas na fase de treinamento. Todavia, muitas implementações permitem arbitrar essas probabilidades.

## Random Forest

Antes de introduzirmos os principais aspectos de um classificador baseado no algoritmo *Random Forest* (RF), traremos alguns conceitos subjacentes que permitirão uma melhor compreensão de sua constituição e funcionamento. Versaremos sobre o algoritmo *Classification And Regression Tree* (CART), a técnica de combinação (*ensemble*) de classificadores *Bagging* e o próprio algoritmo *Random Forest*.

### *Classification And Regression Tree*

O algoritmo CART, proposto por Breiman *et al.* em 1984 [55], é um algoritmo de árvore de decisão capaz de processar atributos tanto nominais quanto contínuos e cujo resultado pode ser também contínuo ou categórico [56]. Seu funcionamento consiste no particionamento recursivo binário dos dados de treinamento, o que leva à geração de uma estrutura de *árvore binária*.

O treinamento de um modelo classificatório CART constitui-se de duas etapas: crescimento da árvore de decisão e subsequente poda. O processo de crescimento inicia no *nó raiz*, quando todos os dados de treinamento são particionados em dois *nós filhos*. Para cada novo nó filho, particiona-se novamente os dados de modo a gerar mais dois novos nós, porém agora utilizando-se apenas o subconjunto de dados que lhe foi atribuído. O procedimento se repete até não ser possível efetuar novo particionamento, o que ocorre ou por falta de observações a serem divididas ou porque o subconjunto de dados associado ao nó folha é homogêneo, *i.e.*, todas as observações do nó folha pertencem à mesma classe.

A cada nova divisão, é preciso antes escolher o atributo que leve à maior redução da entropia ou grau de impureza dos dados em relação à variável de classe [39, 56]. O critério de seleção baseia-se no índice *Gini*, de sorte que para cada atributo é calculado o índice de *Gini* sobre o conjunto de possíveis pontos de divisão. O par ‘atributo - ponto de divisão’ que resultar no menor índice de *Gini* será o escolhido para particionamento dos dados.

O índice *Gini* mede a impureza de um conjunto de dados em relação à variável alvo [57]. Quando aplicado na geração de uma árvore de decisão, ele mede o grau de impureza das novas partições com relação à variável de classe. O índice *Gini* pode ser calculado com a

seguinte equação:

$$Gini(D) = 1 - \sum_{i=1}^m p_i^2 \quad (2.13)$$

em que  $D$  é o conjunto de dados da partição,  $p_i$  é a probabilidade de uma observação em  $D$  pertencer à classe  $c_i$  e  $m$  é o número de classes, logo  $i = [1, 2, \dots, m]$ .

Considerando um particionamento binário — como ocorre em árvores CART —, o índice *Gini* final será a soma ponderada dos índices calculados sobre as novas partições [57]. Para exemplificar, caso divida-se a partição  $D$  em duas partições  $D_1$  e  $D_2$ , o índice final será calculado por:

$$Gini_{final}(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) \quad (2.14)$$

em que  $|D|$ ,  $|D_1|$  e  $|D_2|$  são o número de observações das partições  $D$ ,  $D_1$  e  $D_2$  respectivamente.

Depois de a árvore de decisão alcançar seu tamanho máximo, inicia-se o processo de poda. A poda consiste na remoção dos *nós de decisão* da árvore, começando pelo nó de decisão que menos contribui para o desempenho do classificador até atingir o nó raiz. A decisão do nó a ser removido é feita com base no desempenho obtido nos dados de treinamento. A cada remoção de um nó de decisão, uma nova árvore é gerada e, em seguida, separada para posterior avaliação e seleção. Ao final do processo de poda, haverá um conjunto de árvores com diferentes estruturas. Seleciona-se, então, a árvore que obtiver o melhor desempenho nos dados de teste.

A Figura 2.9 apresenta uma árvore de decisão gerada com o algoritmo CART. A base de dados utilizada foi a *Titanic Data* — conjunto de dados contendo informações sobre passageiros que haviam embarcado no *RMS Titanic*. A base contém 1309 registros e seis variáveis: (i) *classeBilhete*: classe do bilhete (1ª, 2ª ou 3ª classe); (ii) *sexo*: sexo do passageiro; (iii) *idade*: idade do passageiro; (iv) *qtdIrmaos*: quantidade de irmãos ou cônjuges do passageiro a bordo; (v) *qtdPais*: quantidade de pais ou filhos do passageiro a bordo; (vi) *classe*: informação se o passageiro sobreviveu ou não ao naufrágio.

Para classificar uma nova observação, basta seguir o fluxo decisório que se inicia no nó raiz e segue até um nó folha. Será, então, atribuída à nova instância a classe que apresentar o maior número de ocorrências no nó folha ao qual foi associada.

### *Bagging*

*Bagging*, ou *bootstrap aggregating*, é um método de geração e agregação de classificadores. O método foi proposto por Breiman [59] e tem como principal característica a geração de modelos com baixa variância a partir da combinação de modelos gerados por algoritmos

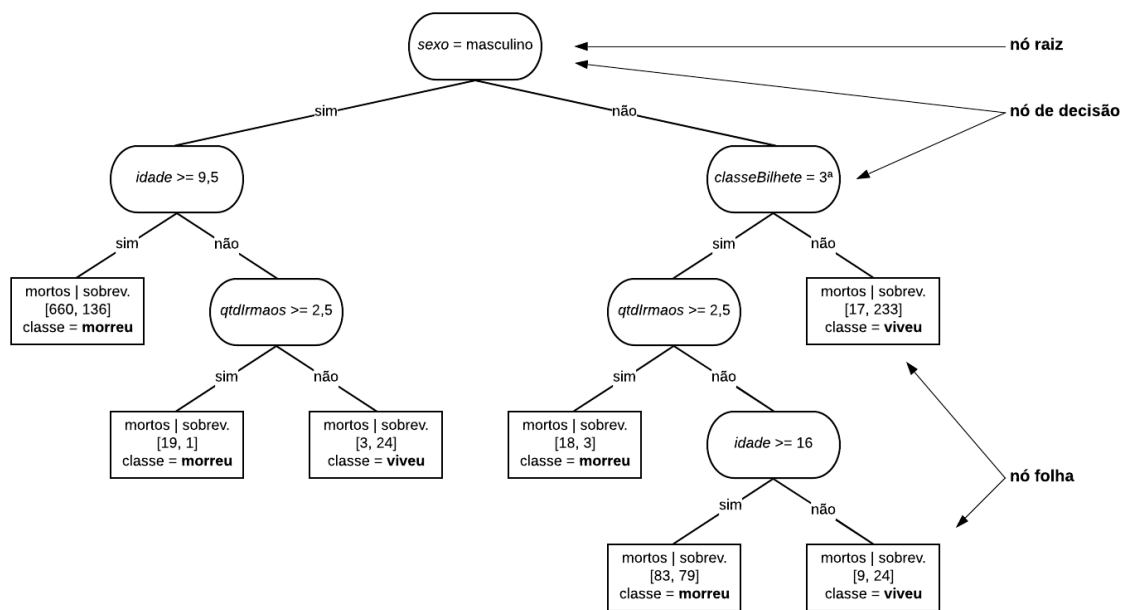


Figura 2.9: Exemplo de uma árvore de decisão gerada com o algoritmo CART após a poda (adaptada a partir de Ma [58]).

instáveis — algoritmos que produzem modelos com previsões significativamente diferentes em razão de ligeiras diferenças entre os conjuntos de dados de treinamento.

A fase de treinamento do método *bagging* consiste na geração de subconjuntos de treinamento por meio da técnica de amostragem *bootstrap* — geração de múltiplas amostras de mesmo tamanho com o uso de reamostragem com reposição — aplicada ao conjunto de dados de treinamento inicial [39]. Depois de gerados os subconjuntos de treinamento, é, então, treinado um classificador para cada novo subconjunto de dados.

Na fase de predição, a nova instância será avaliada por todos os classificadores gerados na etapa de treinamento, de modo que cada classificador deverá atribuir uma classe à nova instância. A classe da nova instância será determinada por meio de votação, *i.e.*, será atribuída à nova instância a classe com maior número de ocorrências.

Segundo Breiman, o *bagging* funciona bem para algoritmos instáveis. A evidência, tanto experimental quanto teórica, sugere que o *bagging* pode contribuir para a melhoria do desempenho de algoritmos instáveis. Por outro lado, ele pode vir a degradar ligeiramente o desempenho de algoritmos estáveis.

### *Random Forest*

Proposto por Breiman [60], *Random Forest* é um classificador composto por uma coleção de árvores de decisão. Ele pode ser descrito como um conjunto de classificadores  $\{h(\vec{x}, \theta_k), k = 1, 2, \dots, m_{tree}\}$ , em que  $\{\theta_k\}$  são vetores aleatórios independentes e identi-



camente distribuídos que determinam o crescimento das respectivas árvores de decisão e  $m_{tree}$  é o número de classificadores.

O processo de modelagem de um classificador *Random Forest* é baseado no método *bagging* descrito acima. Para crescimento das árvores, aplica-se o algoritmo CART de modo parcial, de forma que, enquanto no CART a árvore cresce até o seu limite máximo e, na sequência, é podada, no *Random Forest* as árvores crescem até o seu limite máximo e assim são mantidas, não sendo aplicado, portanto, o procedimento de poda. O Algoritmo 1 descreve de forma estruturada a sequência de passos para modelagem de um classificador *Random Forest*.

---

**Algorithm 1** Modelagem de um classificador *Random Forest*

---

**Entrada:**  $S = \{(\mathbf{x}_i, y_i) \mid i = 1, 2, \dots, n \wedge \mathbf{x} \in \mathbb{R}^d\}$  ▷ conjunto de treinamento  
 $m_{tree}$  ▷ número de subclassificadores  
 $n$  ▷ tamanho da amostra  
 $b$  ▷ número de atributos a selecionar

**Saída:**  $T = \{t_j \mid j = 1, 2, \dots, m_{tree}\}$  ▷ conjunto de árvores de decisão

- 1: **procedure** MODELAGEMRANDOMFOREST( $S, m_{tree}, n, b$ )
- 2:    $T \leftarrow \emptyset$
- 3:   **for**  $j = 1$  to  $m_{tree}$  **do**
- 4:      $s_j \leftarrow$  gera amostra de tamanho  $n$  com o método *bootstrap*
- 5:      $k_j \leftarrow$  seleciona aleatoriamente  $b$  atributos de  $d$  ▷  $b < d$
- 6:      $t_j \leftarrow$  expande árvore CART utilizando  $s_j$  e  $k_j$  ▷ não aplica poda
- 7:     insere  $t_j$  em  $T$
- 8:   **end for**
- 9:   **return**  $T$
- 10: **end procedure**

---

No momento da classificação, a nova instância a submetida a cada um dos subclassificadores do conjunto  $T$ , semelhantemente ao processo de classificação com o método *bagging*. Cada subclassificador vota para uma classe. A classe que obtiver o maior número de votos será atribuída à nova instância.

O algoritmo *Random Forest* pode ser utilizado tanto para classificação quanto para regressão, combinando características do método *bagging* e da seleção aleatório de atributos [39]. A combinação dessas características o torna mais tolerante a ruídos. Outro aspecto positivo é que classificadores *Random Forest* lidam de forma efetiva com o problema de classes desbalanceadas.

Na próxima subseção discorreremos acerca das métricas de desempenho. Apresentaremos aquelas que serão utilizadas nestes trabalho.

### 2.1.5 Métricas de Desempenho

Uma métrica de desempenho pode ser descrita como uma ferramenta que possibilita mensurar o desempenho de um classificador [61]. Métricas diferentes permitem avaliar características diferentes de um mesmo classificador, por conseguinte, a escolha da métrica mais adequada torna-se uma das questões mais importantes no processo de avaliação de um classificador.

Em geral, as métricas de desempenho são aplicadas com os seguintes objetivos: (i) avaliar a capacidade de generalização do classificador — a métrica mensura o desempenho do classificador quando aplicado sobre novas instâncias (conjunto de dados de teste) —; (ii) determinar o melhor modelo dentre os modelos gerados — o valor obtido a partir do cálculo da métrica sobre novas instâncias (conjunto de dados de validação) indicará o classificador cujo desempenho, quando aplicado a novas instâncias, acredita-se ser o melhor —; (iii) escolher a melhor solução dentre as soluções geradas durante o treinamento de um classificador — a métrica é calculada sobre o conjunto de dados de treinamento a cada modificação do classificador que está sendo treinado.

As métricas de desempenho podem ser divididas em três grupos [62]: métricas de limiar (do inglês *thresholds metrics*); métricas de ranqueamento (do inglês *rank metrics*) e métricas de probabilidade (do inglês *probability metrics*). Nas subseções abaixo, abordaremos os principais aspectos das métricas de limiar e ranqueamento, bem como apresentaremos alguns de seus integrantes.

#### Métricas de limiar

As métricas de limiar são métricas sensíveis ao limite (ou ponto de corte) cujo valor influenciará na escolha da classe a qual uma instância pertence. Isso ocorre porque a variação do limite provoca a alteração do resultado da classificação, modificando, também, o desempenho obtido para uma determinada métrica.

Para exemplificar o efeito do limite na classificação de uma nova instância, consideremos a utilização de um modelo de classificação binária, ou seja, com apenas duas classes — classes *negativa* e *positiva* —, cujo resultado esteja compreendido entre 0 e 1 e cujo limite definido seja igual 0,5. Caso o valor produzido pelo classificador seja maior ou igual a 0,5, a nova instância será associada à classe positiva, caso contrário, a nova instância será, então, associada à classe negativa. Percebe-se, portanto, que a modificação para mais ou para menos do limite pode levar a alteração da classe das instâncias cujos resultados estejam próximos a 0,5.

A fim de resumir e simplificar os resultados gerados por um classificador, costuma-se utilizar uma estrutura chamada *matriz de confusão*, também conhecida como *tabela*

de contingência [63]. Em um problema de classificação binária, em que as classes são denominadas positiva e negativa, a matriz de confusão apresenta quatro categorias:  $TP$  — instâncias positivas classificadas corretamente —;  $TN$  — instâncias negativas classificadas corretamente —;  $FP$  — instâncias negativas incorretamente classificadas como positivas —; e  $FN$  — instâncias positivas incorretamente classificadas como negativas. As métricas de limiar são calculadas a partir das categorias da matriz de confusão.

A Tabela 2.1 exibe a estrutura de uma matriz de confusão e a localização de cada uma de suas quatro categorias.

Tabela 2.1: Exemplo de uma matriz de confusão com suas quatro categorias.

		Real	
		Negativa	Positiva
Predita	Negativa	$TN$	$FN$
	Positiva	$FP$	$TP$

Existem diversas métricas de limiar, cada qual com suas características. Nesta subseção, discorreremos sobre a *acurácia*, a *precisão*, a *sensibilidade* (também conhecida como revocação ou *recall*) e a  $F_1$ -score, visto que elas serão utilizadas na avaliação dos modelos classificatórios produzidos nesta pesquisa.

#### *Acurácia*

A acurácia é descrita como a porcentagem das instâncias classificadas corretamente, a despeito da classe a que pertencem [64]. A métrica pode ser calculada por meio da seguinte equação:

$$acuracia = \frac{TP + TN}{TP + FP + TN + FN} \quad (2.15)$$

Embora seja largamente empregada na avaliação do desempenho de classificadores, ela apresenta problemas quando aplicada na avaliação de conjuntos de dados desbalanceados. Nesse cenário, a acurácia pode levar a uma interpretação equivocada acerca do desempenho do classificador.

Para exemplificar, caso o modelo sob avaliação classificasse todas as instâncias como pertencentes à classe majoritária e a proporção de instâncias da classe majoritária fosse, por exemplo, de 99%, a acurácia obtida seria também de 99%. Percebe-se que, embora o modelo não tivesse, de fato, “aprendido” a classificar novas instâncias corretamente, o desempenho aferido indicaria o contrário.

### *Sensibilidade*

A sensibilidade é descrita como a proporção de instâncias positivas classificadas corretamente [61]. Pode ser interpretada como a capacidade de um classificador categorizar corretamente todas as instâncias positivas<sup>2</sup>.

Em geral, sua utilização é feita em conjunto com a precisão, pois a sensibilidade somente não é capaz de determinar a habilidade de um classificador não categorizar uma instância reconhecidamente negativa como positiva. A métrica pode ser calculada por meio da seguinte equação:

$$\text{sensibilidade} = \frac{TP}{TP + FN} \quad (2.16)$$

### *Precisão*

A precisão é descrita como a proporção de instâncias positivas classificadas corretamente, considerando, para o cálculo, apenas o conjunto de instâncias classificadas como positivas [61]. Pode ser interpretada como a capacidade de um classificador não categorizar uma instância negativa como positiva<sup>3</sup>.

Em geral, sua utilização é feita em conjunto com a sensibilidade, pois a precisão somente não é capaz de determinar a habilidade de um classificador de atribuir corretamente a classe de todas as instâncias reconhecidamente positivas. A métrica pode ser calculada por meio da seguinte equação:

$$\text{precisao} = \frac{TP}{TP + FP} \quad (2.17)$$

### *F<sub>1</sub>-score*

*F<sub>1</sub>-score* é uma métrica que combina a precisão e a sensibilidade, sendo descrita como a média harmônica das duas métricas [62]. Seu cálculo é representado por meio da seguinte equação:

$$F_1 = \frac{2 \cdot \text{precisao} \cdot \text{sensibilidade}}{\text{precisao} + \text{sensibilidade}} \quad (2.18)$$

Sua utilização é recomendada quando se deseja avaliar o desempenho de um classificador quando aplicado sobre conjuntos de dados desbalanceados [64]. Um aspecto importante é que, diferentemente da acurácia, a *F<sub>1</sub>-score* mede o desempenho do classi-

---

<sup>2</sup>[https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall\\_score.html#sklearn.metrics.recall\\_score](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html#sklearn.metrics.recall_score)

<sup>3</sup>[https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision\\_score.html#sklearn.metrics.precision\\_score](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html#sklearn.metrics.precision_score)

ficador em relação à capacidade de predição de uma classe específica, e não de todas as classes conjuntamente, como ocorre na acurácia.

## Métricas de ranqueamento

Diferentemente das métricas de limiar, as métricas de ranqueamento dependem somente da ordem da classificação das instâncias, e não dos valores individualmente atribuídos a elas [64]. Contudo que a ordem entre as classes seja preservada, os valores preditos para as instâncias poderão variar sem, contudo, afetar o resultado do cálculo da métrica.

As métricas de ranqueamento medem, portanto, quão bem as instâncias da classe positiva estão posicionadas antes das instâncias da classe negativa. O resultado pode ser interpretado como uma agregação do desempenho do modelo para todos os limites possíveis.

Nesta subseção apresentaremos as características das métricas de ranqueamento *Área sob a Curva ROC* (AUCROC, do inglês *Area Under the Receiver Operating Characteristic Curve*) e a *precisão média* (do inglês *average precision*).

### *Área sob a Curva ROC*

Um gráfico Curva ROC é um gráfico bidimensional no qual a taxa de *TP* ( $tp_r$ ) é plotada no eixo *Y* e a taxa de *FP* ( $fp_r$ ) no eixo *X* [65]. O objetivo do gráfico é apresentar a relação entre o custo (aumento da  $fp_r$ ) e o benefício (aumento da  $tp_r$ ) decorrentes da variação do *threshold* de um classificador. Uma característica importante da Curva ROC é que ela é insensível a mudanças na distribuição de classes. Logo, se a proporção entre as classe positiva e negativa mudarem nos dados de teste, a Curva ROC não será afetada.

A Figura 2.10 apresenta um exemplo de um gráfico Curva ROC com o desempenho de três classificadores: A, B e C. Observa-se que o classificador A é superior aos classificadores B e C. O classificador B, por sua vez, é superior ao C. O classificador C poder ser considerado um classificador que atribui a classe de uma instância de forma aleatória.

A fim de tornar possível a comparação de desempenho de diferentes classificadores utilizando a Curva ROC, é necessário, primeiro, resumir a informação apresentada pelo gráfico a um valor escalar. Para isso, utiliza-se o cálculo da área sob a curva ROC, o qual sempre produzirá um valor no intervalo  $[0, 1]$ , sendo que, quanto mais próximo de um, melhor. Observa-se, contudo, que predições aleatórias produzirão uma linha diagonal entre os pontos  $(0,0)$  e  $(1,1)$ , logo, é improvável que um classificador real obtenha uma AUCROC abaixo de 0,5. A área pode ser calculada a partir da seguinte equação [62]:

$$AUCROC = \int_{t=0}^1 TPr(t)dt \quad (2.19)$$

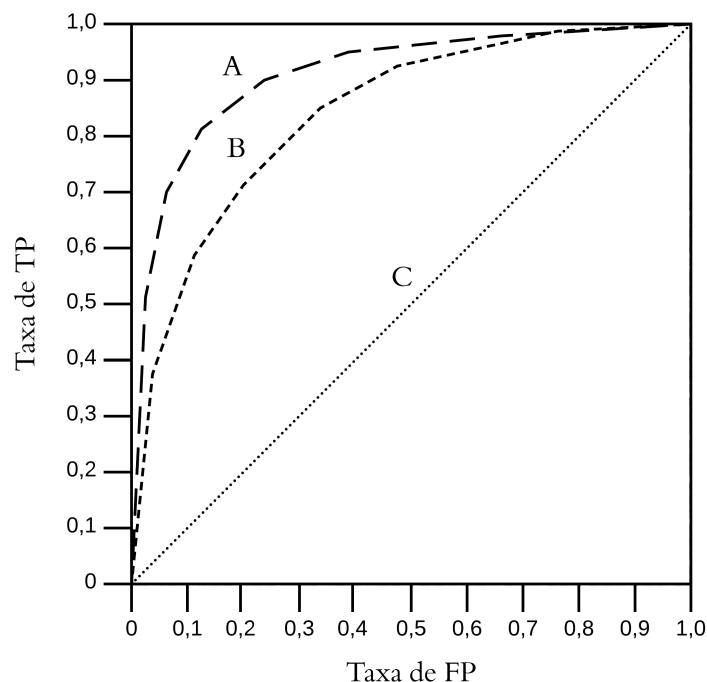


Figura 2.10: Exemplo de um gráfico Curva ROC com os desempenhos de três classificadores: A, B e C.

, em que  $t$  é igual a  $fp_r$ , ou seja,  $t = \frac{FP}{TN+FP}$ , e  $TPr(t)$  é uma função  $f : t \rightarrow tp_r$ , onde  $tp_r$  é definida por  $\frac{TP}{TP+FN}$ .

De acordo com Fawcett [65], o valor da AUCROC de um classificador é equivalente à probabilidade do classificador ranquear uma instância positiva, aleatoriamente escolhida, em posição superior a uma instância negativa — neste caso, assume-se que instâncias positivas devam assumir posições superiores às instâncias negativas.

#### *Precisão média*

Quando utilizada para avaliar um classificador em que o conjunto de dados seja fortemente desbalanceado, a Curva ROC pode apresentar uma visão muito otimista do desempenho esperado [63]. Alguns autores têm citado a adoção da curva precisão-sensibilidade, como alternativa à curva ROC, para os casos em que há classes significativamente desbalanceadas [63, 66, 67].

Um gráfico curva precisão-sensibilidade é um gráfico bidimensional em que a precisão é plotada no eixo  $Y$  e a sensibilidade é plotada no eixo  $X$  [63]. O objetivo, neste caso, é apresentar a relação entre o aumento da sensibilidade e a redução da precisão decorrentes da variação do *threshold* de um classificador. Enquanto na curva ROC os classificadores com curvas mais próximas ao canto superior esquerdo apresentam os melhores desempenhos, na curva precisão-sensibilidade isso ocorrerá quando as curvas estiverem mais

próximas do canto superior direito do gráfico.

A Figura 2.11 apresenta um exemplo de um gráfico curva precisão-sensibilidade com o desempenho de dois classificadores: A e B. Observa-se que o classificador A apresenta desempenho superior ao B. A linha horizontal pontilhada representa a linha de base.

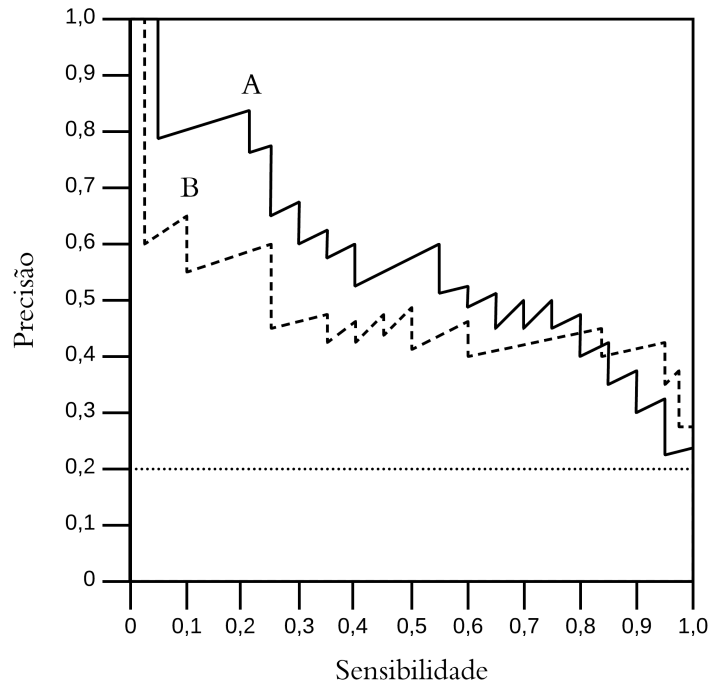


Figura 2.11: Exemplo de um gráfico curva precisão-sensibilidade com os desempenhos de dois classificadores: A e B. A linha horizontal pontilhada indica a linha de base.

Assim como ocorre com a curva ROC, é necessário resumir a informação apresentada pela curva precisão-sensibilidade a um valor escalar a fim de permitir a comparação de desempenho de diferentes classificadores. Neste caso, utiliza-se a equação da *precisão média* (AP, do inglês *average precision*) — uma das formas de aproximação da área sob a curva precisão-sensibilidade —, a qual também produzirá um valor no intervalo  $[0, 1]$ , de modo que, novamente, quanto mais próximo de 1, melhor será o desempenho. A precisão média pode ser calculada a partir da seguinte equação<sup>4</sup>:

$$AP = \sum_n P_n (S_n - S_{n-1}) \quad (2.20)$$

, em que  $P_n$  e  $S_n$  são, respectivamente, a precisão e a sensibilidade calculadas com base no  $n$ -ésimo *threshold*.

Enquanto na curva ROC a linha de base é definida pelo segmento de reta formado entre os pontos  $(0, 0)$  e  $(0, 1)$ , sendo, portanto, igual para qualquer conjunto de dados

<sup>4</sup>[https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average\\_precision\\_score.html#sklearn.metrics.average\\_precision\\_score](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html#sklearn.metrics.average_precision_score)

sob avaliação, *i.e.*, uma linha de base universal; na curva precisão-sensibilidade a linha de base é uma linha horizontal cuja constante que a define dependerá da distribuição de classe do conjunto de dados sob avaliação [68]. No exemplo apresentado na Figura 2.11, a porcentagem de instâncias da classe de interesse é de 20%.

Nesta pesquisa, utilizaremos as métricas  $F_1$ -score e precisão média para selecionar o melhor modelo. As métricas acurácia e AUCROC serão calculadas e reportadas para possibilitar a comparação dos resultados gerados nesta pesquisa com os de outras pesquisas já realizadas ou que porventura venham a ser realizadas.

Na próxima seção serão apresentados os trabalhos correlatos. O levantamento buscou identificar pesquisas semelhantes a esta com o intuito de promover a comparação dos resultados, a identificação das melhores técnicas para enfrentamento do assunto em questão e a prevenção de retrabalhos.

## 2.2 Trabalhos correlatos

De acordo com Ramos [37], em 2011 foi conduzida uma pesquisa de mineração de dados com o objetivo de identificar comunicações de ocorrência de perda indevidas relacionadas ao programa Seguro da Agricultura Familiar – SEAF –, também denominado Proagro Mais. Nesse trabalho, foi proposto o uso de técnicas de inteligência artificial (IA) para detecção de evidências de emissão de comunicados de ocorrência de perdas com base nos laudos de assistências técnica (pré-plantio, plantio e colheita). Ainda segundo Ramos [37], um dos objetivos da pesquisa foi a “proposição de método automático que filtre os laudos técnicos e selecione os que apresentem evidência de COP indevida, para posterior análise pelo *staff* da Coordenadoria do SEAF”.

A pesquisa utilizou 11.743 registros coletados entre 2006 e 2010. Dos 311 atributos coletados dos laudos, 19 foram selecionados para modelagem. Dado que a classe de interesse era desbalanceada, foram construídos quatro conjuntos de dados, além da base original desbalanceada, em que cada conjunto foi rebalanceado com diferentes proporções utilizando-se técnicas de *oversampling*. Com os conjuntos de dados gerado após o pré-processamento, foram obtidas regras de associação e modelos de classificação. As regras de associação foram geradas com base no algoritmo “Apriori” e os modelos de classificação a partir de classificadores probabilísticos *Naive Bayes*, árvore de decisão C4.5, SVM, *K-Nearest Neighbors*, *AdaBoost.M1* e árvore de inferência condicional (CTREE).

Na fase da modelagem, os classificadores foram utilizados individualmente e em conjunto (multiclassificadores). No caso dos multiclassificadores, três abordagens distintas foram aplicadas: multiclassificador em cascata homogêneo, multiclassificador disjunto heterogêneo e multiclassificador ponderado heterogêneo. Para avaliação, foram aplicados o



*cross-validation* com dez partições, o qual foi aplicado sobre o conjunto de dados total, e o método *holdout* na proporção de 70% e 30% para treinamento e teste, respectivamente.

Para determinar o classificador com melhor desempenho, foram utilizadas as métricas *F-measure* e área sob a curva ROC (AUC). O melhor resultado foi obtido com o modelo Cascata *Naive Bayes*, o qual apresentou AUC igual a 0,944, seguido dos modelos Disjuntivo e Ponderado, com AUC iguais a 0,915, e 0,909, respectivamente. Considerando apenas os classificadores individuais, o melhor modelo para a métrica AUC foi o SVM com valor igual 0,911 e, para a métrica *F-measure*, o melhor classificador foi o CTREE, com valor igual a 0,823, seguido dos modelos C4.5 e *Naive Bayes*, com valores iguais a 0,822 e 0,820, respectivamente.

Enquanto a pesquisa conduzida por Ramos [37] tratou da classificação de empreendimentos relacionados com o programa Proagro Mais, a proposta desta pesquisa é avaliar COPs dos programas Proagro e Proagro Mais conjuntamente, aumentando assim a abrangência do estudo. Outra diferença sutil, porém importante, é que o modelo proposto por Ramos busca, apoiado nos laudos, identificar os empreendimentos com maior chance de gerar COPs, enquanto esta pesquisa busca identificar, entre as COPs emitidas, aquelas com maior chance de serem consideradas ilegítimas, portanto tendo como universo de estudo apenas os empreendimentos com COPs já emitidas. Por fim, a pesquisa realizada por Ramos utilizou dados contidos nos laudos de pré-plantio, plantio e colheita na mineração de dados, ao passo que propomos utilizar, além das COPs, informações relativas às características do empreendimento e do mutuário, tais como tipo de lavoura, características geográficas da região do empreendimento, tempo entre a contratação do seguro e a comunicação de ocorrência de perda, etc., bem como fontes de dados externas que tratem de aspectos climáticos.

Destacamos que para esta pesquisa não serão utilizados dados referentes aos laudos técnicos descritos na pesquisa de Ramos visto que (i) os laudos foram coletados apenas para os empreendimentos do programa Proagro Mais e nesta pesquisa avaliaremos COPs de empreendimentos tanto do Proagro Mais quanto do Proagro, (ii) os laudos representam apenas 5% do total de empreendimentos do programa ProagroMais e desejamos construir um modelo capaz de avaliar todas as COPs registradas no Sicor cujas características atendam aos requisitos impostos pelo modelo produto da pesquisa, e (iii) esses laudos não estão disponíveis — provavelmente não estão sendo mais coletados.

Em [14], Rejesus *et al.* propõem ilustrar a utilidade da mineração de dados como ferramenta de detecção de potenciais fraudes, abusos ou excessos no programa de seguro agrícola dos Estados Unidos. Os autores afirmam que técnicas de mineração de dados para conjuntos de dados extremamente grandes e complexos estão em uso em diversos setores da economia. Tecnologias de mineração estão sendo largamente utilizadas em

áreas comerciais, tais como *marketing*, controle de fraude em cartões de crédito, controle de fraude em seguros, análise de crédito e controle de fraude em serviços de comunicação móvel.

Legisladores, criadores de políticas públicas e agências federais de seguro agrícola estadunidenses têm reconhecido o potencial das ferramentas de mineração de dados na detecção de comportamentos atípicos que podem sugerir a existência de fraude no Programa de Seguro Agrícola dos Estados Unidos — *US Crop Insurance Program*. Criadores de política incluíram textos explícitos no *Agricultural Risk Protection Act* declarando que técnicas de mineração devem ser empregadas no auxílio à implementação das disposições antifraude da lei.

Para exemplificar a aplicação de mineração de dados em detecção de anomalias em seguros agrícolas, o autor propôs a realização de uma prova de conceito. A prova de conceito consistiu da criação de um indicador e da seleção de amostras para verificação mais aprofunda a partir desse indicador. Para criação do indicador, foi utilizado um conjunto de dados contendo as informações acerca da produção de milho por município do país entre os anos de 1972 a 2000, o qual continha as seguintes variáveis: estado, município, área administrativa, ano, tipo do plantio, tipos de cultivo, área plantada e área colhida. Primeiro, foram removidas as instâncias duplicadas ou contendo dados faltantes ou evidentemente errados, restando um total de aproximadamente 1,1 milhão de registros. Em seguida, foi criado o novo indicador a partir da divisão da variável “área colhida” pela variável “área plantada”.

Uma vez criado o novo indicador, foram selecionados os municípios que se encontravam abaixo do 5º percentil, o que representa os municípios com a menor razão entre a área colhida e a área plantada dentre todos os municípios. A partir dessa amostra, foram realizadas análises mais profundas a fim de identificar os motivos que poderiam justificar a diferença na produção desses municípios em relação aos demais. As análises não foram conclusivas porque, segundo o autor, seriam necessários dados mais granulares capazes de destacar diferenças na produção por fazenda, e não por município somente.

Embora não tenha sido possível determinar a existência de fraude diretamente a partir dos procedimentos descritos acima, os autores reconhecem que a identificação de municípios com produção atípica permite as agências federais de seguro alocarem de modo mais eficiente recursos na detecção de fraudes, abusos e desperdícios porquanto, ao invés de adotarem amostragem aleatória na seleção de segurados para inspeção, é possível, com base em técnicas de mineração, selecionar entidades com comportamento atípico *a priori*.

Em [69], Kuwata *et al.* propõem construir um índice climático que resulte em um baixo risco de base para os seguros agrícolas (*hedge*) que se baseiam em índices climáticos. Por meio do levantamento de dados climáticos oriundos de diversas fontes e da associação

e correlação desses dados com os índices de produção agrícola para diversos períodos, o estudo buscou identificar os efeitos de variações climáticas severas na lavoura, de modo a alcançar maior acurácia nas estimativas de perdas e ganhos decorrentes dessas variações.

Além disso, a fim de detectar inundações a partir de sensoriamento remoto, foi desenvolvido um modelo com base no algoritmo SVM combinado com a função de bases radial (RBF). Foram utilizados índices *Enhance Vegetation Index* (EVI), *Land Surface Water Index* (LSWI) e *Difference value between EVI and LSWI* (DVEL) e um mapa de inundação da região do Paquistão para o ano de 2010 como conjunto de dados de treinamento.

Embora seguros baseados em índices climáticos não estejam relacionados com o objeto desta pesquisa e muitos dos dados climáticos utilizados no estudo não estejam disponíveis (imagens e dados de sensores obtidos a partir de satélites), o estudo apresenta como um dos resultados a identificação de correlação entre o índice pluviométrico e as temperaturas máximas e mínimas, e o resultado da produção agrícola para a região sob observação (Estado de Illinois, Estados Unidos). Visto que a EMBRAPA coleta dados pluviométricos e de temperatura em diversos pontos do país, é possível tentar correlacionar esses dados com os dados da produção agrícola de cada região do país. Caso seja identificada correlação ou associação entre as variáveis, incluir-se-ão os dados climáticos nos modelos classificatórios.

De acordo com o estudo publicado por Cassimiro *et al.*[38], muitas companhias brasileiras de plano de saúde estão em dificuldades financeiras devido à superação dos custos assistenciais com relação às receitas. Os fatores que mais contribuem para essa situação são a fraude e a utilização excessiva (abusiva). Esse tipo de prática não está circunscrito ao Brasil. Nos Estados Unidos, por exemplo, estima-se que, no ano fiscal de 2009, cerca de 3% a 10% dos gastos dos setores público e privados com saúde estavam relacionados a fraude, o que representa um dispêndio entre US\$ 75 e US\$ 250 bilhões.

A fim de mitigar prejuízos, as companhias de plano de saúde adotam mecanismos que buscam prevenir a ocorrência de despesas indevidas, como condicionamento de autorização de uso a análise prévia das requisições de serviço por um analista. Todavia, esse tipo de procedimento apresenta elevado custo e, como consequência, acaba por induzir as companhias a empregarem técnicas de aprendizagem de máquina na detecção automática de fraudes e abusos.

A utilização de aprendizagem de máquina no processo de verificação da requisição de autorização é afetada pelo fato de haver mais requisições autorizadas que não autorizadas, o que gera um problema de classes desbalanceadas. Em razão disso, o estudo propõe investigar quão afetado é o desempenho dos classificadores em razão da presença de classes desbalanceadas. Para isso, quatro aspectos foram avaliados: (i) como o desempenho da

predição é afetado na presença de diferentes distribuições de classe e algoritmos de classificação; (ii) qual algoritmo de classificação é mais afetado por classes desbalanceadas; (iii) quanto da perda de desempenho pode ser recuperada mediante aplicação de métodos de tratamento de classes desbalanceadas; e (iv) como as diferentes combinações de algoritmos de classificação e métodos de tratamento de classes desbalanceadas influenciam o desempenho de predição.

No experimento, foram utilizados três bases de dados reais e desbalanceadas contendo requisições médicas e odontológicas. Os algoritmos de classificação avaliados foram *RIPPER*, *C4.5*, *Random Forest*, *Naive Bayes* e *SVM*. Os métodos de tratamento de dados desbalanceados aplicados e analisados foram o *Random Oversampling*, o *SMOTE* e o *MetaCost*. O experimento avaliou o desempenho para diferentes proporções de cada classe, variando de 1:99 até 99:1, intervalos de dez (1:99, 10:90, 20:80, etc.).

Os resultados mostram que todos os algoritmos de classificação aplicados no experimento são afetados por classes desbalanceadas, mesmo que de formas diferentes. Os algoritmos *RIPPER*, *C4.5* e *SVM* apresentaram valores de perda de desempenho mais elevados que *Random Forest* e *Naive Bayes*, e, de modo geral, o percentual de recuperação de desempenho fornecida pelos métodos de tratamento de classe desbalanceada tendem a diminuir conforme a proporção de desbalanceamento de classe aumenta. Por último, observou-se benefício na combinação do método *Random Oversampling* com os algoritmos *RIPPER*, *C4.5* e *SVM*; ao passo que não se observou diferença estatisticamente significativa com a adoção de conjuntos de dados tratados ou não tratados para o algoritmo *Random Forest*.

Ainda que o estudo não trate de fraude ou abusos em instrumentos de seguro, o tema discutido assemelha-se ao desta proposta pois aborda a utilização de sistemas de saúde por meio de planos de saúde, os quais apresentam características semelhantes aos seguros agrícolas, por exemplo. Entres as características semelhantes, podemos destacar a existência de classes desbalanceadas, visto que, tanto na comunicação de perda de um seguro agrícola quanto na requisição de um serviço de saúde, as classes comportam-se de forma desbalanceada, conseqüentemente afetando o desempenho de algoritmos de classificação.

No estudo elaborado por Tukman *et al.* [70], propõe-se um sistema de detecção de fraude com cartões de crédito baseado na abordagem *Relative to an Identified Distribution* (RIDIT). De acordo com os autores, a maioria dos métodos supervisionados e não supervisionados empregados na detecção de fraude são dependentes da estrutura dos dados, não sendo, em geral, adequados para lidar com dados categóricos ordenados.

Uma forma de contornar o problema pode se dar por meio de aplicação do método RIDIT. O método consiste na transformação do conjunto de valores da uma variável de

resposta ordenada em uma escala de probabilidade, de modo a tornar-se mais apropriada à aplicação e à análise de testes estatísticos. A transformação é realizada por meio do cálculo do *RIDIT score* da variável de resposta. Os resultados do cálculo do *RIDIT score* se assemelham aos resultados obtidos de uma função de distribuição acumulada.

Por meio da aplicação do teste qui-quadrado sobre as médias RIDIT, as quais foram calculadas para diferentes grupos de referência, foi possível determinar os valores da variável resposta “Histórico de Crédito” - utilizada para descrever o histórico de pagamento do requerente - que indicam maior potencial de fraude. Em razão disso, o estudo conclui que bancos e instituições financeiras podem utilizar a análise RIDIT como ferramenta de detecção de fraude.

Segundo Li *et al.* [39], detecção de fraude em seguros depende fortemente de inspeção especializada e de auditoria. Tais formas de detecção são realizadas manualmente, revelando-se custosas e ineficientes. Técnicas de mineração de dados, por sua vez, têm o potencial de detectar, de modo tempestivo, casos de fraude suspeitos, reduzindo significativamente perdas econômicas de companhias de seguro.

A partir dessa avaliação, os autores do estudo propuseram um novo sistema de múltiplos classificadores baseado nos algoritmos *Random Forest*, PCA e *Potential Nearest Neighbor* (PNN). Objetivo do sistema é melhorar a acurácia do modelo na classificação de instâncias obtidas a partir de conjuntos de dados desbalanceados. A ideia por trás da proposta é que classificadores baseados em *Random Forest* podem resolver, de forma efetiva, problemas de classificação envolvendo bases de dados desbalanceadas, o que, em geral, ocorre com dados relacionados à reivindicação de seguro de automóveis.

O sistema foi construído a partir de técnicas de *ensemble*, *bagging* mais especificamente, aplicadas na combinação de classificadores derivados de árvore de decisão do tipo CART (do inglês *Classification And Regression Tree*), da aplicação de PCA imediatamente antes do treinamento de cada classificador e da utilização do PNN como mecanismo de votação, em substituição ao tradicional voto majoritário. Os resultados mostram que o algoritmo proposto obteve melhor desempenho que o tradicional Random Forest ao ser aplicado à um conjunto de dados de reivindicação de seguro automobilístico. Os dados foram obtidos de uma companhia de seguros da China e constituem seguros reivindicados no ano de 2015.

Ainda que seguro automobilístico não guarde relação direta com seguro agrícola, ambos apresentam semelhanças no contexto deste trabalho, especialmente no que concerne a existência de classes desbalanceadas. Em razão disso, é possível concluir que a aplicação de técnicas e algoritmos iguais ou semelhantes aos aplicados pelos autores podem produzir resultados parecidos, como, por exemplo, a obtenção do ranqueamento dos atributos em função de sua capacidade explicativa – quanto melhor estiver classificado maior será sua

influência na detecção de fraude.

# Capítulo 3

## Método proposto

Este capítulo apresenta a metodologia e os métodos adotados neste projeto de pesquisa. A pesquisa foi conduzida em fases, seguindo a estrutura proposta pelo modelo de referência CRISP-DM. As atividades da pesquisa foram compartimentadas nas seguintes fases: entendimento do negócio, compreensão dos dados, preparação dos dados e modelagem. A Figura 3.1 apresenta o relacionamento entre as atividades a serem executadas durante a pesquisa.

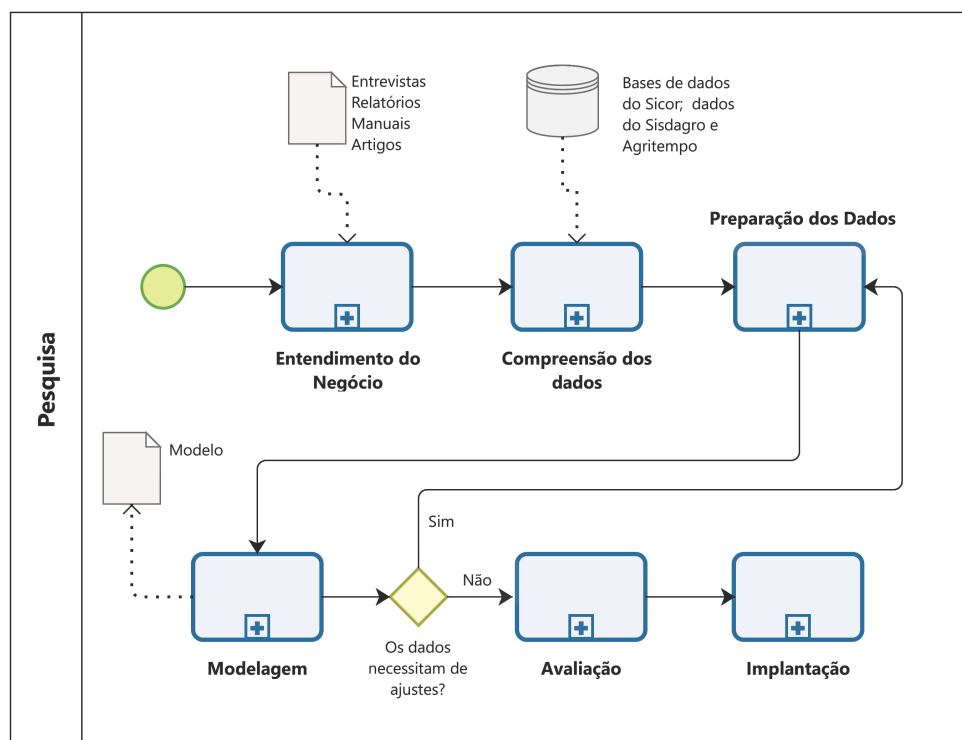


Figura 3.1: Diagrama com as fases do modelo CRISP-DM executadas nesta pesquisa.

## **Entendimento do negócio**

Na fase de compreensão do negócio são levantadas informações relativas ao Proagro e sua gestão. Esse levantamento envolve a realização de entrevistas e coleta de documentos tais como manuais de procedimento, relatórios de auditorias, relatórios de gestão, etc. São também realizadas pesquisas bibliográficas a fim de identificar artigos publicados que tratam do assunto.

É também nesta fase que se definem os resultados esperados para o projeto. A área de negócio responsável pela avaliação e operacionalização da solução deverá estabelecer o desempenho mínimo a ser alcançado para que a solução que possa ser posta em produção.

## **Compreensão dos dados**

O processo de compreensão dos dados tem início com acesso ao sistema Sidor. A partir da visualização de suas telas e campos, é possível inferir quais informações são coletadas e geradas pelo sistema. Uma vez ambientados ao sistema, passa-se a analisar o modelo de dados que contém a representação das entidades que compõem o sistema. Por meio da análise do modelo de dados, é possível identificar os relacionamentos entre as entidades e os tipos de dados contidos nos campos.

Em seguida, são analisados os dados dos sistemas Sisdagro e Agritempo. Durante a análise, são identificados os conteúdos dos campos dos sistemas e seus significados no contexto de negócio de cada sistema e, posteriormente, no contexto do Proagro. Aqui, ocorre também a avaliação da forma de integração dos dados dos diferentes sistemas.

Ao final da análise, são, então, determinadas quais serão as tabelas e os respectivos campos a serem extraídos. Uma vez definido o que deve ser coletado, extraem-se os dados que serão utilizados na próxima etapa.

## **Preparação dos dados**

Na fase de preparação dos dados, são aplicadas as transformações sobre os dados coletados na fase anterior. As atividades executadas nesse momento são: limpeza, remoção de dados duplicados ou faltantes, atribuição de valores para os casos de dados faltantes, criação de novos atributos, seleção de atributos, remoção de colinearidade, projeção e redução de dimensionalidade, balanceamento de classes, integração dos conjuntos dados, etc. Como produto desta fase, obtém-se um conjunto ou conjuntos de dados próprios para modelagem.



## **Modelagem**

Durante a fase de modelagem, algoritmos de aprendizagem de máquina são aplicados sobre o conjunto de dados de treinamento pré-processados. Para cada algoritmo, os hiperparâmetros são ajustados adotando-se a estratégia cartesiana, na qual todas as combinações possíveis para um dado conjunto de hiperparâmetros são testadas.

Ainda nesta fase, são revistos os procedimentos realizados até a construção do modelo com o objetivo de identificar possível necessidade de ajustes e, conseqüentemente, de execução de novas iterações. Ao final do processo, os modelos que apresentaram os melhores desempenhos na validação cruzada, segundo as métricas escolhidas, serão selecionados para análise na fase seguinte.

## **Avaliação**

A avaliação consiste na análise e validação do modelo gerado e selecionado na fase anterior sob a perspectiva dos objetivos de negócio. Aspectos como eficácia e escalabilidade do modelos são analisados para determinar sua viabilidade. As “descobertas” são exploradas e, posteriormente, estabelece-se conclusões.

## **Implantação**

Por fim, na fase de implantação são apresentados os resultados às partes interessadas do projeto. Após avaliação das partes interessadas, são definidos os próximos passos para implantação do modelo e a forma como será conduzida sua manutenção.

As próximas seções detalham as avaliações e ações realizadas em cada uma das fases descritas acima, com exceção das fases de avaliação e de implantação.

## **3.1 Entendimento do negócio**

Conforme descrito no Capítulo 1, o Proagro foi criado em 1973 com o objetivo de desonerar o produtor rural do cumprimento de obrigações financeiras originadas da contratação de crédito rural, para os casos em que houvesse perda de receita decorrente de eventualidades que comprometessem a lavoura, tais como pragas, doenças, estiagens prolongadas e chuvas excessivas. Em 1991, por meio da Lei nº 8.171/91, ficou estabelecido que cabe ao Banco Central do Brasil a administração do Proagro, devendo exercê-la em conformidade com as normas, os critérios e as condições definidas pelo Conselho Monetário Nacional (CMN).

Por meio dessa mesma lei, estabeleceu-se que cabe aos agentes financeiros autorizados a atuar em crédito rural a operação do Proagro. Entre as competências definidas na lei, destacamos as seguintes: formalizar a adesão do mutuário ao Programa; realizar as

análises dos pedidos de cobertura; e decidir sobre os pedidos de cobertura (centrais de análise). A decisão sobre o pagamento ou não de uma COP deve ser feita com base nos relatórios de comprovação de perda, os quais são emitidos por peritos contratados pelo próprio agente do Proagro, conforme determina o MCR. Cabe destacar que a remuneração do perito/técnico é feita com recursos do programa, embora o pagamento seja efetuado pelo agente.

A fim de garantir o cumprimento do regulamento do Proagro e mitigar o risco de pagamento indevido de COPs — entre outras coisa —, foi criada uma divisão de monitoramento das operações do Proagro. As atribuições do monitoramento envolvem a análise das informações existentes nos sistemas do BCB e de informações disponíveis em fontes externas, criação de rotinas de monitoramento e geração de sinalizações. As sinalizações, cujo destinatário é a área de supervisão do Proagro, devem apontar possíveis irregularidades ou desvios. Espera-se, com o monitoramento, tornar a atuação da supervisão mais tempestiva, bem como direcionar seus esforços.

Nesse contexto, consideramos que as técnicas de aprendizagem de máquina podem alavancar os resultados do monitoramento. A aplicação de técnicas avançadas de aprendizagem de máquina sobre os dados gerados a partir da operacionalização do Programa, combinados com outras fontes de dados cujo conteúdo seja relevante para caracterização de um evento em que há ocorrência de perdas na lavoura, pode levar à criação de um modelo classificatório capaz de identificar irregularidades nos pedidos de pagamentos de coberturas.

Por meio de entrevista com integrantes da divisão de monitoramento, fomos informados de qual seria o desempenho esperado de um classificador que pudesse contribuir com a alcance dos objetivos acima definidos. Segundo nos foi reportado, o modelo dever ser capaz de identificar COPs irregulares com precisão igual ou superior a 80%. Em função disso foram conduzidas as ações desta pesquisa, especialmente quanto à preparação dos dados e à seleção do modelo ou modelos finais.

Na próxima seção descreveremos as fontes de dados utilizados neste trabalho, seu conteúdo e as ações necessárias para sua obtenção.

## **3.2 Compreensão dos dados**

Nesta seção discutiremos acerca das bases do Sicor e dos dados disponibilizados no portal do Agritempo e no portal do Sisdagro. Essas foram os sistemas das quais as bases foram extraídas para formarem o conjunto de dados utilizado na fase de modelagem.

### 3.2.1 Sicor

Desde a contratação do crédito rural pelo mutuário até o pagamento de uma COP, todas as operações são registradas no Sicor. É por meio do Sicor que os analistas do BCB acompanham os lançamentos e interagem com os agentes do Proagro. Entre as informações inseridas no Sicor, temos: dados das operações de crédito, características do empreendimento, dados da COP, conteúdo parcial dos relatórios de comprovação de perdas e dados das súmulas de julgamento. As informações são inseridas pelos agentes financeiros do Proagro.

Para entendimento dos dados disponíveis no Sicor e posterior seleção/transformação, acessamos o sistema e passamos a percorrer seus campos. Uma vez que estávamos interessados na classificação de COPs, buscou-se identificar todos os dados relacionados às COPs que porventura pudessem contribuir no processo de predição, ou seja, que fossem contextualmente informativos.

O resultado da investigação levou à identificação de 40 variáveis candidatas, as quais foram distribuídas em três grupos. Os grupos estão relacionados aos seguintes aspectos do programa: empreendimento — onde são armazenadas as informações do financiamento, como juros, valor do crédito, valor do aportado pelo mutuário etc., e os parâmetros do empreendimento, como produto a ser cultivado, programa ao qual o empreendimento está vinculado, previsão de produção etc. —; comunicação de ocorrência de perda — onde são registrados os dados iniciais de uma COP, como data da COP, evento amparado, etc. —; relatório de comprovação de perda — o qual contém os dados levantados pela periciadora, tais como evento da COP, perda de qualidade da produção, data ou período do evento etc. As Tabelas 3.1 a 3.3 listam as variáveis identificadas e apresentam a descrição, o tipo e um exemplo de valor possível de cada variável.

Tabela 3.1: Variáveis do empreendimento

Nome	Descrição	Tipo	Exemplo conteúdo
emp_programa	Programa ao qual o empreendimento está vinculado	Nominal	Pronaf
emp_localidade	Município do empreendimento	Nominal	Cod. do município 04512
emp_empreendimento	Produto a ser cultivado	Nominal	Milho
emp_rct_brt_esperada	Receita bruta esperada	Numérica	R\$ 65.000,00
emp_area	Área do empreendimento (ha)	Numérica	30,05
emp_previsao_produc	Previsão de produção	Numérica	178,55
emp_tipo_cultivo	Tipo de cultivo/exploração	Nominal	Plantio direto
emp_tipo_irrigacao	Tipo de irrigação (quando aplicada)	Nominal	Aspersão
emp_vl_parc_credito	Valor da parcela de crédito	Numérica	R\$ 50.000,00
emp_juros	Juros aplicado à parcela de crédito	Numérica	5,50%
emp_vl_rec proprio	Valor do recurso próprio	Numérica	R\$ 3.000,00
emp_vl_rec proprio_serv	Valor do recurso próprio gasto com serviços	Numérica	R\$ 2.000,00
emp_dt_emis_ced_cred	Data da emissão da cédula de crédito	Data	01/02/2018
emp_mut_sexo	Sexo do mutuário	Nominal	Feminino
emp_mut_possui_dap	Possuir Declaração de Aptidão ao Pronaf	Nominal	Sim

Tabela 3.2: Variáveis da COP

Nome	Descrição	Tipo	Exemplo conteúdo
cop_dt_cop	Data da COP	Data	22/05/2018
cop_grp_ciclo_cultivar	Grupo do ciclo cultivar	Nominal	Grupo I
cop_tipo_solo	Tipo do solo	Nominal	Solo Argiloso
cop_dt_ini_evento	Data de início do evento	Data	01/01/2018
cop_dt_fim_evento	Data de término do evento	Data	07/01/2018
cop_evento	Descrição do evento	Nominal	Seca
cop_status	Status da COP (classe de interesse)	Nominal	Indeferida

Tabela 3.3: Variáveis do relatório de comprovação de perdas

Nome	Descrição	Tipo	Exemplo conteúdo
rcp_dt_ini_colheita	Data de início da colheita	Data	25/02/2018
rcp_dt_fim_colheita	Data de término da colheita	Data	28/02/2018
rcp_dt_ini_plantio	Data de início do plantio	Data	09/09/2017
rcp_dt_fim_plantio	Data de término do plantio	Data	11/09/2017
rcp_dt_ini_evento	Data de início do evento	Data	12/12/2017
rcp_dt_fim_evento	Data de término do evento	Data	14/12/2017
rcp_evento	Evento da COP	Nominal	Seca
rcp_perda_qualid	Perda da qualidade	Nominal	Sim
rcp_prod_estimada	Produção estimada	Numérica	0,00
rcp_rct_estimada	Receita estimada	Numérica	0,00
rcp_dt_vst_perito	Data da visita do perito	Data	26/12/2018
rcp_dt_entrega	Data da entrega do relatório	Data	28/12/2018
rcp_area_medida	Área do empreendimento medida pelo perito (ha)	Numérica	27,00
rcp_ciclo_cultivar	Ciclo cultivar em dias	Numérica	90
rcp_loc_gps	Pontos limítrofes da localização do empreendimento (Gleba)	Contínua (mult. pontos)	Latitude/Longitude/Altitude -27,01.../-53,01.../440,1 -27,02.../-53,01.../440,5

A seleção preliminar das variáveis ocorreu com base em consultas a integrantes da área de monitoramento, os quais já utilizam algumas dessas variáveis na criação de sinalizações. Além disso, a seleção baseou-se também em consultas a publicações que tratam do assunto, principalmente a pesquisa elaborada por Ramos, e em julgamento do pesquisador. Ramos, em sua pesquisa, apresentou diversas variáveis já validadas por especialistas e filtradas por meio de métodos estatísticos. Algumas dessas variáveis integram o conjunto de variáveis selecionadas aqui, por exemplo, *emp\_previsao\_produc*, *rcp\_prod\_estimada*, *rcp\_area\_medida*, *rcp\_ciclo\_cultivar*, *emp\_empreendimento*, *emp\_tipo\_cultivo* e *rcp\_evento*. Contudo, é importante observar que algumas variáveis utilizadas na pesquisa de Ramos não estão disponíveis para esta pesquisa e, por isso, não foram utilizadas. Outro exemplo de variável selecionada com base em publicações é a *emp\_mutsexo*. Algumas pesquisas [71, 72] informam que o sexo do indivíduo poder ser relevante na detecção de fraude.

Algumas variáveis apresentadas nas Tabelas 3.1 a 3.3 não foram utilizadas diretamente na modelagem. Elas ou foram utilizadas para captura de informações externas, como as variáveis *rcp\_loc\_gps*, *rcp\_dt\_ini\_evento* e *rcp\_dt\_fim\_evento*, ou foram combinadas

com outras para criação de uma nova variável. A Seção 3.3 trará mais detalhes dessas transformações.

Uma vez definidas as variáveis (campos) a serem recuperadas do Sicor, procedeu-se à coleta dos registros do sistema. Foram, então, extraídos 56.513 registros relativos a COPs emitidas entre janeiro de 2017 e janeiro de 2019. Não foram coletados registros anteriores a 2017 em razão de alguns campos utilizados na pesquisa não serem de preenchimento obrigatório até essa data, o que pode comprometer a qualidade dos dados.

Os dados revelaram que a distribuição de classes da variável de interesse (*cop\_status*) é desbalanceada. Dos 56.513 registros coletados, apenas 6.525 são “indeferidas”, o que representa 11,55% do total de COPs. Os outros 49.988 registros são de COPs “deferidas”.

Outro aspecto relevante observado a partir dos registros é a distribuição dos eventos de COP. A maior parte dos registros estão concentrados em apenas dois eventos, “seca” e “chuva excessiva”, que juntos representam 78,58% do total de COPs. Ao todo, são onze tipos de eventos, sendo dez relacionados a fenômenos climáticos — “seca”, “chuva excessiva”, “geada”, “vento forte”, “granizo”, “variação excessiva de temperatura”, “vento frio”, “vendaval”, “tromba d’água” e “chuva na colheita” — e um associado a fenômenos biológicos — “doença ou praga”. Essa informação deverá, mais à frente, guiar o processo de preparação dos dados, quando devermos decidir quais instâncias deverão ser utilizadas na modelagem, ou mais especificamente, para quais categorias de eventos o classificador deverá ser construído.

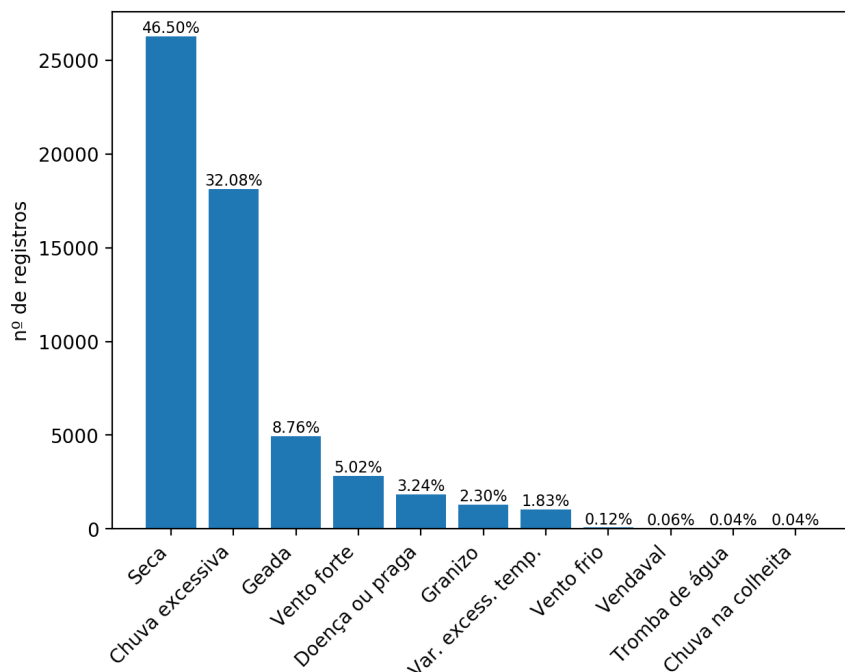


Figura 3.2: Distribuição das COPs por tipo de evento.

Como dito pouco acima, foram coletadas outras variáveis oriundas de fontes de dados externas cujo conteúdo julgou-se poder contribuir na predição. As subseções a seguir detalham os sistemas dos quais os dados foram extraídos e o processo de extração/agrupamento desses dados.

### 3.2.2 Agritempo

De acordo com o portal<sup>1</sup> do Sistema de Monitoramento Agrometeorológico, o “Agritempo é um sistema que permite aos usuários o acesso, via Internet, às informações meteorológicas e agrometeorológicas de diversos municípios e estados brasileiros. Além de informar a situação climática atual, o sistema alimenta a Rede Nacional de Agrometeorologia (RNA) do Ministério da Agricultura, Pecuária e Abastecimento (MAPA) com informações básicas que orientam o zoneamento agrícola brasileiro”.

O sistema contém dados climáticos (temperaturas e precipitação) capturados por centenas de estações meteorológicas espalhadas pelo país. Os dados são enviados diariamente por diversas instituições e, ao serem recebidos, são validados e armazenados nas bases de dados.

De acordo com o Manual do Agritempo [73], o sistema é provido com dados de mais de 1.400 estações meteorológicas distribuídas pelo Brasil. As estações estão concentradas na parte longitudinal leste do Brasil, conseqüentemente, deixando a região noroeste com uma densidade de estações significativamente menor. Isso leva à necessidade de complementariedade das informações, a qual tem sido alcançada a com utilização de imagens de satélites provenientes do sistema *Tropical Rain Meteorological Mission* (TRMM).

O sistema TRMM provê dados de 11.332 pontos de grade relativos a nuvens, precipitações, fluxo de calor, raios solares etc., a partir dos quais são criadas as estações virtuais. Para estimar a temperatura onde estão localizadas as estações virtuais, são utilizadas informações provenientes das estações de superfície mais próximas. Os dados do TRMM também são utilizados para preenchimento de dados faltantes de chuva. A Figura 3.3 apresenta a disposição geográfica das estações meteorológicas reais e das estações virtuais.

Ainda de acordo com o manual, o processamento e interpolação dos dados são feitos utilizando-se o método de krigagem ordinária. A combinação dos dados de superfície com dados de satélites permite uma interpolação mais segura, robusta e confiável.

Para esta pesquisa, estamos interessados nos dados climáticos disponibilizados pelo Agritempo, a saber: temperatura máxima diária, temperatura mínima diária, temperatura média diária e precipitação média diária. Essas informações serão utilizadas na construção de novas variáveis que comporão o vetor de características.

---

<sup>1</sup><https://www.agritempo.gov.br/agritempo/sobre.jsp>



(a) Rede de estações físicas (superfície, convencionais e automáticas).



(b) Rede de estações virtuais.

Figura 3.3: Distribuição geográfica das (a) estações físicas (pontos vermelhos) e (b) virtuais (pontos verdes). Fonte: Manual do Agritempo.

A coleta dos dados climáticos teve início com a cópia das informações das estações registradas no portal do Agritempo. O endereço para download utilizado foi o [https://www.agritempo.gov.br/agritempo/controlador?objeto=Estacao&acao=Indice&siglaUF=sigla\\_uf](https://www.agritempo.gov.br/agritempo/controlador?objeto=Estacao&acao=Indice&siglaUF=sigla_uf)<sup>2</sup>. A consulta retorna um conjunto de informações, em formato *JSON*, para cada estação. Entre as informações retornadas, temos o *id* da estação, a localização geográfica — definida por meio da longitude e latitude —, a altitude em que se encontra a estação e a situação da estação (ativa ou desativada). Foram baixados os dados das estações de todos os estados do país.

Na sequência, para cada COP extraída do Sicor, foram calculadas as seguintes variáveis: *cli\_temp\_max* — temperatura máxima —; *cli\_temp\_med* — temperatura média —; *cli\_temp\_min* — temperatura mínima —; *cli\_precip\_med* — precipitação média —; *cli\_precip\_med\_max* — precipitação média diária máxima. O cômputo das variáveis combina valores diários de temperatura e precipitação, considerando, para isso, o período de ocorrência do evento.

Descreveremos os passos utilizadas para determinar *cli\_temp\_max* como forma de ilustrar o processo de cômputo das variáveis. Primeiro, calcula-se o centro de massa da gleba e o adiciona ao conjunto de coordenadas que definem seu limite. No segundo passo, recuperam-se os dados climáticos das estações que estão a um raio de 37 km das coordenadas. O valor do raio foi definido com base no estudo de Campanhola *et al.*, onde foi verificado que o alcance (*range*) do semivariograma para precipitação é de aproximadamente 37 km. No terceiro passo, estima-se a temperatura máxima para as coordenadas

<sup>2</sup>a expressão *sigla\_uf* deve ser substituída pela sigla do estado no qual se encontram as estações



da gleba utilizando as temperaturas máximas coletadas das estações<sup>3</sup>. A estimativa é feita com o uso de interpolação por krigagem ordinária. O segundo e terceiro passos são repetidos para todos os dias do período de duração do evento. O último passo consiste na identificação da maior temperatura estimada. Esse será o valor atribuído á variável *cli\_temp\_max*. O Algoritmo 2 apresenta, de forma estruturada, os passos para o cômputo de *cli\_temp\_max*.

---

**Algorithm 2** Cálculo da variável *cli\_temp\_max* par uma gleba

---

**Entrada:**

$G = \{(x_g, y_g) \mid (x_g, y_g) \in rcp\_loc\_gps\}$       ▷ conjunto de coordenadas que definem os limites da gleba

$E = \{(x_e, y_e, temp\_max, data\_medicao) \in EstacoesAgritempo\}$       ▷ conjunto em que cada elemento contém as coordenadas da estação, a temperatura máxima e a data da coleta

*dataInicio*      ▷ data de início do evento

*dataFim*      ▷ data final do evento

**Saída:**

*cli\_temp\_max*      ▷ temperatura máxima na gleba

```

1: procedure CALCULATEMPMAX(G, E, dataInicio, dataFim)
2:   Temp_max ← ∅
3:   (x_c, y_c) ← calcula o centro de massa da gleba
4:   insere (x_c, y_c) em G
5:   for data = dataInicio to dataFim do
6:     for (x_g, y_g) ∈ G do      ▷ percorre todos os pontos de G
7:       TempCoor ← {(temp_max, (x_e, y_e)) | dist((x_g, y_g), (x_e, y_e)) ≤ 37km ∧
         data_medicao = data}      ▷ dist() é a distância
8:       temp ← interpolaComKrigagem(TempCoor, (x_g, y_g))
9:       insere temp em Temp_max
10:    end for
11:  end for
12:  return max(Temp_max)      ▷ retorna a maior temperatura em Temp_max
13: end procedure

```

---

O cálculo das demais variáveis segue a mesma sequência de passos adotada no cômputo de *cli\_temp\_max*. A diferença reside nas informações utilizadas para estimação e na função utilizada no passo 12 do Algoritmo 2. Para estimar *cli\_temp\_med*, foi utilizada a temperatura média diária e a função aplicada foi a média. Já para estimar *cli\_temp\_min*, foi utilizada a temperatura mínima diária e a função mínimo, a qual retorna o menor valor de um conjunto. Por fim, para estimar *cli\_precip\_med* e *cli\_precip\_med\_max* foi

---

<sup>3</sup>Exemplo de URL de coleta: <https://www.agritempo.gov.br/agritempo/controlador?objeto=ClimaDiario&acao=PesquisaWeb&idEstacao=9000454&dataInicial=2017-01-01&dataFinal=2017-01-02>

utilizada a precipitação média diária e foram aplicadas, respectivamente, a média e o máximo.

Visto que o número de coordenadas geográficas por gleba varia de 4 a 1.246, foi utilizado o algoritmo *Ramer–Douglas–Peucker* para reduzir o número de pontos e, consequentemente, atenuar o custo computacional envolvido na coleta das temperaturas e precipitação e no cálculo de interpolação.

O algoritmo de simplificação de polilinha *Ramer–Douglas–Peucker*, também conhecido como algoritmo de simplificação *Ramer*, funciona selecionando um subconjunto de vértices de uma polilinha, de modo que, com esse subconjunto, seja possível reconstruir uma nova polilinha simplificada que deve estar a uma distância máxima  $\varepsilon$  da polilinha original [75]. A Figura 3.4 demonstra o efeito da aplicação do algoritmo *Ramer* nos vértices que definem a área de um empreendimento. Inicialmente, o número de vértices era de 1.146. Depois de aplicado o algoritmo, o total de vértices foi reduzido para 25. Percebe-se que, embora com significativamente menos pontos, o polígono final manteve as principais características do polígono original.

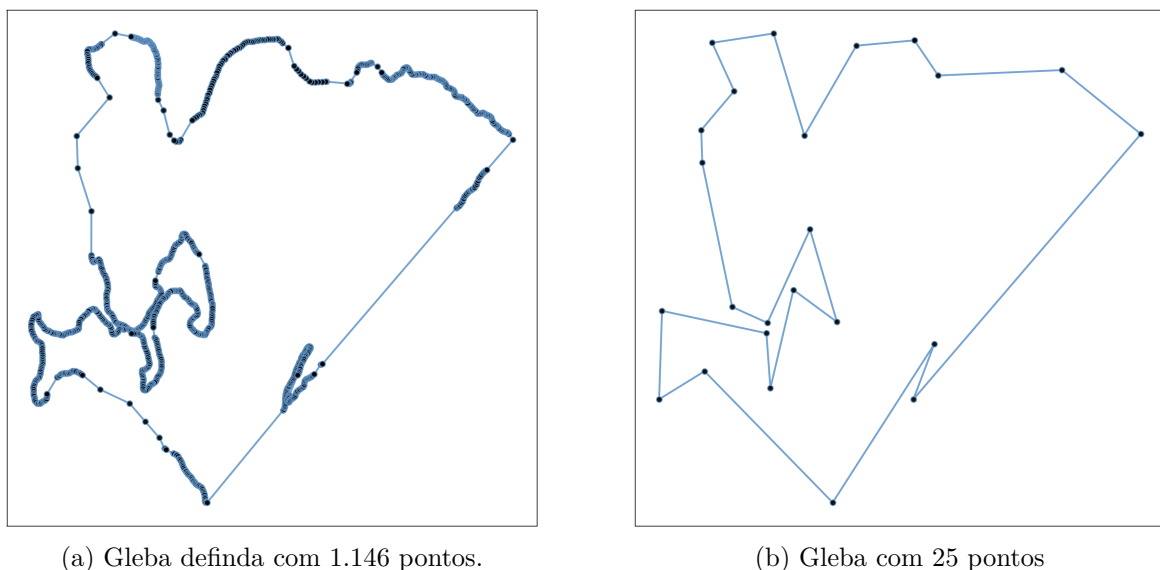


Figura 3.4: Gleba antes (a) e depois (b) da redução do número de vértices com o algoritmo *Ramer*.

Ao final da coleta, foram adicionadas cinco variáveis ao conjunto de variáveis descritas na Subseção 3.2.1, totalizando 45. Na próxima subseção, abordaremos o Sisdagro e as informações disponibilizadas pelo sistema. Discutiremos, também, o processo de extração de dados adotado.

### 3.2.3 Sisdagro

Em conversa com a equipe de monitoramento, fomos informados de que algumas sinalizações são geradas com base em informações colhidas do Sisdagro, mais especificamente, da ferramenta *Balanço Hídrico de Cultivo*. Falaremos, portanto, nesta subseção sobre as informações fornecidas pelo Balanço Hídrico de Cultivo e sobre o processo de extração de seus dados.

O Sistema de Suporte à Decisão na Agropecuária é uma solução *web* desenvolvida pelo Instituto Nacional de Meteorologia (INMET) “com o objetivo de apoiar usuários do setor agrícola em suas tomadas de decisão, auxiliando no planejamento e manejo agropecuário” [76]. O Sisdagro disponibiliza diversas ferramentas para monitoramento agrometeorológico, entre elas o *Balanço Hídrico e Perda de Produtividade* para os cultivos de algodão, arroz, aveia, café, cana-de-açúcar, citros, feijão, girassol, milho, soja e trigo, e o *Índice de Vegetação*, cujo valor é determinado por meio de imagens de satélites [77].

O Balanço Hídrico de Cultivo de uma cultura “visa calcular o balanço de água no solo levando-se em consideração tanto o tipo de vegetação quanto a sua fase de crescimento e desenvolvimento” [77]. A partir do cálculo do balanço hídrico (estimativas de excesso e deficiência hídricas), são produzidas diversas informações relativas aos seus efeitos sobre o cultivar. Entre as informações produzidas, temos o excesso hídrico, o déficit hídrico e a estimativa da produtividade (em porcentagem).

O excesso e déficit hídricos são estimados com base em informações de precipitação, radiação solar, vento, umidade do ar, tipo de cultivar e capacidade de armazenamento d’água do solo. Com base no déficit hídrico, estima-se também o impacto sobre a produtividade. A Figura 3.5 apresenta um gráfico com as estimativas de excesso e déficit hídricos para o algodão com ciclo de 130 dias, compreendendo desde a emergência até a maturidade fisiológica. Já a Figura 3.6 apresenta a estimativa da produtividade ao longo ciclo.

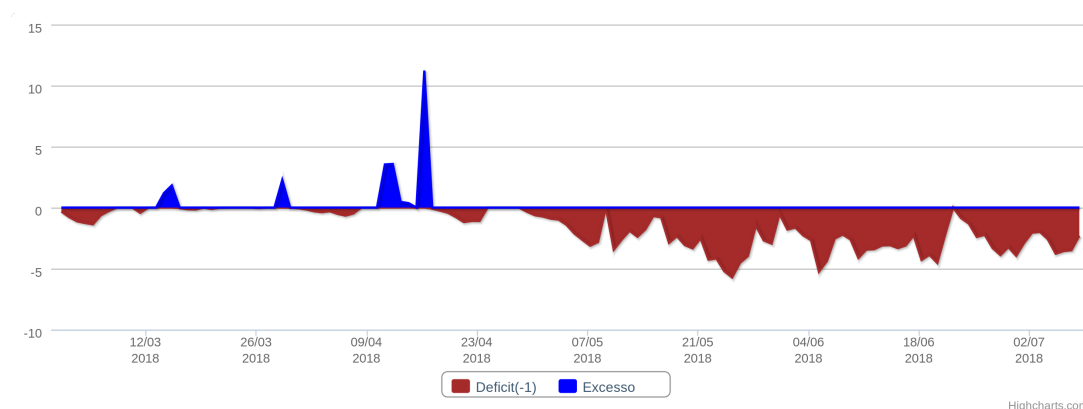


Figura 3.5: Excesso e déficit hídricos ao longo do ciclo da cultura. Fonte: Sisdagro

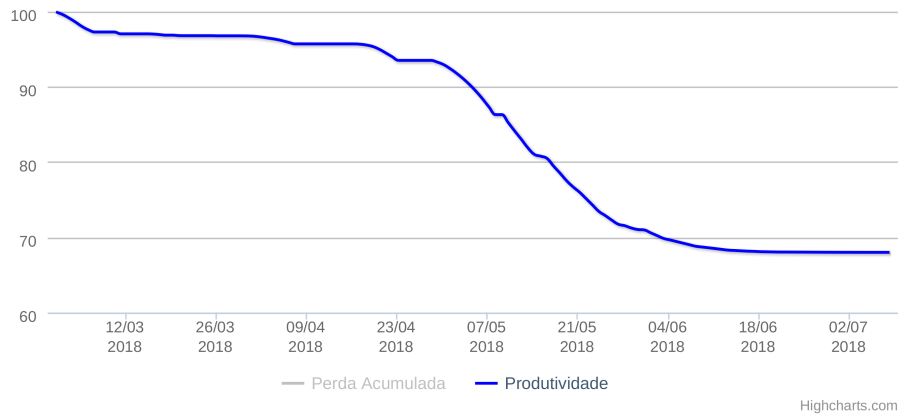


Figura 3.6: Estimativa de impacto na produção ao longo do ciclo da cultivar. Fonte: Sisdagro

O processo de extração dos dados e construção das novas variáveis teve início com a identificação de quais culturas cadastradas na base de COPs do Sicor eram suportadas no Balanço Hídrico do Cultivo. De 67 produtos, apenas cinco eram suportados: trigo, milho, feijão, soja e aveia. Desses cinco, apenas quatro produtos aparecem em um número substancial de COPs: trigo, milho, feijão e soja. Juntos, eles correspondem a 86,10% do total de registros. Foram extraídos, portanto, dados do Sisdagro com base nas COPs cujo produto constava entre os quatro.

Para a extração, foram utilizados a data de plantio, o tipo de solo, o ciclo e o tipo de cultivar coletados do Sicor como parâmetros de entrada para o Sisdagro. Foi realizada consulta para cada registro de COP. O retorno da consulta é um conjunto de registros — um para cada dia do ciclo da cultivar a contar da emergência —, em que cada registro contém, entre outras informações, o excesso hídrico diário, o déficit hídrico diário e o impacto estimado na produtividade até a respectiva data.

Com base nessas informações, foram criadas as variáveis *bhd\_defic\_hidr\_media* — média do déficit hídrico —, *bhd\_defic\_hidr\_total* — déficit hídrico total —, *bhd\_exces\_hidr\_media* — média do excesso hídrico —, *bhd\_exces\_hidr\_total* — excesso hídrico total —, todas calculadas sobre o período do ciclo da cultivar. Também foi criada a variável *bhd\_produtividade*, que corresponde à produtividade final estimada. Ao final do processo, as cinco variáveis foram adicionadas às demais já coletadas, totalizando 50 variáveis.

Na próxima seção, versaremos sobre a fase de Preparação dos Dados, quando ocorrem as atividades de limpeza de dados, criação de novas variáveis e instâncias, extração de características, etc.

### 3.3 Preparação dos dados

A fase de pré-processamento é composta por um conjunto de atividades cujo objetivo é tornar os dados próprios para modelagem. As atividades executadas nesta fase foram organizadas nos seguintes grupos: construção de novos atributos, imputação e agregação de valores, descarte de atributos e instâncias, padronização dos dados, transformação dos dados, redução de dimensionalidade e multicolinearidade, e balanceamento de classes.

Nas próximas subseções, discorreremos sobre cada um destes grupos.

#### 3.3.1 Construção de novos atributos

Uma vez coletados os dados, deu-se início à fase de criação de novos atributos valendo-se dos atributos originais. Primeiramente, foram criados os atributos *nov\_num\_cop\_deferida* e *nov\_num\_cop\_indeferida*, que representam, respectivamente, o número de COPs deferidas e indeferidas para o mutuário da COP. A construção foi feita com base no histórico de COPs dos últimos cinco anos.

A partir dos atributos do tipo data presentes nas Tabelas 3.1 a 3.3, foram criados cinco novos atributos: *nov\_dif\_dias\_emissaoecd\_cop* — dias entre a emissão da cédula de crédito e a comunicação de ocorrência de perdas —, *nov\_dif\_dias\_visitperit\_envrelat* — dias entre a visita do perito e o envio do relatório —, *nov\_dif\_dias\_fimevento\_cop* — dias entre o fim do evento e a comunicação de perdas —, *nov\_duracao\_colheita* — duração da colheita em dias —, *nov\_dias\_durac\_plantio* — duração do planto em dias —, *nov\_dias\_durac\_evento* — duração do evento em dias — e *nov\_dif\_dias\_cop\_visitperit* — dias entre a comunicação de perdas e a visita do perito.

As variáveis do tipo data *cop\_dt\_cop*, *cop\_dt\_ini\_evento*, *rcp\_dt\_ini\_colheita* e *rcp\_dt\_ini\_plantio* foram transformadas e passaram a ser representadas por meio de coordenadas cartesianas  $(x, y)$ . Para cada data, foi extraído o dia do ano,  $d$ , e, na sequência, convertido o valor para uma escala de 0 a  $2\pi$ , de modo que o 1º dia é aproximadamente  $0,02$  e o 365º dia é igual a  $2\pi$ . Em seguida, aplicaram-se as funções seno e cosseno sobre os valores convertidos —  $d'$  —, obtendo-se as coordenadas  $(\cos(d'), \sin(d'))$ . As abcissas e as ordenadas passaram a formar, cada qual, novos atributos. Ao final do processo, foram gerados oito novos atributos, dois para cada uma das variáveis listadas acima, a saber: *cop\_sen\_cop*, *cop\_cos\_cop*, *cop\_sen\_ini\_event*, *cop\_cos\_ini\_event*, *rcp\_sen\_ini\_colh*, *rcp\_cos\_ini\_colh*, *rcp\_sen\_ini\_plant* e *rcp\_cos\_ini\_plant*.

A atributo *rcp\_loc\_gps* não foi utilizado diretamente na modelagem, mas sim na extração de informações de outras fontes de dados, conforme relatado nas Subseções 3.2.2 e 3.2.3, e na criação dos novos atributos *igbe\_mapa\_cli\_umidade* e *igbe\_mapa\_cli\_temp*.

Os atributos *igbe\_mapa\_cli\_umidade* e *igbe\_mapa\_cli\_temp* representam, respectivamente, a zona climática para umidade e temperatura na qual a gleba está contida.

Para determinar a qual zona climática uma gleba está associada, foram utilizados os dados vetoriais do Mapa de Clima do Brasil disponibilizados pelo IBGE<sup>4</sup>. Para isso, primeiro encontramos as coordenadas que apontam para o centro de massa da gleba. Em seguida, com o uso do pacote *GeoPandas*<sup>5</sup>, do Python, identificamos em qual polígono do mapa de clima as coordenadas estão contidas. Uma vez identificado o polígono, extraímos as informações de variedade térmica e grau de umidade às quais o polígono está associado. Atribuímos, então, essas informações às variáveis *igbe\_mapa\_cli\_umidade* e *igbe\_mapa\_cli\_temp*.

A Figura 3.7 apresenta as diferentes zonas climáticas do país agrupadas por temperatura e umidade. A mapeamento das cores e suas respectivas zonas está detalhado na Tabela 3.4. A partir das informações constantes da tabela, é possível notar os valores que a variável *igbe\_mapa\_cli\_umidade* pode assumir, os quais são: semi-árido, semi-úmido, úmido e super úmido; bem como os valores que a variável *igbe\_mapa\_cli\_temp* pode assumir: mesotérmico mediano, mesotérmico brando, subquente e quente.

Por último, foram criadas mais três variáveis, aqui denominadas *nov\_var1*, *nov\_var2* e *nov\_var3*, com base em alguns campos presentes nas bases do Sicor. Essas variáveis são utilizadas pela equipe de monitoramento na geração de sinalizações. Em razão de sua natureza e finalidade, informações acerca das variáveis são consideradas restritas e, portanto, não serão fornecidos maiores detalhes.

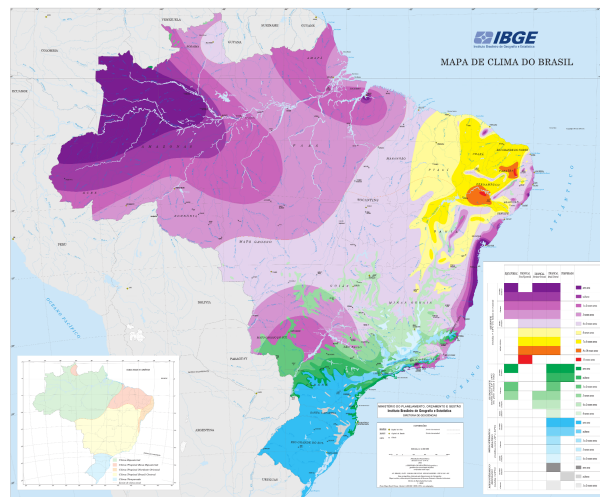













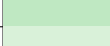
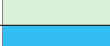


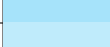
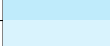






Figura 3.7: Mapa de Clima do Brasil em escala 1:5.000.000 representando as diferentes zonas climáticas do país. Fonte: IBGE<sup>6</sup>.

<sup>4</sup>[ftp://geoftp.ibge.gov.br/informacoes\\_ambientais/climatologia/vetores/brasil/Clima\\_5000mil.zip](ftp://geoftp.ibge.gov.br/informacoes_ambientais/climatologia/vetores/brasil/Clima_5000mil.zip)

<sup>5</sup><http://geopandas.org/>

Tabela 3.4: Associação das cores às suas respectivas zonas climáticas. Fonte: IBGE.

Variedade Térmica	Grau de Umidade	Regime de chuvas	Cor
Quente (média > 18° C em todos os meses)	super úmido	sem seca	
		subseca	
	úmido	1 a 2 meses secos	
		3 meses secos	
	semi-úmido	4 a 5 meses secos	
	semi-árido	6 meses secos	
		7 a 8 meses secos	
		9 a 10 meses secos	
11 meses secos			
Subquente (média entre 15° e 18° C em pelo menos 1 mês)	super úmido	sem seca	
		subseca	
	úmido	1 a 2 meses secos	
		3 meses secos	
	semi-úmido	4 a 5 meses secos	
	semi-árido	6 meses secos	
Mesotérmico Brando (média entre 10° e 15° C)	super úmido	sem seca	
		subseca	
	úmido	1 a 2 meses secos	
		3 meses secos	
	semi-úmido	4 a 5 meses secos	
Mesotérmico Mediano (média > 10° C)	super úmido	sem seca	
		subseca	
	úmido	1 a 2 meses secos	

Na próxima subseção, abordaremos as ações empregadas para correção de valores incorretos ou faltantes.

### 3.3.2 Imputação e agregação de valores

Depois de criados os novos atributos, iniciamos as ações de imputação e agregação de valores. As ações consistiram em identificar valores faltantes e, quando possível, realizar sua imputação, bem como agregar valores de variáveis nominais a fim de reduzir o número de categorias e, conseqüentemente, as chances de sobreajuste.

Primeiramente, identificamos que diversas instâncias não continham valores para os atributos *emp\_vl\_rec proprio\_serv* e *emp\_vl\_rec proprio*. Uma vez que, do ponto de vista do pagamento de coberturas, valores ausentes ou iguais a zero resultam igualmente

<sup>6</sup>[ftp://geofp.ibge.gov.br/informacoes\\_ambientais/climatologia/mapas/brasil/Map\\_BR\\_c lima\\_2002.pdf](ftp://geofp.ibge.gov.br/informacoes_ambientais/climatologia/mapas/brasil/Map_BR_c lima_2002.pdf)

no não pagamento das parcelas de recurso próprio em eventual pagamento de cobertura, foi imputado o valor zero para os casos em que havia valores faltantes.

Em seguida, observou-se que o atributo *emp\_tipo\_irrigacao* é utilizado para informar se o empreendimento aplica ou não algum tipo de irrigação e, caso aplique, qual o tipo de irrigação empregada (aspersão, microaspersão, superfície, etc.). A fim de reduzir o número de categorias, foram agrupados todos os valores que indicavam algum tipo de irrigação em um único valor “irrigado”, de modo que, ao final, o atributo *emp\_tipo\_irrigacao* continha três categorias: “irrigado”, “não irrigado” e “não se aplica”.

Por fim, verificamos que um número significativo de instâncias não continham valores para o atributo *rcp\_ciclo\_cultivar* e, visto que essa informação é necessária para extração dos dados do Sisdagro (ver Subseção 3.2.3), foi necessário preencher os dados faltantes utilizando o valor contido no campo *cop\_grp\_ciclo\_cultivar* (grupo do ciclo do cultivar). Para isso, foi necessário cruzar as informações do atributo *cop\_grp\_ciclo\_cultivar*, juntamente com a informação do ano da safra, da localidade do empreendimento e do produto do empreendimento, com as portarias<sup>7</sup> de Zoneamento Agrícola de Risco Climático do Ministério do Agricultura, Pecuária e Abastecimento para determinar o ciclo, em dias, do cultivar. Uma vez obtido o ciclo, ele foi inserido na campo *rcp\_ciclo\_cultivar* para aquelas instâncias em que havia valor faltante. Ressalta-se que nem todas as instâncias possuíam o campo *cop\_grp\_ciclo\_cultivar* preenchido, razão pela qual adotou-se a estratégia de complementação de um campo com informações de outro.

Nos demais casos em que havia valores faltantes ou incorretos, procedeu-se à remoção das instâncias ou atributos, conforme será apresentado na próxima subseção.

### 3.3.3 Descarte de atributos e instâncias

Uma vez imputados valores para os casos em que os atributos estavam vazios, procedemos à remoção das instâncias cujos atributos não estavam preenchidos ou que possuíam valores reconhecidamente incorretos. Por exemplo, foram removidas 25 instâncias em que o atributo *emp\_mutsexo* estava vazio. Da mesma forma, foram descartadas 18 instâncias cujo valor do atributo *emp\_previsao\_produc* era igual a 0.

Conforme descrito na Subseção 3.2.3, foram coletados dados do Sisdagro apenas para as COPs cujo produto está entre os suportados pelo sistema. Em razão disso, parte das instâncias extraídas do Sidor não continham informações acerca da estimativa de produtividade ou de excesso e déficit hídricos. Essas instâncias foram, portanto, descartadas.

Dos onze eventos de COP listados no Sidor, apenas cinco — seca, chuva excessiva, geadas, variação excessiva de temperatura e chuva na colheita — podem ser evidenciados

---

<sup>7</sup><http://www.agricultura.gov.br/assuntos/riscos-seguro/risco-agropecuario/portarias>



pelas informações existentes nos sistemas Sisdiagro e Agritempo. Dessarte, foram mantidas as instâncias em que o evento de COP está associado a um desses eventos, com exceção do evento chuva na colheita, o qual está associado a somente 24 COPs e, portanto, contém número insuficiente de instâncias para treinamento de um modelo. As demais instâncias foram descartadas.

Em seguida foram removidos atributos redundantes, como *cop\_evento*, *cop\_dt\_ini\_evento*, *cop\_dt\_fim\_evento* e *cop\_grp\_ciclo\_cultivar*, em razão desses atributos corresponderem às mesmas informações apresentadas nos *atributos rcp\_evento*, *rcp\_dt\_ini\_evento*, *rcp\_dt\_fim\_evento* e *rcp\_ciclo\_cultivar*. De acordo com integrantes da equipe de monitoramento, as informações registradas pelos peritos (Tabela 3.3) devem prevalecer sobre as informações apresentadas pelos mutuários ou agentes do Proagro.

Também foram removidos atributos em que foi constatado um número elevado de valores incorretos e não foram encontrados meios de correção desses valores, como a altitude contida na variável *rcp\_loc\_gps*. Embora seja considerada uma variável de grande relevância por Ramos, o mesmo problema identificado em seu estudo foi observado nesta pesquisa: diversos registros com valores significativamente diferentes dos reais. Como não dispúnhamos de recursos para correção tempestiva dos dados, decidimos descartar o atributo. No Capítulo 5 discutiremos sobre possíveis soluções, a serem aplicadas em trabalhos futuros, que permitam sua utilização.

Por fim, foram descartados os atributos do tipo data, bem como os atributos cujo conteúdo é utilizado para identificação da localidade da gleba (*emp\_localidade* e *rcp\_loc\_gps*). Com isso, buscamos prevenir o sobrejeste dos dados que poderia ser causado pela utilização de atributos capazes de identificar unicamente cada instância.

Ao final do processo, restaram 41.409 instâncias e 53 variáveis preditoras, sendo 12 nominais e 41 numéricas. Do total de instâncias, 89,54% correspondem a COPs definitivas e 10,46% a indeferidas. A Tabela 3.5 relaciona os atributos remanescentes e seus respectivos tipos. A Figura 3.8 apresenta a distribuição de classes das COPs por tipo de evento.

Tabela 3.5: Conjunto de variáveis finais

<b>Nome</b>	<b>Tipo</b>	<b>Nome</b>	<b>Tipo</b>
<i>emp_programa</i>	Nominal	<i>bhd_defic_hidr_total</i>	Numérica
<i>emp_empreendimento</i>	Nominal	<i>bhd_exces_hidr_media</i>	Numérica
<i>emp_rct_brt_esperada</i>	Numérica	<i>bhd_exces_hidr_total</i>	Numérica
<i>emp_area</i>	Numérica	<i>bhd_produtividade</i>	Numérica
<i>emp_previsao_produc</i>	Numérica	<i>nov_num_cop_deferida</i>	Numérica
<i>emp_tipo_cultivo</i>	Nominal	<i>nov_num_cop_indeferida</i>	Numérica
<i>emp_tipo_irrigacao</i>	Nominal	<i>nov_dif_dias_emissaoecd_cop</i>	Numérica
<i>emp_vl_parc_credito</i>	Numérica	<i>nov_dif_dias_visitperit_envrelat</i>	Numérica
<i>emp_juros</i>	Numérica	<i>nov_dif_dias_fimevento_cop</i>	Numérica
<i>emp_vl_rec proprio</i>	Numérica	<i>nov_duracao_colheita</i>	Numérica
<i>emp_vl_rec proprio_serv</i>	Numérica	<i>nov_dias_durac_plantio</i>	Numérica
<i>emp_mut_sexo</i>	Nominal	<i>nov_dias_durac_evento</i>	Numérica
<i>emp_mut_possui_dap</i>	Nominal	<i>nov_dif_dias_cop_visitperit</i>	Numérica
<i>cop_tipo_solo</i>	Nominal	<i>nov_var1</i>	Numérica
<i>rcp_evento</i>	Nominal	<i>nov_var2</i>	Numérica
<i>rcp_perda_qualid</i>	Nominal	<i>nov_var3</i>	Numérica
<i>rcp_prod_estimada</i>	Numérica	<i>cop_sen_cop</i>	Numérica
<i>rcp_rct_estimada</i>	Numérica	<i>cop_cos_cop</i>	Numérica
<i>rcp_area_medida</i>	Numérica	<i>cop_sen_ini_event</i>	Numérica
<i>rcp_ciclo_cultivar</i>	Numérica	<i>cop_cos_ini_event</i>	Numérica
<i>cli_temp_max</i>	Numérica	<i>rcp_sen_ini_colh</i>	Numérica
<i>cli_temp_med</i>	Numérica	<i>rcp_cos_ini_colh</i>	Numérica
<i>cli_temp_min</i>	Numérica	<i>rcp_sen_ini_plant</i>	Numérica
<i>cli_precip_med</i>	Numérica	<i>rcp_cos_ini_plant</i>	Numérica
<i>cli_precip_med_max</i>	Numérica	<i>igbe_mapa_cli_umidade</i>	Nominal
<i>bhd_defic_hidr_media</i>	Numérica	<i>igbe_mapa_cli_temp</i>	Nominal

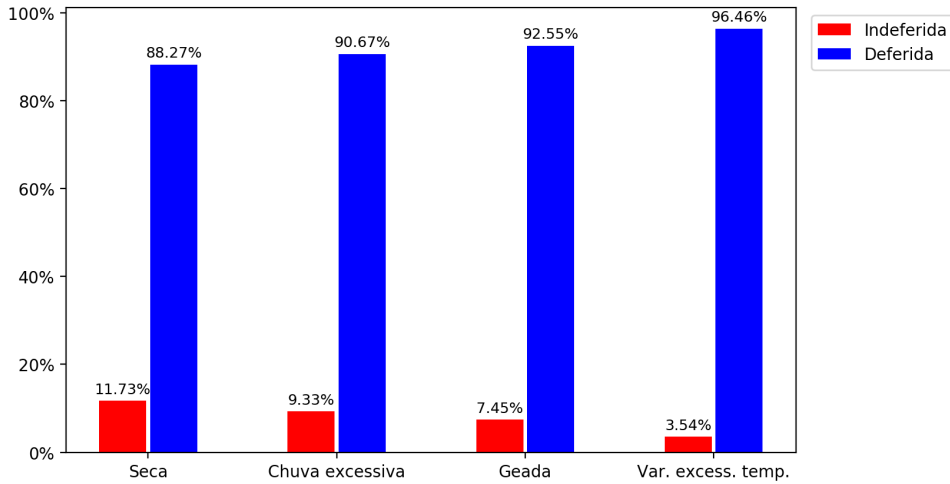


Figura 3.8: Distribuição de classes das COPs por tipo de evento.

Concluídos os procedimentos de descarte, o conjunto de dados foi, então, particionado em dois subconjuntos: partição de treinamento, contendo 75% das observações; e partição de teste, contendo os 25% restantes. O particionamento foi realizado de maneira estratificada, em que foram utilizados como parâmetro para estratificação as classes e os tipos de evento de COP. Desse modo, a partição de treinamento e a de teste continham as mesmas proporções para as classes (deferida e indeferida) e para os tipos de evento (chuva, seca, geada e variação excessiva de temperatura).

### 3.3.4 Padronização dos dados

Uma vez que as variáveis e instâncias haviam sido selecionadas e tratadas, procedemos à padronização dos dados. A padronização consistiu em centralizar os dados e em ajustar a variância de modo a torná-la igual a um. A centralização é alcançada por meio da subtração da média da variável, enquanto a variância unitária é obtida dividindo-se os dados pelo seu desvio padrão.

O cálculo utilizado na padronização pode ser representado pela seguinte equação:

$$\vec{x}_p = \frac{\vec{x}_o - \text{media}(\vec{x}_o)}{dp(\vec{x}_o)} \quad (3.1)$$

em que  $\vec{x}_o$  é o vetor contendo os valores da variável  $X$ ;  $\text{media}()$ , a função da média;  $dp()$ , a função do desvio padrão; e  $\vec{x}_p$  o vetor contendo os valores padronizados.

Concluída a padronização, as variáveis nominais, cujos valores seguem uma distribuição de probabilidade binomial ou multinomial, foram convertidas em variáveis *dummies*.

### 3.3.5 Transformação dos dados

Com os dados padronizados, demos início a sua transformação com o objetivo de aproximar a distribuição dos dados à distribuição gaussiana. As atividades desta etapa foram empregadas somente sobre os conjuntos de dados utilizados no treinamento dos modelos gerados com o *Naive Bayes*, uma vez que, dos quatro algoritmos de aprendizagem de máquina utilizados nesta pesquisa, apenas o Naive Bayes é paramétrico [78]. Quando estamos lidando com variáveis contínuas, a implementação do *Scikit-learn* para o *Naive Bayes*<sup>8</sup> assume normalidade na distribuição de probabilidade dos dados.

Para emprego da transformação, foram, primeiramente, criados dois novos conjuntos de dados de modo a permitir a avaliação de desempenho de modelos treinados com conjuntos de dados resultantes de diferentes tipos de transformação. No primeiro conjunto foram aplicados os algoritmos de transformação de potência *Box-Cox* e *Yeo-Johnson*.

A transformação *Box-Cox* pode ser utilizada apenas quando as variáveis assumem valores estritamente positivos. Já a *Yeo-Johnson* pode ser aplicada tanto sobre valores positivos quanto negativos. A Equação 3.2 descreve o cálculo utilizado pelo *Box-Cox* [79]:

$$x_{i\lambda} = \begin{cases} \frac{x_i^\lambda - 1}{\lambda} & \lambda \neq 0; \\ \ln x_i & \lambda = 0. \end{cases} \quad (3.2)$$

sujeito a  $x_i > 0$

, em que  $x_i$  é o valor a ser transformado, e  $\lambda$  o parâmetro de transformação, o qual pode ser estimado por máxima verossimilhança. A Equação 3.3, por sua vez, apresenta o cálculo empregado na transformação *Yeo-Johnson* [80]:

$$x_{i\lambda} = \begin{cases} \frac{(x_i + 1)^\lambda - 1}{\lambda} & x_i \geq 0, \lambda \neq 0; \\ \ln(x_i + 1) & x_i \geq 0, \lambda = 0; \\ \frac{-(-x_i + 1)^{2-\lambda} - 1}{2 - \lambda} & x_i < 0, \lambda \neq 2; \\ -\ln(-x_i + 1) & x_i < 0, \lambda = 2; \end{cases} \quad (3.3)$$

, em que  $x_i$  e  $\lambda$  representam a mesma variável e parâmetro utilizados na Equação 3.2.

No segundo conjunto de dados, foi aplicada a transformação quantílica, cujo funcionamento consiste em primeiro estimar a função de distribuição acumulada da variável original,  $G(x)$ , para, então, aplicar a inversa da função de distribuição acumulada,  $F^{-1}(p)$ , sobre o resultado de  $G(x)$  [81]. A Equação 3.4 descreve o cálculo utilizado na transfor-

---

<sup>8</sup>[https://scikit-learn.org/stable/modules/naive\\_bayes.html#gaussian-naive-bayes](https://scikit-learn.org/stable/modules/naive_bayes.html#gaussian-naive-bayes)

mação quantílica:

$$x_{iq} = F^{-1}(G(x_i)). \quad (3.4)$$

A Figura 3.9 apresenta os efeitos das transformações de potência e quantílica sobre a variável *cli\_temp\_med*. Os impactos podem ser observados no formato do histograma e no gráfico de probabilidades Q-Q, cuja distribuição teórica segue uma normal.

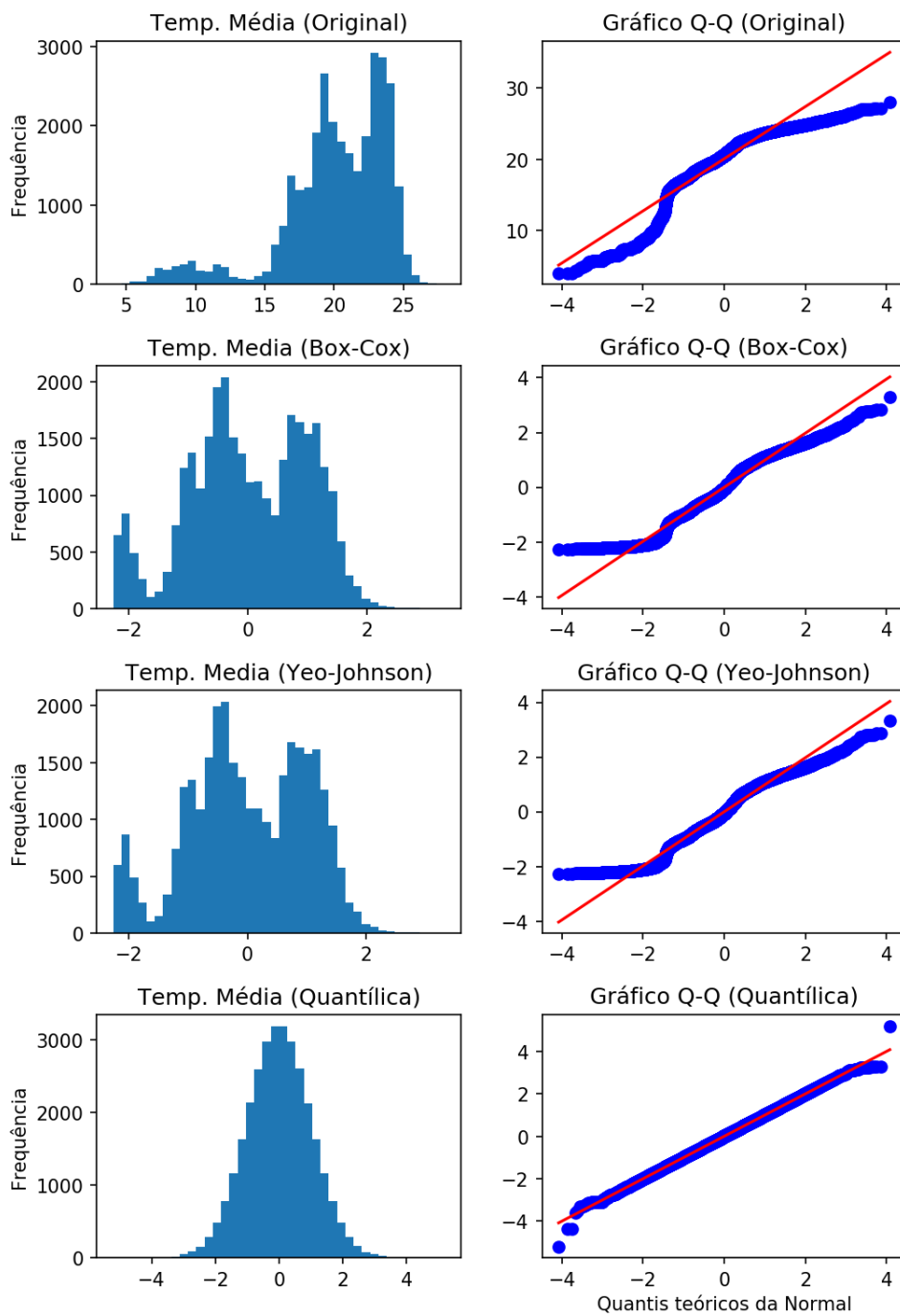


Figura 3.9: Alterações na distribuição de probabilidade da variável *cli\_temp\_med* depois de aplicadas as transformações de potência e quantílicas.

### 3.3.6 Redução de dimensionalidade e de multicolinearidade

Uma vez realizados os ajustes e as transformações necessárias dos dados, efetuamos a redução de dimensionalidade e de multicolinearidade com o uso do PCA. Primeiro separamos as variáveis numéricas das nominais. Em seguida, aplicamos o PCA sobre as 41 variáveis numéricas. Depois de calculados os componentes principais, foram selecionados, na ordem de maior autovalor, aqueles cujas variâncias quando somadas representassem 95% do total de variância das variáveis originais.

O processo resultou em 23 variáveis numéricas descorrelacionadas [82], as quais foram novamente padronizadas por meio do cálculo descrito na Equação 3.1. Por fim, as variáveis numéricas foram reagrupadas às nominais.

### 3.3.7 Balanceamento de classes

Depois de aplicado o PCA, passamos às atividades de balanceamento de classes. O objetivo desta etapa é tornar o número de instâncias de treinamento da classe minoritária próximo ao da classe majoritária. Para isso, foi empregada a abordagem *oversampling* seguida da *undersampling*.

Na abordagem *oversampling*, foi utilizado o método *Smote* implementado no pacote *imbalanced-learn* [83]. O método foi parametrizado para gerar instâncias sintéticas da classe minoritária em quantidade suficiente para igualar a proporção de instâncias em cada classe. Já na abordagem *undersampling*, foi utilizado o método *Tomek*, também implementado pelo pacote *imbalanced-learn*. Por meio do método, foram removidas as instâncias que participavam de um par *tomek link*.

Concluído o balanceamento, cada classe passou a conter 50% das instâncias. Como consequência, o número de instâncias do conjunto de dados de treinamento foi acrescido de 78% de seu valor original. Nesse ponto, haviam sido concluídas as atividades de processamento.

As atividades de transformação dos dados, redução de dimensionalidade e de multicolinearidade, e balanceamento de classes foram executadas seguindo a abordagem proposta por Santos *et al.* [84], em que tais atividades, em especial o balanceamento, são executadas durante a validação cruzada, a fim de evitar que o desempenho obtido na validação seja superestimado. Para isso, foram utilizadas, de forma conjunta, as bibliotecas *Pipeline*<sup>9</sup> e *GridsearchCV*<sup>10</sup>.

---

<sup>9</sup><https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.pipeline.Pipeline.html>

<sup>10</sup>[https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.GridSearchCV.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html)

O efeito prático dessa abordagem na transformação dos dados e na redução de dimensionalidade é que, enquanto os parâmetros de ambos são calculados com base exclusivamente nas partições de treinamento, a transformação e redução são efetuadas tanto nas partições de treinamento quanto na partição de teste. Por sua vez, o efeito no balanceamento de classes consiste na criação de novas instâncias da classe minoritária e na remoção dos *tomek links* presentes nas partições de treinamento, ao passo que a partição de teste é mantida sem acréscimo ou remoção de instâncias.

Outrossim, a utilização conjunta dos métodos *Pipeline* e *GridsearchCV* permitiu testar mais facilmente diferentes combinações de procedimentos de pré-processamento e verificar quais deles resultaram em melhoria do desempenho do classificador. Foram construídos conjuntos de dados com e sem a aplicação da transformação, da redução de dimensionalidade e do balanceamento de classes. Ao todo, foram gerados 12 conjuntos de dados para modelagem. A Tabela 3.6 relaciona os conjuntos, os grupos de atividades sobre eles aplicados e os algoritmos com os quais foram utilizados.

Tabela 3.6: Conjuntos de dados gerados no pré-processamento.

Algoritmos	Conjunto	Transformação	Redução D. (PCA)	Balancem. (SMOTE-TK)
Random Forest, RNA, Naive Bayes e SVM	conjunto_original	Nenhuma	Não	Não
	conjunto_pca	Nenhuma	Sim	Não
	conjunto_smtk	Nenhuma	Não	Sim
	conjunto_pca_smttk	Nenhuma	Sim	Sim
Naive Bayes	conjunto_pt	Potência	Não	Não
	conjunto_pt_pca	Potência	Sim	Não
	conjunto_pt_smttk	Potência	Não	Sim
	conjunto_pt_pca_smttk	Potência	Sim	Sim
	conjunto_qt	Quantílica	Não	Não
	conjunto_qt_pca	Quantílica	Sim	Não
	conjunto_qt_smtk	Quantílica	Não	Sim
	conjunto_qt_pca_smttk	Quantílica	Sim	Sim

Na próxima seção, versaremos sobre a fase de modelagem, quando são definidos os hiperparâmetros e treinados os modelos.

### 3.4 Modelagem

Na fase de modelagem, foram treinados os modelos usando os conjuntos de dados produzidos na fase de pré-processamento. O processo teve início com a definição das abordagens

de classificação a serem avaliadas. Foram propostas duas abordagens: a *monolítica* e a *hierárquica*. A abordagem monolítica consiste na utilização de um modelo classificatório de COPs originadas de qualquer tipo de evento climático. A abordagem hierárquica, por sua vez, funciona primeiro identificando o tipo de evento gerador da COP, para, em seguida, encaminhá-la ao modelo cujos dados de treinamento continham apenas COPs do tipo de evento identificado. Enquanto na primeira abordagem há um único modelo classificador, na segunda abordagem teremos quatro modelos — um para cada tipo de evento. As Figuras 3.10 e 3.11 descrevem, respectivamente, os fluxos das abordagens monolítica e hierárquica.

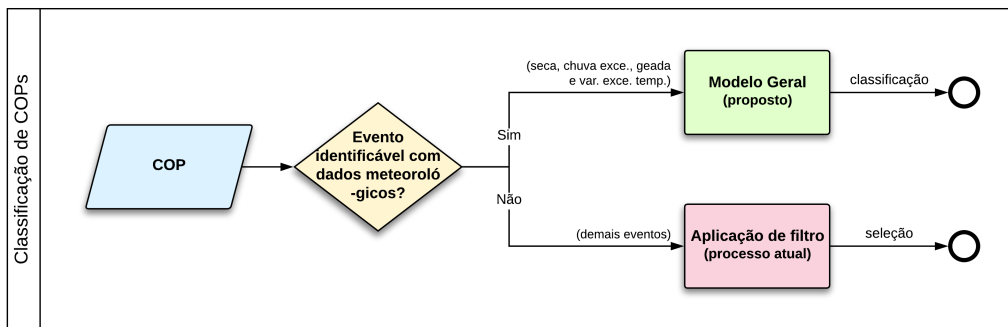


Figura 3.10: Fluxo do processo de classificação na abordagem monolítica.

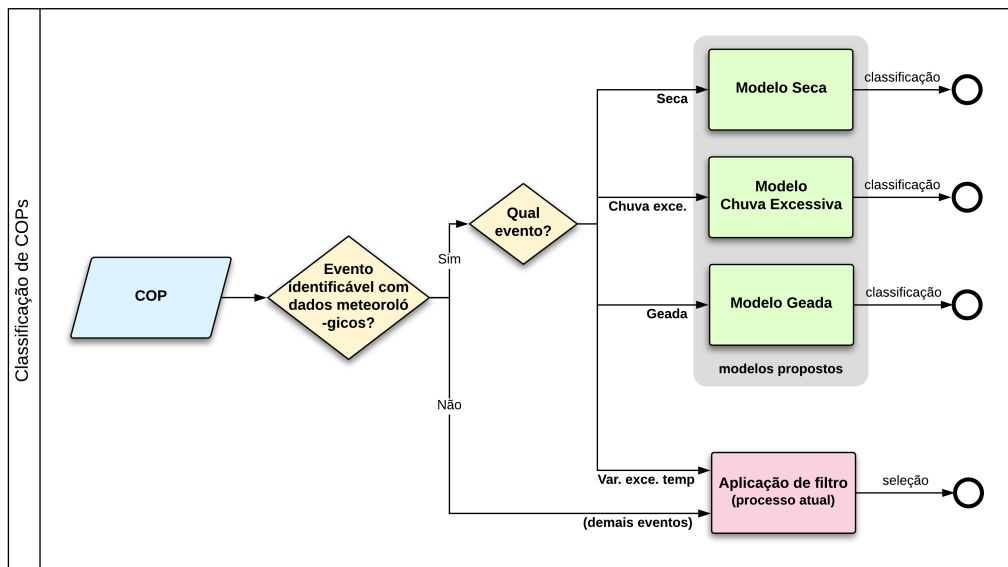


Figura 3.11: Fluxo do processo de classificação na abordagem hierárquica.

Observando a Figura 3.11, é possível perceber que não há modelo para as COPs cujo evento é “variação excessiva de temperatura”, sendo essas COPs tratadas no processo atual em que são utilizados filtros. Isso indica que, embora tivéssemos planejado criar



um modelo para cada um dos quatro tipos de eventos selecionados no pré-processamento, isto acabou por não se concretizar. No momento em que iniciamos a criação dos modelos para o evento “variação máxima de temperatura”, percebemos que havia somente nove instâncias da classe de interesse no conjunto de dados de treinamento e três no conjunto de dados de teste. Conseqüentemente, não seria possível aplicar a validação cruzada com 10 partições, visto que ao menos uma das partições não conteria qualquer instância da dessa classe, tampouco seria possível utilizar o *SMOTE*, uma vez que, é necessário ter ao menos duas instâncias da classe a ser aumentada por partição. Além do mais, caso fosse reduzido o número de partições na validação cruzada para que pudéssemos dar seguimento à criação do modelo para esse evento, os modelos provavelmente estariam sobreajustados em razão de pequena quantidade de exemplos. Portanto, foram criados modelos somente para os eventos “seca”, “chuva excessiva” e “geada”. No Capítulo 4, discutiremos sobre o possível efeito decorrente da não criação desse modelo sobre a apuração dos resultados obtidos na abordagem hierárquica e sobre a comparação desses com os resultados obtidos na abordagem monolítica.

Na geração dos modelos, foram empregados quatro algoritmos de aprendizagem de máquina: *Naive Bayes*, SVM, RNAs e *Random Forest*. Para o *Naive Bayes*, SVM e *Random Forest* foram utilizadas as implementações disponíveis no pacote *Scikit-learn*. Quanto ao algoritmo RNAs, foi utilizada a implementação do pacote *Keras*<sup>11</sup>.

Embora o *Scikit-learn* disponibilize implementações do *Naive Bayes* para as distribuições de probabilidade binomial, multinomial e normal — as quais estão presentes nas variáveis dos dados utilizados nesta pesquisa —, não há uma implementação específica que combine todas essas distribuições. Foi necessário, portanto, implementar uma classe que combinasse os métodos do *Naive Bayes*, produzindo, por sua vez, um único resultado.

Cada algoritmo possui um conjunto próprio de hiperparâmetros. Para definição da configuração mais apropriada dos valores desses hiperparâmetros, foi adotada a estratégia de busca em grade (busca exaustiva). A estratégia consiste na combinação de valores pré-configurados, de modo a criar um conjunto de arranjos contendo todas combinações possíveis, e na geração de um modelo para cada um dos arranjos. O objetivo da estratégia é encontrar o arranjo que resulta no modelo com o maior desempenho. Para criação dos arranjos, foi utilizado o pacote *Pipeline*, o qual foi também empregado na geração dos conjuntos de dados destinados à modelagem. A Tabela A.1 apresenta os hiperparâmetros de cada algoritmo e seus respectivos valores.

Os conjuntos de dados e arranjos foram combinados, e, para cada par “arranjo-conjunto”, foi gerado um modelo, o qual foi treinado com o conjuntos de dados de treinamento, e estimado seu desempenho. O processo de geração e validação do modelo foi

---

<sup>11</sup><https://keras.io/>

executado em duas etapas. A primeira etapa consistiu em estimar a capacidade de generalização dos modelos. Para isso, foi utilizada a validação cruzada com 10 partições. Dessarte, para cada par “arranjo-conjunto”, foram treinados 10 modelos e, para cada modelo, foram coletadas as métricas *F1-score*, Precisão Média, ROC-AUC e Acurácia. Na segunda etapa foi gerado um novo modelo para cada par “arranjo-conjunto”, mas, desta vez, considerando todo o conjunto de dados de treinamento, *i.e.*, o modelo foi treinado sobre 100% do conjunto de dados de treinamento, e não sobre 90%, como foi feito na validação cruzada. As médias das métricas calculadas para os 10 modelos passaram a representar a estimativa de desempenho deste último modelo.

Enquanto na abordagem monolítica foram utilizados, para treinamento, os conjuntos de dados listados na Tabela 3.6 de forma integral, na abordagem hierárquica cada conjunto de dados da referida tabela deu origem a quatro novos conjuntos de treinamento. Os novos conjuntos foram obtidos separando-se as instâncias conforme o tipo de evento climático de modo que cada novo conjunto fosse constituído de instâncias de um único tipo de evento.

No próximo capítulo, reportaremos os desempenhos obtidos com a validação cruzada na fase de modelagem, bem como o desempenho do melhor modelo para cada par “algoritmo-métrica” calculado sobre o conjunto de dados de teste.

# Capítulo 4

## Resultados

Este capítulo é destinado a apresentar os resultados da pesquisa obtidos com a execução do método proposto. Primeiro, são apresentados os resultados alcançados na validação cruzada. Na sequência, são reportados os resultados desses mesmos modelos quando aplicados sobre os dados de teste. Por fim, o desempenho dos modelos são comparados por meio da plotagem da curva ROC e da curva *precisão-sensibilidade*.

Tanto na seção que aborda os resultados obtidos na validação cruzada (Seção 4.1) quanto na seção que aborda os resultados obtidos sobre os dados de teste (Seção 4.2) são apresentados os desempenhos para as abordagens hierárquica e monolítica. Em todos os casos, o cálculo das métricas  $F_1$ -score, *precisão média* e *área sob a curva ROC* foi realizado considerando a classe de interesse, *i.e.*, COPs indeferidas.

Na próxima seção, apresentaremos os resultados obtidos na validação cruzada.

### 4.1 Resultados Validação Cruzada

Conforme explicitado na Seção 3.4, durante a modelagem foram coletadas métricas de desempenho dos modelos. Depois de gerados os modelos e estimados os respectivos desempenhos com a validação cruzada, foram selecionados aqueles que alcançaram os melhores resultados para terem seus desempenhos reportados nesta seção.

Apresentaremos, a seguir, os resultados obtidos na validação cruzada para os modelos gerados nas abordagens monolítica e hierárquica.

#### 4.1.1 Abordagem monolítica

Na abordagem monolítica, foram selecionados os modelos que apresentaram os melhores resultados por métrica e algoritmo. Dessa forma, foram selecionados 16 modelos: 4 métricas vezes 4 algoritmos. Destacamos que, em alguns casos, o mesmo modelo obteve

o melhor desempenho para mais de uma métrica. A Tabela 4.1 apresenta os resultados obtidos na validação cruzada para os 16 modelos enquanto a Tabela 4.2 traz os seus intervalos de confiança. O cálculo dos intervalos foi realizado a partir de uma distribuição  $t$  de *Student* com 9 graus de liberdade considerando um nível de confiança de 95%.

Tabela 4.1: Melhores desempenhos por algoritmo e métrica obtidos com os modelos monolíticos.

Algoritmo	Métricas			
	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
<i>Random Forest</i>	<b>0.491</b>	<b>0.921</b>	<b>0.554</b>	<b>0.829</b>
Redes Neurais Artificiais	0.401	0.908	0.439	0.780
SVM	0.371	0.903	0.386	0.750
<i>Naive Bayes</i>	0.325	0.862	0.304	0.725

Tabela 4.2: Intervalos de confiança das médias calculadas na validação cruzada para a abordagem monolítica (Tabela 4.1)

Algoritmo	Intervalo de Confiança ( $\alpha = 0.05$ , $t_{\alpha/2} = 2.262$ , $df = 9$ )			
	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
<i>Random Forest</i>	[0.478, 0.503]	[0.919, 0.923]	[0.535, 0.573]	[0.821, 0.836]
Redes Neurais Artificiais	[0.384, 0.418]	[0.906, 0.911]	[0.412, 0.466]	[0.768, 0.793]
SVM	[0.360, 0.382]	[0.901, 0.905]	[0.361, 0.412]	[0.737, 0.763]
<i>Naive Bayes</i>	[0.312, 0.338]	[0.854, 0.870]	[0.283, 0.325]	[0.710, 0.740]

Os valores em negrito localizados na Tabela 4.1 indicam os melhores resultados obtidos para cada métrica. Observa-se, portanto, que o algoritmo *Random Forest* produziu os modelos com os maiores desempenhos para todas as métricas. Os valores máximos obtidos para as métricas  $F_1$ -score e precisão média foram, respectivamente, 0,491 e 0,554. Em segundo, terceiro e quarto lugar, vieram, nessa ordem, as RNAs, o SVM e o *Naive Bayes*, que, assim como a *Random Forest*, obtiveram a mesma posição para todas as métricas. Uma vez que os intervalos de confiança não se sobrepõem, pode-se afirmar que as diferenças de desempenhos entre os modelos são estatisticamente significativas.

A Tabela A.2 traz os arranjos dos hiperparâmetros e os conjuntos de dados utilizados no treinamento dos modelos. Ela informa também os resultados que cada modelo obteve em todas as quatro métricas. Analisando os resultados da tabela, identificamos que a utilização do SMOTE-TK contribuiu para o aumento do desempenho dos modelos gerados a partir das RNAs e do *Random Forest* no que concerne à métrica  $F_1$ -score. Isso ocorreu em razão do aumento da *sensibilidade*, o que era esperado uma vez que o SMOTE tende a expandir a fronteira de decisão em favor da classe minoritária. Um efeito negativo

observado foi que, com o aumento da sensibilidade, houve simultaneamente uma redução da *precisão*. Quanto ao PCA, foi possível observar incremento dos resultados com sua utilização em dois modelos: um derivado do *Naive Bayes* e o outro do SVM. Em ambos os casos o benefício foi em favor da métrica acurácia.

Outro aspecto observado foi que tanto a transformação quantílica como a de potência resultaram nos modelos com os maiores desempenhos dentre aqueles oriundos do *Naive Bayes*, confirmando, portanto, a hipótese de que, quando se trata de variáveis contínuas, a implementação do *Naive Bayes* no *Scikit-learn* se beneficia de dados com distribuições mais próximas da normal.

### 4.1.2 Abordagem hierárquica

De forma semelhante à abordagem monolítica, na abordagem hierárquica foram selecionados os modelos que apresentaram os melhores resultados por métrica e algoritmo. Contudo, nesta abordagem a seleção foi realizada considerando também o tipo de evento climático. Visto que foram criados modelos para três tipos de eventos, o total de modelos selecionados foi 48. As Tabelas 4.3, 4.5 e 4.7 exibem os desempenhos obtidos na validação cruzada para os eventos *seca*, *chuva excessiva* e *geada*, respectivamente. Já as Tabelas 4.4, 4.6 e 4.8 trazem os intervalos de confiança correspondentes.

Tabela 4.3: Melhores desempenhos por algoritmo e métrica obtidos com modelos treinados com COPs do evento seca.

Algoritmo	Métricas			
	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
<i>Random Forest</i>	<b>0.500</b>	<b>0.913</b>	<b>0.575</b>	<b>0.832</b>
Redes Neurais Artificiais	0.401	0.893	0.428	0.775
SVM	0.394	0.893	0.410	0.763
<i>Naive Bayes</i>	0.346	0.848	0.335	0.726

Tabela 4.4: Intervalos de confiança das médias das métricas obtidas com os modelos do evento seca (Tabela 4.3).

Algoritmo	Intervalo de Confiança ( $\alpha = 0.05$ , $t_{\alpha/2} = 2.262$ , $df = 9$ )			
	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
<i>Random Forest</i>	[0.477, 0.522]	[0.911, 0.915]	[0.550, 0.600]	[0.820, 0.843]
Redes Neurais Artificiais	[0.379, 0.423]	[0.890, 0.896]	[0.404, 0.451]	[0.764, 0.786]
SVM	[0.382, 0.406]	[0.891, 0.896]	[0.389, 0.430]	[0.748, 0.778]
<i>Naive Bayes</i>	[0.333, 0.359]	[0.842, 0.853]	[0.311, 0.360]	[0.712, 0.740]

Tabela 4.5: Melhores desempenhos por algoritmo e métrica obtidos com modelos treinados com COPs do evento chuva excessiva.

Algoritmo	Métricas			
	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
<i>Random Forest</i>	<b>0.468</b>	<b>0.928</b>	<b>0.530</b>	<b>0.824</b>
Redes Neurais Artificiais	0.383	0.915	0.393	0.766
SVM	0.359	0.913	0.361	0.743
<i>Naive Bayes</i>	0.326	0.864	0.293	0.731

Tabela 4.6: Intervalos de confiança das médias das métricas obtidas com os modelos do evento chuva excessiva (Tabela 4.5).

Algoritmo	Intervalo de Confiança ( $\alpha = 0.05$ , $t_{\alpha/2} = 2.262$ , $df = 9$ )			
	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
<i>Random Forest</i>	[0.450, 0.487]	[0.926, 0.930]	[0.505, 0.556]	[0.806, 0.842]
Redes Neurais Artificiais	[0.365, 0.401]	[0.913, 0.918]	[0.374, 0.412]	[0.743, 0.790]
SVM	[0.338, 0.380]	[0.911, 0.915]	[0.339, 0.384]	[0.727, 0.759]
<i>Naive Bayes</i>	[0.300, 0.353]	[0.856, 0.871]	[0.270, 0.316]	[0.707, 0.754]

Tabela 4.7: Melhores desempenhos por algoritmo e métrica obtidos com modelos treinados com COPs do evento geada.

Algoritmo	Métricas			
	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
<i>Random Forest</i>	<b>0.408</b>	<b>0.941</b>	<b>0.445</b>	<b>0.781</b>
Redes Neurais Artificiais	0.313	0.933	0.345	0.749
SVM	0.260	0.928	0.303	0.703
<i>Naive Bayes</i>	0.308	0.906	0.277	0.729

Tabela 4.8: Intervalos de confiança das médias das métricas obtidas com os modelos do evento geada (Tabela 4.7).

Algoritmo	Intervalo de Confiança ( $\alpha = 0.05$ , $t_{\alpha/2} = 2.262$ , $df = 9$ )			
	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
<i>Random Forest</i>	[0.354, 0.462]	[0.936, 0.946]	[0.393, 0.497]	[0.751, 0.810]
Redes Neurais Artificiais	[0.235, 0.392]	[0.926, 0.940]	[0.259, 0.396]	[0.690, 0.766]
SVM	[0.203, 0.316]	[0.922, 0.933]	[0.239, 0.367]	[0.662, 0.743]
<i>Naive Bayes</i>	[0.271, 0.344]	[0.897, 0.915]	[0.231, 0.323]	[0.705, 0.754]

Observa-se das informações das tabelas que os modelos gerados pelo *Random Forest* novamente superaram os demais em todas as métricas e, nesse caso, para todos os tipos de evento. Os valores máximos obtidos para as métricas  $F_1$ -score e precisão média foram, respectivamente, 0,500 e 0,575 para seca, 0,467 e 0,530 para chuva excessiva e 0,408 e 0,445 para geada. A ordem de desempenho dos demais algoritmos foi a mesma observada no modelos monolíticos: RNAs em segundo lugar, SVM em terceiro e *Naive Bayes* com os piores resultados — as exceções são as métricas  $F_1$ -score e ROC AUC para o SVM, cujos valores ficaram abaixo dos observados para o *Naive Bayes*. É importante destacar, contudo, que os intervalos de confiança calculados para as médias das métricas muitas vezes se sobrepõem, especialmente para os modelos do evento geada, o que impossibilita afirmar, apenas pela observação dos intervalos, que as diferenças sejam estatisticamente significativas.

As Tabelas A.3 a A.5 apresentam os valores dos hiperparâmetros e os conjuntos de dados utilizados no treinamento dos modelos para os eventos de seca, chuva excessiva e geada. Para cada modelo são apresentados também os resultados obtidos nas quatro métricas. Efeitos semelhantes aos observados nos modelos monolíticos decorrentes da utilização do SMOTE-TK são visto nos resultados dos modelos hierárquicos. Tanto as RNAs quanto o *Random Forest* produziram modelos com melhor desempenho no que concerne à métrica  $F_1$ -score quando o conjunto de dados de treinamento havia sido submetido às técnicas de *oversampling* e *undersampling*. A exceção foi o modelo gerado para o evento geada a partir das RNAs. Nesse caso, o melhor resultado para a métrica  $F_1$ -score foi obtido por um modelo cujos dados de treinamento não haviam sido balanceados.

Quanto ao PCA, nove modelos tiveram o desempenho melhorado com sua utilização, sendo todos eles derivados ou do *Naive Bayes* ou do SVM. Diferentemente do ocorrido com os modelos monolíticos, em que só a acurácia foi beneficiada com a utilização do PCA, nesta abordagem os modelos apresentaram ganhos em diversas métricas com sua aplicação.

Por fim, novamente as transformações quantílica e de potência resultaram nos modelos de maior desempenho dentre os gerados a partir do *Naive Bayes*. Não houve qualquer

modelo do *Naive Bayes* treinado sobre dados não transformados que tivesse apresentado desempenho superior àqueles cujos dados de treinamento haviam sido transformados. É conveniente ressaltar que, dos 12 modelos selecionados, 9 foram treinados com dados modificados com a transformação quantílica, enquanto 3 foram treinados com dados modificados com a transformação de potência, o que revela, portanto, a proeminência da transformação quantílica frente à de potência.

Na próxima seção, relataremos os resultados obtidos sobre os dados de teste para os modelos selecionados na validação cruzada. Apresentaremos também uma comparação entre os melhores modelos da abordagem monolítica e os da hierárquica.

## 4.2 Resultados Dados de Teste

A fim de estimar a capacidade de generalização dos modelos selecionados na etapa anterior, considerando, para isso, que pode ter ocorrido sobreajuste na validação cruzada [84], avaliamos o desempenho dos modelos quando aplicados sobre os dados de teste, os quais são constituídos de 10.353 instâncias, *i.e.*, 25% do total de observações.

A avaliação foi realizada com os modelos das duas abordagens. Na avaliação dos modelos monolíticos, os dados de teste foram aplicados integralmente. Já na avaliação dos modelos da abordagem hierárquica, primeiro foram separadas as instâncias conforme o tipo de evento, em seguida os modelos de cada evento foram avaliados com seus respectivos subconjuntos de teste.

A seguir, mostraremos os resultados obtidos nos dados de teste para os modelos selecionados na validação cruzada. Primeiro serão reportados os desempenhos dos modelos monolíticos. Em seguida, serão reportados os desempenhos dos modelos de abordagem hierárquica. Por fim, serão comparados os melhores modelos das duas abordagens.

### 4.2.1 Abordagem monolítica

Foram calculadas as quatro métricas para os modelos monolíticos gerados ao final da validação cruzada, utilizando, para a avaliação, os dados de teste. Os valores das métricas estão dispostos na Tabela 4.9. Comparando os valores obtidos nos dados de teste com os da validação cruzada (Tabela 4.1), observa-se que os desempenhos são similares e, portanto, assumimos que não houve sobreajuste dos dados.



Tabela 4.9: Desempenho dos modelos monolíticos estimado a partir dos dados de teste.

Algoritmo	Métricas			
	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
<i>Random Forest</i>	<b>0.495</b>	<b>0.919</b>	<b>0.549</b>	<b>0.833</b>
Redes Neurais Artificiais	0.406	0.907	0.426	0.780
SVM	0.342	0.904	0.385	0.739
<i>NaiveBayes</i>	0.316	0.888	0.296	0.728

As quatro métricas calculadas para cada um dos modelos podem ser visualizadas na Tabela A.6. Já a Tabela 4.10 traz a matriz de confusão para a métrica  $F_1$ -score. Nela, as COPs indeferidas correspondem ao número 1.

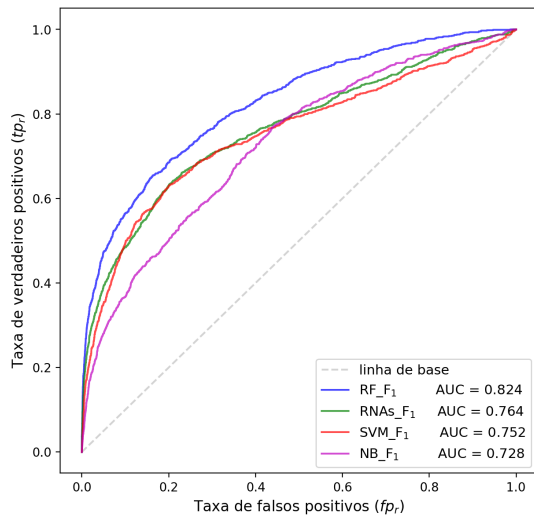
Tabela 4.10: Matriz de confusão para os modelos monolíticos com melhor resultado na métrica  $F_1$ -score.

		<i>Random Forest</i>		RNAs		SVM		<i>Naive Bayes</i>	
		<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>
Real	<b>0</b>	8846	424	8194	1076	9039	231	7550	1720
	<b>1</b>	587	496	533	550	812	271	558	525

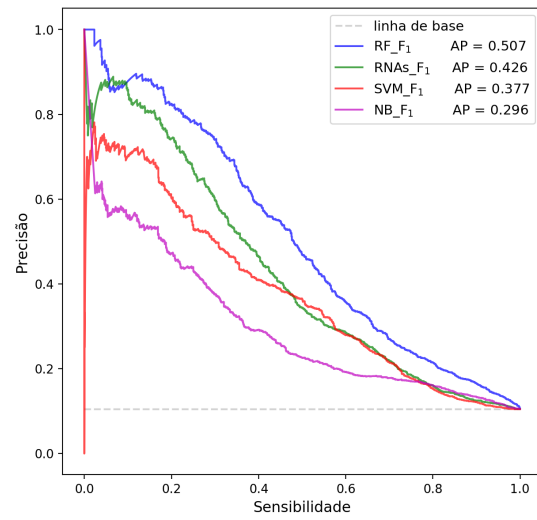
Foram plotadas a curva ROC e a curva *precisão-sensibilidade* para comparação dos melhores modelos de cada algoritmo. As plotagens foram efetuadas para os modelos que apresentaram melhor desempenho na precisão média. A escolha da precisão média como métrica de referência para otimização dos algoritmos e, conseqüente, para comparação dos desempenhos por meio da curva *precisão-sensibilidade* é justificada pela necessidade de obtenção de um classificador com precisão igual ou superior a 80%, conforme comunicado pela área de negócio, associada ao fato de que, quanto maior for a precisão média, maior será a chance de incremento da sensibilidade dada uma precisão e vice-versa.

Nesta subseção, serão apresentadas as duas curvas para as métricas  $F_1$ -score, acurácia, precisão média e ROCAUC por meio das Figuras 4.1, 4.2, 4.3 e 4.4, nessa ordem, a fim de demonstrar como a métrica de interesse pode alterar o comportamento das curvas. Nas seções seguintes, serão apresentadas somente as curvas ROC e *precisão-sensibilidade* para a métrica precisão média, sendo as demais exibidas no Apêndice B.

Para a curva *precisão-sensibilidade*, a linha de base foi definida em 0,10, o que corresponde à porcentagem de instâncias da classe de interesse (COPs indeferidas).

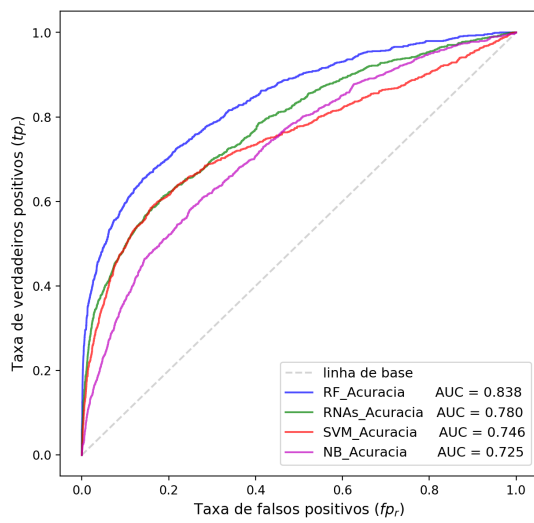


(a) Curva ROC.

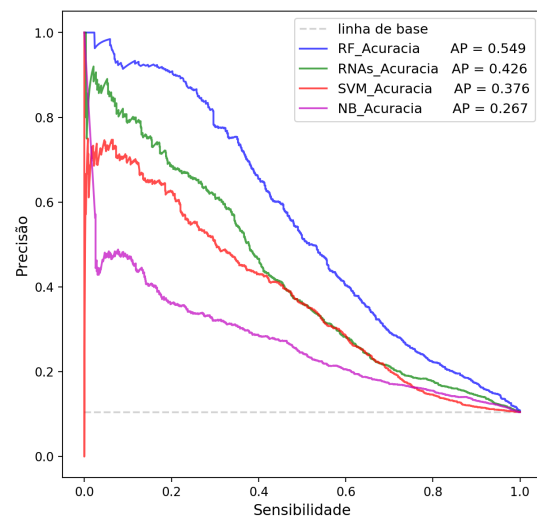


(b) Curva precisão-sensibilidade.

Figura 4.1: Curvas (a) ROC e (b) precisão-sensibilidade para a métrica  $F_1$ -score.

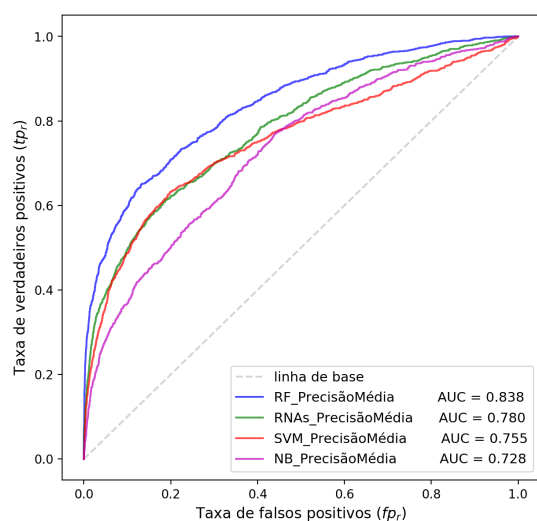


(a) Curva ROC.

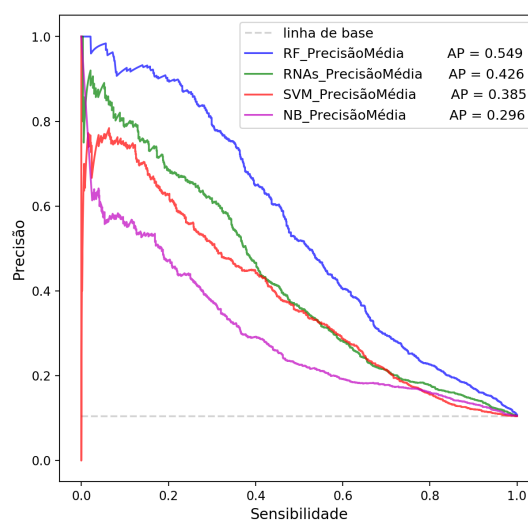


(b) Curva precisão-sensibilidade.

Figura 4.2: Curvas (a) ROC e (b) precisão-sensibilidade para a métrica acurácia.

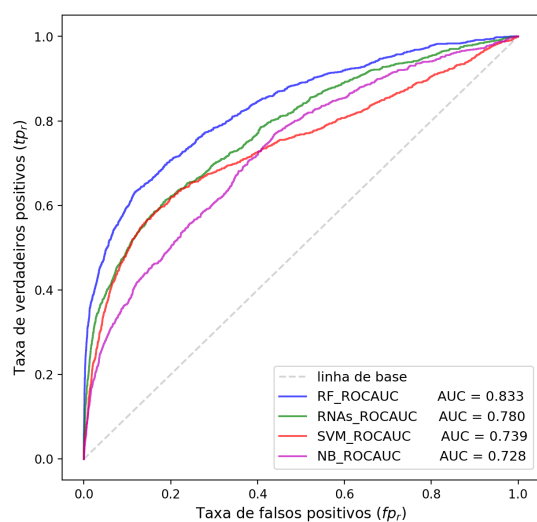


(a) Curva ROC.

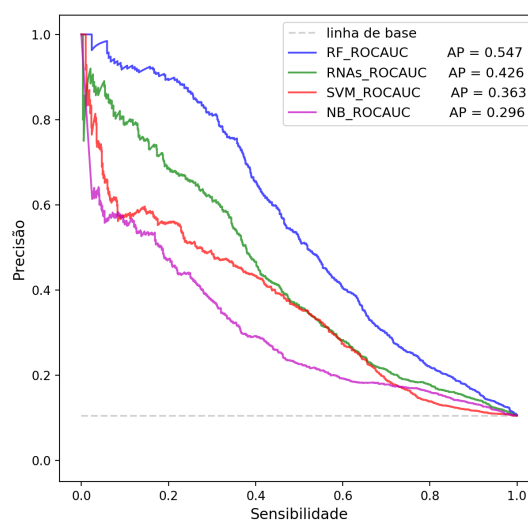


(b) Curva precisão-sensibilidade.

Figura 4.3: Curvas (a) ROC e (b) precisão-sensibilidade para a métrica precisão média.



(a) Curva ROC.



(b) Curva precisão-sensibilidade.

Figura 4.4: Curvas (a) ROC e (b) precisão-sensibilidade para a métrica ROCAUC.

Na próxima subseção, apresentaremos os resultados para a abordagem hierárquica.

## 4.2.2 Abordagem hierárquica

Assim como na abordagem monolítica, foram calculadas as quatro métricas com dados de teste para os modelos hierárquicos gerados ao final da validação cruzada. Os valores das métricas para os três tipos de eventos estão dispostos nas Tabelas 4.11, 4.12 e 4.13.

Comparando os valores obtidos nos dados de teste com os da validação cruzada (Tabelas 4.3, 4.5 e 4.7), observa-se que os desempenhos são similares, embora sejam encontradas diferenças maiores do que aquelas observadas nos modelos monolíticos. Uma hipótese para o aumento da diferença entre as estimativas da validação cruzada e as dos dados de teste é que, por conta da redução do número de observações decorrentes da separação dos tipos de eventos, houve um aumento da variância.

Tabela 4.11: Desempenho dos modelos hierárquicos para o evento seca estimados a partir dos dados de teste.

Algoritmo	Métricas			
	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
<i>Random Forest</i>	<b>0.495</b>	<b>0.908</b>	<b>0.555</b>	<b>0.838</b>
Redes Neurais Artificiais	0.389	0.890	0.402	0.764
SVM	0.385	0.890	0.379	0.747
<i>Naive Bayes</i>	0.344	0.872	0.321	0.712

Tabela 4.12: Desempenho dos modelos hierárquicos para o evento chuva excessiva estimados a partir dos dados de teste.

Algoritmo	Métricas			
	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
<i>Random Forest</i>	<b>0.467</b>	<b>0.926</b>	<b>0.535</b>	<b>0.830</b>
Redes Neurais Artificiais	0.352	0.912	0.344	0.763
SVM	0.317	0.911	0.337	0.737
<i>Naive Bayes</i>	0.328	0.864	0.270	0.753

Tabela 4.13: Desempenho dos modelos hierárquicos para o evento geada estimados a partir dos dados de teste.

Algoritmo	Métricas			
	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
<i>Random Forest</i>	<b>0.385</b>	<b>0.938</b>	<b>0.436</b>	<b>0.801</b>
Redes Neurais Artificiais	0.304	0.931	0.387	0.712
SVM	0.280	0.927	0.340	0.709
<i>Naive Bayes</i>	0.344	0.907	0.307	0.745

As quatro métricas calculadas para cada um dos modelos podem ser visualizadas nas Tabelas A.7 a A.9. Já as Tabelas 4.14, 4.15 e 4.16 trazem as matrizes de confusão para a métrica  $F_1$ -score, cada qual construída a partir de um tipo de evento. Novamente, as COPs indeferidas correspondem ao número 1.

Tabela 4.14: Matriz de confusão para os modelos hierárquicos com melhor resultado na métrica  $F_1$ -score construídos com base no evento seca.

		<i>Random Forest</i>		RNAs		SVM		<i>Naive Bayes</i>	
		<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>
Real	<b>0</b>	4806	253	4230	829	4533	526	4069	990
	<b>1</b>	368	304	310	362	386	286	327	345

Tabela 4.15: Matriz de confusão para os modelos hierárquicos com melhor resultado na métrica  $F_1$ -score construídos com base no evento chuva excessiva.

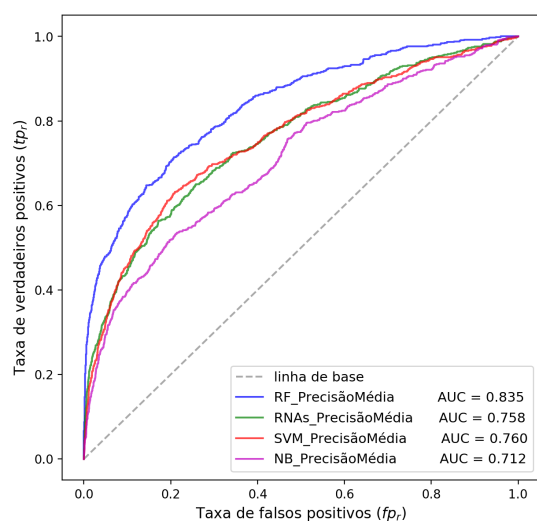
		<i>Random Forest</i>		RNAs		SVM		<i>Naive Bayes</i>	
		<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>
Real	<b>0</b>	3209	156	2821	544	3223	142	3037	327
	<b>1</b>	193	153	156	190	254	92	214	132

Tabela 4.16: Matriz de confusão para os modelos hierárquicos com melhor resultado na métrica  $F_1$ -score construídos com base no evento geada.

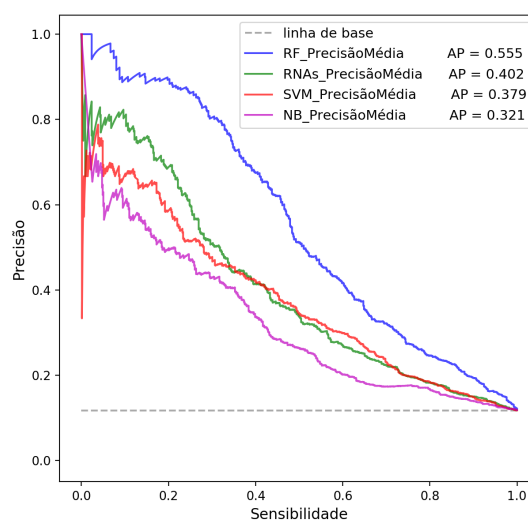
		<i>Random Forest</i>		RNAs		SVM		<i>Naive Bayes</i>	
		<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>
Real	<b>0</b>	738	26	759	5	734	30	672	92
	<b>1</b>	41	21	50	12	47	15	30	32

Tal como na Subseção 4.2.1, foram plotadas a curva ROC e a curva *precisão-sensibilidade* para comparação dos resultados dos melhores modelos de cada algoritmo, porém, desta vez, as curvas foram construídas por tipo de evento, conforme estrutura da abordagem hierárquica. As linhas de base das curvas *precisão-sensibilidade* foram definidas com base nas porcentagens de COPs indeferidas encontradas em cada grupo de evento.

A Figura 4.5 exhibe as duas curvas para os modelos do evento *seca* otimizados para a métrica *precisão média*. Para a curva *precisão sensibilidade*, a linha de base foi definida em 0,117. As curvas para as demais métricas podem ser observadas nas Figuras B.1 a B.3.



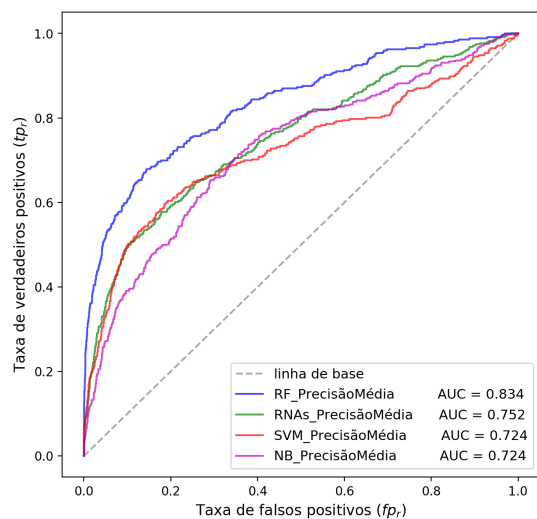
(a) Curva ROC.



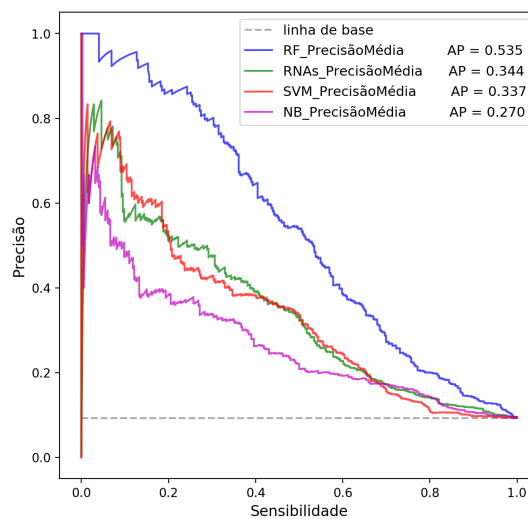
(b) Curva precisão-sensibilidade.

Figura 4.5: Curvas (a) ROC e (b) precisão-sensibilidade dos modelos do evento seca que obtiveram os melhores desempenho na métrica precisão média.

Já a Figura 4.6 exibe as duas curvas para os modelos do evento *chuva excessiva*. A linha de base para a curva precisão-sensibilidade foi definida em 0,093. As Figuras B.4 a B.6 trazem as duas curvas para as demais métricas.



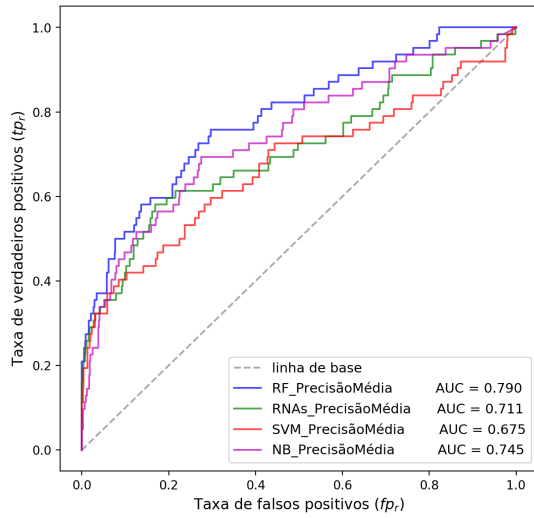
(a) Curva ROC.



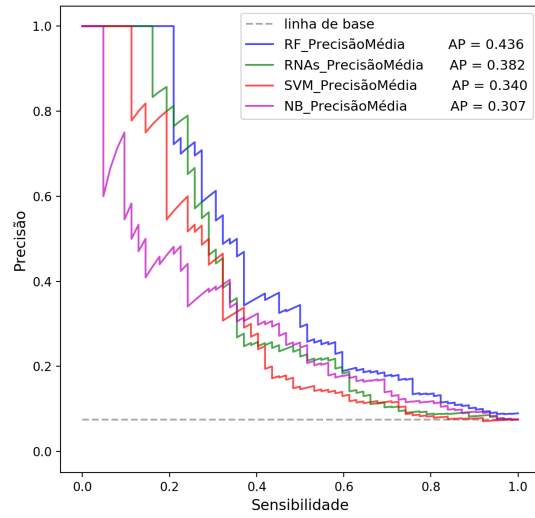
(b) Curva precisão-sensibilidade.

Figura 4.6: Curvas (a) ROC e (b) precisão-sensibilidade dos modelos do evento chuva excessiva que obtiveram os melhores desempenho na métrica precisão média.

Por fim, a figura 4.7 exibe as duas curvas para os modelos do evento *geada*. A linha de base para a curva precisão-sensibilidade foi definida em 0,075. As Figuras B.7 a B.9 exibem as curvas para as demais métricas.



(a) Curva ROC.



(b) Curva precisão-sensibilidade.

Figura 4.7: Curvas (a) ROC e (b) precisão-sensibilidade dos modelos do evento geada que obtiveram os melhores desempenho na métrica precisão média.

Na próxima subseção, compararemos os resultados dos melhores modelos das duas abordagens.

### 4.2.3 Comparação melhores modelos das abordagens monolítica e hierárquica

Como dito anteriormente, os conjuntos de dados utilizados nos modelos da abordagem monolítica divergem daqueles utilizados na abordagem hierárquica, pois, no primeiro caso, os dados foram aplicados integralmente, *i.e.*, com todos os tipos de eventos, enquanto, no segundo, os dados foram separados por tipo de evento. Além disso, conforme foi reportado na Seção 3.4, não houve modelo hierárquico treinado para eventos de variação excessiva de temperatura, logo, não há estimativa de desempenho para esse tipo de evento dentro da abordagem hierárquica. Conseqüentemente, não é possível realizar a comparação direta entre os resultados das duas abordagens.

Para realizar a comparação dos desempenhos entre as abordagens, combinamos as saídas de classificação dos modelos hierárquicos de modo que o produto final contivesse a classificação dos três tipos de eventos, semelhantemente ao que ocorre nos modelos monolíticos, com a ressalva de que nos dados utilizados pelos modelos monolíticos havia também o evento variação excessiva de temperatura. Depois de combinadas as saídas dos modelos hierárquicos, as quatro métricas foram, então, recalculadas.

Recalculamos também os resultados dos modelos monolíticos, mas dessa vez removendo-se primeiro as instâncias do evento variação excessiva de temperatura.

Dessarte, tanto o desempenho dos modelos monolíticos quanto o dos hierárquicos puderam ser estimados sobre o mesmo conjunto de dados.

Os recálculos da abordagem hierárquica e da monolítica foram efetuados apenas para os modelos que obtiveram o melhor resultado em uma das métricas na validação cruzada, não importando o algoritmo. Dessa forma, foram recalculados os desempenhos de quatro modelos em cada abordagem — um para cada métrica. Os modelos aplicados nesta etapa foram os gerados a partir do *Random Forest*, visto que foi esse o algoritmo vencedor em ambas as abordagens.

A Tabela 4.17 apresenta os resultados da abordagem hierárquica combinada e os da monolítica ajustada (sem evento variação excessiva de temperatura) para cada métrica. O detalhamento dos desempenhos pode ser observado na Tabela A.10, em que são apresentados os resultados por métrica de otimização. Já a Tabela 4.18 apresenta a matriz de confusão gerada com a aplicação dos modelos otimizados para a métrica  $F_1$ -score de ambas as abordagens.

Tabela 4.17: Desempenho dos modelos das abordagens hierárquica e monolítica calculado sobre o mesmo conjunto dados de teste.

Abordagem	Métricas			
	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
Monolítica Ajustada	<b>0.496</b>	<b>0.918</b>	<b>0.550</b>	0.834
Hierárquica Combinada	0.480	0.917	0.540	<b>0.835</b>

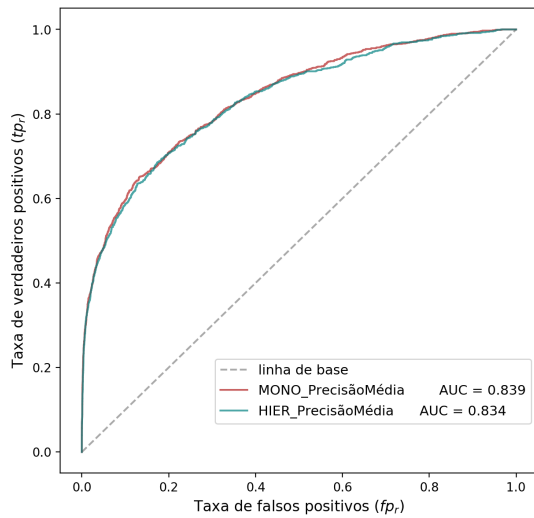
Tabela 4.18: Matriz de confusão para os melhores modelos das abordagens hierárquica e monolítica com relação à métrica  $F_1$ -score.

		Monolítica Ajustada		Hierárquica Combinada	
		0	1	0	1
Real	0	8768	420	8753	435
	1	585	495	602	478

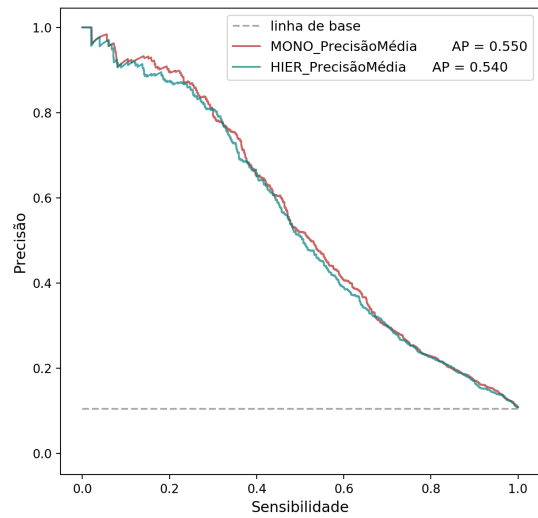
Comparando os valores da Tabela 4.9, algoritmo *Random Forest*, com os da Tabela 4.9, abordagem Monolítica Ajustada, verifica-se que a remoção das COPs do evento “variação excessiva de temperatura” alterou os resultados de forma quase imperceptível. Isso já era esperado em razão do pequeno número de instâncias que esse evento representa em relação ao total de instâncias presentes no conjunto de dados de teste original.

A Figuras 4.8 exibe as curvas ROC e precisão-sensibilidade dos resultados recalculados para os melhores modelos de cada abordagem no que concerne a métrica precisão média. A linha de base da curva precisão-sensibilidade foi estabelecida em 0,105. As curvas para as demais métricas de referência estão disponíveis nas Figuras B.10 a B.12.





(a) Curva ROC.



(b) Curva precisão-sensibilidade.

Figura 4.8: Curvas (a) ROC e (b) precisão-sensibilidade dos resultados das abordagens monolítica e hierárquica recalculados a partir do mesmo conjunto de dados. Os modelos utilizados foram aqueles otimizados para a métrica precisão média.

A partir da Tabela 4.17 e da Figura 4.8a é possível observar que os resultados recalculados das abordagens são muito próximos. As pequenas diferenças encontradas podem ser atribuídas a aleatoriedade existente no processo de geração das árvores de decisão. Aplicando o princípio da Navalha de Occam, foi selecionado o modelo construído na abordagem monolítica, visto que ele apresenta menor complexidade no processo de classificação de uma COP.

Por último, buscou-se identificar qual seria a sensibilidade máxima, dado um limite inferior para precisão, do modelo com a melhor precisão média. Visto que o modelo vencedor foi o gerado pela abordagem monolítica, foi, novamente, plotada a curva precisão-sensibilidade com os resultados do modelo monolítico otimizado para a métrica precisão média. Porém, desta vez, acrescentou-se o ponto em que a sensibilidade é a maior possível dada uma precisão de 80%. O ponto laranja entre as duas linhas pontilhadas da Figura 4.9, cujo valor da sensibilidade é 0.30, representa essa situação.

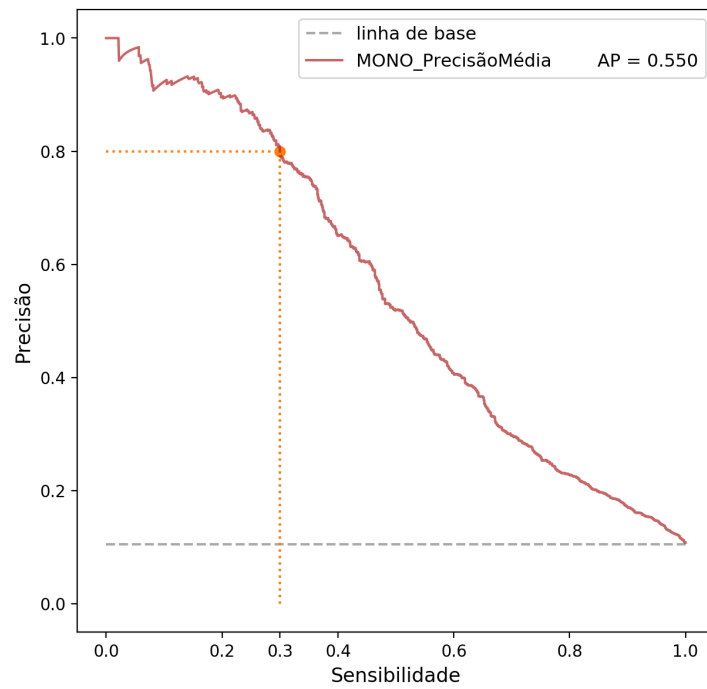


Figura 4.9: Curva precisão-sensibilidade com ponto (laranja) identificando a sensibilidade máxima de 30% para uma precisão de 80%.

# Capítulo 5

## Conclusão

Neste trabalho foi realizado estudo e aplicação de técnicas de aprendizagem de máquina para criação de modelos classificatórios destinados à identificação de COPs irregulares ou indevidas geradas no contexto do Proagro. Para treinamento e validação dos modelos, foram utilizados dados dos sistemas Sicor, Sisdagro e Agritempo. Os dados coletados compreendem aqueles gerados entre janeiro de 2017 e janeiro de 2019.

O modelo de referência CRISP-DM foi utilizado no processo de mineração de dados. Foram empregadas as atividades das etapas de entendimento do negócio, compreensão dos dados, preparação dos dados e modelagem. Na fase de modelagem, foram aplicados os algoritmos *Naive Bayes*, SVM, RNAs e *Random Forest*. Para escolha do melhor modelo, foram utilizadas as métricas  $F_1$ -score e *precisão média* calculadas na validação cruzada com dez partições. O desempenho final foi estimado sobre o conjunto de dados de teste, cujas instâncias representam 25% do conjunto de dados original.

Duas abordagens foram utilizadas na construção dos modelos: monolítica e hierárquica. Em ambas o algoritmo *Random Forest* superou os demais, o que pode ser facilmente evidenciado por meio de análise das curvas ROC e precisão-sensibilidade apresentadas no Capítulo 4 e no Apêndice B. Em todos os casos apresentados, as curvas geradas pelo *Random Forest* superaram as curvas produzidas pelos demais algoritmos.

A abordagem monolítica, que adota um único modelo para todos os tipos de evento climático, apresentou resultados ligeiramente superiores aos da abordagem hierárquica, o que pode ter ocorrido em razão da diferença no número de instâncias de treinamento utilizadas em cada abordagem. Em razão do processo de separação dos eventos empregado na abordagem hierárquica, os tamanhos de seus conjuntos de treinamento e teste foram inferiores aos dos conjuntos utilizados na abordagem monolítica, conseqüentemente, aumentando as chances de sobreajuste dos modelos.

O modelo vencedor, o qual foi gerado na abordagem monolítica a partir do algoritmo *Random Forest*, obteve precisão média de 0.550 nos dados de teste. Para alcance de

uma precisão igual ou maior que 80% com o uso do modelo, a sensibilidade máxima será de 30%, ou seja, para que ao menos 80% das instâncias classificadas pelo modelo como irregulares sejam de fato irregulares, o percentual de instâncias irregulares recuperadas será igual ou inferior a 30%.

Além da criação de um modelo classificatório que possa contribuir com a identificação de COPs irregulares e, conseqüentemente, tornar mais preciso e tempestivo o trabalho da equipe de monitoramento, a pesquisa produziu um conjunto de *scripts* que automatizam o processo de coleta e tratamento dos dados oriundo de fontes externas (Sisdagro e Agritempo).

Outrossim, a partir dos dados do temperatura mínima extraídos do Agritempo, foi possível identificar 75 COPs deferidas que haviam sido emitidas em razão de geada em que a temperatura mínima para o período de ocorrência do evento no local da gleba estava acima de 9° C, chegando, em alguns casos, a ser superior a 16° C. Esses casos foram reportados à equipe de monitoramento para análise. Vislumbra-se a possibilidade de utilização de filtros baseados em regras que identifiquem e reportem automaticamente situações semelhantes a essas.

Nos próximos meses, deverão ser conduzidas as atividades das etapas de avaliação e implantação. Os modelos serão validados pela área de negócio e, uma vez definida a estratégia de utilização, o modelo deverá ser implantado. Durante a implantação, será necessário construir interfaces que permitam a consulta pelos analistas das COPs classificadas juntamente com os dados capturados dos sistemas Agritempo e Sisdagro.

Como trabalho futuro, espera-se pode fazer uso dos dados de altitude disponíveis na base do Sicor. Acreditamos que seja possível confrontar a informação de altitude da base com as estimativas produzidas pela ferramenta *Google Earth*. Para os casos em que houvesse discrepância elevada, seriam realizadas análises individuais, para os demais casos seriam mantidas as informações do Sicor. Por fim, esperamos incrementar os dados com informações do Índice de Vegetação por Diferença Normalizada (NDVI, do inglês *Normalized Difference Vegetation Index*) disponíveis no Sistema de Análise Temporal da Vegetação (SATVeg) da Embrapa<sup>1</sup>. Com o NDVI, acreditamos que seja possível estimar se e quando ocorreu o plantio. Uma vez estimada a data de plantio, confrontaríamos a estimativa com a informação disponibilizada pelo Sicor.

---

<sup>1</sup><https://www.satveg.cnptia.embrapa.br>

# Referências

- [1] Chengguo Weng, Lysa Porth, Ken Seng Tan, and Ryan Samaratunga. Modelling the Sustainability of the Canadian Crop Insurance Program: A Reserve Fund Process Under a Public–Private Partnership Model. *The Geneva Papers on Risk and Insurance - Issues and Practice*, 42(2):226–246, April 2017. ISSN 1018-5895, 1468-0440. doi: 10.1057/s41288-017-0044-5. URL <http://link.springer.com/10.1057/s41288-017-0044-5>. 1, 6, 7
- [2] Programa de Garantia da Atividade Agropecuária – PROAGRO: Relatório Circunstanciado - 1991 a 1996. Technical report, BANCO CENTRAL DO BRASIL, 1997. URL <https://www.bcb.gov.br/htms/proagro/1996/re101.asp>. 1, 2
- [3] BRASIL. Decreto n. 175, de 10 de maio de 1991., . p. 13661, 11 jul. 1991. 1, 2
- [4] BRASIL. Lei n. 8.171, de 17 de janeiro de 1991, . p. 4477, 12 mar. 1991. 2
- [5] Christopher M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, 2006. ISBN 978-0-387-31073-2. 3, 9, 10, 25
- [6] Programa de Garantia da Atividade Agropecuária PROAGRO: Relatório Circunstanciado 2013 a 2016. Technical report, BANCO CENTRAL DO BRASIL, 2016. URL <https://www.bcb.gov.br/htms/proagro/1996/re101.asp>. Brasília. 4
- [7] David Coderre and Royal Canadian Mounted Police. Global technology audit guide: continuous auditing implications for assurance, monitoring, and risk assessment. *The Institute of Internal Auditors*, pages 1–34, 2005. 4
- [8] Milton Boyd, Jeffrey Pai, Zhang Qiao, and Wang Ke. Crop Insurance Principles and Risk Implications for China. *Human and Ecological Risk Assessment: An International Journal*, 17(3):554–565, May 2011. ISSN 1080-7039, 1549-7860. doi: 10.1080/10807039.2011.571072. URL <http://www.tandfonline.com/doi/abs/10.1080/10807039.2011.571072>. 6
- [9] Pedro Abel Vieira Junior. Um Modelo Integrado de Gestão do Risco Agrícola para o Brasil. 4(8):40, 2009. 6
- [10] Ke Wang, Qiao Zhang, Shingo Kimura, and Suraya Akter. Is the crop insurance program effective in China? Evidence from farmers analysis in five provinces. *Journal of Integrative Agriculture*, 14(10):2109–2120, October 2015. ISSN 20953119. doi: 10.1016/S2095-3119(14)60842-X. URL <http://linkinghub.elsevier.com/retrieve/pii/S209531191460842X>. 6

- [11] Joseph W. Glauber, Keith J. Collins, and Peter J. Barry. Crop insurance, disaster assistance, and the role of the federal government in providing catastrophic risk protection. *Agricultural Finance Review*, 62(2):81–101, November 2002. ISSN 0002-1466. doi: 10.1108/00214900280001131. URL <https://www.emeraldinsight.com/doi/10.1108/00214900280001131>. 7
- [12] Andrew Schmitz. Canadian Agricultural Programs and Policy in Transition. *Canadian Journal of Agricultural Economics/Revue canadienne d'agroeconomie*, 56(4):371–391, December 2008. ISSN 00083976, 17447976. doi: 10.1111/j.1744-7976.2008.00136.x. URL <http://doi.wiley.com/10.1111/j.1744-7976.2008.00136.x>. 7
- [13] Jesús Antón, Shingo Kimurai, and Roger Martini. Risk Management in Agriculture in Canada. OECD Food, Agriculture and Fisheries Papers 40, February 2011. URL [https://www.oecd-ilibrary.org/agriculture-and-food/thematic-review-on-risk-management-canada\\_5kgj0d6189wg-en](https://www.oecd-ilibrary.org/agriculture-and-food/thematic-review-on-risk-management-canada_5kgj0d6189wg-en). 7
- [14] Roderick M. Rejesus, Bertis B. Little, and Ashley C. Lovell. Using data mining to detect crop insurance fraud: is there a role for social scientists? *Journal of Financial Crime*, 12(1):24–32, January 2005. ISSN 1359-0790. doi: 10.1108/13590790510625052. URL <http://www.emeraldinsight.com/doi/10.1108/13590790510625052>. 8, 40
- [15] Bertis B. Little, Walter L. Johnston, Ashley C. Lovell, Roderick M. Rejesus, and Steve A. Steed. Collusion in The U.S. Crop Insurance Program: Applied Data Mining. In Robert Grossman, Jiawei Han, Vipin Kumar, Heikki Mannila, and Rajeev Motwani, editors, *Proceedings of the 2002 SIAM International Conference on Data Mining*, pages 583–597. Society for Industrial and Applied Mathematics, Philadelphia, PA, April 2002. ISBN 978-0-89871-517-0 978-1-61197-272-6. doi: 10.1137/1.9781611972726.34. URL <http://epubs.siam.org/doi/abs/10.1137/1.9781611972726.34>. 8
- [16] Rekha Bhowmik. Data Mining Techniques in Fraud Detection. *Journal of Digital Forensics, Security and Law*, 2008. ISSN 15587223. doi: 10.15394/jdfsl.2008.1040. URL <http://commons.erau.edu/jdfsl/vol3/iss2/3/>. 8
- [17] R.S Michalski, J.G Carbonell, and T.M Mitchell. *Machine Learning An Artificial Intelligence Approach*. Springer Berlin, Berlin, 2013. ISBN 978-3-662-12407-9. OCLC: 864590508. 8, 9
- [18] Jaime G. Carbonell, Ryszard S. Michalski, and Tom M. Mitchell. AN OVERVIEW OF MACHINE LEARNING. In *Machine Learning*, pages 3–23. Elsevier, 1983. ISBN 978-0-08-051054-5. doi: 10.1016/B978-0-08-051054-5.50005-4. URL <http://linkinghub.elsevier.com/retrieve/pii/B9780080510545500054>. 9
- [19] I. H. Witten, Eibe Frank, and Mark A. Hall. *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann series in data management systems. Morgan Kaufmann, Burlington, MA, 3rd ed edition, 2011. ISBN 978-0-12-374856-0. OCLC: ocn262433473. 9, 10, 26, 27

- [20] Rüdiger Wirth and Jochen Hipp. CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, pages 29–39. Citeseer, 2000. 10
- [21] Sergio Moro, Raul Laureano, and Paulo Cortez. Using data mining for bank direct marketing: An application of the crisp-dm methodology. In *Proceedings of European Simulation and Modelling Conference-ESM'2011*, pages 117–121. Eurosis, 2011. 10
- [22] Ana Azevedo and Manuel Filipe Santos. KDD, SEMMA and CRISP-DM: a parallel overview. In *IADIS European Conf. Data Mining*, 2008. 10
- [23] Jie Cai, Jiawei Luo, Shulin Wang, and Sheng Yang. Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79, July 2018. ISSN 09252312. doi: 10.1016/j.neucom.2017.11.077. URL <http://linkinghub.elsevier.com/retrieve/pii/S0925231218302911>. 11, 12, 13
- [24] Jiliang Tang, Salem Alelyani, and Huan Liu. Feature Selection for Classification: A Review. *Data classification: Algorithms and applications*, page 33, 2014. 11, 12
- [25] Shifei Ding, Hong Zhu, Weikuan Jia, and Chunyang Su. A survey on feature extraction for pattern recognition. *Artificial Intelligence Review*, 37(3):169–180, March 2012. ISSN 0269-2821, 1573-7462. doi: 10.1007/s10462-011-9225-y. URL <http://link.springer.com/10.1007/s10462-011-9225-y>. 12, 13
- [26] Karl Pearson. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. 13
- [27] Sang-Bum Kim, Kyoung-Soo Han, Hae-Chang Rim, and Sung Hyon Myaeng. Some Effective Techniques for Naive Bayes Text Classification. *IEEE Transactions on Knowledge and Data Engineering*, 18(11):1457–1466, November 2006. ISSN 1041-4347. 13, 27
- [28] N. A. Campbell and W. R. Atchley. The Geometry of Canonical Variate Analysis. *Systematic Biology*, 30(3):268–280, September 1981. ISSN 1063-5157, 1076-836X. doi: 10.1093/sysbio/30.3.268. URL <https://academic.oup.com/sysbio/article-lookup/doi/10.1093/sysbio/30.3.268>. 13, 14
- [29] Sofia Visa and Anca Ralescu. Issues in Mining Imbalanced Data Sets - A Review Paper. page 7, 2005. 16
- [30] Wei Mei Zhi, Hua Ping Guo, and Ming Fan. Sample Size on the Impact of Imbalance Learning. *Advanced Materials Research*, 756-759:2547–2551, September 2013. ISSN 1662-8985. doi: 10.4028/www.scientific.net/AMR.756-759.2547. URL <https://www.scientific.net/AMR.756-759.2547>. 16
- [31] Enislay Ramentol, Yailé Caballero, Rafael Bello, and Francisco Herrera. SMOTE-RSB \*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory. *Knowledge and Information Systems*, 33(2):245–265, November 2012. ISSN 0219-1377, 0219-3116.

- doi: 10.1007/s10115-011-0465-6. URL <http://link.springer.com/10.1007/s10115-011-0465-6>. 16, 19
- [32] Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20, June 2004. ISSN 19310145. doi: 10.1145/1007730.1007735. URL <http://portal.acm.org/citation.cfm?doi=1007730.1007735>. 16, 19, 21
- [33] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. page 37, 2002. 17, 18
- [34] Ivan Tomek. Two Modifications of CNN. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(11):769–772, November 1976. ISSN 0018-9472, 2168-2909. doi: 10.1109/TSMC.1976.4309452. URL <http://ieeexplore.ieee.org/document/4309452/>. 19
- [35] Youngtae Park. A comparison of neural net classifiers and linear tree classifiers: Their similarities and differences. *Pattern Recognition*, 27(11):1493–1503, November 1994. ISSN 00313203. doi: 10.1016/0031-3203(94)90127-9. URL <https://linkinghub.elsevier.com/retrieve/pii/0031320394901279>. 19
- [36] Miroslav Kubat and Stan Matwin. Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. In *In Proceedings of the Fourteenth International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann, 1997. 19
- [37] Rafael Marconi Ramos. *IDENTIFICAÇÃO DE COMUNICADO DE OCORRÊNCIA DE PERDAS EM SEGURO AGRÍCOLA UTILIZANDO ALGORITMOS DE INTELIGÊNCIA ARTIFICIAL*. Dissertação, Universidade Federal de Brasília, Brasília - DF, 2011. 21, 39, 40, 52, 64
- [38] Jackson Cunha Cassimiro, Andre Macedo Santana, Pedro Santos Neto, and Ricardo Lira Rabelo. Investigating the effects of class imbalance in learning the claim authorization process in the Brazilian health care market. pages 3265–3272. IEEE, May 2017. ISBN 978-1-5090-6182-2. doi: 10.1109/IJCNN.2017.7966265. URL <http://ieeexplore.ieee.org/document/7966265/>. 21, 42
- [39] Yaqi Li, Chun Yan, Wei Liu, and Maozhen Li. A principle component analysis-based random forest with the potential nearest neighbor method for automobile insurance fraud identification. *Applied Soft Computing*, July 2017. ISSN 15684946. doi: 10.1016/j.asoc.2017.07.027. URL <http://linkinghub.elsevier.com/retrieve/pii/S1568494617304386>. 21, 29, 31, 32, 44
- [40] Xiaojun Zhai, Amine Ait Si Ali, Abbes Amira, and Faycal Bensaali. MLP Neural Network Based Gas Classification System on Zynq SoC. *IEEE Access*, 4:8138–8146, 2016. ISSN 2169-3536. doi: 10.1109/ACCESS.2016.2619181. URL <http://ieeexplore.ieee.org/document/7605493/>. 21, 26



- [41] Aji Mubalrike Mubarek and Esref Adali. Multilayer perceptron neural network technique for fraud detection. In *2017 International Conference on Computer Science and Engineering (UBMK)*, pages 383–387, Antalya, October 2017. IEEE. ISBN 978-1-5386-0930-9. doi: 10.1109/UBMK.2017.8093417. URL <http://ieeexplore.ieee.org/document/8093417/>. 21
- [42] Chuan Liu, Wenyong Wang, Meng Wang, Fengmao Lv, and Martin Konan. An efficient instance selection algorithm to reconstruct training set for support vector machine. *Knowledge-Based Systems*, 116:58–73, January 2017. ISSN 09507051. doi: 10.1016/j.knosys.2016.10.031. URL <http://linkinghub.elsevier.com/retrieve/pii/S0950705116304257>. 22
- [43] HuaJuan Huang, Xiuxi Wei, and Yongquan Zhou. Twin support vector machines: A survey. *Neurocomputing*, 300:34–43, July 2018. ISSN 09252312. doi: 10.1016/j.neucom.2018.01.093. URL <http://linkinghub.elsevier.com/retrieve/pii/S0925231218302923>. 22
- [44] Hsin-Hsiung Huang, Zijing Wang, and Wingyan Chung. Efficient parameter selection for SVM: The case of business intelligence categorization. pages 158–160. IEEE, July 2017. ISBN 978-1-5090-6727-5. 23
- [45] Christina Oberlin. Sparsity in the Context of Support Vector Machines. page 10, 2004. 24
- [46] Loris Nanni and Alessandra Lumini. Coding of amino acids by texture descriptors. *Artificial intelligence in medicine*, 48:43–50, 2009. 24
- [47] Christopher M. Bishop. *Neural networks for pattern recognition*. Clarendon Press ; Oxford University Press, Oxford : New York, 1995. ISBN 978-0-19-853849-3 978-0-19-853864-6. 25, 26, 27
- [48] Guoqiang Peter Zhang. Neural Networks for Classification: A Survey. 30(4):12, 2000. 25
- [49] Kurt Hornik. Approximation Capabilities of Multilayer Feedforward Networks. page 7, 1990. 25
- [50] Michael D Richard and Richard P Lippmann. Neural network classifiers estimate Bayesian a posteriori probabilities. *Neural computation*, 3(4):461–483, 1991. 25, 26
- [51] Mohammad Reza Mohammadi, Sayed Alireza Sadrossadat, and Behzad Nouri. A brief review over neural network modeling techniques. page 4, 2017. 25, 26
- [52] F. Fernández-Navarro, Maria Angeles de la Cruz, P. A. Gutiérrez, A. Castaño, and C. Hervás-Martínez. Time series forecasting by recurrent product unit neural networks. *Neural Computing and Applications*, 29(3):779–791, February 2018. ISSN 0941-0643, 1433-3058. doi: 10.1007/s00521-016-2494-2. URL <http://link.springer.com/10.1007/s00521-016-2494-2>. 25

- [53] Franco Scarselli and Ah Chung Tsoi. Universal Approximation Using Feedforward Neural Networks: A Survey of Some Existing Methods, and Some New Results. *Neural Networks*, page 23, 1998. 26
- [54] Claude Sammut and Geoffrey I. Webb, editors. *Encyclopedia of machine learning*. Springer, New York ; London, 2010. ISBN 978-0-387-30768-8 978-0-387-34558-1 978-0-387-30164-8. OCLC: ocn651073009. 28
- [55] Leo Breiman, Jerome Friedman, Charles j. Stone, and R.A. Olshen. *Classification and regression trees*. Chapman & Hall [u.a.], Boca Raton, repr edition, 1984. ISBN 978-0-412-04841-8. OCLC: 247053926. 29
- [56] Xindong Wu. *The top ten algorithms in data mining*. Chapman & Hall, London, 2009. ISBN 978-1-4200-8964-6 978-1-4200-8965-3. OCLC: 890620202. 29
- [57] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining, Southeast Asia Edition: Concepts and Techniques*. Elsevier Science, Burlington, 2006. ISBN 978-0-08-047558-5. URL <http://qut.eblib.com.au/patron/FullRecord.aspx?p=291712>. OCLC: 1044715961. 29, 30
- [58] Min Ma. Classification and Regression Trees (CART) with rpart and rpart.plot, August 2014. URL [https://rpubs.com/minma/cart\\_with\\_rpart](https://rpubs.com/minma/cart_with_rpart). 31
- [59] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996. ISSN 0885-6125, 1573-0565. doi: 10.1007/BF00058655. URL <http://link.springer.com/10.1007/BF00058655>. 30, 31
- [60] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL <https://doi.org/10.1023/A:1010933404324>. 31
- [61] Hossin Mohammad and Sulaiman M.N. A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):01–11, March 2015. ISSN 2231007X, 22309608. doi: 10.5121/ijdkp.2015.5201. URL <http://www.airconline.com/ijdkp/V5N2/5215ijdkp01.pdf>. 33, 35
- [62] Yangguang Liu, Yangming Zhou, Shiting Wen, and Chaogang Tang. A Strategy on Selecting Performance Metrics for Classifier Evaluation:. *International Journal of Mobile Computing and Multimedia Communications*, 6(4):20–35, October 2014. ISSN 1937-9412, 1937-9404. doi: 10.4018/IJMCMC.2014100102. URL <http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/IJMCMC.2014100102>. 33, 35, 36
- [63] Jesse Davis and Mark Goadrich. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning - ICML '06*, pages 233–240, Pittsburgh, Pennsylvania, 2006. ACM Press. ISBN 978-1-59593-383-6. doi: 10.1145/1143844.1143874. URL <http://portal.acm.org/citation.cfm?doid=1143844.1143874>. 34, 37

- [64] Laszlo A. Jeni, Jeffrey F. Cohn, and Fernando De La Torre. Facing Imbalanced Data—Recommendations for the Use of Performance Metrics. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 245–251, Geneva, Switzerland, September 2013. IEEE. ISBN 978-0-7695-5048-0. doi: 10.1109/ACII.2013.47. URL <http://ieeexplore.ieee.org/document/6681438/>. 34, 35, 36
- [65] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006. ISSN 01678655. doi: 10.1016/j.patrec.2005.10.010. URL <https://linkinghub.elsevier.com/retrieve/pii/S016786550500303X>. 36, 37
- [66] Parag Singla and Pedro Domingos. Discriminative Training of Markov Logic Networks. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2, AAAI'05*, pages 868–873. AAAI Press, 2005. ISBN 1-57735-236-X. URL <http://dl.acm.org/citation.cfm?id=1619410.1619472>. event-place: Pittsburgh, Pennsylvania. 37
- [67] Mark Goadrich, Louis Oliphant, and Jude Shavlik. Learning ensembles of first-order clauses for recall-precision curves: A case study in biomedical information extraction. In *International Conference on Inductive Logic Programming*, pages 98–115. Springer, 2004. 37
- [68] Peter Flach and Meelis Kull. Precision-Recall-Gain Curves: PR Analysis Done Right. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 838–846. Curran Associates, Inc., 2015. URL <http://papers.nips.cc/paper/5867-precision-recall-gain-curves-pr-analysis-done-right.pdf>. 39
- [69] Kentaro Kuwata, Faizan Mahmood, and Ryosuke Shibasaki. Weather index for crop insurance to mitigate basis risk. pages 4656–4659. IEEE, July 2015. ISBN 978-1-4799-7929-5. doi: 10.1109/IGARSS.2015.7326867. URL <http://ieeexplore.ieee.org/document/7326867/>. 41
- [70] Norbaiti Tukiman, Norhaiza Ahmad, Suhana Mohamed, Zarith Sofiah Othman, CT Munirah Niesha Mohd Shafee, and Zairi Ismael Rizman. Credit Card Detection System Based on Rudit Approach. *International Journal on Advanced Science, Engineering and Information Technology*, 7(6):2071, December 2017. ISSN 2460-6952, 2088-5334. doi: 10.18517/ijaseit.7.6.1316. URL [http://ijaseit.insightsociety.org/index.php?option=com\\_content&view=article&id=9&Itemid=1&article\\_id=1316](http://ijaseit.insightsociety.org/index.php?option=com_content&view=article&id=9&Itemid=1&article_id=1316). 43
- [71] Stijn Viaene, Mercedes Ayuso, Montserrat Guillen, Dirk Van Gheel, and Guido Dedene. Strategies for detecting fraudulent claims in the automobile insurance industry. *European Journal of Operational Research*, 176(1):565–583, January 2007. ISSN 03772217. doi: 10.1016/j.ejor.2005.08.005. URL <https://linkinghub.elsevier.com/retrieve/pii/S0377221705006405>. 52
- [72] Fatma Ulucan Özkul and Ayşe Pamukçu. Fraud Detection and Forensic Accounting. In Kiyem Çaliyurt and Samuel O. Idowu, editors, *Emerging Fraud*, pages 19–41.

- Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-20825-6 978-3-642-20826-3. doi: 10.1007/978-3-642-20826-3\_2. URL [http://link.springer.com/10.1007/978-3-642-20826-3\\_2](http://link.springer.com/10.1007/978-3-642-20826-3_2). 52
- [73] Martha Delphino Bambini, Ariovaldo Luchiar Junior, Luciana Alvim Santos Romani, Adriano Franzoni Otavian, Luciano Vieira Koenigkan, and Silvio Roberto de Medeiros Evangelista. Manual on-line do sistema Agritempo versão 2.0, July 2015. URL [https://www.agritempo.gov.br/agritempo/arquivos/Manual\\_Agritempo.pdf](https://www.agritempo.gov.br/agritempo/arquivos/Manual_Agritempo.pdf). ISSN 1677-9274. 54
- [74] Clayton Campanhola, Gustavo Kauark Chianca, Herbert Cavalcante de Lima, José Gilberto Jardine, Tércia Zavaglia Torres, Sônia Ternes Frassetto, and Álvaro Seixas Neto. Diretoria Executiva da Embrapa. page 20. ISSN 1677-9266. URL <https://www.infoteca.cnptia.embrapa.br/bitstream/doc/8851/1/bp9.pdf>. 55
- [75] G. Dettori and B. Falcidieno. An algorithm for selecting main points on a line. *Computers & Geosciences*, 8(1):3–10, January 1982. ISSN 00983004. doi: 10.1016/0098-3004(82)90031-0. URL <https://linkinghub.elsevier.com/retrieve/pii/S0098300482900310>. 57
- [76] Instituto Nacional de Meteorologia. Balanço Hídrico, . URL <http://sisdagro.inmet.gov.br/sisdagro/app/balancoHidrico>. 58
- [77] Instituto Nacional de Meteorologia. SISDAGRO, . URL <http://sisdagro.inmet.gov.br/sisdagro/app/balancoHidrico>. 58
- [78] Daniele Soria, Jonathan M. Garibaldi, Federico Ambrogi, Elia M. Biganzoli, and Ian O. Ellis. A ‘non-parametric’ version of the naive Bayes classifier. *Knowledge-Based Systems*, 24(6):775–784, August 2011. ISSN 09507051. doi: 10.1016/j.knosys.2011.02.014. URL <https://linkinghub.elsevier.com/retrieve/pii/S0950705111000414>. 67
- [79] R. M. Sakia. The Box-Cox Transformation Technique: A Review. *The Statistician*, 41(2):169, 1992. ISSN 00390526. doi: 10.2307/2348250. URL <https://www.jstor.org/stable/10.2307/2348250?origin=crossref>. 67
- [80] I.-K. Yeo. A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4):954–959, December 2000. ISSN 0006-3444, 1464-3510. doi: 10.1093/biomet/87.4.954. URL <https://academic.oup.com/biomet/article-lookup/doi/10.1093/biomet/87.4.954>. 67
- [81] Roman Krzysztofowicz. Transformation and normalization of variates with specified distributions. *Journal of Hydrology*, 197(1-4):286–292, October 1997. ISSN 00221694. doi: 10.1016/S0022-1694(96)03276-3. URL <https://linkinghub.elsevier.com/retrieve/pii/S0022169496032763>. 67
- [82] Kuang Hongyu and Vera Lúcia Martins Sandanielo. Análise de Componentes Principais: resumo teórico, aplicação e interpretação Principal Component Analysis: theory, interpretations and applications. *Engineering and Science*, 1:8, 2016. 69

- [83] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017. URL <http://jmlr.org/papers/v18/16-365.html>. 69
- [84] Miriam Seoane Santos, Jastin Pompeu Soares, Pedro Henriques Abreu, Helder Araujo, and Joao Santos. Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches [Research Frontier]. *IEEE Computational Intelligence Magazine*, 13(4):59–76, November 2018. ISSN 1556-603X, 1556-6048. doi: 10.1109/MCI.2018.2866730. URL <https://ieeexplore.ieee.org/document/8492368/>. 69, 79

# Apêndice A

## Tabelas

Tabela A.1: Hiperparâmetros e seus valores

Algoritmos	Hiperparâmetro	Valores	Descrição
Naive Bayes	prior	None, [0.8, 0.2], [0.7, 0.3], [0.6, 0.4], [0.5, 0.5]	Define a probabilidade <i>a priori</i> de cada classe. <i>None</i> indica que ela será estimada no treinamento.
SVM	C	0.001, 0.01, 1, 10, 100, 200, 300	Custo do erro de violação das margens.
RNAs	hidden_layers	<sup>1</sup> [50], [50, 50], [50, 50, 50], [50, 50, 50, 50], [90], [90, 90], [90, 90, 90], [90, 90, 90, 90]; <sup>2</sup> [70], [70,70], [70,70,70], [70,70,70,70], [120], [120,120], [120,120,120], [120,120,120,120]	Número de camadas ocultas e de neurônios.
	dropout_hidden	0, 0.2, 0.5	Porcentagem de <i>dropout</i> nas camadas ocultas.
	dropout_input	0, 0.2	Porcentagem de <i>dropout</i> na camada de entrada.
	reg_l1_l2	[0,0], [0.0001,0.0001]	Regularização L1 e L2
Random Forest	n_estimators	200, 400, 600, 800	Número de árvores de decisão.
	max_depth	None, 20, 30, 40	Profundidade máxima de cada árvore de decisão.
	min_sample_leaf	1, 5, 10	Quantidade mínima de instâncias no nó folha.

<sup>1</sup>Parâmetros aplicados sobre os conjunto de dados com redução de dimensionalidade.

<sup>2</sup>Parâmetros aplicados sobre os conjunto de dados sem redução de dimensionalidade.

Tabela A.2: Arranjos dos hiperparâmetros dos modelos monolíticos e suas respectivas métricas calculadas na validação cruzada.

Modelos		Métricas			
Algoritmo	Hiperparâmetros / Conjuntos de Dados	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
Random Forest	depth: 20 min_leaf: 5 trees: 400 SMOTE-TK: Sim PCA: Não	<b>0.491</b>	0.902	0.515	0.819
	depth: 30 min_leaf: 1 trees: 400 SMOTE-TK: Não PCA: Não	0.449	<b>0.921</b>	0.554	0.826
	depth: 30 min_leaf: 1 trees: 600 SMOTE-TK: Não PCA: Não	0.448	0.921	<b>0.554</b>	0.827
	depth: 20 min_leaf: 1 trees: 800 SMOTE-TK: Não PCA: Não	0.444	0.921	0.554	<b>0.829</b>
Redes Neurais Artificiais	hidden_layers: [120] dropout_hidden: 0.0 dropout_input: 0.2 reg_l1_l2: [0, 0] SMOTE-TK: Sim PCA: Não	<b>0.401</b>	0.829	0.420	0.763
	hidden_layers: [120, 120] dropout_hidden: 0.2 dropout_input: 0.0 reg_l1_l2: [0.0001, 0.0001] SMOTE-TK: Não PCA: Não	0.338	<b>0.908</b>	<b>0.439</b>	<b>0.780</b>
SVM	C: 200 SMOTE-TK: Não PCA: Não	<b>0.371</b>	0.855	0.327	0.708
	C: 100 SMOTE-TK: Não PCA: Sim	0.286	<b>0.903</b>	0.368	0.724
	C: 100 SMOTE-TK: Não PCA: Não	0.329	0.903	<b>0.386</b>	0.744
	C: 1 SMOTE-TK: Não PCA: Não	0.347	0.765	0.319	<b>0.750</b>
Naive Bayes	prior: None transformação: 'Quantílica' SMOTE-TK: Não PCA: Não	<b>0.325</b>	0.780	0.304	0.725
	prior: None transformação: 'Potência' SMOTE-TK: Não PCA: Sim	0.220	<b>0.862</b>	<b>0.222</b>	<b>0.705</b>

Tabela A.3: Arranjos dos hiperparâmetros dos modelos hierárquicos treinados com COPs do evento seca e suas respectivas métricas calculadas na validação cruzada.

Modelos		Métricas			
Algoritmo	Hiperparâmetros / Conjuntos de Dados	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
Random Forest	depth: 30 min_leaf: 5 trees: 600 SMOTE-TK: Sim PCA: Não	<b>0.500</b>	0.894	0.538	0.824
	depth: 40 min_leaf: 1 trees: 200 SMOTE-TK: Não PCA: Não	0.469	<b>0.913</b>	0.571	0.828
	depth: 20 min_leaf: 1 trees: 800 SMOTE-TK: Não PCA: Não	0.463	0.912	<b>0.575</b>	0.832
	depth: 50 min_leaf: 5 Trees: 800 SMOTE-TK: Não PCA: Não	0.449	0.911	0.571	<b>0.832</b>
Redes Neurais Artificiais	layers: [120] dropout_hidden: 0.0 dropout_input: 0.2 reg_l1_l2: [0, 0] SMOTE-TK: Sim PCA: Não	<b>0.401</b>	0.805	0.416	0.765
	layers: [70, 70] dropout_hidden: 0.2 dropout_input: 0.2 reg_l1_l2: [0, 0] SMOTE-TK: Não PCA: Não	0.262	<b>0.893</b>	0.405	0.768
	layers: [70] dropout_hidden: 0.0 dropout_input: 0.2 reg_l1_l2: [0, 0] SMOTE-TK: Sim PCA: Não	0.398	0.809	<b>0.428</b>	0.767
	layers: [120, 120, 120] dropout_hidden: 0.0 dropout_input: 0.0 reg_l1_l2: [0.0001, 0.0001] SMOTE-TK: Não PCA: Não	0.278	0.891	0.408	<b>0.775</b>



Algoritmo	Hiperparâmetros / Conjuntos de Dados	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
SVM	C: 100 SMOTE-TK: Sim PCA: Não	<b>0.394</b>	0.839	0.358	0.721
	C: 10 SMOTE-TK: Não PCA: Não	0.225	<b>0.893</b>	0.403	0.742
	C: 100 SMOTE-TK: Não PCA: Não	0.368	0.891	<b>0.410</b>	0.752
	C: 1 SMOTE-TK: Sim PCA: Não	0.375	0.755	0.332	<b>0.763</b>
Naive Bayes	prior: None transformação: 'Quantílica' SMOTE-TK: Não PCA: Não	<b>0.346</b>	0.769	<b>0.335</b>	<b>0.726</b>
	prior: None transformação: 'Potência' SMOTE-TK: Não PCA: Sim	0.173	<b>0.848</b>	0.230	0.696

Tabela A.4: Arranjos dos hiperparâmetros dos modelos hierárquicos treinados com COPs do evento chuva excessiva e suas respectivas métricas calculadas na validação cruzada.

Modelos		Métricas			
Algoritmo	Hiperparâmetros / Conjuntos de Dados	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
Random Forest	depth: 30 min_leaf: 5 trees: 400 SMOTE-TK: Sim PCA: Não	<b>0.468</b>	0.906	0.490	0.815
	depth: 30 min_leaf: 1 trees: 600 SMOTE-TK: Não PCA: Não	0.428	<b>0.928</b>	0.530	0.823
	depth: 30 min_leaf: 1 trees: 800 SMOTE-TK: Não PCA: Não	0.428	0.928	<b>0.530</b>	0.822
	depth: None min_leaf: 5 Trees: 200 SMOTE-TK: Não PCA: Não	0.414	0.928	0.529	<b>0.824</b>
Redes Neurais Artificiais	layers: [70] dropout_hidden: 0.0 dropout_input: 0.2 reg_l1_l2: [0.0001, 0.0001] SMOTE-TK: Sim PCA: Não	<b>0.383</b>	0.841	<b>0.393</b>	0.750
	layers: [120, 120] dropout_hidden: 0.2 dropout_input: 0.2 reg_l1_l2: [0.0001, 0.0001] SMOTE-TK: Não PCA: Não	0.295	<b>0.915</b>	0.384	0.757
	layers: [120, 120] dropout_hidden: 0.0 dropout_input: 0.2 reg_l1_l2: [0.0001, 0.0001] SMOTE-TK: Não PCA: Não	0.302	0.915	0.391	<b>0.766</b>
SVM	C: 300 SMOTE-TK: Não PCA: Não	<b>0.359</b>	0.901	0.323	0.716
	C: 10 SMOTE-TK: Não PCA: Não	0.229	<b>0.913</b>	<b>0.361</b>	0.738

Algoritmo	Hiperparâmetros / Conjuntos de Dados	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
	C: 1 SMOTE-TK: Sim PCA: Não	0.328	0.800	0.315	<b>0.743</b>
Naive Bayes	prior: None transformação: quantile SMOTE-TK: Não PCA: Sim	<b>0.326</b>	0.854	0.267	0.730
	prior: None transformação: power SMOTE-TK: Não PCA: Sim	0.299	<b>0.864</b>	0.241	0.728
	prior: [0.8, 0.2] transformação: 'Quantílica' SMOTE-TK: Sim PCA: Sim	0.317	0.825	<b>0.293</b>	0.709
	prior: None transformação: 'Quantílica' SMOTE-TK: Não PCA: Não	0.307	0.786	0.273	<b>0.731</b>

Tabela A.5: Arranjos dos hiperparâmetros dos modelos hierárquicos treinados com COPs do evento geada e suas respectivas métricas calculadas na validação cruzada.

Modelos		Métricas			
Algoritmo	Hiperparâmetros / Conjuntos de Dados	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
Random Forest	depth: 20 min_leaf: 10 trees: 800 SMOTE-TK: Sim PCA: Não	<b>0.408</b>	0.907	0.368	0.770
	depth: 30 min_leaf: 1 trees: 400 SMOTE-TK: Não PCA: Não	0.383	<b>0.941</b>	0.436	0.772
	depth: 30 min_leaf: 10 trees: 200 SMOTE-TK: Não PCA: Não	0.250	0.936	<b>0.445</b>	0.777
	depth: 20 min_leaf: 5 trees: 600 SMOTE-TK: Não PCA: Não	0.298	0.937	0.442	<b>0.781</b>
Redes Neurais Artificiais	layers: [70, 70] dropout_hidden: 0.0 dropout_input: 0.0 reg_l1_l2: [0.0001, 0.0001] SMOTE-TK: Não PCA: Não	<b>0.313</b>	0.926	0.325	0.726
	layers: [70, 70] dropout_hidden: 0.2 dropout_input: 0.2 reg_l1_l2: [0.0001, 0.0001] SMOTE-TK: Não PCA: Não	0.272	<b>0.933</b>	0.334	0.723
	layers: [120] dropout_hidden: 0.0 dropout_input: 0.2 reg_l1_l2: [0.0001, 0.0001] SMOTE-TK: Não PCA: Não	0.271	0.930	<b>0.345</b>	<b>0.749</b>
SVM	C: 200 SMOTE-TK: Não PCA: Sim	<b>0.260</b>	0.911	0.249	0.661
	C: 10 SMOTE-TK: Não PCA: Sim	0.163	<b>0.928</b>	0.299	0.702

Algoritmo	Hiperparâmetros / Conjuntos de Dados	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
	C: 10 SMOTE-TK: Não PCA: Não	0.165	0.928	<b>0.303</b>	0.681
	C: 0.01 SMOTE-TK: Sim PCA: Sim	0.217	0.854	0.209	<b>0.703</b>
Naive Bayes	prior: [0.6, 0.4] transformação: 'Quantílica' SMOTE-TK: Não PCA: Sim	<b>0.308</b>	0.834	0.263	0.727
	prior: None transformação: 'Potência' SMOTE-TK: Não PCA: Sim	0.222	<b>0.906</b>	0.239	0.682
	prior: None transformação: 'Quantílica' SMOTE-TK: Não PCA: Não	0.298	0.856	<b>0.277</b>	<b>0.729</b>

Tabela A.6: Arranjos dos hiperparâmetros dos modelos monolíticos e suas respectivas métricas calculadas sobre os dados de teste.

Modelos		Métricas			
Algoritmo	Hiperparâmetros / Conjuntos de Dados	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
Random Forest	depth: 20 min_leaf: 5 trees: 400 SMOTE-TK: Sim PCA: Não	<b>0.495</b>	0.902	0.507	0.824
	depth: 30 min_leaf: 1 trees: 400 SMOTE-TK: Não PCA: Não	0.427	0.919	0.549	0.838
	depth: 30 min_leaf: 1 trees: 600 SMOTE-TK: Não PCA: Não	0.427	<b>0.919</b>	<b>0.549</b>	<b>0.838</b>
	depth: 20 min_leaf: 1 trees: 800 SMOTE-TK: Não PCA: Não	0.425	0.919	0.547	0.833
Redes Neurais Artificiais	hidden_layers: [120] dropout_hidden: 0.0 dropout_input: 0.2 reg_l1_l2: [0, 0] SMOTE-TK: Sim PCA: Não	<b>0.406</b>	0.845	0.426	0.764
	hidden_layers: [120, 120] dropout_hidden: 0.2 dropout_input: 0.0 reg_l1_l2: [0.0001, 0.0001] SMOTE-TK: Não PCA: Não	0.307	<b>0.907</b>	<b>0.426</b>	<b>0.780</b>
SVM	C: 200 SMOTE-TK: Não PCA: Não	<b>0.342</b>	0.899	0.377	0.752
	C: 100 SMOTE-TK: Não PCA: Sim	0.277	<b>0.904</b>	0.376	0.746
	C: 100 SMOTE-TK: Não PCA: Não	0.320	0.903	<b>0.385</b>	<b>0.755</b>
	C: 1 SMOTE-TK: Não PCA: Não	0.045	0.897	0.363	0.739

Algoritmo	Hiperparâmetros / Conjuntos de Dados	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
Naive Bayes	prior: None transformação: 'Quantílica' SMOTE-TK: Não PCA: Não	<b>0.316</b>	0.780	<b>0.296</b>	<b>0.728</b>
	prior: None transformação: 'Potência' SMOTE-TK: Não PCA: Sim	0.223	<b>0.888</b>	<b>0.267</b>	<b>0.725</b>

Tabela A.7: Arranjos dos hiperparâmetros dos modelos hierárquicos treinados com COPs do evento seca e suas respectivas métricas calculadas nos dados de teste.

Modelos		Métricas			
Algoritmo	Hiperparâmetros / Conjuntos de Dados	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
Random Forest	depth: 30 min_leaf: 5 trees: 600 SMOTE-TK: Sim PCA: Não	<b>0.495</b>	0.892	0.525	0.829
	depth: 40 min_leaf: 1 trees: 200 SMOTE-TK: Não PCA: Não	0.434	0.908	0.552	0.837
	depth: 20 min_leaf: 1 trees: 800 SMOTE-TK: Não PCA: Não	0.430	<b>0.909</b>	0.555	0.835
	depth: 50 min_leaf: 5 Trees: 800 SMOTE-TK: Não PCA: Não	0.415	0.908	<b>0.557</b>	<b>0.838</b>
Redes Neurais Artificiais	layers: [120] dropout_hidden: 0.0 dropout_input: 0.2 reg_l1_l2: [0, 0] SMOTE-TK: Sim PCA: Não	0.389	0.801	0.383	0.758
	layers: [70, 70] dropout_hidden: 0.2 dropout_input: 0.2 reg_l1_l2: [0, 0] SMOTE-TK: Não PCA: Não	0.175	<b>0.890</b>	0.380	0.759
	layers: [70] dropout_hidden: 0.0 dropout_input: 0.2 reg_l1_l2: [0, 0] SMOTE-TK: Sim PCA: Não	<b>0.396</b>	0.813	<b>0.402</b>	0.758
	layers: [120, 120, 120] dropout_hidden: 0.0 dropout_input: 0.0 reg_l1_l2: [0.0001, 0.0001] SMOTE-TK: Não PCA: Não	0.175	0.887	0.377	<b>0.764</b>



Algoritmo	Hiperparâmetros / Conjuntos de Dados	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
SVM	C: 100 SMOTE-TK: Sim PCA: Não	<b>0.385</b>	0.841	0.337	0.712
	C: 10 SMOTE-TK: Não PCA: Não	0.180	<b>0.890</b>	<b>0.387</b>	0.756
	C: 100 SMOTE-TK: Não PCA: Não	0.322	0.887	0.379	<b>0.760</b>
	C: 1 SMOTE-TK: Sim PCA: Não	0.368	0.759	0.304	0.747
Naive Bayes	prior: None transformação: 'Quantílica' SMOTE-TK: Não PCA: Não	<b>0.344</b>	0.770	<b>0.321</b>	<b>0.712</b>
	prior: None transformação: 'Potência' SMOTE-TK: Não PCA: Sim	0.178	<b>0.872</b>	0.254	0.702

Tabela A.8: Arranjos dos hiperparâmetros dos modelos hierárquicos treinados com COPs do evento chuva excessiva e suas respectivas métricas calculadas nos dados de teste.

Modelos		Métricas			
Algoritmo	Hiperparâmetros / Conjuntos de Dados	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
Random Forest	depth: 40 min_leaf: 5 trees: 400 SMOTE-TK: Sim PCA: Não	<b>0.467</b>	0.906	0.478	0.817
	depth: 50 min_leaf: 1 trees: 600 SMOTE-TK: Não PCA: Não	0.409	0.926	0.533	<b>0.835</b>
	depth: 20 min_leaf: 1 trees: 800 SMOTE-TK: Não PCA: Não	0.413	0.926	<b>0.535</b>	0.834
	depth: 40 min_leaf: 5 Trees: 200 SMOTE-TK: Não PCA: Não	0.411	<b>0.927</b>	0.530	0.830
Redes Neurais Artificiais	layers: [70] dropout_hidden: 0.0 dropout_input: 0.2 reg_l1_l2: [0.0001, 0.0001] SMOTE-TK: Sim PCA: Não	<b>0.352</b>	0.811	0.344	0.752
	layers: [120, 120] dropout_hidden: 0.2 dropout_input: 0.2 reg_l1_l2: [0.0001, 0.0001] SMOTE-TK: Não PCA: Não	0.208	<b>0.912</b>	0.362	0.762
	layers: [120, 120] dropout_hidden: 0.0 dropout_input: 0.2 reg_l1_l2: [0.0001, 0.0001] SMOTE-TK: Não PCA: Não	0.241	0.910	<b>0.367</b>	<b>0.763</b>
SVM	C: 300 SMOTE-TK: Não PCA: Não	0.317	0.893	0.278	0.691
	C: 10 SMOTE-TK: Não PCA: Não	0.211	<b>0.911</b>	<b>0.337</b>	0.724

Algoritmo	Hiperparâmetros / Conjuntos de Dados	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
	C: 1 SMOTE-TK: Sim PCA: Não	<b>0.335</b>	0.800	0.277	<b>0.737</b>
Naive Bayes	prior: None transformação: quantile SMOTE-TK: Não PCA: Sim	<b>0.328</b>	0.854	<b>0.272</b>	0.745
	prior: None transformação: power SMOTE-TK: Não PCA: Sim	0.270	<b>0.864</b>	0.240	0.740
	prior: [0.8, 0.2] transformação: 'Quantílica' SMOTE-TK: Sim PCA: Sim	0.315	0.825	0.270	0.724
	prior: None transformação: 'Quantílica' SMOTE-TK: Não PCA: Não	0.307	0.782	0.266	<b>0.753</b>

Tabela A.9: Arranjos dos hiperparâmetros dos modelos hierárquicos treinados com COPs do evento geada e suas respectivas métricas calculadas nos dados de teste.

Modelos		Métricas			
Algoritmo	Hiperparâmetros / Conjuntos de Dados	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
Random Forest	depth: 20 min_leaf: 10 trees: 800 SMOTE-TK: Sim PCA: Não	<b>0.385</b>	0.919	0.363	0.785
	depth: 30 min_leaf: 1 trees: 400 SMOTE-TK: Não PCA: Não	0.338	0.938	0.442	0.789
	depth: 50 min_leaf: 10 trees: 200 SMOTE-TK: Não PCA: Não	0.301	0.938	0.436	0.790
	depth: 20 min_leaf: 5 trees: 600 SMOTE-TK: Não PCA: Não	0.347	<b>0.941</b>	<b>0.446</b>	<b>0.801</b>
Redes Neurais Artificiais	layers: [70, 70] dropout_hidden: 0.0 dropout_input: 0.0 reg_l1_l2: [0.0001, 0.0001] SMOTE-TK: Não PCA: Não	0.304	0.933	0.362	0.703
	layers: [70, 70] dropout_hidden: 0.2 dropout_input: 0.2 reg_l1_l2: [0.0001, 0.0001] SMOTE-TK: Não PCA: Não	0.345	0.931	0.381	<b>0.718</b>
	layers: [120] dropout_hidden: 0.0 dropout_input: 0.2 reg_l1_l2: [0.0001, 0.0001] SMOTE-TK: Não PCA: Não	<b>0.354</b>	<b>0.938</b>	<b>0.387</b>	0.712
SVM	C: 200 SMOTE-TK: Não PCA: Sim	280	0.907	0.242	0.685
	C: 10 SMOTE-TK: Não PCA: Sim	0.211	0.927	0.285	0.665

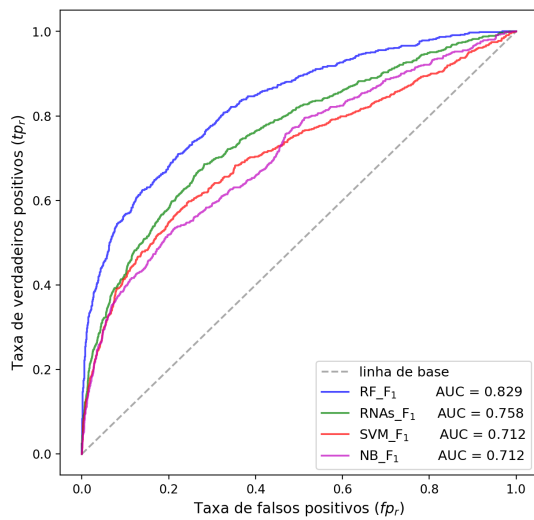
Algoritmo	Hiperparâmetros / Conjuntos de Dados	$F_1$ -score	Acurácia	Precisão Média	ROC AUC
	C: 10 SMOTE-TK: Não PCA: Não	<b>0.308</b>	<b>0.935</b>	<b>0.340</b>	0.675
	C: 0.01 SMOTE-TK: Sim PCA: Sim	0.241	0.855	0.177	<b>0.709</b>
Naive Bayes	prior: [0.6, 0.4] transformação: 'Quantílica' SMOTE-TK: Não PCA: Sim	0.344	0.852	0.270	<b>0.754</b>
	prior: None transformação: 'Potência' SMOTE-TK: Não PCA: Sim	0.280	<b>0.907</b>	0.262	0.725
	prior: None transformação: 'Quantílica' SMOTE-TK: Não PCA: Não	<b>0.348</b>	0.873	<b>0.307</b>	0.745

Tabela A.10: Métricas calculadas com os melhores modelos das abordagens monolítica e hierárquica sobre os dados de teste contendo os eventos seca, chuva excessiva e geadas.

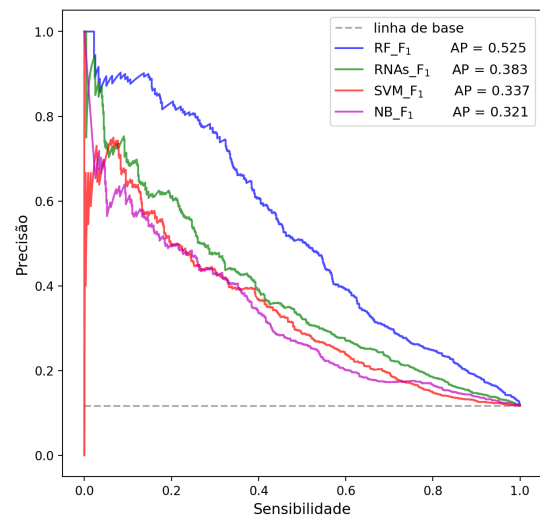
Abordagem	Métrica de otimização dos modelos	Métricas			
		$F_1$ -score	Acurácia	Precisão Média	ROC AUC
Monolítica	$F_1$ -score	<b>0.496</b>	0.902	0.508	0.824
	Acurácia	0.428	0.918	0.550	0.838
	Precisão Média	0.428	<b>0.919</b>	<b>0.550</b>	<b>0.839</b>
	ROC AUC	0.426	0.918	0.549	0.834
Hierárquica combinada	$F_1$ -score	<b>0.480</b>	0.899	0.499	0.823
	Acurácia	0.421	0.917	0.539	<b>0.835</b>
	Precisão Média	0.418	<b>0.918</b>	0.540	0.834
	ROC AUC	0.411	0.917	<b>0.541</b>	0.835

# Apêndice B

## Figuras

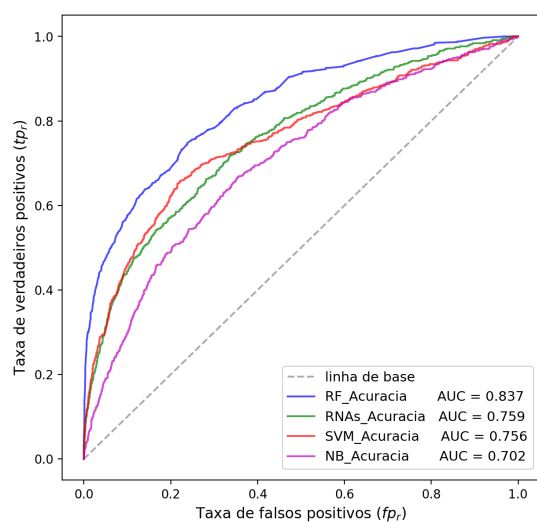


(a) Curva ROC.

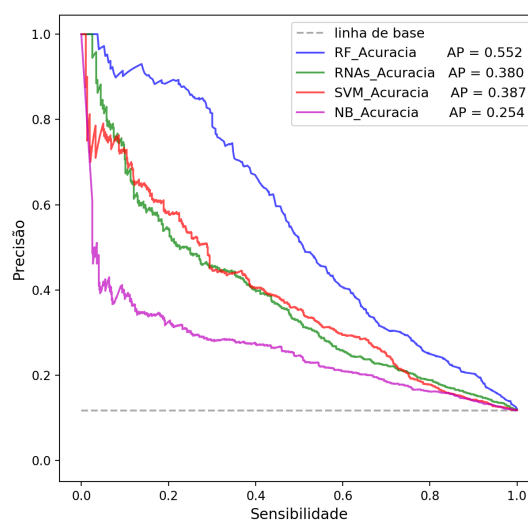


(b) Curva precisão-sensibilidade.

Figura B.1: Curvas (a) ROC e (b) precisão-sensibilidade dos modelos do evento seca que obtiveram os melhores desempenho na métrica  $F_1$ -score.

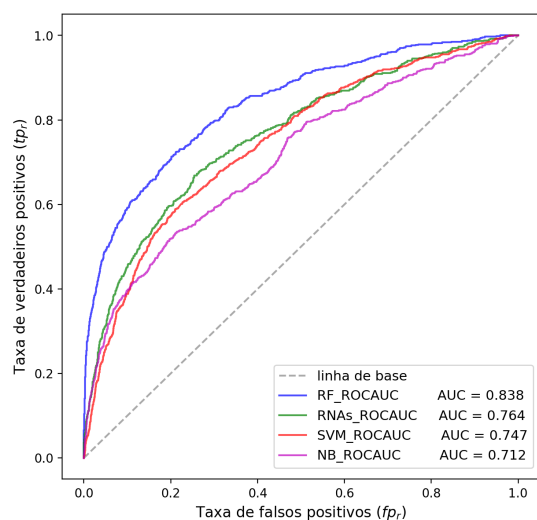


(a) Curva ROC.

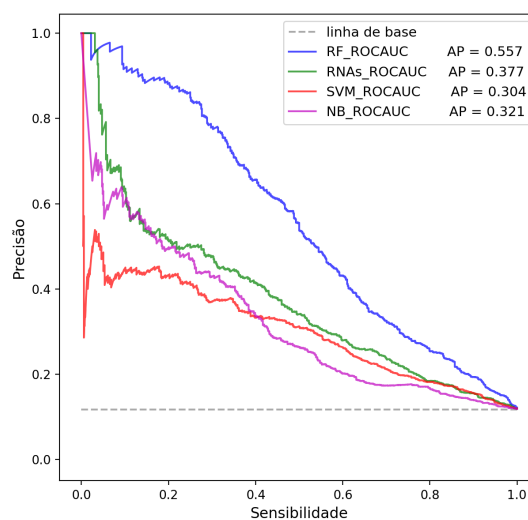


(b) Curva precisão-sensibilidade.

Figura B.2: Curvas (a) ROC e (b) precisão-sensibilidade dos modelos do evento seca que obtiveram os melhores desempenho na métrica acurácia.



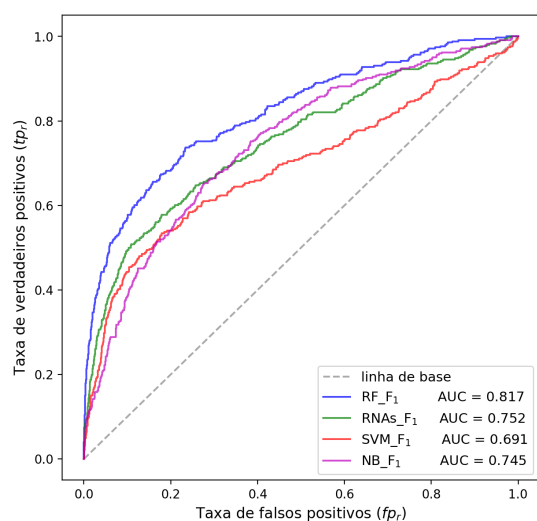
(a) Curva ROC.



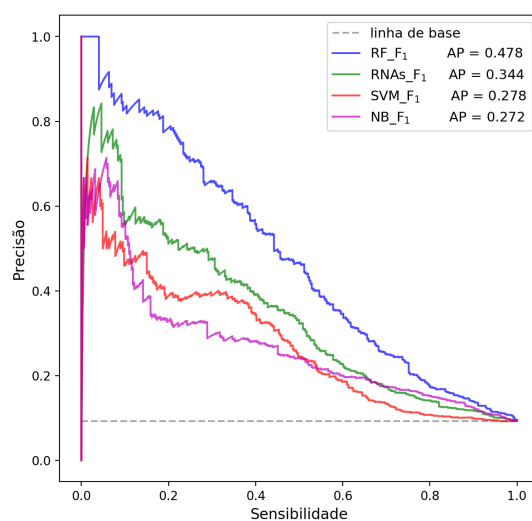
(b) Curva precisão-sensibilidade.

Figura B.3: Curvas (a) ROC e (b) precisão-sensibilidade dos modelos do evento seca que obtiveram os melhores desempenho na métrica ROCAUC.



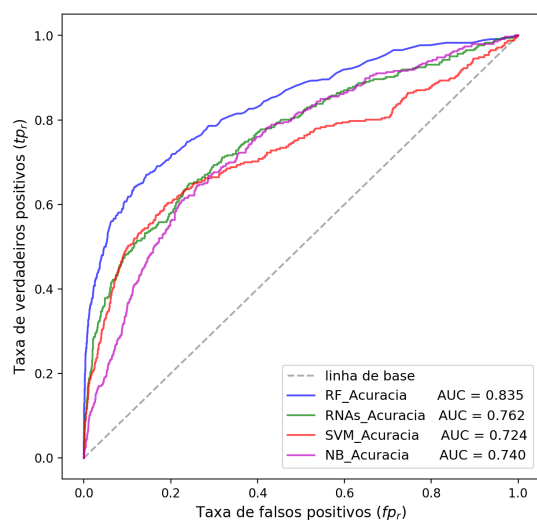


(a) Curva ROC.

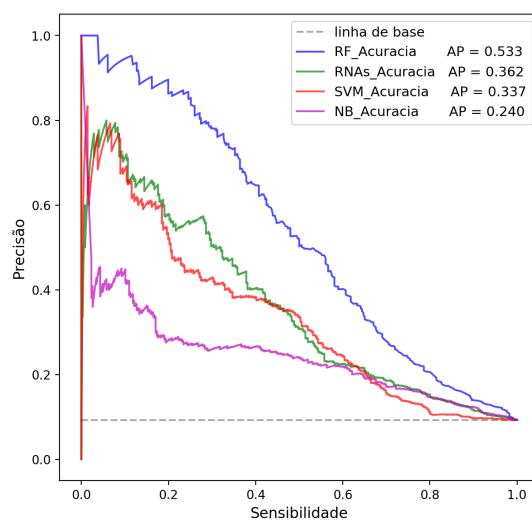


(b) Curva precisão-sensibilidade.

Figura B.4: Curvas (a) ROC e (b) precisão-sensibilidade dos modelos do evento chuva excessiva que obtiveram os melhores desempenho na métrica  $F_1$ -score.

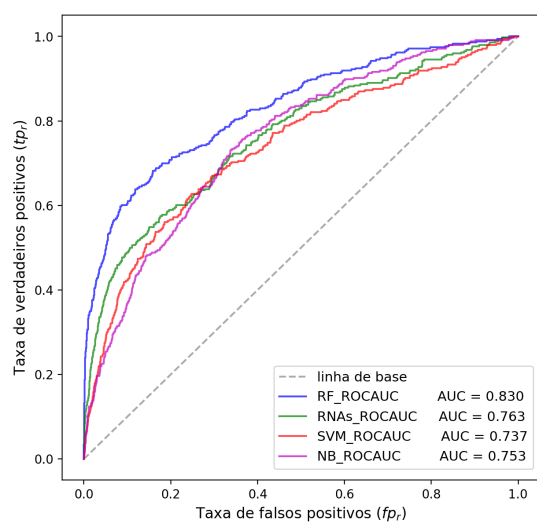


(a) Curva ROC.

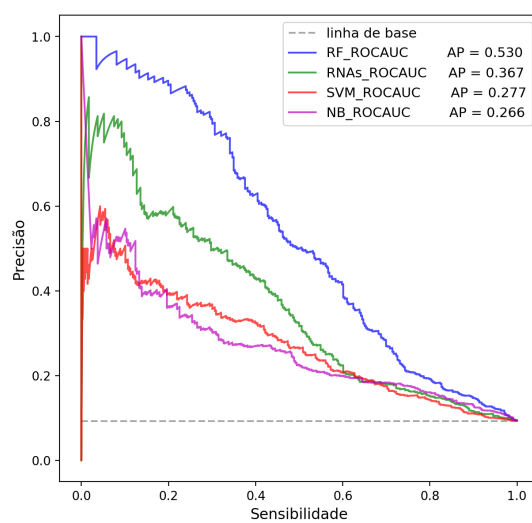


(b) Curva precisão-sensibilidade.

Figura B.5: Curvas (a) ROC e (b) precisão-sensibilidade dos modelos do evento chuva excessiva que obtiveram os melhores desempenho na métrica acurácia.

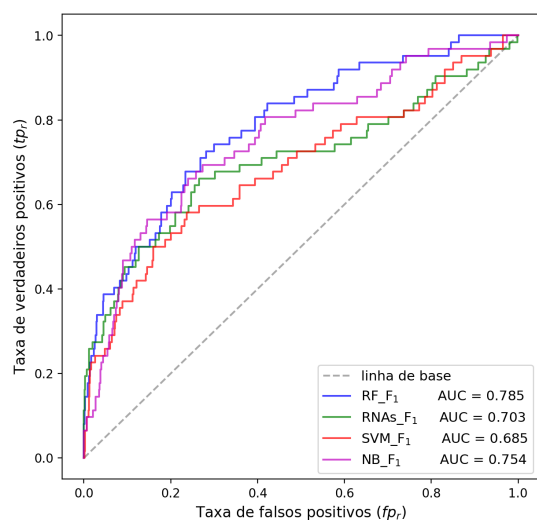


(a) Curva ROC.

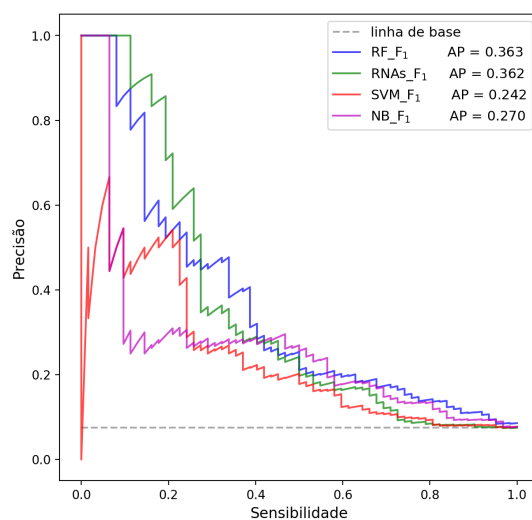


(b) Curva precisão-sensibilidade.

Figura B.6: Curvas (a) ROC e (b) precisão-sensibilidade dos modelos do evento chuva excessiva que obtiveram os melhores desempenho na métrica ROCAUC.

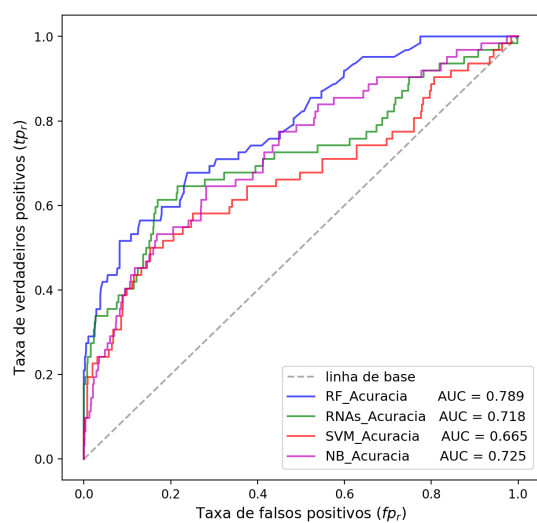


(a) Curva ROC.

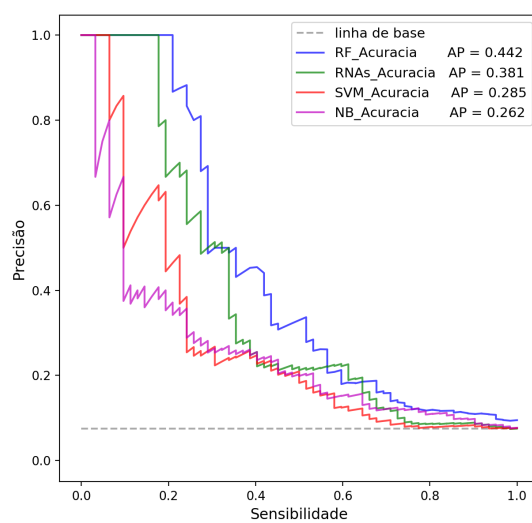


(b) Curva precisão-sensibilidade.

Figura B.7: Curvas (a) ROC e (b) precisão-sensibilidade dos modelos do evento geada que obtiveram os melhores desempenho na métrica  $F_1$ -score.

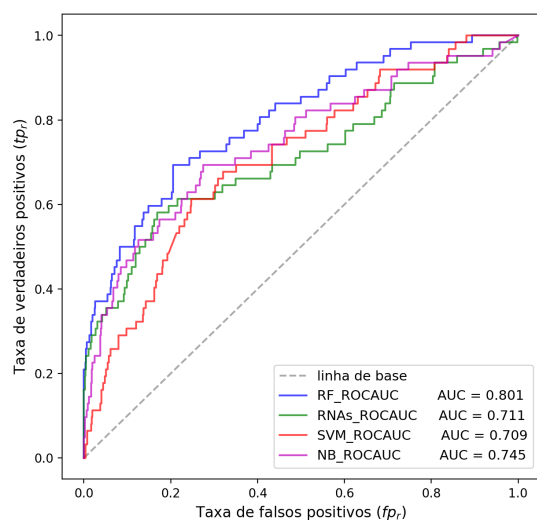


(a) Curva ROC.

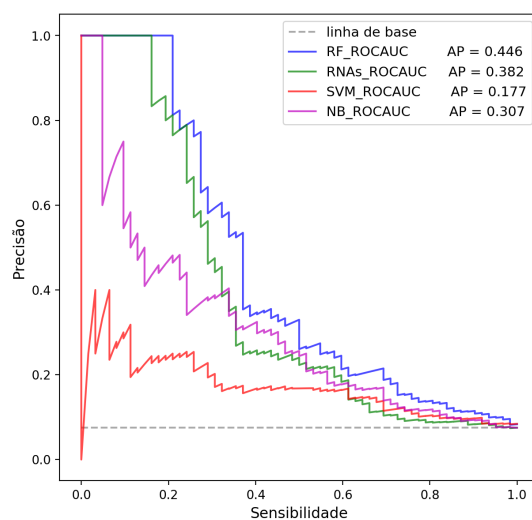


(b) Curva precisão-sensibilidade.

Figura B.8: Curvas (a) ROC e (b) precisão-sensibilidade dos modelos do evento geada que obtiveram os melhores desempenho na métrica acurácia.

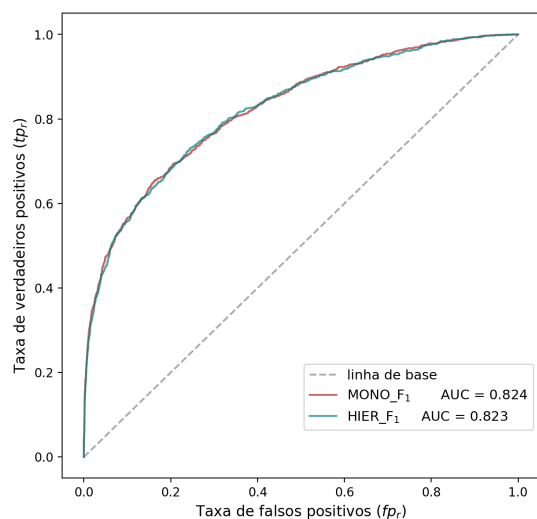


(a) Curva ROC.

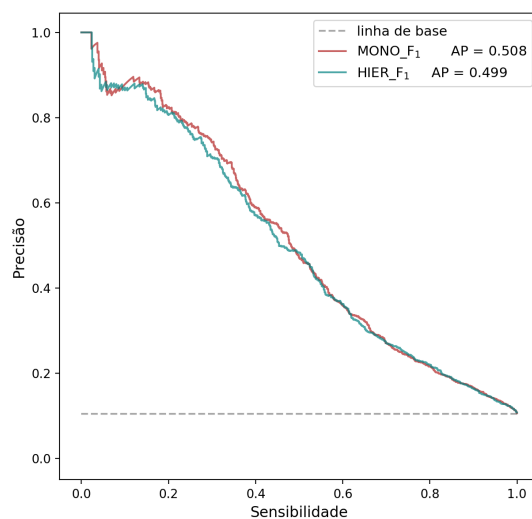


(b) Curva precisão-sensibilidade.

Figura B.9: Curvas (a) ROC e (b) precisão-sensibilidade dos modelos do evento geada que obtiveram os melhores desempenho na métrica ROCAUC.

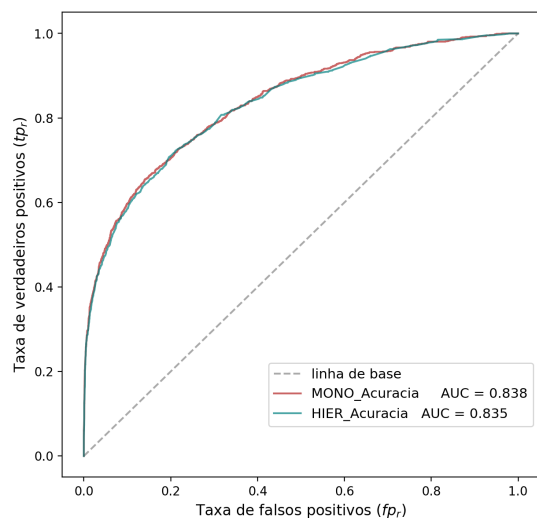


(a) Curva ROC.

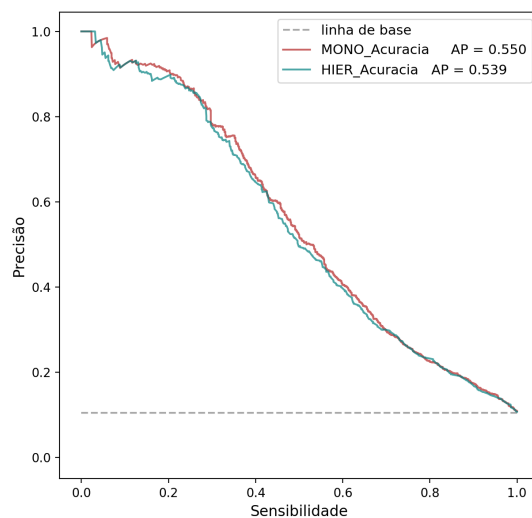


(b) Curva precisão-sensibilidade.

Figura B.10: Curvas (a) ROC e (b) precisão-sensibilidade dos resultados das abordagens monolítica e hierárquica recalculados a partir do mesmo conjunto de dados. Os modelos utilizados foram aqueles otimizados para a métrica  $F_1$ -score.

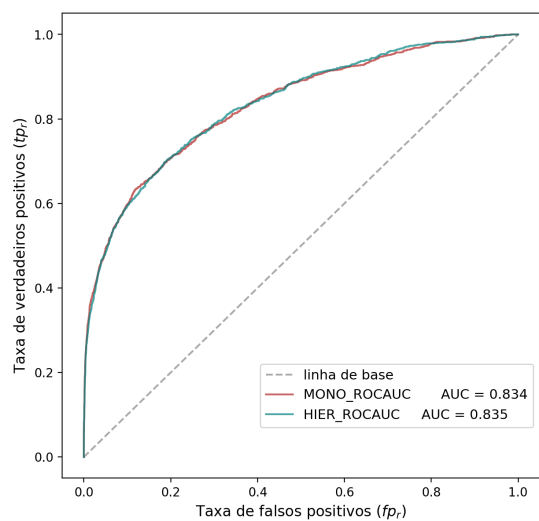


(a) Curva ROC.

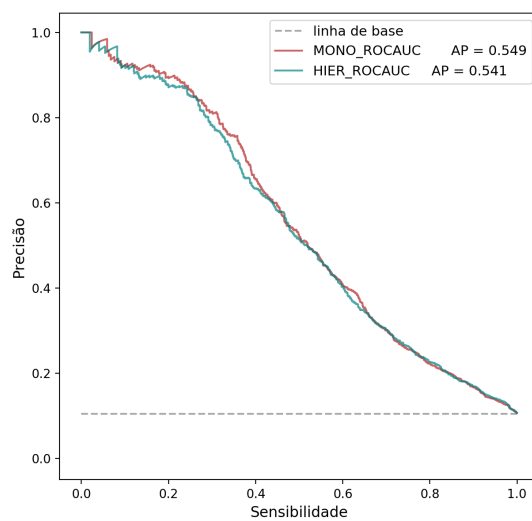


(b) Curva precisão-sensibilidade.

Figura B.11: Curvas (a) ROC e (b) precisão-sensibilidade dos resultados das abordagens monolítica e hierárquica recalculados a partir do mesmo conjunto de dados. Os modelos utilizados foram aqueles otimizados para a métrica acurácia.



(a) Curva ROC.



(b) Curva precisão-sensibilidade.

Figura B.12: Curvas (a) ROC e (b) precisão-sensibilidade dos resultados das abordagens monolítica e hierárquica recalculados a partir do mesmo conjunto de dados. Os modelos utilizados foram aqueles otimizados para a métrica ROCAUC.