UNIVERSIDADE DE BRASÍLIA
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
MESTRADO EM INFORMÁTICA

IDENTIFICAÇÃO DE COMUNICADO DE OCORRÊNCIA DE PERDAS EM SEGURO AGRÍCOLA UTILIZANDO ALGORITMOS DE INTELIGÊNCIA ARTIFICIAL

por

Rafael Marconi Ramos

Dissertação apresentada como requisito parcial
para conclusão do Mestrado em Informática

Prof. Dr. Marcelo Ladeira

Orientador

Brasília
2011

Universidade de Brasília
Reitor: Prof. Dr. José Geraldo de Sousa Junior

Instituto de Ciências Exatas
Diretor: Prof. Dr. Noraí Romeu Rocco

Departamento de Ciência da Computação
Chefe de Departamento: Profª Drª Priscila América Solís Mendez Barreto

Mestrado em Informática
Coordenador de Pós-Graduação: Prof. Dr. Maurício Ayala Rincón

Banca examinadora composta por:

Prof. Dr. Marcelo Ladeira (Orientador) – CIC/UnB
Prof. Dr. Li Weigang – CIC/UnB
Prof. Dr. Remis Balaniuk – Universidade Católica da Brasília

CIP — Catalogação Internacional na Publicação

Ramos, Rafael Marconi.

Identificação de Comunicado de Ocorrência de Perdas em Seguro Agrícola Utilizando Algoritmos de Inteligência Artificial / Rafael Marconi Ramos. – Brasília: UnB, 2011.

- 80 p.: il.; 29,5 cm.

Dissertação (Mestrado) — Universidade de Brasília. Programa de Pós-Graduação em Informática, Brasília, 2011. Orientador: Ladeira, Marcelo.

1. Computação, 2. Inteligência Artificial, 3. Mineração de Dados, 4. Agricultura Familiar, 5. Seguro de Agricultura Familiar, 6. Comunicado de Ocorrência de Perdas. I. Ladeira, Marcelo. II. Título.

CDU 004.8

Endereço: Universidade de Brasília

Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

UNIVERSIDADE DE BRASÍLIA
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
MESTRADO EM INFORMÁTICA

IDENTIFICAÇÃO DE COMUNICADO DE OCORRÊNCIA DE PERDAS EM SEGURO AGRÍCOLA UTILIZANDO ALGORITMOS DE INTELIGÊNCIA ARTIFICIAL

Rafael Marconi Ramos

Dissertação apresentada como requisito parcial
para conclusão do Mestrado em Informática

Prof. Dr. Marcelo Ladeira (Orientador)
CIC/UnB

Prof. Dr. Li Weigang
CIC/UnB

Prof. Dr. Remis Balaniuk
Universidade Católica da Brasília

Prof. Dr. Maurício Ayala Rincón
Coordenador do Mestrado em Informática

Brasília, 28 de junho de 2011

Navegar é preciso, minerar não é preciso!
Marcelo Ladeira

Resumo

O Ministério do Desenvolvimento Agrário – MDA, através da Secretaria de Agricultura Familiar – SEAF, possui estrutura e metodologia para acompanhamento dos agricultores familiares assistidos por programas governamentais como o PRONAF. A safra esperada pode ser segurada por meio de um agente financeiro (banco) com carteira agrícola. Quando um agricultor acredita que teve perdas na safra, recorre ao banco solicitando pagamento de seguro agrícola. O banco então emite um Comunicado de Ocorrência de Perda – COP para o Banco Central. Em geral eles são pagos sem averiguação. Se o agricultor alegou perdas e solicitou o pagamento do seguro agrícola, no laudo emitido após a colheita, constará a informação da emissão de COP. Essa pesquisa propõe um método automático que viabilize analisar um maior número de COP, contribuindo para minimizar os custos do pagamento de COP indevidas, sem ser necessário ampliar os atuais recursos humanos disponíveis na SEAF. O método proposto é baseado na construção de modelos com técnica de inteligência artificial para inferir se há evidências de que pode ocorrer COP, a partir dos dados dos laudos obtidos pelo MDA para as safras de agricultura familiar dos anos de 2006 a 2010, totalizando 11.743 registros. Com a aplicação de técnicas estatísticas foram selecionados 19 atributos dentre os 311 atributos coletados por meio dos laudos. A partir desses 19 atributos foram construídas regras de associação e classificadores para COP. As regras de associação foram obtidas com o algoritmo "Apriori" e os modelos de classificação foram baseados na construção de classificadores probabilísticos (*Naive Bayes* e árvore de inferência condicional), árvore de decisão (C4.5) e máquina de vetores de suporte (SVM). Os classificadores construídos foram considerados isoladamente e em comitês, constituídos multiclassificadores com as técnicas de votação, *boosting* e *bagging*. Foram propostas novas abordagens para construção de multiclassificadores baseados em disjunção, ponderação pelos índices de desempenho e cascata. Todos os classificadores foram avaliados com as métricas de sensibilidade, especificidade, acurácia e F-*measure*, além da análise de curva ROC. Os resultados obtidos dão suporte experimental para se concluir que os multiclassificadores propostos – disjunção, ponderação e cascata – apresentam melhor desempenho para esse tipo de problema do que os obtidos com a abordagem tradicional, tendo alcançado o índice de 0,944 para AUC.

Palavras chaves: multiclassificadores, perdas agrícolas, mineração de dados, seguro agrícola.

Abstract

The Ministry of Agrarian Development in Brazil (MDA), through the Secretariat for Family Agriculture (SEAF), has structure and methodology to monitor family farmers through the government program known as National Program for the Strengthening of Family Agriculture (PRONAF). The SEAF is responsible by PRONAF. The expected crop can be insured by a commercial bank with agricultural portfolio. When a farmer believes that has a loss of crop, he can request the bank the payment of the crop insurance. The bank issues a Communication of Occurrence of Losses - COP to the Central Bank of Brazil. If the farmer claimed losses and sought payment of the crop insurance, the technical report issued after the harvest, contains the information of COP. This research proposes an automatic method that allows the analyses of a larger number of occurrences of COP, contributing to minimize the costs of undue payment of COPs, without the need to expand the current technical staff of SEAF. The proposed method is based on building models which uses artificial intelligence techniques in order to conclude if there are enough evidence that COPs can occurs. Those models are built based on data extracted from technical reports about the family farm harvests in the years from 2006 to 2010. The results obtained in this research were based on 11,743 of these technical reports available in the MDA. With the application of statistical analysis the 19 attributes more relevant, among the 311 attributes available, were identified. Based on the 19 more relevant attributes, association rules and classifiers were built to identify the possibility of occurrence of COP. The association rules were obtained with the Apriori algorithm and the models for classification were built with probabilistic classifiers (Naive Bayes), decision tree (C4.5), and Support Vector Machine (SVM). The classifiers were considered alone and in committees. A classifier formed by a committee of classifiers was called multi-classifier. The last one used the techniques of vote, boosting and bagging. New proposals have been made for the construction of multi-classifiers based on disjunction of simple classifiers, the weighting of individual performance indicators, and on making a cascade of a set of simple classifiers. Both the classifiers and the multi-classifiers were evaluated with the metrics of sensitivity, specificity, accuracy, F-measure, and analysis of ROC curve. The results achieved provide experimental support for concluding that our multi-classifiers proposed have had better performance on this task than those obtained with the traditional approach. The better performance index achieved by our best multi-classifier was 0.944 for AUC.

Keywords: multi-classifiers, lost agricultural, data mining, crop insurance.

Agradecimentos

Primeiramente gostaria de agradecer a Deus por ter permitido que eu caminhasse até essa etapa de minha vida com muita benção. Aos meus pais que me incentivaram e apoiaram durante toda minha vida. A minha namorada que abdicou de seu tempo para me fazer companhia durante todo o tempo de estudo.

Agradeço ao meu orientador Marcelo Ladeira por ter sido um orientador nota mil, com muita paciência e conhecimento.

A todos que de alguma forma participaram ou permitiram que eu desenvolvesse esse trabalho, em especial a equipe do MDA, meu chefe José Zukowisk, o Diretor João Guadagnin e meu parceiro de desenvolvimento Victor Ferreira Leite.

A CAPES, pelo importante apoio financeiro dado a esta pesquisa.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

ATER	Sistema de Assistência Técnica
AUC	Área sob a curva ROC (do inglês, *Area under curve*)
COP	Comunicado de Ocorrência de Perdas
CRISP-DM	Modelo de Referência para Mineração de Dados (do *Cross-Industry Standard Process for Data Mining*)
FN	Falso Negativo (do inglês, *False Negative*)
FNR	Taxa de Falso Negativo
FP	Falso Positivo
IA	Inteligência Artificial
Curva ROC	Curvas Características de Operação do Receptor (do inglês, *Receiver Operating Characteristics Curve*)
MDA	Ministério do Desenvolvimento Agrário
SAF	Secretaria de Agricultura Familiar do MDA
SEAF	Seguro de Agricultura Familiar
	Coordenação do Seguro de Agricultura Familiar da SAF/MDA
TN	Verdadeiro Negativo (do inglês, *True Negative*)
TNR	Taxa de Verdadeiro Negativo
TP	Verdadeiro Positivo (do inglês, *True Positivo*)
TPR	Taxa de Verdadeiro Positivo

1 Introdução

Este capítulo descreve o problema objeto dessa pesquisa, sua relevância e as áreas de pesquisas relacionadas. A Seção 1.1 introduz o problema a ser abordado. A Seção 1.2 apresenta o objetivo desta pesquisa, a Seção 1.3 aborda as áreas de pesquisa relacionadas e a Seção 1.4 apresenta a estrutura desse documento.

1.1 Definição do Problema

Criado em 31 de agosto de 2005, originalmente com a denominação **Proagro Mais**, o Seguro da Agricultura Familiar – SEAF é um dos mais importantes instrumentos de política para a agricultura familiar, proporcionando garantia de renda para atividades agropecuárias. A implementação desse instrumento exige, por parte dos tomadores de decisão, o exercício de rigorosos procedimentos de acompanhamento, como forma de garantir a eficiência, eficácia e efetividade das ações, lastreados em padrões de transparência e qualidade.

A sustentabilidade de um seguro depende dos mecanismos criados para evitar e prevenir perdas e fraudes. No caso do SEAF, existe um mecanismo de acompanhamento da lavoura através do processo de assistência técnica.

Esse processo consiste em orientar e acompanhar, com técnicos agrônomos, os diferentes momentos da plantação. O resultado desse processo é a criação de laudos de assistência técnica emitidos durante o pré-plantio, plantio e colheita. No último laudo, se houve quebra da safra esperada, é informado o pedido de perda através da informação da COP – Comunicado de Ocorrência de Perda. A safra esperada pode ser segurada por meio de um agente financeiro (banco) com carteira agrícola. Quando um agricultor acredita que teve perdas na safra, recorre ao banco solicitando pagamento de seguro agrícola. O banco então emite um COP para o BACEN – Banco Central. Como a SEAF não dispõe de recursos humanos suficiente, apenas sob demanda específica, tal Coordenadoria analisa COP. Em geral eles são pagos sem averiguação. Se o agricultor alegou perdas e solicitou o pagamento do seguro agrícola, no terceiro laudo, emitido após a colheita, constará a informação da emissão de COP.

O objetivo do seguro agrícola SEAF é dar segurança ao produtor durante a safra, não se limitando a apenas cobrir eventuais quebras de safra. Resumidamente o SEAF proporciona uma garantia de renda, mas essa renda deve vir em primeiro lugar de um trabalho consistente de produção. Se o produtor adotou procedimentos adequados na condução da lavoura e tomou as medidas preventivas que seriam cabíveis, mas houve perdas por evento amparado, então ele poderá recorrer ao Seguro e receber a indenização. Dessa forma é fundamental que o SEAF produza sua base de informações que permita estudos que garantam a segurança da produção agrícola através da extração de informações para melhora do procedimento de plantio, provisionamento de recursos, garantia de constituição de fundos de proteção para a carteira do seguro e outros.

Na safra 2004-2005, a Secretaria da Agricultura Familiar (SAF) iniciou um conjunto de ações de monitoramento do programa, envolvendo acompanhamento de diversos processos de

operacionalização junto às agências locais das instituições financeiras, para acompanhamento do desempenho das lavouras seguradas. Os seguintes sistemas estão em desenvolvimento:

- Sistema de periciamento

  Responsável pelo controle dos peritos autorizados a realizar perícias.

- Sistema de supervisão

  Responsável pela supervisão dos peritos

- Sistemas de COP

  Sistema de notificação de perdas

- Sistema de Assistência Técnica – ATER

  Sistema de acompanhamento de assistência técnica

A Figura 1 ilustra o relacionamento entre eles. Essa pesquisa está vinculada ao esforço de desenvolvimento de sistemas para gerenciamento de COP, em especial, para a previsão da possibilidade de ocorrências de comunicados de COP.

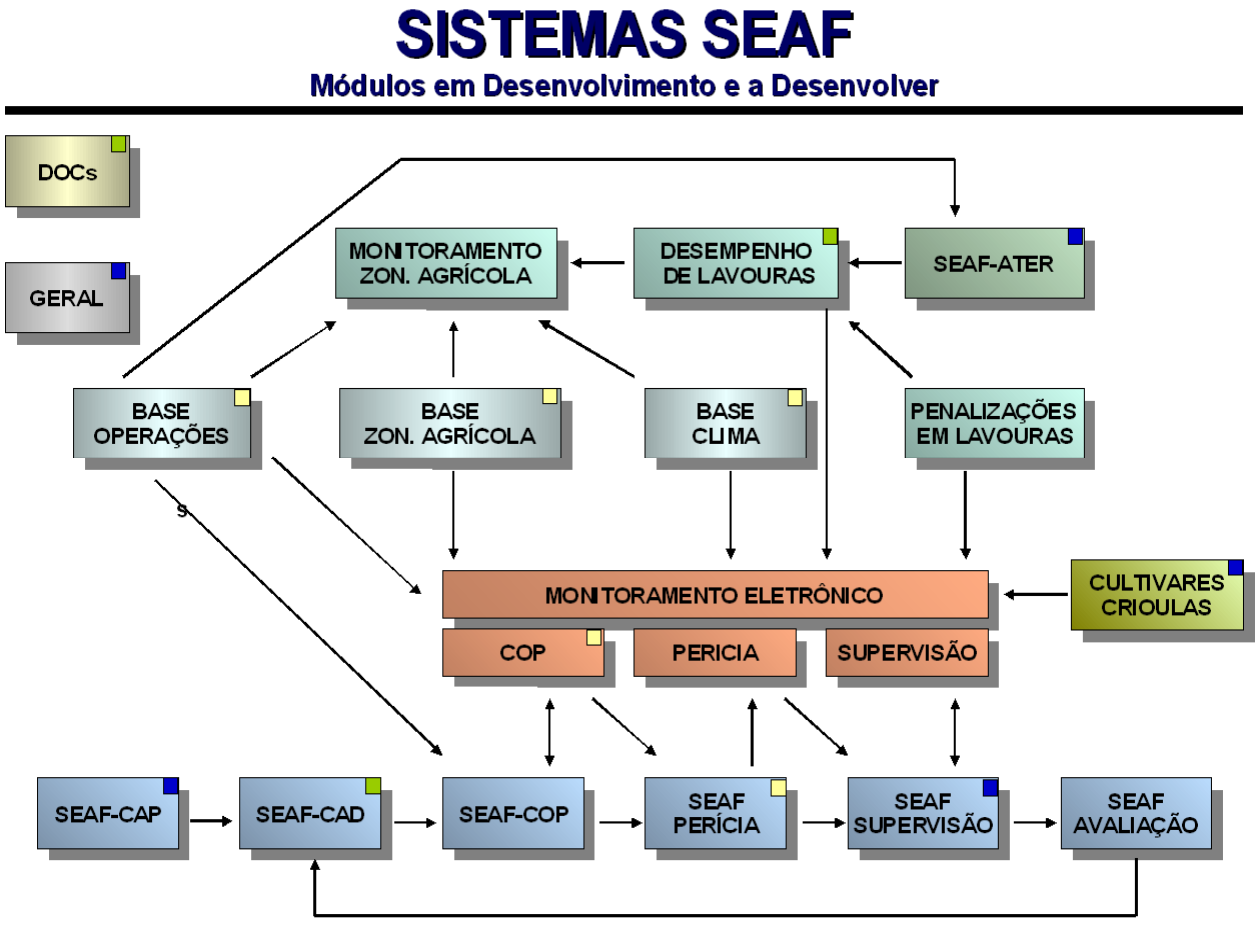


Figura 1 - Estrutura de Sistemas do SEAF

Devido à sua natureza ligada ao setor público, o seguro SEAF não é regido pela legislação de seguros privados, sendo um programa governamental criado com base nos princípios de seguros agrícolas. Não é um programa de renda mínima, nem um seguro de emergência assim como o **Bolsa Família** e **Garantia Safra**. O público alvo do SEAF compõe-se de produtores da agricultura familiar, que embora sejam de pequeno porte, podem tomar financiamento de até R$ 3.000 no Grupo C, de até R$ 6.000 no Grupo D ou valores maiores no Grupo E. Tem potencial para conduzir suas lavouras como um empreendimento e produzir receita para pagar o financiamento, cobrir os custos de manutenção familiar e realizar investimentos. Os números do SEAF giram em torno de 543,8 mil produtores amparados com valor total segurado em média por safra a R$ 2,5 bilhões.

Um dos princípios básicos de um seguro agrícola é que o prêmio deve ser suficiente para cobrir os gastos com indenizações e custos administrativos e ainda prover recursos para constituição de fundo de reserva. Baseado nisso o subsídio do governo é fundamental, mas o SEAF têm se mostrado como instrumento mais adequado do que programas de emergência, porque possibilita um tratamento mais sistematizado dos problemas causados por eventos agroclimáticos e falhas técnicas podendo ser um instrumento indutor da adoção de medidas de prevenção e redução de riscos. Além desse fato, um seguro, quando é possível constituir um fundo de reserva, possibilita uma melhor gestão orçamentária, evitando impactos abruptos nas contas públicas.

Além do desenvolvimento de cálculos atuariais e da definição das coberturas, são fatores críticos de sucesso no ramo de seguros a sistematização de informações sobre riscos e a especificação de regras e procedimentos visando definir os riscos cobertos dentro de limites e de condições administráveis. Destacam-se como carências nos seguros agrícolas brasileiros as bases de informações e tecnologias na área de gestão de riscos. É necessário avançar em um trabalho sistematizado nessa área. Isso envolve algumas linhas de ação que em grande parte dependem fundamentalmente de um conhecimento especializado na agronomia e agroclimatologia que compõem a realidade da agricultura familiar.

O planejamento da SAF prevê a cooperação entre o SEAF e a ATER a qual vem sendo implementada desde o ano agrícola 2005-2006, com base no desenvolvimento de:

a) trabalho de campo,
b) convênio com a Embrapa,
c) estruturação da base de informações do SEAF.

O trabalho de campo contempla um conjunto de ações que vão da orientação em campo com cuidados e procedimentos ao longo das diversas fases da lavoura, seleção de 5% dos empreendimentos para acompanhamento detalhado através dos três laudos de assistência técnica emitidos nas fases de pré-plantio, plantio e colheita, visitas locais a esses empreendimentos, e coleta sistematizada de informações para constituição da base de informações.

Com o trabalho de campo realizado é possível estruturarmos a base de informações para subsidiar o desenvolvimento de cálculos atuariais e diversos outros ações em gestão de riscos na agricultura familiar como metodologias de avaliação de impacto de eventos agroclimáticos sobre produtividades, sistemas de produção e estimativas de rendimento, indicativos de plantio,

normatizações sobre técnicas de cultivo, metodologias de periciamento, antecipação de perdas e provisionamento de COP.

1.2 Objetivo

Baseado no objetivo e justificativas do seguro SEAF, essa pesquisa foca o uso de técnicas de inteligência artificial (IA) para a detecção de evidências de emissão de comunicados de ocorrência de perdas em seguros agrícolas, com base em regras e padrões extraídos da base de laudos de assistência técnica. A partir dos dados contidos nestes três laudos de acompanhamento da safra, essa pesquisa propõe um método automático que viabilize analisar um maior número de laudos, contribuindo para minimizar os custos do pagamento de COP indevidas, sem ser necessário ampliar os atuais recursos humanos disponíveis na SEAF. O método proposto é baseado na construção de modelos de inteligência artificial para inferir se há evidências de que poderia ocorrer COP, a partir dos dados disponíveis nos laudos técnicos.

Os objetivos específicos dessa pesquisa são:

- 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,

- implementação em scripts na linguagem R, de prova de conceito de aplicativo inteligente que se baseie no método automático proposto, visando posterior avaliação pela SAF/MDA como ferramenta para acompanhamento de COP,

- proposição de novos algoritmos para construção de multiclassificadores

- desenvolvimento de aplicativos para motivar o staff da SEAF a utilizar técnicas de mineração de dados no acompanhamento do seguro SEAF.

1.3 Metodologia

Serão utilizados dados reais disponibilizados pela SEAF/MDA. O modelo de referência CRISP-DM (Seção 2.1) será utilizado como metodologia para mineração da base de dados. Os algoritmos a serem desenvolvidos serão programados em scripts R (Apêndice A). O problema de previsão de ocorrência de COP será abordado com a construção de classificadores e regras de associação, sendo COP a variável de interesse. Em complementação à fase de pré-processamento dos dados disponíveis, prevista no CRISP-DM, serão aplicadas técnicas de balanceamento de base de dados, pois a percentagem de ocorrência de COP na base disponível é de apenas 5%. Para avaliação dos modelos propostos, será utilizada a análise ROC e os usuais índices de performance: sensibilidade, especificidade, precisão e *F-measure*.

As etapas a serem executadas são:

a) extração dos laudos das bases de dados dos sistemas da SEAF/MDA

b) pré-processamento dos dados, incluindo tratamento de *outliers* e dos erros de preenchimento (Seção 2.2.1)
c) aplicação de técnicas estatísticas para seleção de variáveis visando identificar variáveis co-relacionadas e quais têm maior influência na explicação de COP (Seção 2.2.2)
d) aplicação de técnicas de balanceamento de classes (Seção 2.2.3),
e) avaliação de técnicas de mineração de dados para identificar quais apresentam melhor desempenho neste domínio (Seção 2.3)
f) proposição de métodos de combinação de classificadores individuais para construção de multiclassificadores com melhor desempenho (Seção 2.4).
g) avaliação dos resultados (2.7)

1.4 Áreas de Pesquisas Relacionadas

Agronomia é o conjunto das ciências e dos princípios que regem a prática da agricultura. Ela visa basicamente à produção, conservação, comercialização e consumo de alimentos. Para o desenvolvimento agronômico, é comum utilizarmos a extensão rural. A extensão rural, basicamente, é a distribuição do conhecimento tecnológico da área para a melhoria do setor agrícola. Em países com dimensões continentais como o Brasil, onde o setor agrícola é muito importante, a extensão rural é fundamental, mas possui vários complicadores. Um dos métodos de extensão rural é a promoção da visita de técnicos especialistas para realização de análises. No caso da SEAF/MDA, a realização de visita de técnicos é dificultada pela quantidade de produtores familiares e pelo alto custo de cada visita técnica.

Outra área vinculada a esse estudo é a de seguros, especificamente seguros agrícolas. A extensão do Brasil e a variedade de solos, climas e práticas agrícolas impedem a uniformização de tecnologias de extensão rural, bem como os processos de fiscalização e avaliações de fraudes em seguros agrícolas. O custo de deslocamento e a dificuldade operacional são fatores que favorecem trabalhar com amostragens. No caso da SEAF/MDA não é economicamente viável avaliar todas as COP, sendo bastante usual pagar a todas elas, visto que os valores segurados são pequenos.

1.5 Estrutura do Documento

Esta dissertação está organizada da seguinte forma: o Capítulo 2 descreve o modelo de referência CRISP-DM, focando técnicas de pré-processamento de dados, técnicas de balanceamento de classes, classificadores, algoritmos de associação e sistema multiclassificadores. O Capítulo 3 formaliza o problema abordado e a solução proposta para resolvê-lo. No Capítulo 4, os resultados obtidos com as abordagens propostas são apresentados e analisados através dos métodos de avaliações de índices de desempenho e análise ROC. Por fim, conclusões e trabalho futuro são descritos no Capítulo 5. O Apêndice A descreve todos os comandos e bibliotecas do R utilizadas e no Apêndice B são apresentados todos os scripts R desenvolvidos.

2 Fundamentação Teórica

Este capítulo apresenta a fundamentação teórica, descrevendo de forma concisa aspectos das técnicas e metodologias que serão utilizadas nesta pesquisa.

Para a realização desse estudo, realizamos uma pesquisa ampla de vários assuntos descritos ao longo desse capítulo. Utilizamos o *CiteSeer* como ferramenta de pesquisa de artigos e trabalhos relacionados ao nosso estudo.

Dentre os trabalhos e artigos pesquisados temos:

2.1 Estado da Arte

Para a construção dessa pesquisa foi realizado um estudo sobre assuntos semelhantes com objetivo de orientar o desenvolvimento desse trabalho. Os artigos mais relevantes seguem descritos nesse tópico.

A Brief Introduction to Boosting: Robert E. Scharpire (Schapire R. E., 1999)

Este artigo introduz o conceito de utilização do *Boosting* como técnica para melhorar a precisão de qualquer algoritmo de aprendizagem. Apresenta especificamente o algoritmo *AdaBoost*, explicando seu funcionamento e justifica o motivo por não acontecer *overfitting* na maioria dos casos.

An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting and Variants (Bauer & Kohavi, 1999)

Apresenta uma comparação empírica de algoritmos de classificação por votação: *Bagging*, *Boosting* e variantes. Utilizou como algoritmo de aprendizagem a árvore de decisão. Utilizou o algoritmo *p*roposto por (Breiman, 1996) para multiclassificadores por votação com diferentes amostras geradas por *Bootstrap* (Efron & Tibshirani 1993).

Bagging and Boosting with Dynamic Integration of Classifiers (Tsymba & Puuronen, 2000)

Apresenta com solução para reduzir erros de classificação a cooperação de classificadores em comitê de decisão. A principal decisão atual com aprendizagem de comissão impulsionar o conjunto de treinamento em máquinas com diferentes técnicas que derivam classificadoras bases. *Boosting* usa uma espécie de votação ponderada e *Bagging* usa o mesmo peso de voto como uma combinação de métodos. Ambos não levam em conta os aspectos locais que as bases classificadoras podem ter dentro do espaço do problema. A técnica proposta neste artigo para integração dinâmica é aplicado com *AdaBoost* e *Bagging*. Os resultados da comparação com vários dos conjuntos de dados mostram que o aumento da aprendizagem com a integração dinâmica de classificadores resulta em resultados melhores que os apenas aplicando *Bagging* e *Boosting*.

Bagging, Boosting and C4.5 (Quinlan, Bagging, Boosting and C4.5, 1996)

Conforme (Breiman, 1996) e (Schapire Y. F., 1996), *Bagging* e *Boosting* são métodos recentes para melhorar o poder preditivo de sistemas classificadores. Ambos formam um conjunto de classificadores que são combinados através do voto, através da geração de amostras *Bootstrap* dos dados, e uso de pesos ajustados de acordo com os resultados de instâncias. Este artigo apresenta os resultados da aplicação de ambas às técnicas a um sistema que aprende árvores de decisão e de testes em uma coleção representativa de conjuntos de dados onde apresentaram melhoras nos resultados. Por outro lado, produziram grave degradação em alguns conjuntos de dados. Uma pequena mudança na forma de combinação de classificadores reduz essa desvantagem e também leva a resultados ligeiramente melhores na maioria dos conjuntos de dados considerados.

Bagging Predictors (Breiman, 1996)

Apresenta o *Bagging*, como um método para melhorar resultado de classificação. As médias de agregação sobre as versões para predizer um resultado é feita por voto majoritário em múltiplas classificações realizadas com diferentes amostras de treinamento geradas com *Bootstrap*. Testado em dados reais com árvores de classificação e regressão mostram que o *Bagging* pode dar ganhos substanciais na precisão. O elemento vital é a instabilidade do método de previsão. A alteração da geração dos conjuntos de aprendizado gera significativa mudança no classificador construído e pode melhorar a precisão.

Boosting a weak learning algorithm by majority (Freund, Boosting a weak learning algorithm by majority, 1995)

Esse artigo apresenta um algoritmo para melhorar a precisão dos algoritmos de aprendizagem de conceitos binários. A melhoria é possível através da combinação de um grande número de hipóteses, cada uma é gerada pelo treinamento do algoritmo de aprendizado em um conjunto diferente de exemplos. Este algoritmo é baseado em idéias apresentadas por Shapire em seu artigo *"The strength of weak learnability"* e representa uma melhoria sobre seus resultados. A análise deste algoritmo fornece limites superiores gerais sobre os recursos necessários para a aprendizagem em polinômio *Valiant*, que são os melhores em geral de limites superiores. Mostra que o número de hipóteses que são combinados pelo o algoritmo é o menor número possível. Outros resultados da análise são os resultados sobre o poder de representação dos circuitos limiar, a relação entre a capacidade de aprendizado e de compressão, e um método para paralelização de algoritmos de aprendizagem *PAC*.

Experiments with a New Boosting Algorithm (Freund & Schapire, 1996)

Apresenta experimentos com algoritmo *AdaBoost* que, teoricamente pode ser usado para reduzir significativamente o erro de qualquer algoritmo de aprendizagem que constantemente gera classificadores cujo desempenho é um pouco melhor do que adivinhar aleatoriamente. Durante os estudos foram realizados dois experimentos. No primeiro em comparação entre o *AdaBoost* e *Bagging* de Breiman. Foi comparado o desempenho dos dois métodos em uma coleção de benchmarks de aprendizagem de máquina. No segundo conjunto de experimentos, foi estudado

mais detalhadamente o desempenho Do *Boosting* usando um classificador mais próximo de um problema de *OCR*.

Generalized Boosted Models (Ridgeway, 2007)

Foi possível avaliar que *Boosting* assume várias formas, com diferentes programas utilizando diferentes perdas de funções. Além disso, alguns algoritmos implementados no pacote *gbm* difere a implementação padrão. O algoritmo *AdaBoost* tem um função de perda particular e um algoritmo de otimização específicos associados. A implementação de *gbm AdaBoost* adota perda exponencial do *AdaBoost* função (seu limite na taxa de erro de classificação), mas utiliza a descida de Friedman gradiente ao invés do original proposto.

ipred : Improved Predictors (Peters, Hothorn, & Lause, 2002)

Existem várias tentativas para criar regras de atribuir observações futuras para certas classes em problemas de classificação. Os métodos mais comuns são análise discriminante linear ou árvores de classificação. Desenvolvimentos recentes levam a redução substancial do erro em classificação com a utilização de combinação de classificadores treinados em amostras *bootstrap* dos dados originais. Outra forma é a classificação indireta, incorpora um conhecimento a priori em uma regra de classificação. Os critérios para avaliar as técnicas de classificação são através do *cross-validation* ou erro de classificação. Esse artigo apresenta principalmente essas técnicas de calculo de erro, utilizadas para comparação entre classificadores. Essas técnicas estão implementadas na biblioteca *ipred*.

A Short Introduction to Boosting (Freund & Schapire, 1999)

Boosting é um método geral para melhorar a precisão de qualquer algoritmo de aprendizagem. Este artigo apresenta a visão do *AdaBoost*, e explica teoricamente do porque do *Boosting* muitas vezes não sofrem de *overfitting* como as *SVM*. Alguns exemplos de aplicações recentes de *Boosting* são descritos. Depois de introduzir *AdaBoost*, foi falado sobre teorias básica subjacente do *Boosting*, incluindo uma explicação do motivo pelo qual, muitas vezes, tende a não *overfit*. Descrito também algumas experiências e aplicações usando *Boosting*.

An Introduction to Boosting and Leveraging (Meir & Rätsch, 2003)

Este artigo mostra uma introdução aos aspectos teóricos e práticos do *boosting*, fornecendo uma referência útil para pesquisadores da área. Foi ilustrada a utilidade de aumentar algoritmos, dando uma visão geral de alguns aplicativos existentes, as principais idéias sobre o problema de classificação binária, embora várias extensões sejam discutidas.

mboost Illustrations (Hothorn & Buhlmann, 2007)

Este artigo reproduz as análises de dados apresentados em (Buhlmann & Hothorn, 2007). Para uma descrição da teoria por trás de aplicativos. Os resultados difere ligeiramente, devido a alterações técnicas, problemas no *mboost* que têm sido implementadas. Mais importante *gamboost*, usa penalizado *B-splines* em vez de suavização de *splines* como *baselearners*. Os cálculos são muito mais rápidos e o resultado difere um pouco do original.

Why Does Bagging Work? A Bayesian Account and its implications (Domingos, 1997)

A taxa de erro de árvore de decisão e alunos em outra classificação muitas vezes pode ser muito reduzida por ensacamento: aprendendo com vários modelos a partir de amostras *bootstrap* do banco de dados, e combiná-los através do voto uniforme.

Multi-Classifier Systems - A Review and Roadmap for Developers (Ranawana & Palade, 2006)

Este artigo apresenta um pouco sobre Sistemas Multiclassificador (MCS) e como vem ganhando uma rápida popularidade entre os pesquisadores por sua capacidade de saídas múltiplas para classificação, melhor precisão e classificação. O artigo apresenta um panorama atual da MCS e as tentativas de fornecer um roteiro para designers MCS. Foi identificado todas as principais decisões que um designer teria que fazer sobre o projeto de um sistema de MCS e lista as opções mais úteis disponíveis em cada tomada de decisão.

A decision-theoretic generalization of on-line learning and a application to boosting. (Freund & Schapire, 1995)

Este artigo considera o problema da repartição de recursos dinamicamente entre um conjunto de opções em um pior quadro. O modelo que pode ser interpretado como uma extensão, resumo do modelo de previsão bem estudada para uma configuração de decisão teórico-geral. Aplicar as técnicas de peso multiplicativo para derivar um novo algoritmo de reforço. Este algoritmo de *boosting* não exige qualquer conhecimento prévio sobre o desempenho de algoritmo inferior.

2.2 Modelo de Referência CRISP-DM

Este pesquisa adota como metodologia de mineração de dados o modelo de referência criado em 1996 por um grupo de especialistas do mercado de *Data Mining*, o CRISP-DM (*CRoss-Industry Standard Process for Data Mining*), o qual surgiu com o intuito de promover a padronização de conceitos e técnicas na busca de informações específicas para tomada de decisões. Seu objetivo é auxiliar administradores e responsáveis no processo geral de planejar e executar a mineração de dados, englobando a especificação do processo até a apresentação dos resultados.

O CRISP-DM se baseia na existência de um ciclo de vida para um projeto de mineração de dados, seguindo uma seqüência de seis fases, conforme mostra a Figura 2. Pode ser necessário retornar a alguma fase já executada. O círculo externo representa a natureza cíclica própria da mineração de dados, pois os processos subseqüentes no *Data Mining* se beneficiarão da experiência adquirida dos anteriores (CRISP, 2009). As flechas representam as dependências mais importantes e freqüentes entre as fases.

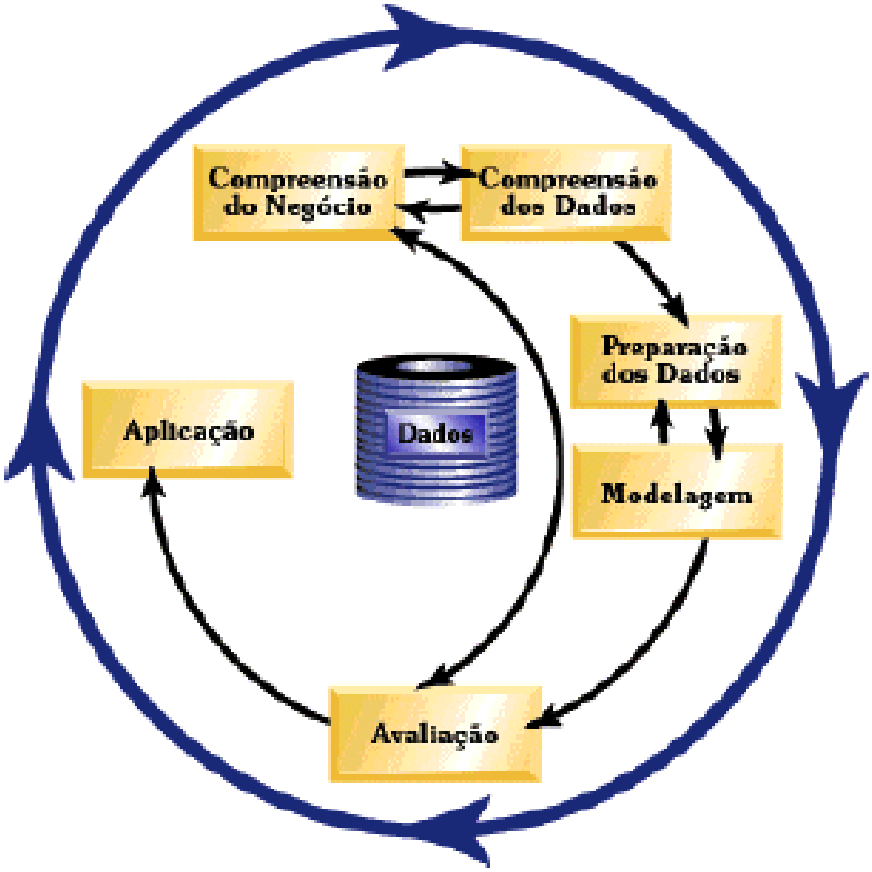


Figura 2 - Ciclo de Vida do CRISP-DM

1ª Fase: Compreensão do Negócio

Esta fase visa identificar as necessidades e os objetivos do negócio do cliente, convertendo este conhecimento em uma tarefa de mineração de dados. Também procura detectar algum problema ou restrição que, se desconsiderados, poderão implicar em perda de tempo e esforço em obter respostas corretas para questões erradas, prejudicando o projeto. Planos de contingência para estes casos devem ser elaborados. As principais tarefas relacionadas a esta fase são:

- avaliação dos recursos disponíveis, tais como *hardware*, *software* etc.
- listagem das obrigações do projeto, como cronograma e requerimentos.
- construção da análise de custo-benefício do projeto.
- determinação de onde serão obtidos os dados para treinamento e teste, assim como quais ferramentas e técnicas serão utilizadas para realizar o processo de *Data Mining*.

2ª Fase: Compreensão dos Dados

Nesta fase seleciona-se um conjunto de dados ou foca-se sobre um conjunto de variáveis ou amostra de dados, nos quais será realizado o processo de descoberta. Abrange a coleta inicial, geralmente feita na base de dados operacional, que pode ser feita de várias fontes. Esta coleta inicial procura adquirir as informações com as quais se irão trabalhar, relacionando a forma como os dados foram adquiridos, o procedimento de leitura e os problemas detectados durante a extração. Deve-se

fazer ainda uma descrição dos dados (formato, quantidade de registros e campos) e detalhar uma exploração dos mesmos, contendo: distribuição de atributos chave, relacionamento entre pares de atributos, agregações, propriedades de sub-populações e análise estatística. Neste passo, também é analisada a qualidade dos dados com a ajuda da Estatística, evitando que se trabalhe com uma amostra de dados viciada ou não representativa do domínio, por exemplo.

3ª Etapa: Preparação dos Dados

A preparação ou pré-processamento dos dados consiste em tarefas destinadas a obter o conjunto final dos dados, a partir do qual será criado e validado(s) o(s) modelo(s). Essas tarefas de pré-processamento incluem: seleção, limpeza, construção, integração e formatação dos dados de entrada para o tipo de ferramenta que será utilizada. Operações como remoção de dados estranhos são realizadas, além da coleta da informação necessária ao modelo, definição de estratégias para lidar com valores faltantes, produção de atributos derivados, criação de novos registros, integração de tabelas, operações de agregações, reformatação dos dados e discretização dos dados numéricos, se for necessário.

4ª Etapa: Modelagem

Após o entendimento do domínio do negócio e dos dados e da preparação dos mesmos, chega-se à fase de modelagem. Nela deve-se selecionar e aplicar as tarefas de mineração de dados mais apropriadas, definindo se o objetivo do processo é a classificação, regressão, agrupamento (*clustering*), dentre outros. Isto orientará a escolha dos algoritmos de mineração de dados.

A Tabela 1 descreve resumidamente as tarefas de mineração de dados. Neste contexto, tarefa é um tipo de problema de descoberta de conhecimento a ser solucionado.

TAREFA	**DESCRIÇÃO**	**EXEMPLOS**
Classificação	Constrói um modelo de algum tipo que possa ser aplicado a dados não classificados a fim de categorizá-los.	Classificar pedidos de crédito ou identificar a melhor forma de tratamento de um paciente.
Regressão	Constrói um modelo para estimar um valor para alguma variável continua cujo valor é desconhecido.	Estimar o número de filhos ou a renda total de uma família. Prever a demanda de um consumidor para um novo produto.
Associação	Usada para determinar quais itens tendem a ocorrer (serem adquiridos juntos) em uma mesma transação.	Determinar quais produtos costumam ser colocados juntos em um carrinho de supermercado.
Agrupamento	Processo de partição de uma população heterogênea em vários	Agrupar clientes por região do país. Agrupar clientes com comportamento de

	subgrupos mais homogêneos.	compra similar.
Sumarização	Envolve métodos para encontrar descrição compacta para dados.	Tabular o significado, média e desvio padrão para todos os itens de dados.

Tabela 1- Tarefas realizadas por Técnicas de Mineração de Dados.

A escolha de determinado método de mineração de dados deve levar em consideração os critérios globais do sistema. O usuário final normalmente está mais interessado na compreensibilidade do modelo do que na sua capacidade de previsão.

A criação de um conjunto de dados para teste permite construir mecanismo para comprovar a qualidade e validar os modelos que serão obtidos. É necessário então dividir a massa de dados em conjuntos de treinamento, testes e validação.

Logo, esta etapa representa a fase central da mineração, incluindo escolha, parametrização e execução de técnicas sobre a massa de dados a fim de obter um ou vários modelos.

5ª Etapa: Avaliação

A avaliação do(s) modelo(s) consiste na revisão dos passos seguidos, verificando se os resultados obtidos condizem com os objetivos propostos previamente na compreensão do negócio. De acordo com os resultados alcançados, na revisão do processo, decide-se pela continuidade do projeto e definem-se quais as próximas tarefas ou se correções devem ser efetuadas, e se é preciso retornar a algum dos passos anteriores.

Nesta revisão do processo devem-se identificar também atividades que possam ter sido esquecidas ou que devam ser repetidas.

6ª Etapa: Aplicação

A Aplicação é o conjunto de ações que conduzem à organização do conhecimento obtido e à sua disponibilização de forma que possa ser utilizado eficientemente pelo cliente. Nesta etapa, consolidam-se o conhecimento descoberto, sendo necessário apresentar um relatório para explicar tempo gasto, custo envolvido, experiências, além da sugestão de trabalho futuro a fim de melhorar os resultados obtidos. A elaboração deste relatório final e a respectiva apresentação ao usuário oficializam a conclusão do projeto.

2.3 Técnicas Utilizadas para o Pré-Processamento dos Dados

Nesse tópico iremos abordar algumas técnicas utilizadas na fase de pré-processamento dos dados com objetivo de limpar e preparar a base para ser utilizada na criação de modelos em inteligência artificial.

2.3.1 *Outliers*

Em inteligência artificial modelos podem ser aprendidos por indução através da aplicação de algoritmos numéricos a base de dados. Quando essa base de dados encontra-se com informações erradas, podemos gerar modelos que não condizem com a realidade. Por isso é necessário realizar um tratamento para evitar esse problema.

Em geral, as instâncias de dados que apresentam um grande desvio em relação às demais são consideradas inconsistentes e designadas *outliers*. Estas observações são também chamadas de observações anormais, contaminantes, estranhas, extremas ou aberrantes (Figueira, 1998).

As principais causas que levam ao aparecimento de *outliers* são erros de medição, erros de execução, e variabilidade inerente dos elementos da população (Figueira, 1998).

Alguns métodos para identificação de *outliers* são: gráfico de Box-Plot, modelos de discordância, teste de Dixon, teste de Grubbs, e Z-scores. Cada método de identificação de *outliers* tem suas características e se adapta melhor em certas situações. Dessa forma, é necessário estudar cada um deles para identificar qual se aplica melhor ao caso em questão.

Para identificação de *outliers* também é importante observarmos a necessidade de repetirmos o procedimento até atingirmos um nível de erros consideravelmente aceitáveis. É importante observar que toda a remoção de erros deve ser confirmada pelo especialista da área, pois os algoritmos de identificação de *outliers* podem identificar erros equivocadamente.

Gráfico de Box-Plot

O gráfico de Box é calculado da seguinte forma (Figura 3):

- Calcula-se a mediana, o quartil inferior (Q_1) e o quartil superior (Q_3);
- $L = Q_3 - Q_1$
- Os valores que estiverem no intervalo entre $Q_3+1,5L$ e Q_3+3L e no intervalo entre Q_1-1,5L e Q_1-3L, serão considerados *outliers* e podem ser aceitos com alguma suspeita
- Os valores maiores que Q_3+3L e menores que Q_1-3L são chamados de extremos e devem ser considerados suspeitos, deve-se ser investigada a origem da dispersão.

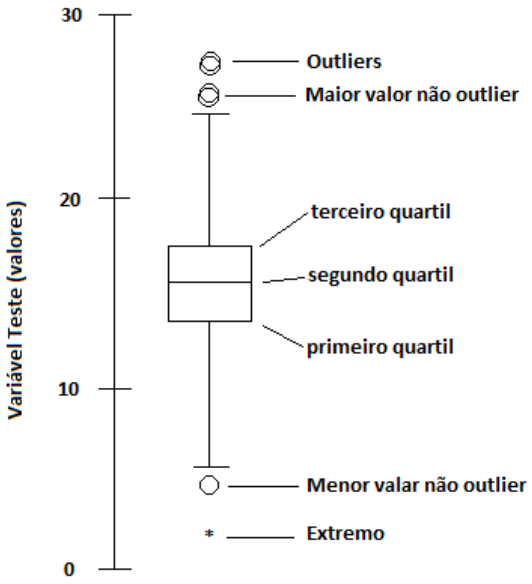


Figura 3 - Exemplo Gráfico Box

Teste de Dixon

É utilizado para identificar se um valor de uma pequena base (até 10 elementos) deve ser ou não considerado *outlier*. O teste de Dixon funciona da seguinte forma:

- Distribuição normal; teste bilateral.
- Ordenar os valores de forma crescente de 1 a H (máximo). (identifica apenas se os extremos são *outliers)*
- Supor a hipótese de que o menor valor, 1, ou o maior valor, H, são suspeitos como valores *outliers*.
- Tabela Q crítico (D.B. Rorabacher, 1991)

N	Q_{crit} (CL:90%)	Q_{crit} (CL:95%)	Q_{crit} (CL:99%)
3	0.941	0.970	0.994
4	0.765	0.829	0.926
5	0.642	0.710	0.821
6	0.560	0.625	0.740
7	0.507	0.568	0.680
8	0.468	0.526	0.634
9	0.437	0.493	0.598
10	0.412	0.466	0.568

Tabela 2- Q_{crit} - Teste Dixon / N tamanho da base; CL confiança

- Calcular Q
  - Q inferior (limite inferior)

$$Q = \frac{x_2 - x_1}{x_n - x_1}$$

- Q superior (limite superior)

$$Q = \frac{x_n - x_{n-1}}{x_n - x_1}$$

- Se o Q calculado for menor que o Q crítico do tamanho da base com confiança de 95% então ele deverá ser considerado *outlier*.

Teste de Grubbs

O teste de Grubbs (Grubbs, 1969) (Stefansky, 1972) funciona avaliando os extremos do conjunto de dados verificando se o valor é disperso. Se o valor for disperso ele é removido e o teste é refeito. Esse teste pode remover vários *outliers* em um único teste.

Os seus passos são:

- Distribuição Normal
- Cálculo do desvio d_i em relação à média

$$d_i = |x_i - \bar{x}|$$

- Calcular o desvio padrão *s*
- Calcular G

$$G = \frac{|x_i - \bar{x}|}{s}$$

- O valor G calculado é considerado *outiler* se for maior do que a tabela G crítico

n	G_{crit} 95 %
3	1,15
4	1,48
5	1,72
6	1,89
7	2,02
8	2,13
9	2,22
10	2,29
11	2,36
12	2,41
14	2,51
16	2,59
18	2,65
20	2,71
50	3,13

Tabela 3 - Tabela G crítico teste de Grubbs

Z-score

Com essa técnica basta calcular os *z-scores* (Iglewicz, 1993) de cada valor da base e seguir as orientações abaixo para identificar se é ou não um *outlier*:

- Se o conjunto dos dados é pequeno (inferior a 50), valores que tenham *z-scores* inferiores a -2.5 ou superiores a 2.5 devem ser considerados *outliers*.
- Se o conjunto dos dados é grande, valores que tenham *z-scores* inferiores a -3.3 ou superiores a 3.3 são tipicamente considerados *outliers*.
- Se o conjunto dos dados é muito grande (1000 ou mais), também valores mais extremos do que +-3.3 podem ser considerados dados normais e não *outliers*. Para isso é bom avaliar especificamente a variável em questão.

O cálculo do *z-score* (z_i) é definio abaixo:

$$z_i = \frac{(x_i - \overline{x})}{s}$$

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}}$$

2.3.2 Correlação entre variáveis – Método Kendall

De forma geral é possível afirmar que modelos com maior número de variáveis são mais difíceis de serem compreendidos e utilizados por pessoas do que modelos com menos variáveis. Nesse caso é fundamental realizar uma verificação da relação entre variáveis para remover as que possam trazer a informação. Outro fato importante é que a chance de termos variáveis irrelevantes em um conjunto grande de variáveis é maior. Variáveis irrelevantes geralmente confundem o algoritmo de aprendizado, gerando perda de eficiência do modelo.

A correlação do Kendall avalia o grau de similaridade entre dois conjuntos informando um coeficiente de correlação. Este coeficiente depende do número de inversões de pares de objetos que seriam necessárias para transformar a ordem. Para isso a ordem de classificação é representada pelo conjunto de todos os pares de objetos atribuindo 1 para par correspondente e 0 para par não correspondente. (KENDALL, 1955) apud (Abdi, 2007)

Então o coeficiente Kendall *t* é definido como:

$$t = \frac{número\ de\ pares\ correspondentes - número\ de\ pares\ não\ correspondentes}{\frac{1}{2}n * (n-1)}$$

O coeficiente Kendall *t* é definido no intervalo [-1,1] onde -1 representa não correspondência perfeita e 1 representa a correspondência perfeita entre dois *rankings*.

2.3.3 Técnicas de Balanceamento

No mundo real é comum encontrarmos casos onde o desbalanceamento faz parte da realidade do problema (Visa, 2005). Como na tarefa de classificação temos apenas uma variável de classe que é a variável de interesse, podemos, sem perda de generalidade, também utilizar o termo classe para designar os valores que a variável de classe assume. Se classe de menor freqüência em uma base de dados for a de interesse, o desempenho dos classificadores construídos por indução será afetado

pois qualquer algoritmo utilizado para construí-los tenderá a reproduzir as classes mais freqüentes. Assim é necessário aplicar técnicas de balanceamento quando existir desbalanceamento de classes. Dizemos que uma base de dados é desbalanceada quando a distribuição das classes for muito assimétrica. Num exemplo com duas classes, dizemos que a base de dados está balanceada se a proporção entre elas variar entre 60% a 40%.

Existem diferentes formas e técnicas para realizarmos o balanceamento de classe, sendo as mais utilizadas as seguintes:

- *under-sampling*. Consiste em diminuir os casos da classe majoritária tentando obter equilíbrio entre todas as classes (Kubat, 1997). Essa técnica possui uma desvantagem muito grande que é a perda de informação, visto que teremos que diminuir o número de casos da classe majoritária. Para minimizar essa perda realizamos o seguinte procedimento:
    - agruparmos os registros iguais e criamos uma coluna de contagem de casos,
    - selecionamos os registros com maior número de casos e diminuímos sua freqüência,
    - não devemos optar em remover casos que tenham contagem baixa para não perdemos informação ao ponto de não existirem na nova base.

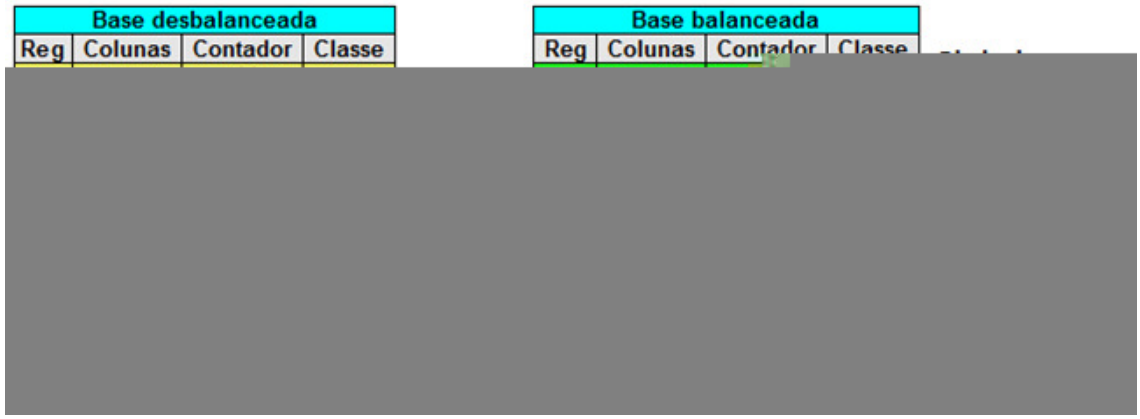


Figura 4- *Under-sampling*

- *over-sampling*: consiste em aumentar os casos da classe minoritária tentando aproximar a representação de todas as classes (Ling, 1998).

  Essa técnica possui a desvantagem de não agregar novas informações e possibilidade de criar *overfitting*, isto é, o modelo se tornar muito aderente aos dados atuais e perder a capacidade de generalização ao ser aplicado em novos conjuntos de dados. O *overfitting* é um problema visto que o modelo criado perde a generalidade, se restringindo a apenas repetir padrões existentes nos dados de entrada. A conseqüência disso é o mau funcionamento do modelo em novas bases de dados.

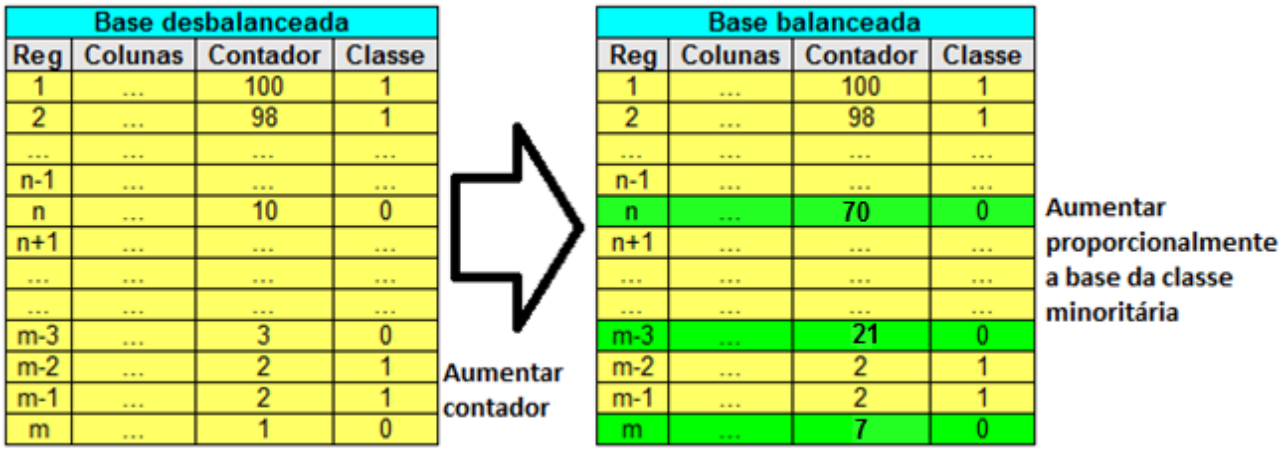


Figura 5 - *over-sampling*

- combinação do *under-sampling* e *over-sampling.*
  Nessa técnica combinamos as duas técnicas anteriores diminuindo os casos da classe majoritários e aumentando os casos da classe minoritária.

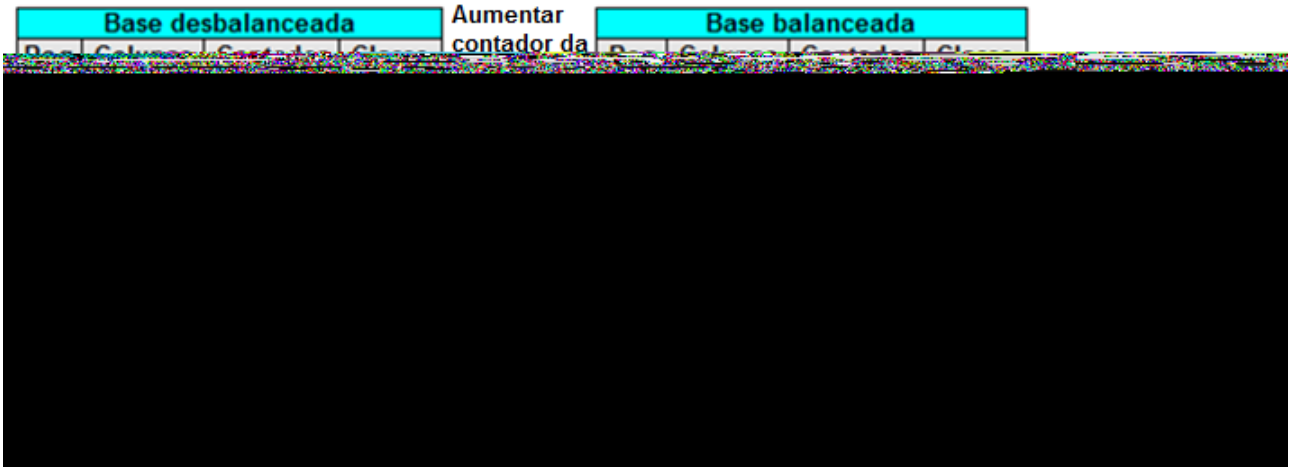


Figura 6 - combinação do *under-sampling* e *over-sampling*

2.4 Classificadores

A tarefa de classificação em mineração de dados consiste em aprender um classificador que mapeia um item de dado para uma entre as várias classes pré-definidas da variável de classe. A construção de um classificador é baseada na indução de um modelo (classificador) a partir de um conjunto de exemplos de dados. Os algoritmos de indução considerados nessa pesquisa foram: árvore de decisão (algoritmo C4.5), *Naive Bayes*, árvore de inferência condicional, SVM – *Support Vector Machine*, e KNN – *K-nearest neighbor*, sendo que árvore de decisão, SVM e KNN são exemplos de classificadores determinísticos. Já *Naive Bayes* e árvore de inferência condicional são exemplos de classificadores probabilísticos.

2.4.1 Árvore de Decisão – C4.5

O algoritmo ID3 (Quinlan, 1986) constrói uma árvore na qual os nós correspondem a variáveis utilizadas (atributos) no modelo para determinar a classe (valor) da variável de classe (variável de interesse). Os nós folha são formados por classes da variável de classe. A base de dados (conjunto de instâncias) não pode ter dados faltantes para nenhum atributo. O ID3 efetua um teste estatístico sobre cada um dos atributos para determinar qual deles melhor classifica os exemplos, ou seja, qual o que separa melhor os exemplos. Para tanto, ele determina para cada atributo o ganho da informação que se obtém caso se utilize este atributo para realizar a partição. O algoritmo usa esta medida de ganho de informação para selecionar, entre os atributos candidatos, qual será utilizado como sub-raiz, a cada passo da construção da árvore. O ganho da informação, que mede o quanto um atributo separa o conjunto de treinamento de acordo com a melhor classificação alvo, se baseia na medida entropia da teoria da informação. A entropia caracteriza a pureza ou impureza de uma coleção arbitrária de dados, ou seja, mede a homogeneidade dos dados. Dado um conjunto de exemplos *S*, em que o atributo alvo (variável de classe) pode ter c valores diferentes, *H*, a entropia de *S* em relação à variável de classe, será dada por:

$$Entropia(S) = \sum_{i=1}^{c} (-p_i \log_2 p_i)$$

onde p_i é a proporção de exemplos de *S* que pertencem à classe i. A entropia é 0, mínima (homogeneidade máxima), se todos os membros de S pertencem à mesma classe, e 1, máxima (homogeneidade mínima), quando as classes da variável de classe possuem uma distribuição uniforme, isto é, cada classe contém um número igual de instâncias (exemplos) (Quinlan, 1986).

Para um melhor entendimento do significado desta medida, considere o caso de dois eventos com probabilidades *p* e *(1 – p)*, então se tem:

$$H = [-p \log_2 p - (1-p) \log_2 (1-p)]$$

que pode ser melhor visualizado na Figura 7. Como os logaritmos das freqüências p_i são negativos, então a entropia se torna positiva. A entropia é medida em unidades denominadas de bits. O ID3 é um algoritmo recursivo que a cada passo seleciona uma variável para compor um nó na árvore de decisão. O atributo que tiver o maior ganho de informação é o escolhido para compor o nó da árvore. Os valores desse atributo impõem um particionamento dos dados disponíveis (as instâncias). O procedimento de seleção de novo atributo é repetido para as partições de dados (isto é, para os subconjuntos de dados com mesmo valor para esse atributo) até que esse subconjunto seja homogêneo com relação às classes (valores) da variável de classe.

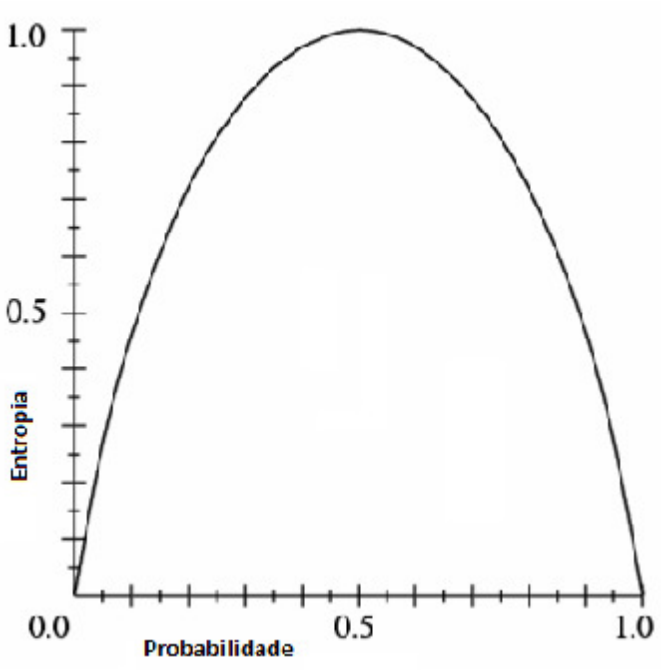


Figura 7 - Entropia x Probabilidade

O ganho da informação é dado por:

$$Ganho(S,A) \equiv Entropia(S) - \sum_{v \in V(A)} \left(\frac{|S_v|}{|S|} Entropia(S_v) \right)$$

V(A) é o conjunto de todos os valores possíveis para o atributo *A*, e S_v é o subconjunto de *S* para o qual o atributo A tem valor v. Observe que na equação de ganho da informação o primeiro termo é só a entropia da coleção original S, dada a variável de classe, e o segundo termo é o valor esperado da entropia depois que S é particionado usando o atributo A. Então o ganho da informação é a diferença entre estes valores.

O C4.5 é um algoritmo usado para gerar uma árvore de decisão desenvolvido por Ross Quilan (Quinlan, 1993) como uma extensão do ID3 que permite base de dados com dados faltantes e poda da árvore para evitar super ajuste aos dados. No C4.5, o critério de seleção do atributo para a partição das instâncias, maior ganho de informação, é substituído pelo critério de maior taxa de ganho de informação.

O C4.5 é um algoritmo recursivo que cria árvores de decisão, a partir de um conjunto de dados de treinamento, de forma similar ao ID3, usando o conceito de entropia de informação.

Em cada nó da árvore, o C4.5 escolhe um atributo dos dados que mais efetivamente divide seu conjunto de amostras em subgrupos enriquecido em uma classe ou outra. Seu critério é a taxa de ganho de informação.

O algoritmo C4.5 introduz a possibilidade de trabalhar com valores faltantes (*missing values*), com valores contínuos, podar árvores de decisão e derivar regras (Ingargiola, 1996), (Quinlan, 1993).

Trabalhar com dados faltantes pode ser um problema na hora da criação de árvore de decisão. No C4.5 na criação da árvore de decisão, os registros dados faltantes tanto podem ser descartados quanto classificados pela estimativa da probabilidade dos vários valores possíveis (Ingargiola, 1996).

Para a utilização de valores contínuos é necessária a ordenação dos valores de forma crescente para que possa ser feito a divisão. Dependendo do tamanho da base a ordenação pode exigir muitas computações para a ordenação (Ingargiola, 1996).

Com o surgimento de subárvores complexas a solução é um mecanismo de poda. O método de podar é realizado substituindo uma subárvore por um nodo folha. Este método é realizado se uma regra de decisão estabelecer que a taxa de erro prevista na subárvore é muito grande, em relação a utilização de um único nodo folha. A substituição de partes da árvore é realizada considerando que estas não contribuem à exatidão da classificação em determinados casos, produzindo algo menos complexo e assim mais compreensível (Ingargiola, 1996), (Quinlan, 1993). Nesta pesquisa foi utilizado apenas o algoritmo C4.5 para indução de árvores de decisão.

2.4.2 Naive Bayes

O classificador *Naive Bayes* (Friedman, 1997) também pode ser referenciado como classificador bayesiano ingênuo ou rede bayesiana ingênua. Assim como as árvores de decisão e as redes neurais, o classificador *Naive Bayes* é um dos métodos de aprendizagem mais prático.

O classificador bayesiano ingênuo se baseia na aplicação do teorema de Bayes. Entretanto, ele assume que os valores dos atributos são condicionalmente independentes, dado um valor da variável de classe. Esta suposição reduz a complexidade de aprendizagem da distribuição de probabilidades da variável de classe.

O procedimento de construção de uma rede bayesiana ingênua é composto por dois passos. Primeiramente deve-se deixar o nó de classe C ser o pai de todos os outros nós. A seguir aprende-se os parâmetros (valores das distribuições de probabilidade de cada atributo, dado um valor da variável de classe) e produz a *Naive Bayes*. A abaixo exemplifica uma estrutura de árvore gerada por um classificador *Naive Bayes*.

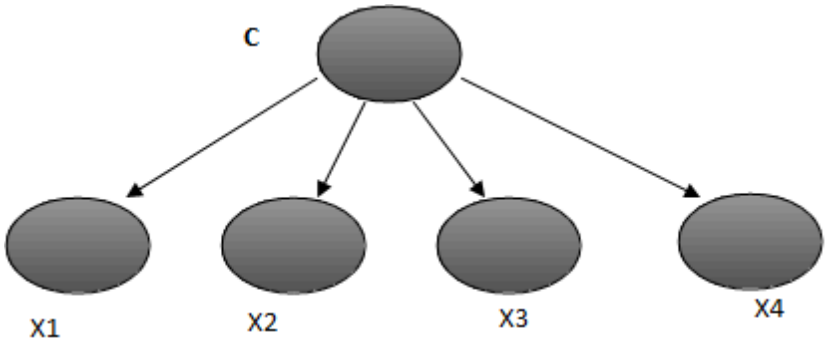


Figura 8 - Árvore Gerada pelo *Naive Bayes*

Em geral, existe um espaço de hipóteses (V), e deseja-se determinar a hipótese ($v_i \in V$) mais provável, observados os dados de treino (D). No caso, v_i são as classes possíveis, ou seja os valores que a variável de classe pode assumir. Supondo que uma instância $I \epsilon D$, onde D são as instâncias de dados, consiste em um vetor de valores relativos aos seus m atributos $(a_1, a_2, \dots, a_m)$, pretende-se obter o V_{map}, – estimador da Hipótese de máxima probabilidade a *posteriori* – que é o valor alvo (classe) mais provável, dentre os classificadores possíveis de *V*, isto é:

$$V_{map} = arg\ max_v\ \mathrm{P}(v|a_1, a_2, \dots, a_m), v \epsilon V$$

mas, pelo teorema de *Bayes*, a igualdade apresentada é equivalente a:

$$V_{map} = arg\ max_v \frac{\mathrm{P}(a_1, a_2, \dots, a_m \mid v)P(v)}{\mathrm{P}(a_1, a_2, \dots, a_m)}, v \epsilon V$$

Como o termo que está no denominador, o normalizador, não interfere no resultado final, por ser independente do classificador em causa (*v*), pode ser suprimido tendo-se apenas:

$$V_{map} = arg\ max_v\ \mathrm{P}(a_1, a_2, \dots, a_m \mid v)P(v), v \epsilon V$$

Admitindo agora a independência de $a_1, a_2, \dots, a_m$ chegamos à fórmula do classificador *Naive Bayes*:

$$V_{map} = V_{nb} = \arg max_v P(v) \prod_{i=1}^{m} P(a_i / v), v \epsilon V$$

Esta última fórmula é mais aplicável, visto que só é necessário calcular $k * m$ probabilidades $(k = |V|)$, enquanto que para o classificador V_{map}, mais geral, esse valor é da ordem m^k, simplificando muito em termos de complexidade computacional.

O classificador V_{nb} é um classificador amplamente utilizado e, em certos domínios, seu desempenho pode ultrapassar o obtido por outros classificadores como as redes neurais e as árvores de decisão.

2.4.3 Árvore de Inferência Condicional (CTREE)

Árvore de inferência condicional estima uma relação de regressão de particionamento recursivo binário em um quadro de inferência condicional. O algoritmo (Figura 9) funciona da seguinte forma (Hothorn T. K., 2006):

- teste a hipótese nula global de independência entre quaisquer das variáveis de entrada e a resposta. Pare se essa hipótese não puder ser rejeitada. Essa associação é medida por um *p-value* correspondente a um teste para a hipótese de nulidade parcial de uma variável de entrada única e da resposta.
- divisão binária na variável de entrada selecionada.
- repetir recursivamente as etapas anteriores.

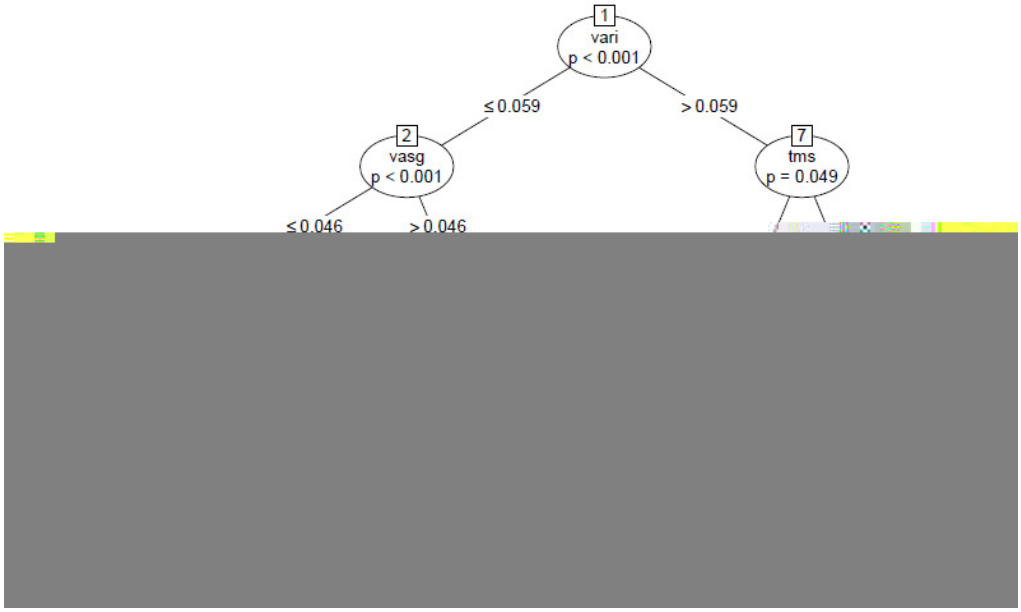


Figura 9 - Exemplo de Árvore de Inferência Condicional

2.4.4 SVM

Os algoritmos de aprendizagem de máquina SVM têm como objetivo a determinação de limites de decisão que produzam uma separação ótima entre classes por meio da minimização dos erros (Vapnik, 1995). O SVM consiste em uma técnica computacional de aprendizado para problemas de reconhecimento de padrão. A classificação SVM é baseada no principio de separação ótima entre classes, tal que se as classes são separáveis, a solução é escolhida de forma a separar ao máximo as classes (Figura 10).

O algoritmo pode ser descrito da seguinte forma: dadas D amostras de treinamento $\{x_i, y_i\}$, com i =1,2..., D, onde $x_i \in R^M$ é uma representação vetorial de um conjunto e $y_i \in \{-1,1\}$ é sua classe associada. Neste processo existe uma distribuição de probabilidade $P(x, y)$ desconhecida da qual os dados de treinamento serão retirados. Ou seja, o processo de treinamento consiste em treinar um classificador de forma que este aprenda um mapeamento de x em y por meio de exemplos de treinamento $\{x_i, y_i\}$ de forma que a máquina seja capaz de classificar um exemplo (x, y) ainda não visto que siga a mesma distribuição de probabilidade (P) dos exemplos de treinamento.

No SVM a expectativa de erro de classificação é dada por:

$$\varepsilon(\delta) = \int \frac{1}{2} \mid y - f(x, \delta)| \, dP(x, y)$$

A distribuição de probabilidade $P(x, y)$ não é conhecida e por isso não é possível resolver esse equação. Dessa forma definimos o risco empírico que é a média da taxa de erro nos elementos do conjunto de treinamento representado por:

$$\varepsilon_{\varphi}(\delta) = \frac{1}{2D} \sum_{i=1}^{D} |y_i - f(x_i, \delta)|$$

Sendo que $\varepsilon_{\varphi}(\delta)$ é fixo para um *V* arbitrário e um conjunto de treinamento $\{x_i, y_i\}$.

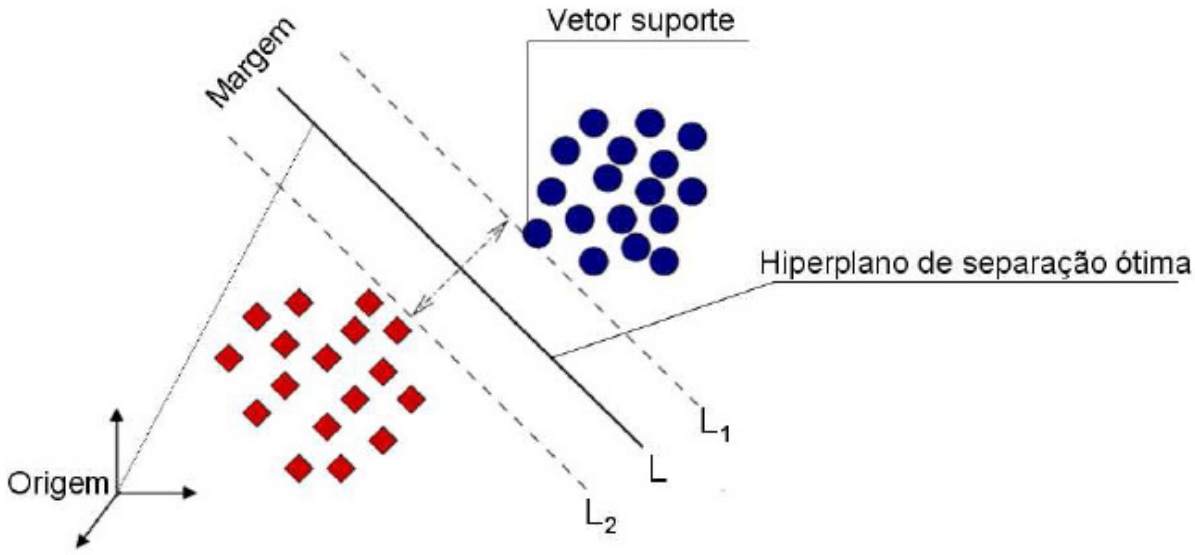


Figura 10 - Esquema de classificação por meio do SVM

2.4.5 KNN

Uma abordagem alternativa à indução de um modelo a partir de instâncias de casos classificados é classificar um novo caso comparando-o com casos conhecidos e classificá-lo de acordo com a classificação do caso mais próximo. Como exemplo ilustrativo, seria dizer que a partir da definição de sintomas um médico poderia pesquisar em suas fichas de atendimento antigas e compará-las, sendo que a ficha que apresentar maior similaridade será a que definirá o diagnóstico desse novo paciente.

O algoritmo *k-nearest neighbor* ou "k-vizinhos mais próximos" é um algoritmo de aprendizado supervisionado (Aha, 1991). O algoritmo propõe identificar os k casos conhecidos mais próximos e, com base nesses k casos, realizar a classificação do novo caso. Os pontos importantes desse algoritmo são: definição dos casos de treinamento a ser considerados; medida para quantificar similaridade, e quantos/quais vizinhos mais próximos a ser considerados. Uma questão importante no treinamento é definição dos casos que mais são relevantes para a classificação de novos casos. Nesse aspecto o *knn* pode apresentar problemas para classificar novos casos, pois se corre o risco de comparar o novo caso a um conjunto de casos conhecido muito grande. O ideal é minimizar o conjunto utilizando menor número de exemplos e mais importantes.

O grau de similaridade é fundamental para esse algoritmo e diversas medidas vêm sendo propostas como medidas de distâncias e correlação. As medidas de proximidades entre pares exemplos (E_i, E_j) devem satisfazer às seguintes propriedades:

a) positividade: $dist(E_i, E_j) \geq 0, \forall(i, j)$

b) identidade: $dist(E_i, E_j) = 0$ se e somente se $E_i = E_j$

c) simetria: $dist(E_i, E_j) = dist(E_i, E_j)$

Além das propriedades anteriores, a medida de similaridade deve respeitar a propriedade de desigualdade triangular:

d) $dist(E_i, E_j) \leq dist(E_i, E_q) + dist(E_q, E_j), \forall(i, j, q)$

Cada atributo da base de treinamento representa uma dimensão de um espaço multidimensional e cada caso conhecido é descrito como um ponto desse espaço $E_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{in})$. *Minkowsky* [ref?] estabeleceu uma maneira genérica de calcular a distância entre pontos no espaço multidimensional R^n de acordo com *d*, que determina a medida utilizada:

$$dist(E_i, E_j) = \left(\sum_{l=1}^{n} |x_{il} - x_{jl}|^d \right)^{\frac{1}{d}}$$

Quando *d=1,* a medida de *Minkowsky* é conhecida como distância de Manhattan e quando *d=2*, ela define a distância Euclidiana. Como na classificação o algoritmo *knn* utiliza um conjunto de *k* vizinhos mais próximos para determinar a classe do caso a classificar, se *k=1* o novo caso é classificado simplesmente como da mesma classe do vizinho mais próximo. Se *k>1*, então são considerados os *k* casos mais próximos para realizar a classificação, e o caso é classificado como da mesma classe majoritária dentre as classes desses *k* vizinhos. A seleção do número *k* influência o resultado, visto que um maior número de vizinhos influenciará a classificação. A Figura 12 exemplifica a aplicação desse algoritmo, para o caso em que se tenha apenas duas classes. A classe do novo caso (círculo) será *triângulo* para *k* até 3, mas se *k* for 5, a classe se torna *quadrado*.

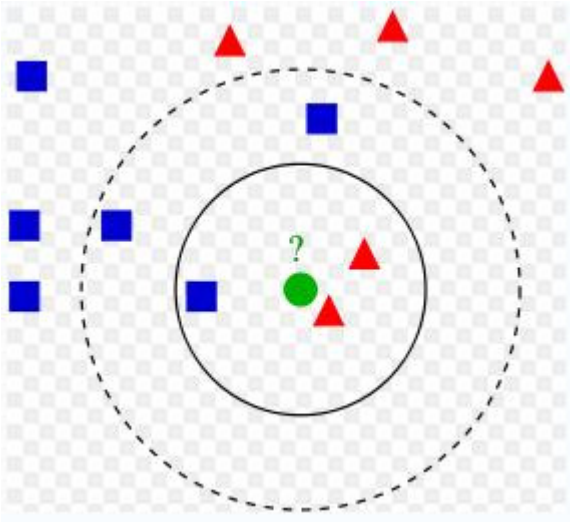


Figura 11 - Exemplo KNN

2.5 Sistemas Multiclassificadores

Sistemas Multiclassificadores são classificadores complexos que combinam diferentes classificadores. São denominados homogêneos se os classificadores combinados são obtidos com o

mesmo algoritmo. Caso contrário, são denominados heterogêneos. Existem diversas técnicas para combinação dos classificadores, dentre elas, as mais usuais são: *bagging* e *boosting*. Também existem diversas abordagens para determinação da classe a partir das atribuições de classes feitas pelos classificadores individuais, dentre elas, a votação majoritária e a ponderação por pesos atribuídos a cada classificador individual.

2.5.1 *Bagging*

Bagging (Breiman, 1996) é um método utilizado para combinar diversos classificadores de um mesmo algoritmo de aprendizado, ou seja, classificadores homogêneos. Esse método explora a instabilidade observada em alguns algoritmos de aprendizado, isto é, classificadores com comportamentos bastante distintos podem ser gerados a partir de pequenas variações do mesmo conjunto de treinamento. Classificadores gerados a partir de diferentes amostras de dados podem captar diferentes regularidades do problema de aprendizado sendo tratado. Neste caso, combinar tais classificadores poderia trazer um ganho na precisão na classificação.

A seqüência de funcionamento desse classificador é:

- a partir da amostra de treinamento é criado aleatoriamente um conjunto de T amostras, (B_1, B_2,..., B_T)
- cada amostra é submetida ao algoritmo de aprendizagem criando um conjunto C com (C_1, C_2,..., C_T) classificadores.
- o classificador final, C_f, é construído através da combinação de todos os classificadores

A combinação dos classificadores para composição de C_f pode ser constituída através do algoritmo de votação majoritária.

O algoritmo de votação majoritária atribui a classe de maior freqüência entre os classificadores, ou seja, o novo caso será classificado de acordo com a classificação majoritária atribuída pelo conjunto de classificadores individuais.

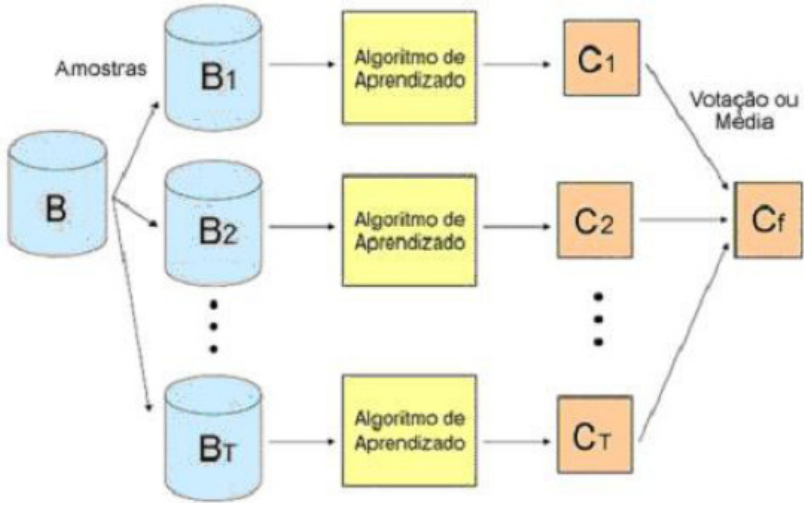


Figura 12 - Ilustração do funcionamento do *Bagging*

2.5.2 *Boosting*

O método *Boosting* (Schapire R. , 2002) é usado para combinar diversos classificadores homogêneos. Em comum com o *Bagging* é que ele também combina classificadores gerados a partir do mesmo algoritmo, usando partições diferentes dos dados de treinamento. No entanto, o *Boosting* é interativo no sentido de que cada amostra de instâncias é selecionada considerando o desempenho dos classificadores construídos anteriormente.

O algoritmo *AdaBoost.M1* (Schapire Y. F., 1996) é utilizado como classificador funcionando da seguinte forma:

- inicialmente é atribuído o mesmo peso a todas as instâncias no conjunto de treinamento (mesma probabilidade na primeira iteração).
- a partir da amostra de dados gerada, um classificador é construído.
- em seguida, o classificador é usado para classificar todas as instâncias de treinamento
- com o resultado dos classificadores o peso das instâncias é modificado onde as instâncias que foram classificadas erroneamente têm o seu peso aumentado, enquanto que as que foram classificadas corretamente têm o seu peso diminuído.

O cálculo do erro do classificador, onde I é o indicador da função que recebe o valor 1 ou 0, dependendo se Ct classificou a i-ésima instância de forma errada ou não, é representado pela fórmula abaixo, onde $\propto_t$ é o peso, c_t é o classificador, x_i uma instância, v_i é uma classe, D_t é a distribuição de probabilidade e Z_t constante de normalização do peso:

$$E_t = \sum_{i=1}^{N} I(c_t(x_i) \neq v_i) D_t(i)$$

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha t}, & se\ c_t(x_i) = v_i \\ e^{\alpha t}, & se\ c_t(x_i) \neq v_i \end{cases} = \frac{D_t(i)\exp\left(-\propto_t v_i c_t(x_i)\right)}{Z_t}$$

Durante a classificação de uma nova instância, cada classificador recebe o peso $\propto_t$:

$$\propto_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

A resposta final será uma combinação linear ponderada das respostas individuais fornecidas pelos classificadores.

$$H(x) = sign\left(\sum_{t=1}^{T} \propto_t c_t(x)\right)$$

2.6 Algoritmo de Associação – Apriori

O algoritmo *Apriori* (Agrawal R. a., 1994) é um algoritmo para mineração de regras de associação em grandes bancos de dados centralizados. Ele encontra todos os conjuntos de itens freqüentes, denominados *itemsets* freqüentes (L_k).

O algoritmo encontra as regras de associação utilizando duas funções básicas, *apriori* para identificar candidatos e eliminar os que não são freqüentes, e a função *rules*, utilizada para extrair as regras de associação usando como entrada o conjunto gerado pela primeira parte do algoritmo.

O primeiro passo do algoritmo é realizar a contagem de ocorrências dos itens para determinar os *itemsets* freqüentes de tamanho unitário. Os k passos seguintes consistem de duas fases, os *itemsets* freqüentes L_{k-1}, encontrados no passo anterior são utilizados para gerar os conjuntos de itens potencialmente freqüentes, os *itemsets* candidatos (C_k). Na seqüência, é realizada uma nova busca contando-se o suporte de cada candidato em C_k.

A geração dos *itemsets* candidatos toma como argumento L_{k-1}, o conjunto de todos *(k-1)-itemsets* freqüentes. Para tal, utiliza-se a função *apriori*, que retorna um conjunto de todos os *k-itemsets* freqüentes. A intuição por trás desse procedimento é que se um *itemset X* tem suporte mínimo, todos os seus subconjuntos também terão (Agrawal R. S., 1996).

O último passo é a descoberta das regras de associação, obtida através da função *rules*. A geração de regras, para qualquer *itemset* freqüente, significa encontrar todos os *subsets* não vazios de *"l"*. Assim, para todo e qualquer *subset "a"*, produz-se uma regra *a → (l - a)* somente se a razão *(suporte (l)/suporte(a))* é igual ou maior que a confiança mínima estabelecida para o problema.

Para gerar regras com maior número de conseqüentes, são considerados todos os *subsets*. Dado um *itemset ABCD*, considera-se primeiro o *subset ABC*, seguido de *AB* e se *ABC*→D não atingir uma confiança suficiente, não é necessário verificar se AB → CD, pois podemos descartá-la visto que a confiança será igual ou menor que a anterior, não atendendo a confiança mínima.

2.7 MDL

Alguns algoritmos de inteligência artificial não aceitam bases de dados com variáveis numéricas. Devido a isso, para que possamos utilizar esse tipo de algoritmo é necessário realizarmos discretização dos valores. A discretização pode ser definida como o processo de transformação de uma variável contínua em uma variável discreta. Em geral, esse método é supervisionado e univariado (Fayyad, 1993). Já o método de discretização baseado na métrica MDL (*Minimum Description Length*) utiliza uma heurística de entropia mínima e não requer nenhum parâmetro. Desta forma a discretização da variável *S* ocorre sem intervenção humana, e associada a uma variável de classe. Dado um conjunto de valores, contendo exemplos rotulados com as classes positiva e negativa, a entropia de S, relativa à classificação booleana onde p_+ é a proporção de exemplos positivos em *S* e p_- é a proporção de exemplos negativos em *S,* é dada por:

$$Entropia(S) = -p_+ log_2 p_+ - p_- log_2 p_-$$

O critério de discretização utilizado baseia-se no Princípio da Descrição de Menor Tamanho – MDLP (*Minimum Description Length Principle*) (Fayyad, 1993). Inicialmente as instâncias da variável *S* são ordenadas. Os pontos médios de cada intervalo formado por valores de *S* que estão relacionados à alternância das classes da variável de classe são considerados candidatos para discretização. O algoritmo de discretização compara se há ganho de informação com a criação desse novo ponto de corte (ponto médio de cada intervalo).

2.8 Métodos de Avaliação

Para compararmos diferentes modelos e algoritmos de classificação são necessárias medidas de comparação. Nesse tópico são abordadas duas técnicas para avaliação de classificadores.

2.8.1 Estrutura de Treinamento e Avaliação

Para a utilização de classificadores em inteligência artificial é necessário definir a estrutura de treinamento e avaliação. Normalmente são utilizadas as abordagens de divisão da base de dados em base de treinamento e base de avaliação, ou a utilização de *Cross-Validation (referência cruzada)* (Stone, 1974).

A primeira abordagem consiste em particionar a base original em duas outras bases: uma utilizada para o aprendizado e denominada base de treinamento, e outra utilizada para avaliação. Essas bases são divididas mantendo a proporção dos casos da variável de classe, ou seja, a proporção original dos estados da variável de classe é mantida tanto na base de avaliação quanto na de treinamento. É comum utilizar para treinamento uma base maior e por isso dividimos a base na proporção de 70% para treinamento e 30% para avaliação.

A outra abordagem de *Cross-Validation* denominada *k-fold-cross-validation* (Haykin, 1999) dividi a base original em *k* bases, *k*>1. O modelo é treinado com *k*-1 bases e avaliado na base restante. Esse procedimento é repetido por *k* vezes, cada rodada utilizando uma base diferente para avaliação. Então o resultado da avaliação do modelo é a média simples de todas as avaliações.

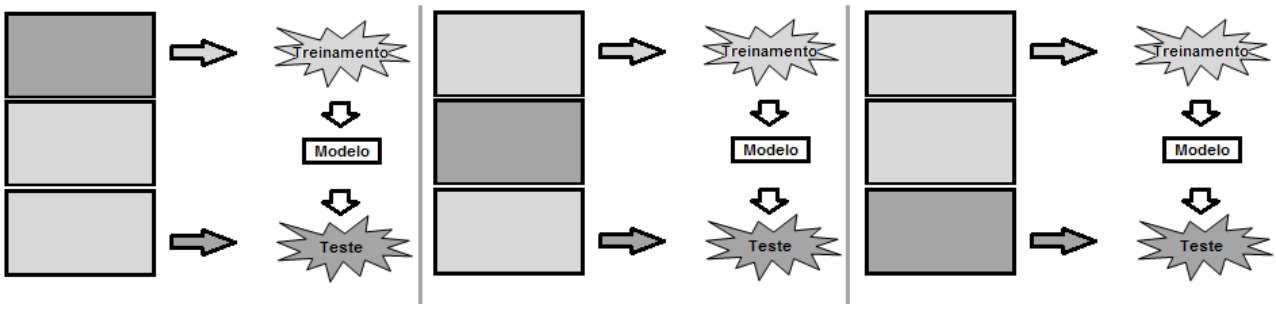


Figura 13 - Procedimento *3-fold Cross-Validation*

2.8.2 F-measure

A avaliação de classificadores é baseada na comparação do desempenho deles com métricas obtidas através da matriz de confusão. A Tabela 4 apresenta um exemplo de matriz de confusão para variável com duas classes apenas.

		Classe Predita	
		positivo	**negativo**
Classe Real	**positivo**	Verdadeiro Positivo (TP)	Falso Negativo (FN)
	negativo	Falso Positivo (FP)	Verdadeiro Negativo (TN)

Tabela 4 - Matriz de Confusão

As métricas de desempenho usuais utilizadas estão apresentadas na Tabela 5.

Sensibilidade (S)	**Especificidade (E)**	**Acurácia (Ac)**
S = TP / (TP+FN)	E = TN / (TN+FP)	Ac = (TP+TN) / (TP+TN+FP+FN)

Tabela 5 - Índices de Desempenho de Classificadores

Um classificador que utilize uma base de dados desbalanceada pode errar todas as classificações de uma classe minoritária e ainda assim obter uma acurácia elevada. A métrica *F-measure* (Rijsbergen, 1979) minimiza esse problema ao utilizar uma média ponderada das métricas de sensibilidade e especificidade. O cálculo do *F-measure* é dado por:

$$F\text{-}measure = \frac{2 \times sensibilidade \times especificidade}{sensibilidade + especificidade}$$

Todas essas métricas associam valores no valor no intervalo [0,1], sendo que quanto o maior seu valor, melhor é o resultado do classificador.

2.8.3 Análise ROC

A análise ROC (*Receiver Operating Characteristics*), também conhecida como análise de curva ROC, é uma técnica para a visualização, organização e seleção de classificadores baseada em seus desempenhos.

As curvas ROC são representadas em gráfico bidimensional com taxa de falso positivo no eixo X e taxa de verdadeiro positivo no eixo Y (Figura 14). Quando uma curva está representada acima de outra, indica que o classificador da primeira curva é melhor que o segundo.

Muitas vezes visualmente não podemos afirmar claramente através do gráfico que um classificador é melhor que o outro, pois as curvas podem estar próximas ou se alternarem na superioridade. Dessa forma podemos, através do cálculo da AUC (área sob a curva ROC), determinar qual classificador tem melhor desempenho. Podemos dizer que se um classificador tiver AUC maior que outro, ele terá desempenho melhor.

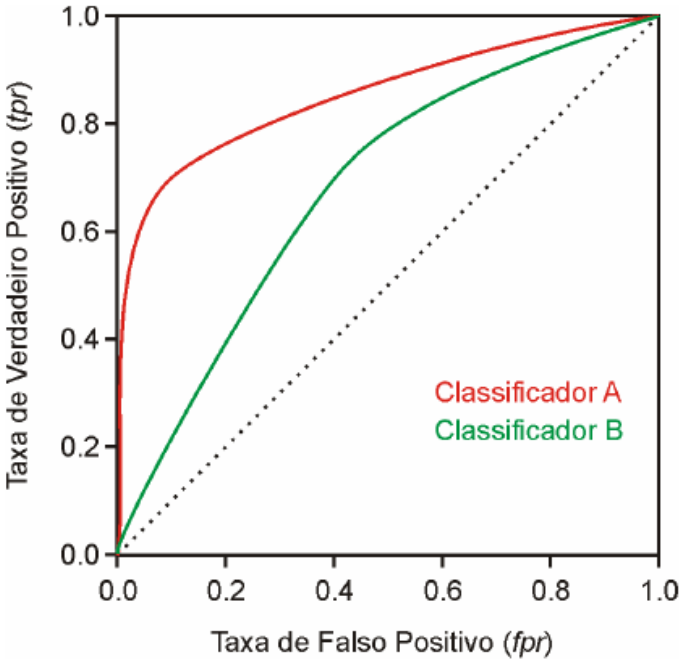


Figura 14 - Exemplo de Curva ROC

2.9 Projeto R

O R é um software estatístico livre, muito utilizado em diversas áreas de pesquisas. Ele tem sido uma boa alternativa a software pagos estabelecidos no mercado, como o *SPSS* ou o *SAS*. Como ele é desenvolvido e mantido pela comunidade global, encontramos um conjunto muito grande de bibliotecas com diversos recursos e técnicas estatísticas implementados.

O R tem sido muito utilizado para a área de mineração de dados, pois disponibiliza um conjunto muito grande de técnicas que vão desde os recursos para pré-processamento até os diferentes tipos de classificadores. Também estão disponíveis bases de dados clássicas para testes e exemplificação com o uso do R. Devido às facilidades que oferece, optamos por utilizar o R como ferramenta para implementação dos algoritmos propostos para identificação de solicitação de COP. Devido a sua popularidade, é possível desenvolver novos recursos e disponibilizá-los rapidamente para a comunidade científica.

3 Solução Proposta

Este capítulo descreve a solução proposta. Ele está dividido da seguinte forma: a Seção 3.1 aborda a contextualização do problema e a Seção 3.2 apresenta a solução proposta.

3.1 Contextualização

Assim como apresentado anteriormente, o seguro SEAF é um seguro agrícola que não é considerado uma fonte de distribuição de renda, seguindo os mesmos princípios de um seguro agrícola normal, e por isso precisa buscar sua própria sustentabilidade através da constituição do fundo de reserva e diminuição dos subsídios governamentais. O seguro também é um instrumento de melhora no processo técnico de plantio com realização de assistência técnica rural. Com esses objetivos em mente, é fundamental a realização de estudos para construir uma base de informações que auxilie a Coordenadoria do SEAF a tornar o seguro SEAF auto-sustentabilidade. Nesse sentido o SEAF criou um sistema com as informações das lavouras assistidas pelo seguro através de laudos de assistência técnica rural. Esses dados foram armazenados, por safras, em uma base de dados que foi disponibilizada para essa pesquisa. São diversas as possibilidades de exploração dessa base mas essa pesquisa foca apenas a aplicação de técnicas de inteligência artificial para a detecção de evidências de emissão de comunicados de ocorrência de perdas (COP) em seguros agrícolas, com base em regras e padrões extraídos da base de laudos de assistência técnica. O foco nesse aspecto é justificado pela necessidade do seguro de prever situações de perda para criação de cálculos atuariais mais específicos e para identificação de problemas no processo agrícola. Outra justificativa é a necessidade de um maior número de avaliações de lavoura. O Ministério do Desenvolvimento Agrário (MDA) não possui recursos suficientes para fiscalizar todos os produtores e devido a isso todas as COP são pagas sem averiguação.

Toda a base de dados do MDA foi disponibilizada para essa pesquisa, correspondendo a 11.743 registros, cada qual com 311 atributos, relativos às safras de agricultura familiar nos anos 2006 a 2010. Tais dados foram obtidos a partir dos laudos técnicos emitidos por técnicos agrícolas para acompanhamento do cultivo em 5% dos segurados. Tais laudos são preenchidos nas fases de pré-plantio, plantio e colheita da safra. Esses segurados foram selecionados por amostragem estatística. Com a aplicação de técnicas estatísticas foram selecionados 19 atributos dentre os 311 atributos coletados originalmente por meio dos laudos técnicos. A partir desses 19 atributos foram construídas regras de associação e classificadores para COP. As regras de associação foram obtidas com o algoritmo "A Priori" e os modelos de classificação foram baseados na construção de classificadores probabilísticos (*Naive Bayes* e árvore de inferência condicional), árvore de decisão (C4.5) e máquina de vetores de suporte (SVM). Os classificadores construídos foram considerados isoladamente e em comitês, constituídos multiclassificadores com as técnicas de votação, *Boosting* e *Bagging*. Foram propostas novas abordagens para construção de multiclassificadores baseados em disjunção, ponderação pelos índices de desempenho e cascata. Todos os classificadores foram avaliados com as métricas de sensibilidade, especificidade, acurácia e F-*measure*, além da análise de curva ROC.

Como os laudos técnicos foram preenchidos por técnicos agrícolas de praticamente todas as regiões do Brasil, com nível de competência bastante variado, os dados coletados apresentaram qualidade variada. A partir dos resultados iniciais obtidos nessa pesquisa que evidenciou os problemas com preenchimento dos laudos técnicos, a Coordenação do SEAF decidiu alterar os laudos e desenvolver pequenos programas para crítica dos dados relativos à safra 2009/2010, mantendo a mesma sistemática de três laudos preenchidos nas fases de pré-plantio, plantio e colheita. Desta forma foi necessário consolidar dados coletados antes e após essas alterações.

A metodologia para mineração de dados seguida foi o modelo de referência CRISP-DM.

3.2 Solução Proposta

A abordagem proposta para facilitar a automatização da detecção de evidências de emissão de COP em seguros agrícolas é a realização de mineração na base de dados para descoberta de regras e padrões, a partir da base de laudos de assistência técnica consolidada, por meio da aplicação de algoritmos de indução para construção de classificadores e regras de associação. Essas tarefas de mineração de dados foram escolhidas por se adaptarem mais às características desse problema, após a realização de testes iniciais que descartaram o uso da tarefa de agrupamento, tendo em vista o grande número de variáveis a considerar, sendo algumas delas categóricas, e o fato da gerência da SEAF está interessada, nesta fase, na predição do valor de apenas uma variável, a ocorrência de COP, variável esta que é categórica. Como todas as ocorrências de COP são pagas sem análise, não é possível realizar qualquer inferência sobre a ocorrência ou não de fraudes que possam ter levado a emissão de COP fraudulentas. Para essa pesquisa foi assumida a hipótese de que todas as COP correspondem a quebras reais na safra e, portanto, são verdadeiras e devidas. Assim, os modelos induzidos a partir dos dados disponíveis visam apenas a descoberta de padrões que evidenciam a possibilidade de emissão ou não de COP, face os dados disponíveis nos laudos técnicos para um dado agricultor que tenha contratado o seguro SEAP.

Inicialmente realizamos estudos estatísticos para as avaliações iniciais da base de dados. Em seguida iniciamos o trabalho de pré-processamento que englobava limpeza de dados, tratamento de *outliers*, consolidação das bases de dados disponíveis, seleção de variáveis e balanceamento da base de dados, indução de modelos e avaliação dos modelos induzidos (Figura 16). Por fim, houve a disponibilização dos resultados obtidos para a Coordenação do SEAF.

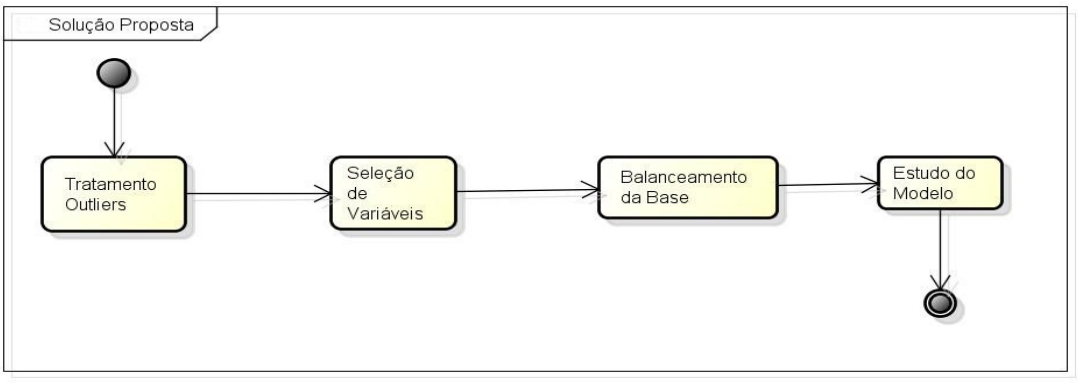


Figura 15 - Fluxo das etapas da solução proposta

3.2.1 Base de Dados

Os dados utilizados nesse estudo são provenientes da assistência técnica realizada pelo SEAF/MDA. Foram utilizados dados das safras 2006-2007, 2007-2008, 2008-2009 e 2009-2010.

Essa base é constituída pelos dados extraídos de três laudos de assistência técnica para acompanhamento amostral da safra agrícola. Essa base possuía originalmente 311 variáveis com duas classes, COP e não-COP, com uma distribuição de 95% não-COP e 5% COP. Além disso, essa base possuía erros graves de preenchimento e muitos dados faltantes devido à falta de preparo técnico dos responsáveis pelo preenchimento no campo. A base total é pequena e possuía 11743 registros onde 697 eram COP.

A base de dados foi fornecida a partir de dois sistemas diferentes, a primeiras safras apresentaram mais erros, pois utilizavam o primeiro sistema que tinha problemas na captação de informação. Em conseqüência desse trabalho foi desenvolvido um novo sistema para captação dos dados com maior qualidade e confiabilidade. O primeiro sistema não possuía um controle de inserção de dados seguro e confiável, permitindo a inclusão de informações com formatos errados e a entrega de laudos incompletos. O novo sistema utiliza metadados para especificação de campos de tabelas de banco de dados, e apresenta um controle mais rígido, com validações individuais e grupais. Por usar metadados, também permite a manipulação dos laudos com maior facilidade, por meio das operações de inclusão, alteração ou remoção de novos campos, sem ser necessário alterar a modelagem do sistema e os laudos já inseridos. A arquitetura do primeiro sistema não permitia essa versatilidade, pois utiliza modelagem entidade-relacionamento estrita, requerendo alterações na base de dados para a inclusão de novos campos.

3.2.2 Pré-processamento

A Figura 17 ilustra as atividades realizadas durante a fase de pré-processamento dos dados da base de dados. Inicialmente foram realizadas as seguintes tarefas: limpeza dos dados, definição de tipos, atribuição de *NA* (*not available*) para os campos da base de dados sem valores. A seguir foi realizada a limpeza dos registros e eliminação de variáveis correlacionadas.

A base de dados original possui 311 variáveis numéricas ou nominais, contendo as informações dos três laudos de assistência técnica rural, pré-plantio (Apêndice B), plantio (Apêndice C), e colheita (Apêndice D).

Observando os laudos foi possível verificar que era possível cadastrar três áreas de produção para cada produtor atendido. Em avaliação junto aos especialistas da área agrônoma e avaliando estatisticamente, verificamos que na prática não era comum o cadastro de mais de uma área de produção no laudo. Dessa forma realizamos nossa primeira grande limpeza na base, a remoção das variáveis que armazenavam informações de outras áreas de produção. Dessa forma reduzimos o número de variáveis para 185. Essas variáveis foram divididas em quatro grupos: variáveis *numéricas* (Tabela 6), variáveis *nominais* (Tabela 7), variáveis de *datas* (Tabela 8) e *outras* variáveis. Os nomes da maioria destas variáveis são autoexplicativo e por isso não serão comentados.

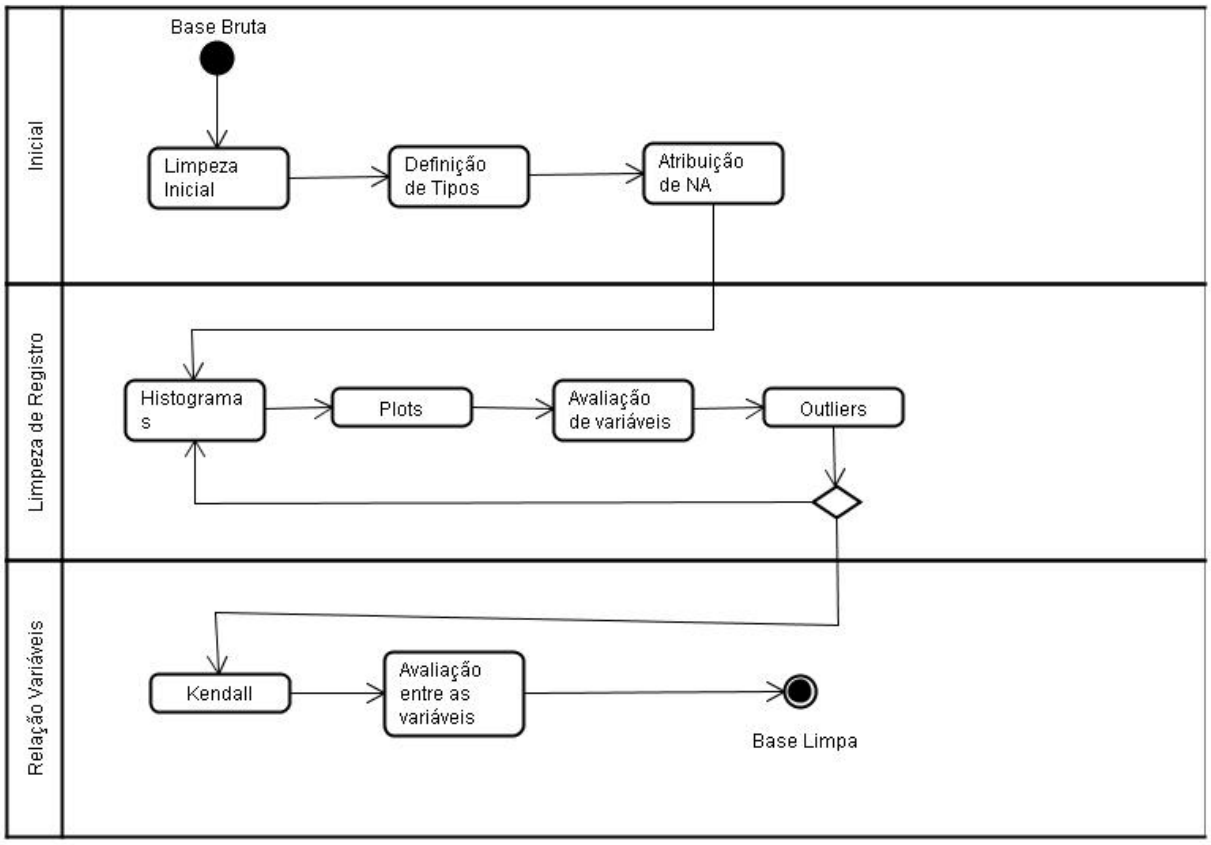


Figura 16 - Etapas do pré-processamento

Nome da Variável	Tipo	Média	Máximo	Mínimo	Unidade
idculturafinanciada	Numérico	5777.5	11146	3	-
valorfinanciado	Numérico	2630.65	65404298	0	R$
valorsegurado	Numérico	2548.23	448500695	0	R$
idformulariopreplantio	Numérico	5607.5	10779	3	-
idpessoafisica	Numérico	54166.5	444894	3	-
idareapreplantio	Numérico	6100	11698	3	-
idpropriedade	Numérico	10390.5	20620	3	-
profundidadesolo	Numérico	1	10000	0	m
declividade	Numérico	10	300	-10	%
altitude	Numérico	535	990990	0	m
ciclofenologico	Numérico	132	14000	0	dias
Stand	Numérico	50000	34000000	0	plantas/ha
produtividadecultivar	Numérico	3600	320000	0	kg/ha
produtividadefinalcultura	Numérico	3500	60000	0	kg/ha
idcontratoseaf	Numérico	5967.5	11311	3	-
idmutuario	Numérico	136830	488271	8	-
idtecnico	Numérico	54166.5	444894	3	-

Nome da Variável	Tipo	Média	Máximo	Mínimo	Unidade
idassistenciatecnica	Numérico	15	32	1	-
idformularioposemergencia	Numérico	4935.5	9453	2	-
idareaposemergencia	Numérico	5338.5	10292	4	-
ciclofenologicocultivarplantada	Numérico	135	250000	0	dias
idtecnico2	Numérico	54200	444894	3	-
idassistenciatecnica2	Numérico	15	32	1	-
idformulariocolheita	Numérico	4484.5	8555	3	-
stand3	Numérico	48000	55556000	0	plantas/ha
produtividadeestimada	Numérico	3000	266667	0	kg/ha

Tabela 6 - Variáveis Numéricas após Primeira Limpeza

Nome da Variável	Tipo	Estados	Nome da Variável	Tipo	Estados
cdculturabacen	Nominal	13	cordoesvegetacaopermanente	Nominal	2
cdgerado	Nominal	7936	faixasbordadura	Nominal	2
foiexcluido	Nominal	1	faixasquebravento	Nominal	2
cdculturaconsorciada	Nominal	27	plantiofaixas	Nominal	2
idusuario	Nominal	1015	rotacaocultura	Nominal	2
idusuarioalterou	Nominal	382	adubacaoverdecultura	Nominal	24
dataultimaalteracao	Nominal	1006	adubacaoverdamassa	Nominal	132
croqui	Nominal	452	preparoprimariosolo	Nominal	7
tamanhoha	Nominal	657	preparosecundariosolo	Nominal	5
tiposolo	Nominal	3	semeadura	Nominal	14
pedregosidade	Nominal	6	adubacao	Nominal	4
coberturasolo	Nominal	3	controleplantasdaninhas	Nominal	10
ultimacultura	Nominal	61	controlepragasdoencas	Nominal	7
presencaerosao	Nominal	4	irrigacao	Nominal	10
analisequimicasolo	Nominal	6	colheita	Nominal	4
phsolo	Nominal	156	adequacaoarea	Nominal	6
macronutrientes	Nominal	1415	codsequencia	Nominal	3
micronutrientes	Nominal	336	comunidade	Nominal	1
existenciapragas	Nominal	57	nomepropriedade	Nominal	4708
impactopragas	Nominal	4	endereco	Nominal	6657
existenciadoencas	Nominal	40	cdagencia	Nominal	1314
impactodoencas	Nominal	5	numerooperacaoenquadrada	Nominal	7489
idade	Nominal	57	anoagricola	Nominal	113
vigor	Nominal	4	cdmunicipioibgeagencia	Nominal	866
estagioproducao	Nominal	5	safra	Nominal	3
nomecultivar	Nominal	2173	cdagentefinanceiro	Nominal	8
tipoagricultura	Nominal	6	sigla	Nominal	8
plantiodireto	Nominal	2	nome	Nominal	8
residuosculturais	Nominal	3	cpfmutuario	Nominal	7670
curvasnivel	Nominal	2	nomemutuario	Nominal	7662
terraceamento	Nominal	2	conselho	Nominal	164

Nome da Variável	Tipo	Estados	Nome da Variável	Tipo	Estados
numeroregistroprofissional	Nominal	1285	acmineralkgha	Nominal	84
cpftecnico	Nominal	1294	acquimicaureia	Nominal	129
nometecnico	Nominal	1292	acquimicanpk	Nominal	274
cnpj	Nominal	17	acquimicanpkkgha	Nominal	206
razaosocial	Nominal	17	codsequencia2	Nominal	3
numerodecredenciamento	Nominal	1	semente	Nominal	7
uf	Nominal	17	emergencia	Nominal	86
cdmunicipioibge	Nominal	1110	b	Nominal	75
nomemunicipio	Nominal	1098	co	Nominal	12
cduf	Nominal	17	cu	Nominal	47
siglauf	Nominal	17	fe	Nominal	9
nomecultura	Nominal	13	mn	Nominal	25
cdgerado2	Nominal	7981	mo	Nominal	13
dataultimaalteracao2	Nominal	266	zn	Nominal	69
croqui2	Nominal	504	conselho2	Nominal	163
areaplantadaha	Nominal	549	numeroregistroprofissional2	Nominal	1288
culturaibgeplantada	Nominal	33	cpftecnico2	Nominal	1294
cultivarrncplantada	Nominal	2321	nometecnico2	Nominal	1292
areaemergenciaplantulas	Nominal	534	cnpj2	Nominal	17
stand2	Nominal	658	razaosocial2	Nominal	17
produtividadeprevista	Nominal	419	numerodecredenciamento2	Nominal	1
tipoevento	Nominal	16	uf2	Nominal	17
cdpragaoudoenca	Nominal	42	cdmunicipioibge2	Nominal	1110
perdaprevista	Nominal	107	nomemunicipio2	Nominal	1098
datainicioevento2	Nominal	254	cduf2	Nominal	17
cop	Nominal	2	siglauf2	Nominal	17
tipoagricultura2	Nominal	5	cdgerado3	Nominal	8014
plantiodireto2	Nominal	2	areacolheitaha	Nominal	540
contrplantasdaninhas	Nominal	10	presencaplantasdaninhas	Nominal	4
contrpragas	Nominal	7	tipoeventoculturaprincipal	Nominal	16
contrdoencas	Nominal	7	perdaestimada	Nominal	238
tipocalagem	Nominal	4	produtorcomunicou perdasagentefina	Nominal	2
prnt	Nominal	49	tecnologia	Nominal	279
calagemkgha	Nominal	172	cddoencaoupraga	Nominal	51
aplantioorganicatipo	Nominal	10	cdmunicipioibge3	Nominal	1110
aplantioorganicakgha	Nominal	134	nomemunicipio3	Nominal	1098
aplantiomineraltipo	Nominal	4	siglauf3	Nominal	17
aplantiomineralkgha	Nominal	116	nomecultura3	Nominal	13
apquimicaureia	Nominal	52	acmineraltipo	Nominal	4
apquimicanpk	Nominal	674	acorganicatipo	Nominal	9
apquimicanpkkgha	Nominal	222	acorganicakgha	Nominal	80

Tabela 7 - Variáveis Nominais após Primeira Limpeza

Nome da Variável	Tipo
datadevisita	Data
datavalidacaoinsercao	Data
datadeinsercao	Data
dtinsercao	Data
datadevisita2	Data
datavalidacaoinsercao2	Data
datadeinsercao2	Data
datainicioplantio	Data
datafimplantio	Data
dataprevistacolheita	Data
datadevisita3	Data
datadeinsercao3	Data
datainiciocolheita	Data
datainicioevento3	Data
datacop	Data

Tabela 8 - Variáveis de Data após Primeira Limpeza

As variáveis do tipo *outras* contemplam as chaves como *idculturafinanciada*, valores únicos como *CPF* e *CNPJ* e campos de textos abertos como *tecnologia* e *nometecnico2*. Como com os modelos que se deseja construir devem capturar padrões e regras existentes nos dados, foram retiradas as variáveis de *datas* e *outras*. Ainda com foco na generalidade do modelo, as variáveis numéricas e nominais foram avaliadas. Variáveis relacionadas à localização como *uf* e *município* e variáveis relacionadas ao tempo como safra foram removidas para evitar que o modelo gerado se relacione com locais e datas. Algumas variáveis eram colhidas em laudos diferentes e por isso essa duplicidade foi removida. Algumas informações nominais apareciam representadas em mais de uma variável de forma diferente como *cdculturabacen* e *nomecultura* (informação redundante). Ambas traziam a mesma informação e por isso essas variáveis foram removidas e mantidas apenas uma delas. Essas operações constituíram a segunda etapa de limpeza.

Após a limpeza das variáveis partimos para a limpeza dos registros. Primeiramente foi realizada uma limpeza inicial que removeu registros que possuíam erros críticos de preenchimento ou não possuíam classificação como no caso do campo *cop* conter o valor *NULL*. As variáveis numéricas tiveram o intervalo de valores corretos definidos e as variáveis nominais tiveram a lista de estados permitidos definidas.

A seguir foram utilizadas funções estatísticas, disponibilizadas pela linguagem R, para emissão de *plots* e histogramas para as 185 variáveis então consideradas. Esses histogramas e gráficos permitem realizar uma verificação visual dos valores assumidos por cada uma dessas variáveis e verificar se, visualmente, há alguma anômalo. Caso seja localizado um valor não esperado, a corretude do mesmo deve ser validada junto a um especialista do domínio. A variável que possuía esse valor foi considerada como "variável problemática". A Figura 17 apresenta um gráfico (*plot*) com valores anômalos (identificados por um círculo) para as variáveis *stand* e *altitude*. A variável *stand* mensura a quantidade de plantas por hectare da cultura principal. No caso dessa pesquisa, os especialistas de domínio consultados foram os técnicos agrônomos do MDA.

Com a identificação das variáveis problemáticas iniciou-se o trabalho de identificação se os valores anômalos são *outliers* ou não. Um *outlier* é um ponto extremo que deve ser descartado antes do processo de indução do modelo a partir da base de dados disponível. Para identificação de *outliers* foram utilizadas as funções *boxplot* e *zcore* disponibilizadas na linguagem R. Ao final dos testes de *outliers* os registros com problemas eram encaminhados para especialistas técnicos do MDA, para avaliação e conseqüente autorização, se for o caso, de remoção do registro que contenha essa variável. Após a remoção dos registros com erro, a variável era submetida novamente ao teste de *outliers*. Esse procedimento se repetia até os testes utilizados não mais indicarem a presença de *outliers*. Como estávamos trabalhando com uma base de dados pequena no número de registros, a orientação era evitar remover registros. Registros que poderiam ser reavaliados e corrigidos eram enviados para avaliação aos técnicos agrônomos, e variáveis que provocavam um grande número de remoções de registros eram removidas após consulta aos especialistas. Por exemplo, *areacolheitaha*, *tamanhoha* e *areaemergenciaplantula* foram removidas após consulta aos especialistas. Por outro lado a variável *altitude* também apresentou muitos erros de preenchimento que poderiam provocar a retirada de muitos registros. Outra abordagem seria manter tais registros, mas retirar a variável *altitude* da base de dados. No entanto, os especialistas consideraram não recomendável a remoção dessa variável porque ela está diretamente ligada à quebra da safra em algumas culturas e, portanto, à possibilidade de emitir COP. Por exemplo, De forma geral, as características de crescimento de cultivares de mamoneira ficam comprometidas quando as plantas são cultivas em baixa altitude. Para minimizar a perda de registro os técnicos agrícolas do MDA foram consultados para correção de valores de *latitude*. A Figura 18 apresenta o histograma e *plot* de *altitude* após a eliminação dos valores considerados como *outliers* pelos especialistas. Esses valores podem corresponder a erros de digitação dos técnicos que preencheram os laudos ou simplesmente a valores lançados sem a vírgula decimal.

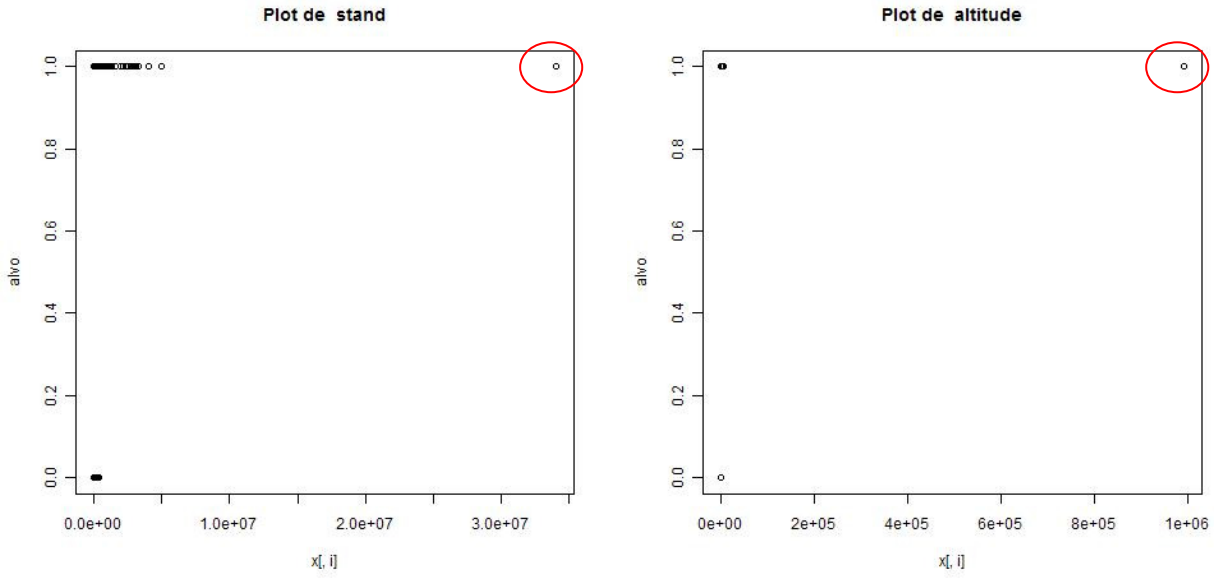


Figura 17 - Plot que apresenta erro das variáveis *stand* e *altitude*

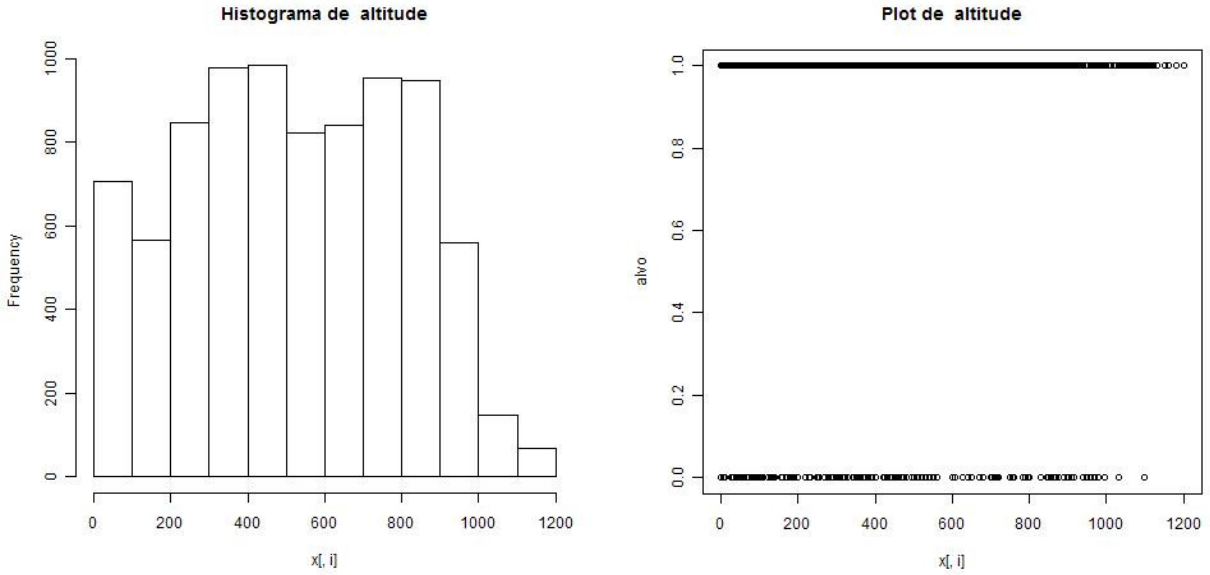


Figura 18 – Variável *altitude* após eliminção de *outliers*

As variáveis *stand*, *stand2* e *stand3* representam a evolução do *stand* (quantidade de plantas por hectare), ao longo dos laudos de pré-plantio, plantio e colheita. Para reduzir o número de variáveis, após conversa com os especialistas do MDA, se optou por manter-se o *stand3* e criar uma nova variável, *difstand*, representando a diferença do maior *stand* para o menor *stand*. O objetivo disso era manter o *stand* observado durante a colheita e descrever o crescimento ou decrescimento do *stand* nas fases anteriores. As variáveis *b, co, cu, fe, mn, mo* e *zn* representam adubação em kg/ha, com os micronutrientes boro, cobalto, cobre, ferro, manganês, molibdênio e zinco e foram substituídas pela variável *adubação* com valor 0 se não houve adubação com micronutrientes, ou 1 se houve. As variáveis *existenciapragas, impactopragas, existenciadoencas* e *impactodoencas* foram substituídas pela variável *impacto* com valor 1 se houve impacto de doenças ou pragas, e 0 caso contrário. Códigos de praga e doença foram removidos por serem ligados à cultura específica.

Durante o processo de avaliação das variáveis verificamos que havia poucas variáveis que tratavam de informações climáticas. Os especialistas informaram que algumas culturas podem apresentar problemas se houver certos eventos climáticos em fases específicas do cultivo. Com a informação de ciclofenológico, cultura e datas de plantio seria possível cruzar com os eventos climáticos descritos na base de informações do **Agritempo** e criar variáveis que permitiriam definir ocorrências críticas em cultivos específicos. Na tentativa de criação dessa ligação verificou-se ser necessário uma definição exata de posição da propriedade mas essa informação não era disponível. Então propusemos a inserção de dados de GPS do local da lavoura. Essa variável entrou de forma optativa na safra 2010-2011, para conhecimento geral, sendo obrigatório seu preenchimento na safra 2011-2012, o que vai permitir mapear clima e eventos climáticos em cada lavoura.

3.2.3 Seleção de variáveis

Após a limpeza da base, o número de variáveis ainda era grande, o que dificultaria a análise e utilização dos modelos que fossem criados. Então foi realizado um estudo de correlação entre as

variáveis visando remoção das com alta correlação, com o método de Kendall, disponibilizado na linguagem R, cuja saída é uma matriz que apresenta os graus de correlação indicados na Tabela 9:

Símbolo	Relação (%)
“ “	0 a 30%
“.”	30% a 60%
“,”	60% a 80%
“+”	80% a 90%
“*”	90% a 95%
“B”	95% a 100%

Tabela 9 - Graus de Correlação Kendall

A matriz de correlações obtida com a função Kendall está apresentada no Apêndice E. A remoção das variáveis com correlações a partir de 80% foi validada com os especialistas do MDA. Com o processo de limpeza concluído e a aplicação do método Kendall, houve redução do número de variáveis de 311 para as 65 apresentadas nas Tabelas 10 e 11.

Variável	Tipo	Media	Max	Mínimo	Unidade
valorfinanciado	Numérico	2.600	10000	1000	R$
valorsegurado	Numérico	2600	10000	1000	R$
profundidadesolo	Numérico	1	100	0	m
declividade	Numérico	10	150	-10	%
perdaprevista	Numérico	0	100	0	%
phsolo	Numérico	1,8	7	0	
stand3	Numérico	45580	70000	2857	plantas/ha
Difstand	Numérico	5088	50000	0	plantas/ha
prnt	Numérico	56	100	0	%
calagemkgha	Numérico	1852	26000	0	kg/ha
aplantiomineralkgha	Numérico	3746	40000	0	kg/ha
apquimicaureia	Numérico	26	2000	0	kg/ha
apquimicanpkkgha	Numérico	212	5000	0	kg/ha
acorganicakgha	Numérico	870	9500	0	kg/ha
altitude	Numérico	550	1800	0	m
produtividadecultivar	Numérico	3600	14000	0	kg/ha
adubacaoverdamassa	Numérico	0	14000	0	kg/ha
produtividadefinalcultura	Numérico	3600	14000	0	kg/ha
areaplantadaha	Numérico	3.54	150	0.5	ha
ciclofenologicocultivarplantada	Numérico	131	965	0	dias
produtividadeprevista	Numérico	3500	14663	0	kg/ha
aplantioorganicakgha	Numérico	0	14500	0	kg/ha

Tabela 10 - Relação de Variáveis Numéricas após Limpeza e Kendall

Nome da Variável	Tipo	Estados	Nome da Variável	Tipo	Estados
cdculturabacen	Nominal	11	preparosecundariosolo	Nominal	5
cdculturaconsorciada	Nominal	19	semeadura	Nominal	14
tiposolo	Nominal	3	adubacao	Nominal	4
pedregosidade	Nominal	6	controleplantasdaninhas	Nominal	10
coberturasolo	Nominal	3	controlepragasdoencas	Nominal	7
ultimacultura	Nominal	54	irrigacao	Nominal	10
presencaerosao	Nominal	4	colheita	Nominal	4
analisequimicasolo	Nominal	7	adequacaoarea	Nominal	7
vigor	Nominal	5	tipoevento	Nominal	16
estagioproducao	Nominal	6	cdpragaoudoenca	Nominal	38
tipoagricultura	Nominal	6	contrplantasdaninhas	Nominal	10
plantiodireto	Nominal	2	contrpragas	Nominal	7
residuosculturais	Nominal	3	contrdoencas	Nominal	7
curvasnivel	Nominal	2	tipocalagem	Nominal	4
terraceamento	Nominal	2	aplantioorganicatipo	Nominal	10
cordoesvegetacaopermanente	Nominal	2	aplantiomineraltipo	Nominal	4
faixasbordadura	Nominal	2	acorganicatipo	Nominal	8
faixasquebravento	Nominal	2	acmineraltipo	Nominal	4
plantiofaixas	Nominal	2	COP	Nominal	2
rotacaocultura	Nominal	2	minerios	Nominal	2
adubacaoverdecultura	Nominal	19	impacto	Nominal	2
preparoprimariosolo	Nominal	7			

Tabela 11 - Relação de Variáveis Nominais após Limpeza e Kendall

Com todas as limpezas e reduções o conjunto de 65 variáveis para criação do modelo ainda era grande e foi utilizada a técnica de árvore de decisão para identificar variáveis irrelevantes com relação à variável de interesse, COP. O processo se resumia em aplicar o algoritmo de construção de árvore de decisão C4.5 aos dados da base de dados e observar quais variáveis que apareciam na árvore criada. As variáveis que não se encontravam na árvore gerada eram eliminadas. Novamente era gerada uma nova árvore e isso se repetia até as variáveis permanecerem a mesma após duas execuções consecutivas. Com tal técnica foi possível reduzir a lista de variáveis para 19 (Tabelas 12 e 13) e 9611 registros na base de dados, sendo 457 (4,7%) de ocorrência de COP.

Nome da Variável	Tipo	Média	Máximo	Mínimo	Unidade
Altitude	Numérico	550	1800	0	M
Produtividadecultivar	Numérico	3600	14000	0	kg/há
Adubacaoverdamassa	Numérico	0	14000	0	kg/há
produtividadefinalcultura	Numérico	3600	14000	0	kg/há
Areaplantadaha	Numérico	3.54	150	0.5	Há
ciclofenologicocultivarplantada	Numérico	131	965	0	Dias
Produtividadeprevista	Numérico	3500	14663	0	kg/há
Aplantioorganicakgha	Numérico	0	14500	0	kg/há

Tabela 12 - Variáveis Numéricas para Indução de Modelos

Nome da Variável	Tipo	Número de Estados
Cdculturabacen	Nominal	19
Tipoagricultura	Nominal	6
adubacaoverdecultura	Nominal	24
Semeadura	Nominal	6
Adubação	Nominal	4
controleplantasdaninhas	Nominal	5
Tipoevento	Nominal	15
Contrpragas	Nominal	3
aplantiomineraltipo	Nominal	4
tipoeventoculturaprincipal	Nominal	16
COP	Nominal	2

Tabela 13 - Variáveis Nominais para Indução de Modelos

A distribuição dessas variáveis pelos laudos está ilustrada na Tabela 14, sendo seis do laudo de pré-plantio, dez do laudo de plantio, e três do laudo de colheita.

Nome	Laudo	Tipo	Nome	Laudo	Tipo
cdculturabacen	L1	Nominal	produtividadefinalcultura	L3	Numérica
altitude	L2	Numérica	areaplantadaha	L2	Numérica
produtividadecultivar	L2	Numérica	ciclofenologicocultivarplantada	L2	Numérica
tipoagricultura	L1	Nominal	produtividadeprevista	L2	Numérica
adubacaoverdecultura	L1	Nominal	tipoevento	L2	Nominal
adubacaoverdamassa	L2	Numérica	contrpragas	L2	Nominal
semeadura	L1	Nominal	aplantioorganicakgha	L2	Numérica
adubacao	L1	Nominal	aplantiomineraltipo	L2	Numérica
controleplantasdaninhas	L1	Nominal	tipoeventoculturaprincipal	L3	Nominal
COP	L3	Alvo			

Tabela 14 - Distribuição das Variáveis por Laudo

3.2.4 Balanceamento

A ocorrência de apenas 4,7% de COP nos obtidos após as etapas anteriores de pré-processamento caracteriza uma base desbalanceada. Para prover um balanceamento de classes foi utilizada de *over-sampling*, onde aumentamos o número de casos da classe minoritária. A escolha dessa técnica se deveu ao fato de não se desejar perder mais informação com a aplicação da técnica de *under-sampling* ou mista, visto que a base de dados já apresentava um tamanho reduzido.

Primeiramente realizamos a divisão da base em duas outras bases, avaliação e treinamento, utilizando os valores usuais na literatura de 30% dos registros para avaliação e os restantes 70% dos registros para a base de treinamento. A seleção dos registros foi feita com amostragem aleatória estratificada de maneira a se manteve as mesmas proporções de COP e não COP em ambas as bases. O balanceamento de classe somente foi feito para a base de treinamento. Foram testados diversos níveis de balanceamento com o aumento gradativo da classe minoritária e comparação do

desempenho dos classificadores Naive Bayes, C4.5 e CTREE (árvore de inferência condicional) gerados via a métrica *F-measure* (Tabela 15)

Base	Naive Bayes	C4.5	CTREE	F-*measure* COP (%)
Sem balanceamento	0.57	0.37	0.28	5,00
2x	0.7240	0.5260	0.50	10,00
5x	0.78	0.70	0.79	23,00
8x	0.80	0.79	0.79	31,00
12x	0.82	0.81	0.79	40,00

Tabela 15 - Balanceamento de Classes da Base de Treinamento

Após este estudo selecionamos balanceamento de *over-sampling* com taxa de 8x e 12x. Bases com balanceamentos maiores apresentaram um diminuição no *F-measure* e não foram utilizadas. A partir dessa avaliação então foram criados duas estruturas de bases para criação dos modelos: uma base integral balanceada para ser utilizada em modelos com *cross-validation* e outra base dividida em avaliação e treinamento, com balanceamento apenas da base de treinamento.

3.2.5 Modelagem

Nessa fase focamos a indução de modelos, com a base de treinamento obtida na fase anterior, para inferir se há evidências de que pode ocorrer COP, a partir dos dados dos laudos obtidos pelo MDA para as safras de agricultura familiar dos anos de 2006 a 2010, considerando as 19 variáveis relacionadas na Tabela 14. O Capítulo 4 discute os resultados obtidos.

Inicialmente executamos o algoritmo *Apriori* para identificação de regras de associação. Para a execução do algoritmo foi necessário realizar a discretização das variáveis numéricas *altitude, produtividadecultivar, adubacaoverdamassa, produtividadefinalcultura, areaplantadaha, produtividadeprevista, aplantioorganicakgha e ciclofenologicocultivarplantada*. A discretização foi baseana na aplicação do princípio MDL. Durante esse processo, as variáveis *adubacaoverdamassa* e *areaplantadaha* foram removidas da base porque a discretizaçao resultou em apenas um único estado. Os estados obtidos para as demais variáveis estão apresentados na Tabela 16.

A execução do *Apriori* foi realizada com a base total não balanceada. Como a COP era minoritária na base, executamos o algoritmo com suporte baixo. Os suportes utilizados foram de 5%, 3% e 2%. Todas as execuções utilizaram confiança de 90%.

Após a execução do *Apriori* partimos para outra abordagem, a criação de classificadores. Utilizamos inicialmente os classificadores tradicionais: os algoritmos C4.5, *Ctree* (árvore de inferência condicional), *Naive Bayes*, *Bagging*, *AdaBoost.M1*, *SVM* e KNN. Eles foram treinados com a base de treinamento balanceada e avaliados com a base de avaliação. Para os classificadores tradicionais também utilizamos *cross-validation* com 10 partições para realizar avaliações. Nesse caso foi utilizada apenas a métrica de F-*measure* para avaliação. No caso da abordagem base de treinamento e base de avaliação foram calculadas a curva ROC e AUC.

Variável	Estado	Corte
Altitude	1	<= 444.42
	2	>444.42
produtividadecultivar	1	<= 4550
	2	> 4550
produtividadefinalcultura	1	<= 4523
	2	> 4523
ciclofenologicocultivarplantada	1	<= 116.5
	2	> 160.5
	3	> 116.5 e <=124.5
	4	>124.5 e <=136.5
	5	>136.5 e <=140.5
	6	> 140.5 e <= 160.5
produtividadeprevista	1	<= 157.5
	2	>157.5 e <= 805.0
	3	>805.0 e <= 1310.0
	4	>1310.0 e <=4523.0
	5	>4523.0
aplantioorganicakgha	1	<=3100
	2	>3100

Tabela 16 - Tabela de discretização

A seguir foi explorada a abordagem de multiclassificação, iniciando com multiclassificação por votação (Figura 19). Essa abordagem utiliza vários classificadores e propõe que o estado mais classificado seja considerado o recomendado pelo multiclassificador. Foram utilizados seis algoritmos e atribuída COP se pelo menos três modelos apontarem isso.

Algoritmo Classificador Votação

```
votação(caso) {Base de dados, 0 – COP , 1 – não-COP}
        n ← total de classificadores
        cop ← 0
        ncop ← 0
        para i = 1 até n faça
                se classificar[i](caso) = verdadeiro então
                        cop ← cop + 1
                caso contrário
                        ncop ← ncop + 1
                fim se
        fim para
        se cop >= ncop então
                retornar 0
        caso constrário
                retornar 1
        fim se
```

Figura 19 - Pseudocódigo votação

Essa abordagem, já utilizada em problemas de classificação, nos estimulou na criação de outras abordagens de multiclassificação para melhora dos resultados. Propomos três novas abordagens de multiclassificação:

- multiclassificador em cascata homogêneo
- multiclassificador disjunto heterogêneo
- multiclassificador ponderado heterogêneo

O multiclassificador em cascata homogêneo surge das características do domínio do problema analisado, onde a base de dados é composta por três laudos que são preenchidos em momentos distintos e consecutivos. Essa solução parecia interessante pelo aspecto de podermos ter uma classificação intermediária, mesmo sem termos todas as variáveis disponíveis para uma classificação final. Isso seria muito interessante ao seguro, pois poderíamos ter um controle evolutivo e até um provisionamento em relação a número de COP esperado.

O multiclassificador em cascata homogêneo é dito homogêneo porque, em todas as fases de classificação, utiliza o mesmo algoritmo de indução. No primeiro momento é induzido um classificador com as variáveis obtidas do laudo 1, no segundo momento, com as variáveis obtidas do laudo 2 e, por fim, no terceiro momento, com as variáveis obtidas do laudo 3. A classificação sugerida pelo primeiro classificador e as variáveis do segundo laudo eram entradas para o segundo classificador. A classificação sugerida por esse segundo classificador e as variáveis do laudo 3 eram entradas para o terceiro classificador, cuja resposta era considerada como a resposta do multiclassificador por cascata (Figura 20). Os algoritmos usados foram C4.5, *Ctree*, *Naive Bayes*, *Bagging* e *AdaBoost.M1, SVM* e KNN. O pseudo-código desse multiclassificador está apresentado na Figura 21.

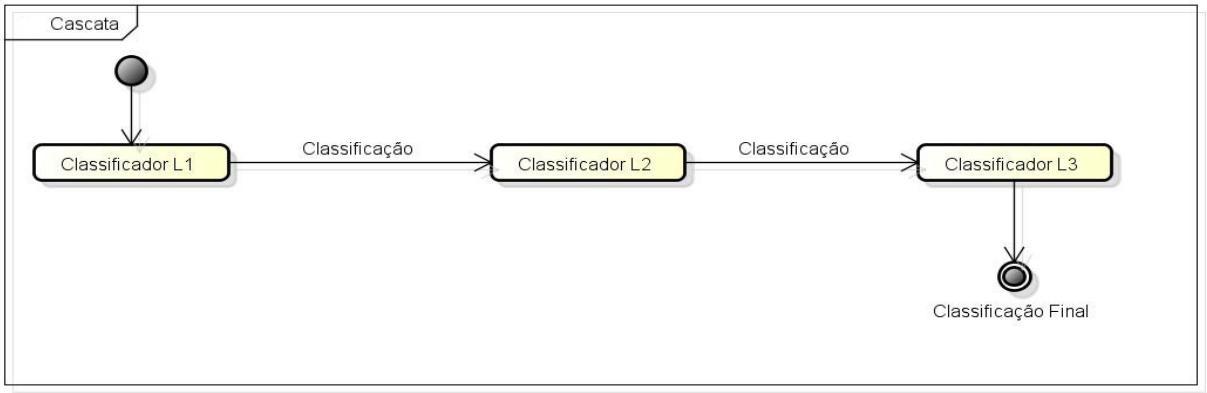


Figura 20- Diagrama do modelo em cascata

Algoritmo Classificador em Cascata
cascata(base, var1[], var2[], var3[]) {Base de dados e lista de variáveis} classificacao1[] ← classificar1(base, var1[]) classificacao2[] ← classificar2(base, combinar(var2[],classificao1)) classificacao3[] ← classificar3(base, combinar(var3[],classificao2)) retornar classificacao3[]

Figura 21 - Pseudocódigo Multiclassificador em Cascata

Como a classe de interessa era COP, propomos um algoritmo que privilegiasse a identificação de COP. Para isso combinamos os algoritmos de classificação C4.5, *Ctree*, *Naive Bayes*, *Bagging*, *AdaBoost* e SVM para compor o multiclassificador disjunto heterogêneo. Cada classificador classificava isoladamente e, se pelo menos um deles classificasse o registro como COP, então o multiclassificador classificaria o registro como COP. Como cada algoritmo tem características específicas, assumimos a premissa de que eles pudessem explorar subespaços diversos do espaço de COP, para tentar classificar o máximo de COP possível. A Figura 22 ilustra a arquitetura desse multiclassificador. A Figura 23 apresenta o seu pseudo-código.

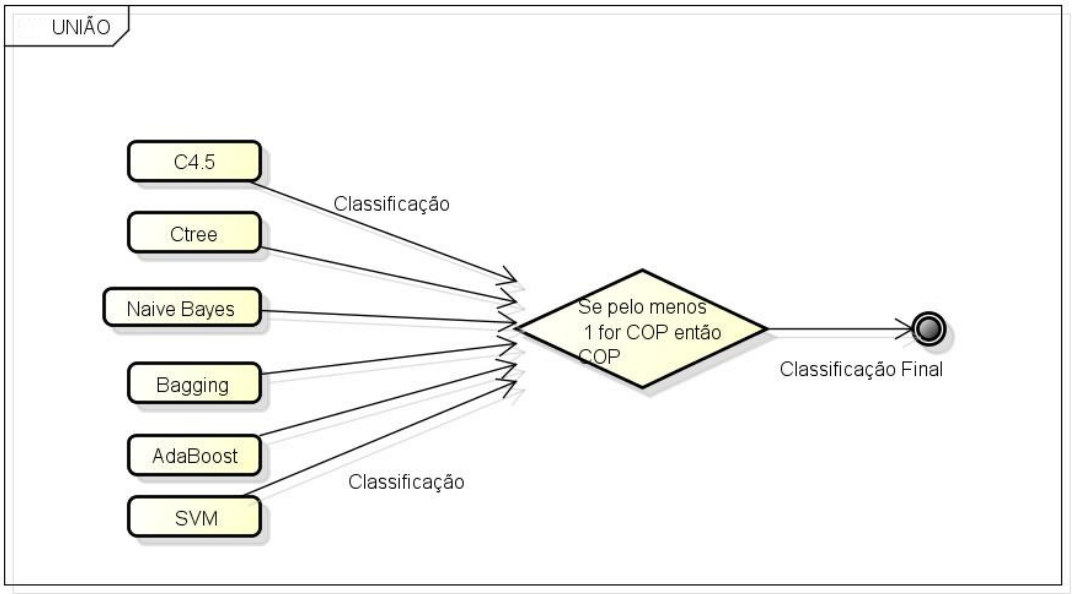


Figura 22 - Diagrama do modelo disjunção heterogênea

Algoritmo Classificador Disjuntivo

```
disjuntivo(caso) {Base de dados, COP – 0 , não-COP –1}
        n ← total de classificadores
        cop ← 1
        para i = 1 até n faça
                se classificar[i](caso) = 0 então
                        cop ← 0
                fim se
        fim para
retornar cop
```

Figura 23 - Pseudocódigo disjuntivo

Por fim, foi desenvolvido o modelo ponderado heterogêneo que faz a união dos diferentes algoritmos através de uma fórmula ponderada. Cada classificador tem um peso definido pelo seu grau de acerto. Nesse caso assuminos a heurística que quanto maior o grau de acerto de um classificador, mais importante será a sua classificação. Se o classificador indicar COP então será usado a sensibilidade como peso caso contrário será usado a especificidade. As equações a seguir (Figura 24) definem o algoritmo utilizado para esse multiclassificador.

Algoritmo Classificador Ponderado Heterogêneo
$sensibilidade = \frac{TP}{TP + FN}$
$especificidade = \frac{TN}{TN + FP}$
$COP(x) = \begin{Bmatrix} 1, se\ x = 0 \\ 0, se\ x = 1 \end{Bmatrix}$
$SOMACOP = \frac{\sum_{i=1}^{n} sensibilidade_i \times COP(classificacao_i)}{\sum_{i=1}^{n} sensibilidade_i}$
$SOMANAOCOP = \frac{\sum_{i=1}^{n} especificidade_i \times classificacao_i}{\sum_{i=1}^{n} especificidade_i}$
$Se\ SOMANAOCOP > SOMACOP, retorna\ 0\ senão\ retorna\ 1$

Figura 24 - Pseudocódigo Ponderado Heterogêneo

4 Análise dos Resultados Obtidos

Este capítulo apresenta os resultados obtidos com a aplicação dos modelos classificadores, multiclassificadores e regras de associação propostos para inferirem se há evidências de que pode ocorrer COP. A Seção 4.1 apresentada a metodologia utilizada para a avaliação desses modelos. A Seção 4.2 apresenta a análise dos resultados e a Seção 4.3 apresenta as considerações finais.

4.1 Metodologia de Avaliação

Os modelos criados foram obtidos através da aplicação de algoritmos de indução onde os classificadores e multiclassificadores foram gerados com base de treinamento contendo 70% das instâncias da base de dados e as regras de associação, com toda a base de dados. Essa base de treinamento foi submetida a técnica de *oversampling* para balanceamento de classes, gerando quatro amostras balanceadas, além da original, não balanceada. Os modelos obtidos foram avaliados por meio do desempenho deles na classificação das instâncias da base de avaliação, contendo os 30% de instâncias restantes da base de dados, sendo avaliados com as métricas de sensibilidade, especificidade, acurácia e *F-measure* (Seção 2.7). Em paralelo, também foram gerados modelos a partir da base de dados não fragmentada, após a mesma ter sido balanceada com a técnica de *over-sampling* e avaliados com análise ROC e métrica AUC (Seção 2.7.3). Os classificadores ou multiclassificadores que obtiveram maior índice *F-measure* ou AUC foram considerados os de melhor desempenho. Apenas os resultados da análise ROC para amostra balanceada cujos multiclassificadores apresentaram melhor desempenho serão apresentados.

4.2 Resultados e Análise das Avaliações

A utilização do algoritmo *Apriori* para identificação de regras de associação parecia interessante, pois poderia apresentar relações entre as variáveis e conseqüentemente alguma associação com a COP. Poderíamos através dos resultados identificar regras de prováveis falhas técnicas do cultivo. Isso não ocorreu na prática. A representação da COP era muito pequena na base de dados e por isso foi necessário rodar o algoritmo com suporte baixo. Com o suporte baixo o número de regras apresentando foi extremamente grande (Tabela 17), mas nenhuma delas estava associada a ocorrência da emissão de COP (COP=0).

Confiança	Suporte	Número de Regras	Regras com COP
90%	5%	515.405	0
90%	3%	1.251.062	0
90%	1%	Estouro de Memória	Não foi possível calcular
80%	3%	9.875.191	Estouro de Memória

Tabela 17 - Regras de Associação obtidas com o *Apriori*

Na implementação dos classificadores foi fundamental o balanceamento. Através da métrica do *F-measure* foi possível verificar a melhora de performance dos classificadores com o balanceamento de classes (Tabela 18)

Base	*Naive Bayes*	C4.5	CTREE	*Bagging*	*AdaBoost.M1*	SVM	KNN
Desbalanceada	0.572	0.371	0.284	0.592	0.54	0.553	0.594
Balanceada	0.826	0.816	0.794	0.645	0.815	0.617	0,671

Tabela 18 - Impacto do balanceamento sobre F-*measure* de Classificadores

Os classificadores individuais foram treinados com a base de treinamento balanceada e avaliados na base de avaliação (Tabela 19) com 2921 registros, correspondentes a 30% do número total de 9611 registros. Note que emissão de COP corresponde a COP=0.

	NB		**C4.5**		**Ctree**		**Bagging**		**AdaBoost**		**SVM**		**KNN**	
Real	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
0	101	30	96	35	82	49	72	59	102	29	59	72	67	64
1	390	2400	315	2475	229	2561	47	2743	438	2352	45	2745	58	2732

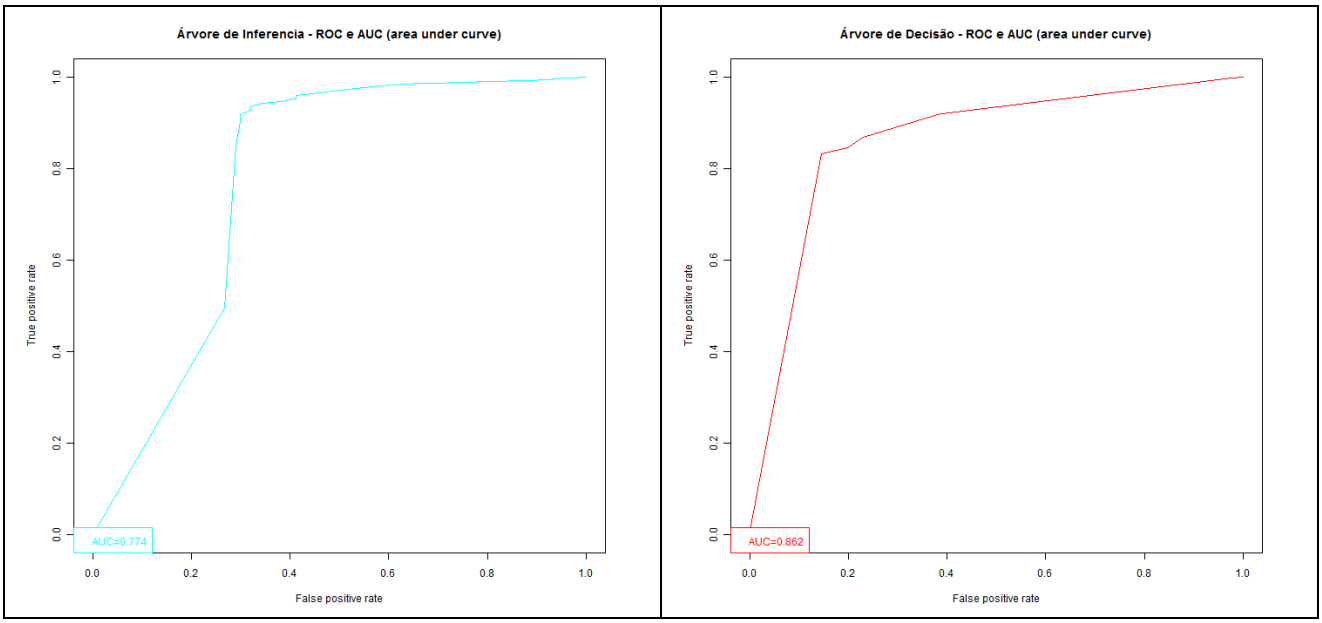
Tabela 19- Matriz de Confusão para Classificadores Individuais

Eles também foram treinados com uma base não particionada balanceada utilizando 10-*fold cross-validation*. O índice F-*measure* médio obtidos está apresentado na Tabela 20. Comparando as Tabelas 18 e 20, concluímos que há evidência experimental de que não houve *over-fitting* na abordagem base de treinamento e base de avaliação para gerar os classificadores individuais.

Naive Bayes	**C4.5**	**Ctree**	**Bagging**	**AdaBoost**	**SVM**	**KNN**
0.820	0.822	0.823	0.745	0.804	0.691	0.732

Tabela 20 - F-*measure* de Classificadores Individuais (*cross-validation*)

As curvas ROC e o desempenho médio medido por meio da AUC para os classificadores individuais estão apresentados na Figura 25.



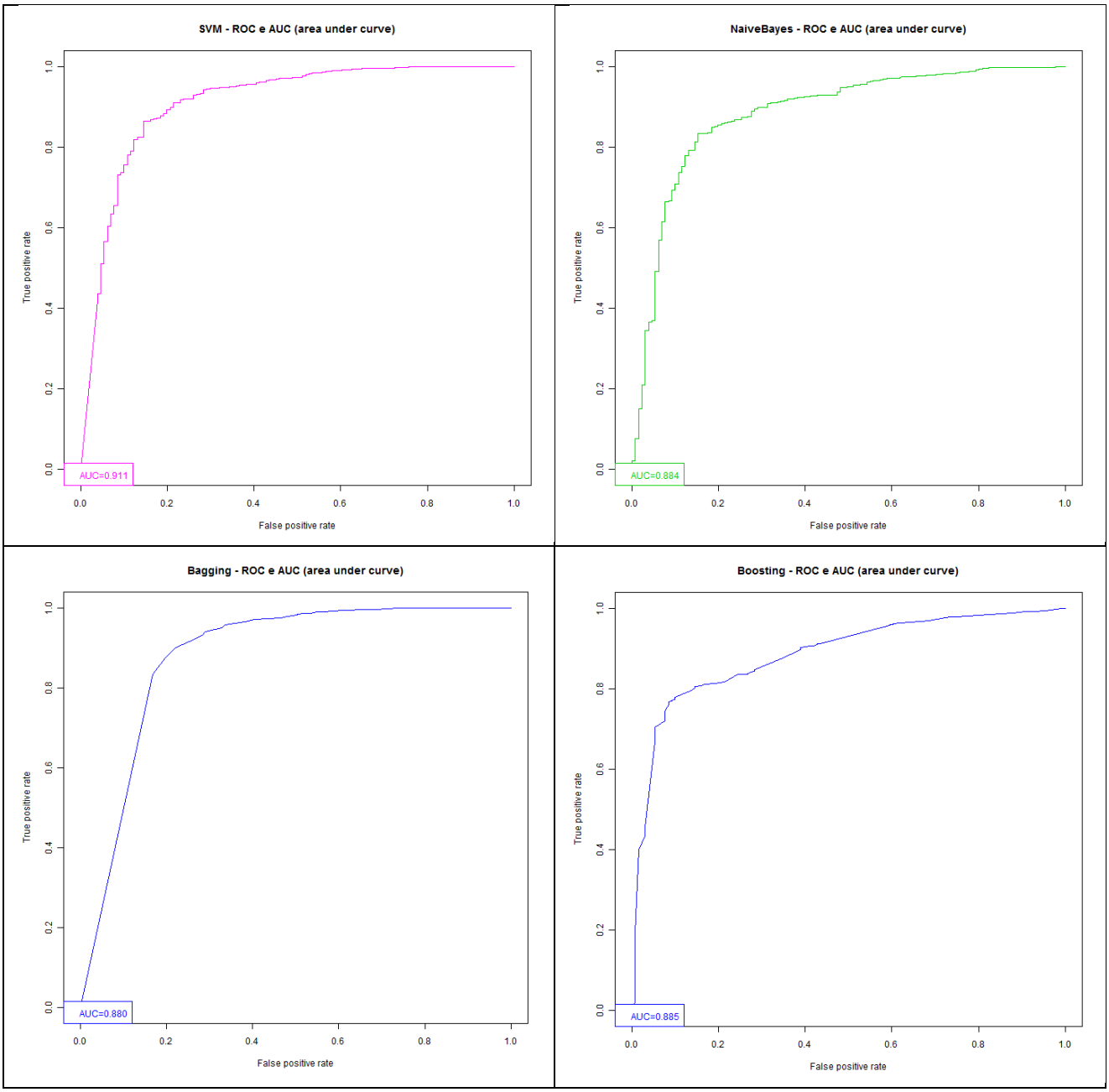


Figura 25- Curvas ROC dos Classificadores individuais

Para facilitar a leitura, a Tabela 21 reproduz os índices de desempenho mensurado através da métrica AUC para esses classificadores. O classificador de melhor desempenho foi o SVM mas note que todos eles obtiveram desempenho similar, exceto os baseados *Ctree* ou KNN que apresentaram desempenho 15,04% e 15,38% menores do que o SVM, respectivamente.

Naive Bayes	C4.5	CTREE	*Bagging*	*AdaBoost.M1*	SVM	KNN
0,884	0,862	0,774	0,880	0,885	0,911	0,748
-2,96%	-5,38%	-15,04%	-3,40	-2,85	Perda % relativa ao SVM	-15,38

Tabela 21 - AUC de Classificadores Individuais

Através da combinação desses classificadores foram gerados multiclassificados. Em geral, os multiclassificadores apresentaram desempenho melhor do que os classificadores individuais.

O modelo multiclassificação **cascata homogêneo** foi avaliado usando a base de avaliação de 30% e o *cross-validation.* O multiclassificador cascata homogêneo de melhor desempenho foi o baseado no classificador *Naive Bayes* para a abordagem base de treinamento e base de avaliação. Houve ganho de performance com relação a abordagem de classificador individual para *Naive Bayes* e Árvore de Inferência Condicional e perda para C4.5 (Tabela 22).

Análises	*Naive Bayes*	C4.5	CTREE	*Bagging*	*AdaBoost.M1*	KNN
	0,944	0,822	0,833	0,884	0,885	0,714
Perda % relativa ao NB	-	-12,9	-11,8	-6,36	-6,25	-24,36
Ganho (%) relativo à Tabela 21	6,79	-4,64	7,62	0,45	0,00	-4,54

Tabela 22 - AUC Multiclassificador Cascata Homogêneo

Os resultados obtidos para multiclassificação **cascata homogêneo** com *cross-validation* estão apresentados na Tabela 23, tendo o *Naive Bayes* também apresentado o melhor desempenho, agora medido em F-*measure*.

Análises	*Naive Bayes*	C4.5	CTREE	*Bagging*	*AdaBoost.M1*	SVM	KNN
	0,811	0,807	0,782	0,649	0,801	0,735	0,644
Perda % relativa ao NB	-	-0,49	-3,58	-19,98	-1,23	-9,37	-20,59
Ganho % relativo à Tabela 20	6,79	-4,64	7,62	0,45	0,00		

Tabela 23 - F-*measure* Multiclassificador Cascata Homogêneo (*Cross-validation*)

O modelo multiclassificação por **votação heterogêneo** foi gerado apenas para a abordagem base de treinamento e base de avaliação. O desempenho inicial mensurado como *F-measure* foi de 0,791. Como nesta pesquisa o foco da Coordenadoria do SEAF é a previsão da emissão de COP, sendo de menor valia classificação de não-COP, esse multiclassificador foi alterado para utilizar peso um para classificação de COP (COP=0) e peso 0,2 para classificação não-COP (COP=1).. Com essa abordagem o *F-measure* aumentou para 0,846. Os classificadores individuais utilizados foram *Naive Bayes*, C4.5, *Ctree*, *Bagging*, SVM e *AdaBoost*.

O modelo multiclassificador **ponderado heterogêneo** foi gerado com a abordagem de base de treinamento e base de avaliação e combinou os seguintes classificadores individuais: *Naive Bayes*, C4.5, *Ctree*, *Bagging* e SVM. A performance obtida foi de 0,909, mensurada em AUC (Figura 26).

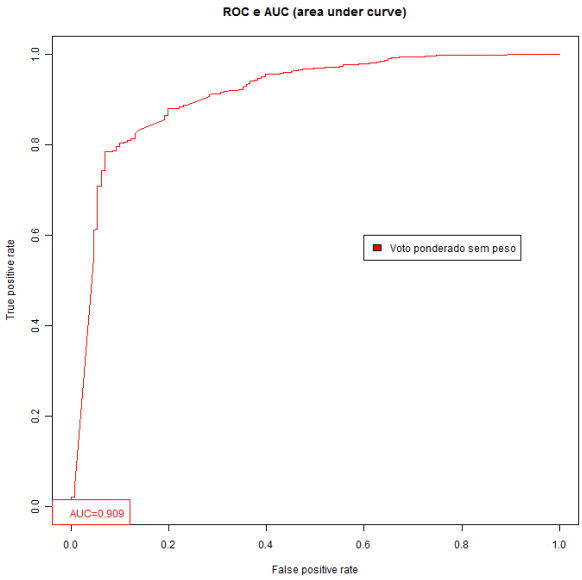


Figura 26 - Curvas ROC Classificador Ponderado Heterogêneo.

O multiclassificador **disjunto heterogêneo** de melhor performance alcançou AUC de 0,915 (Figura 27) e *F-measure* de 0,83.

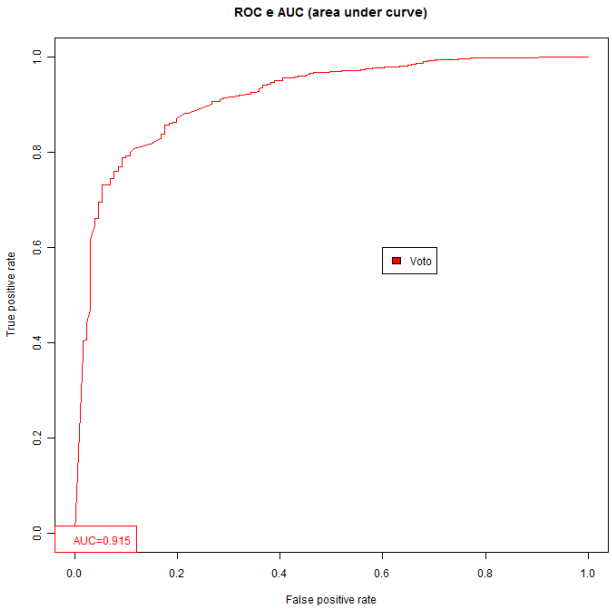


Figura 27 - Curvas ROC Classificador Disjunto Heterogêneo

As matrizes de confusão para os multiclassificadores citados estão apresentadas na Tabela 23. Os multiclassificadores foram obtidos com a abordagem de base de treinamento e base de avaliação. Para o multiclassificador cascata homogêneo, foi considerado apenas o baseado no classificador individual *Naive Bayes*, por ter apresentado o melhor desempenho. O multiclassificador ponderado apresentado utiliza pesos 1 para emissão de COP (COP=0) e 0,2 para não COP (COP=1) pois foi o que apresentou melhores resultados. A partir dos dados dessas

matrizes de confusão é possível calcular as métricas de sensibilidade, especificidade, acurácia e F-*measure*, utilizando as fórmulas apresentadas na Seção 2.7.

	Cascata NB		**Disjunto**		**Ponderado**		**Votação**	
	0	**1**	**0**	**1**	**0**	**1**	**0**	**1**
0	107	24	118	13	116	15	110	21
1	464	2326	558	2232	516	2274	413	2377

Tabela 24 - Matriz de Confusão Multiclassificadores

Na Tabela 25 é apresentado o resultado da execução dos modelos de multiclassificadores propostos com a utilização de *cross-validation.* Os resultados demonstram que os modelos multiclassificadores nessa estrutura de avaliação apresentaram bons resultados.

Cascata NB	**Disjunto**	**Ponderado**	**Votação**
0,811	0,860	0,856	0,856

Tabela 25 - *F-measure* dos multiclassificadores (*cross-validation*)

4.3 Considerações Finais

A execução do *Apriori* não gerou bons resultados, relacionado ao nosso objetivo de obter regras que pudessem associar-se com a COP. Foi necessário definir o suporte menor que 5% (total de COP na base) para que fosse possível surgir regras associada a COP. Isso provocou a geração de um número muito grande de regras e nenhuma dela associada à emissão de COP (COP=0). Essa abordagem não apresentou resultados significativos e sua exploração foi descontinuada.

A Tabela 26 apresenta índices de performance para os classificadores e multiclassificadores estudados, todos eles derivados a partir da abordagem de base de treinamento e base de avaliação.

Modelo	**Tipo1**	**Tipo2**	**F-measure**	**AUC**
Cascata NB	Homogêneo	Multiclassificador	0,811	0,944
Disjunto	Heterogêneo	Multiclassificador	0,847	0,915
Ponderado	Heterogêneo	Multiclassificador	0,848	0,909
Votação	Heterogêneo	Multiclassificador	0,845	0,890
Naive Bayes	-	Classificador Individual	0,820	0,884
SVM	-	Classificador Individual	0,691	0,911
C4.5	-	Classificador Individual	0,822	0,862
CTree	-	Classificador Individual	0,823	0,774
KNN	-	Classificador Individual	0,732	0,748
AdaBoost	Homogêneo	Multiclassificador	0,804	0,885
Bagging	Homogêneo	Multiclassificador	0,745	0,880

Tabela 26 - Resumo dos resultados de classificadores

Por fim, através de vários modelos propostos, alguns multiclassificadores, foi possível verificar seus resultados através da análise do *F-measure*, ROC e AUC. Vimos que o melhor classificador individual foi o SVM. No contesto geral o classificador homogêneo em cascata obteve

o melhor resultado absoluto, AUC = 0,944. Outros modelos também apresentaram grande eficiência, como o modelo disjuntivo e ponderado heterogêneo com AUC = 0,915 e AUC = 0,909 respectivamente. É interessante observar que esses dois últimos modelos possuíam algoritmos de classificação distintos e combinados apresentaram resultado melhor do que trabalhando separadamente.

Com base nos resultados experimentais obtidos, podemos afirmar então que a combinação de algoritmos de classificação pode gerar melhores resultados para a montagem de um classificador único. Os multiclassificadores propostos apresentaram desempenho melhor do que o multiclassificadores convencionais *bagging* e *AdaBoost.M1*.

Com o modelo cascata *Naive Bayes* apresentando resultados muito bons, podemos implementar um classificar evolutivo que pode fornecer classificações intermediárias o que permitira uma classificação antes da conclusão dos laudos. Dessa forma o seguro poderia solicitar alguma ação técnica reparadora. Essa alternativa está sendo discutida junto à Coordenação do SEAF.

5 Conclusões e Trabalho Futuros

Essa pesquisa propõe novas abordagens para construir modelos de inteligência artificial para identificar COP ou apontar indícios de COP em seguros agrícolas do MDA. Para validar experimentalmente as abordagens propostas, os modelos foram construídos utilizando scripts em R e aplicados aos dados disponíveis nas safras de agricultura familiar dos anos 2006 a 2010.

Realizamos os treinamentos e testes utilizando os modelos propostos como: regras de associação com *Apriori,* classificadores individuais, multiclassificador em cascata homogêneo, multiclassificador ponderado heterogêneo, multiclassificador disjuntivo heterogêneo e multiclassificador por votação heterogênea. Os resultados obtidos pelo Apriori não foram bons e não cumpriram o objetivo de identificar indícios de COP. Os modelos propostos de multiclassificadores chegaram a resultados bons com AUC = 0,909 no ponderado, AUC = 0,915 no disjuntivo e AUC = 0,944 no cascata onde o máximo é 1. O multiclassificador disjuntivo que aceitou como COP qualquer registro que tivesse sido classificado COP por qualquer dos classificadores se tornou bem interessante, pois classificou TP em maior número apesar de ter FP com valor superior. Como o valor do seguro é pequeno e a chance de fraude isolada é pequena, esse modelo pode ser bem interessante para ser aplicado pelo MDA.

O modelo em cascata que obteve o melhor resultado é uma aplicação específica para esse caso, pois o acompanhamento é através de laudos técnicos elaborados em três momentos específicos da cultivação. Com isso podemos realizar uma classificação por etapas obtendo classificações parciais ao longo do tempo. Dessa forma é possível antecipar ações técnicas e previsão orçamentária para um melhor funcionamento do seguro.

Esse estudo também propiciou a identificação de novas variáveis que poderiam compor os laudos, como por exemplo, a posição GPS da lavoura. Essa variável permitiria a integração do sistema de laudos com o sistema **Agritempo** permitindo a criação de novas variáveis que pudessem compor a avaliação do laudo. Outra conseqüência desse estudo foi o desenvolvimento de um novo sistema de captação de laudos mais flexível e com maior controle dos dados. Verificamos que os erros de preenchimento dos laudos no sistema antigo era extremamente grande e gerou um trabalho muito grande no pré-processamento. Também foi fornecido ao MDA um conjunto de ferramentas para avaliações estatísticas dos laudos contendo *plots*, histogramas, correlação de variáveis e distribuição de estados por tipo de laudo.

Os resultados foram disponibilizados para o MDA contribuindo para a inserção da mineração de dados, em especial, técnicas de classificação, na sua rotina de trabalho do SEAF. O estudo também contribuiu com a melhoria da qualidade da base de dados com o desenvolvimento e colocação em produção de mecanismos de proteção e filtros na impostação dos laudos da nova safra. Com a criação de uma base de informações e estudos mais aprofundados a partir desse estudo teremos:

- melhora do processo de cultivo pelo agricultor
- maior sustentabilidade do seguro e do princípio de garantia de renda.

- viabilização de prêmios adequados à realidade, evitando custos relativamente altos para o produtor, cálculos mais específicos.
- Melhor conhecimento dos fatores de risco, possibilitando torná-los mais administráveis.
- redução da imprevisibilidade dos gastos.
- redução dos custos com indenizações.
- geração de condições para no futuro vir a ser criado um fundo, possibilitando melhor gestão orçamentária.

Como utilizamos o R para o desenvolvimento desta pesquisa, disponibilizamos para a comunidade científica os recursos, algoritmos e modelos de classificação desenvolvidos, na forma de uma biblioteca disponível para a comunidade de usuários do R.

A combinação de classificadores foi mais eficiente nessa pesquisa e foi necessário aplicar técnicas de balanceamento de classes, pois é inerente ao problema de detecção de COP que as bases sejam desbalanceadas. Dessa forma podemos apontar como um trabalho futuro a criação de um multiclassificador heterogêneo que possa de forma automatizada aplicar técnicas de balanceamento de classes e da seleção da abordagem da combinação dos classificadores individuais para obtenção de um multiclassificador mais eficiente.

Bibliografia

Abdi, H. (2007). *Kendall rank correlation*. Retrieved from Encyclopedia of Measurement and Statistics: http://www.utdallas.edu/~herve/Abdi-KendallCorrelation2007-pretty.pdf

Agrawal, R. a. (1994). Fast algorithms for mining association rules in large databases. *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB*, (pp. 487-499). Santiago, Chile.

Agrawal, R. S. (1996). Parallel mining of association rules. *IEEE Transactions on Knowledge and Data Engineering, vol. 8, NO. 6* .

Aha, D. W. (1991). Instance-based learning algorithms. *Machine Learnig 6* , pp. 37-66.

Bauer, E., & Kohavi, R. (1999). An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting and Variants. *Machine Learning 36* , pp. 105-142.

Breiman, L. (1996). Bagging Predictors. *Machine Learning 24* , pp. 123-140.

Buhlmann, P., & Hothorn, T. (2007). Boosting algorithms: Regularization,prediction and model. *Statistical Science, 22* , pp. 477-505.

CRISP. (2009). *Cross Industry Standard Process for Data Mining. Disponível no site padrão CRISP-DM*. Retrieved from http://www.crisp-dm.org.

D.B. Rorabacher. (1991). *Anal. Chem. 63* , p. 139.

Domingos, P. (1997). Why Does Bagging Work? A Bayesian Account and its Implications. *In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining.*

Fayyad, U. M. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. *Artificial intelligence, 13* , pp. 1022–1027.

Figueira. (1998). M.M.C, Identificação de Outliers. *MILLENIUM nº12* .

Freund, Y. (1995). Boosting a weak learning algorithm by majority.

Freund, Y., & Schapire, R. E. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. *Lecture Notes in Computer Science* , pp. 23-37.

Freund, Y., & Schapire, R. E. (1999). A Short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence, 14* , pp. 771-780.

Freund, Y., & Schapire, R. E. (1996). Experiments with a New Boosting Algorithm. *AT&T Research* .

Friedman, D. G. (1997). *Machine Learning 29* , pp. 131-163.

Grubbs, F. (1969). Procedures for Detecting Outlying Observations in Samples. *Technometrics, 11* , pp. 1-21.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation, 2nd ed.*

Hothorn, T. K. (2006). Unbiased Recursive Partitioning: A Conditional Inference Framework. *Journal of Computational and Graphical Statistics 15(3)* , pp. 651–674.

Hothorn, T., & Buhlmann, P. (2007). mboost Illustrations.

Iglewicz, B. a. (1993). How to Detect and Handle Outliers. *American Society for Quality Control* .

Ingargiola, G. B. (1996). *Classification Models: ID3 and C4.5*. Retrieved from http://www.cis.temple.edu/~ingargio/cis587/readings/id3-c45.html

Kubat, M. a. (1997). Addressing the curse of imbalanced data set: One sided sampling. *Proceedings of the Fourteenth International Conference on Machine Learning* , pp. 179-186.

Ling, C. a. (1998). Data Mining for Marketing: Problems and Solutions. *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining* , pp. 73-79.

Meir, R., & Rätsch, G. (2003). An Introduction to Boosting and Leveraging. *Lecture Notes in Computer Science* , pp. 118-183.

Peters, A., Hothorn, T., & Lause, B. (2002). ipred: Improved predictors. *R-News* , pp. 33-36.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers* .

Quinlan, J. R. (1996). Bagging, Boosting and C4.5. *In Proceedings of the Thirteenth National Conference on Artificial Intelligence* .

Quinlan, J. R. (1986). Induction of Decision Trees. *Mach. Learn. 1* , pp. 81-106.

Ranawana, R., & Palade, V. (2006). Multi-Classifier Systems: Review and a roadmap for developers. *International Journal of Hybrid Intelligent Systems Vol 3* .

Ridgeway, G. (2007). Generalized Boosted Models:A guide to the gbm package.

Rijsbergen, C. J. (1979). Information Retrieval. *Butterworth-Heinemann, London,2nd edition.*

Schapire, R. E. (1999). A Brief Introduction to Boosting. *Proceedings of the Sixteenth International Joint.*

Schapire, R. (2002). The boosting approach to machine learning: an overview. *MSRI Workshop on Nonlinear Estimation and Classification* .

Schapire, Y. F. (1996). Experiments with a new boosting algorithm. *In Proceedings of the International Conference on Machine Learning*, (pp. 148–156).

Stefansky, W. (1972). Rejecting Outliers in Factorial Designs. *Technometrics, 14* , pp. 469-479.

Stone, M. (1974). Cross-validation choice and assessment of statistical predictions. *Journal of the Royal Statistical Society* , pp. B-36, 111–147.

Tsymba, A., & Puuronen, S. (2000). Bagging and Boosting with Dynamic Integration of Classifiers. *Discovery, Proc. PKDD* .

Vapnik, V. (1995). The Nature of Statistical Learning Theory. *New York: Springer-Verlag* .

Visa, S. a. (2005). Issues in Mining Imbalanced Data Sets - A Review Paper. *Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference* , pp. 67-73.

Apêndice A: Tabela Resumo de Comandos R

Comandos	**Descrição**
summary	Descrição detalhada de objetos
lm	Modelo linear
plot	Plotagem gráfica de valores (gráficos)
read.table	Leitura de arquivos externos
factor	Conversão de tipo numérico para nominal
hist	Histograma
outlier	*Outliers* (erros)
numeric	Conversão de tipo nominal para numérico
write.csv	Escrita de arquivo CSV
jpeg	Escrita de arquivo JPEG
cor	Correlação de variáveis
sum	Somatório
max	Máximo
min	Mínimo
length	Comprimento
is.matrix	Verifica se é matriz
is.vector	Verifica se é vetor
as.matrix	Converte para matriz
as.vector	Converte para vetor
c	Codificar números em nomes
rpart	Gerar modelo árvore de decisão
predict	Avaliar modelo (classificar base a partir de modelo)
table	Cruzamento de valores
NA	Dados faltantes (missing values)
cbind	Combinar colunas
text	Escrever textos

Apêndice B: Laudo 1 (Pré-Plantio)

Ministério do Desenvolvimento Agrário
Secretaria da Agricultura Familiar

Seguro da Agricultura Familiar - SEAF
UNIDADES DE REFERÊNCIA **Laudo 1**

1 - Entidade	Sigla/ Nome		Unidade Local	
2 - Técnico	a. CPF	b. Nome		c. CREA/UF
3 - Produtor	a. CPF	b. Nome		c. Fone
4 - Contrato	a. Banco	b. Código da Agência	c. Número Operação/Ano	d. Município da Agência	e. Safra
5 - Lavoura Segurada	a. Cultura Principal	b. Cultura Consorciada		6 – Data Visita	

DADOS DA PROPRIEDADE

7 – Identificação/Características/Localização

a. Nome	e. Coordenadas GPS (sede) DATUM SAD 69	Latitude ° ' "	Longitude ° ' "	
b. Roteiro de Acesso	f. Área Total da Propriedade			
c. Bairro ou Comunidade	g. Regime de Uso da Terra	☐ Proprietário	☐ Posseiro	☐ Arrendatário
d. Município	h. Nº Pessoas Residentes na Propriedade			

08 – Panorama Produtivo da Propriedade

a. Safra		b. Pessoas Trabalhando na Propriedade (homens/ano)	

c. Atividade/Fonte de Renda	d. Área Ocupada	Produção e. Quantidade	Produção f. Unidade	g. Receita
TOTAL				

DADOS DA UNIDADE DE REFERÊNCIA

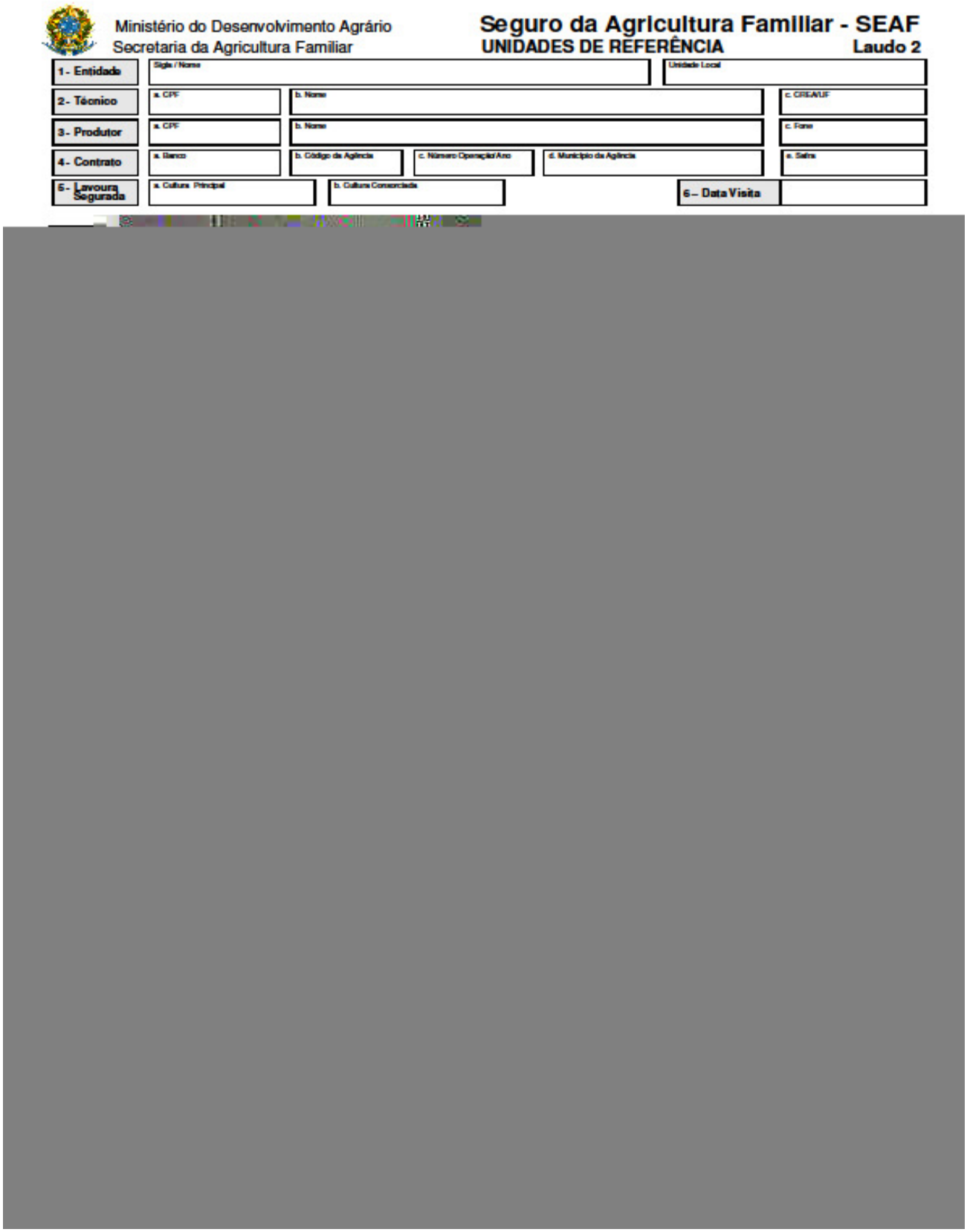
	09 – Cultura Permanente	Área 01	Área 02	Área 03
Fitossanitário	a. Exist. Pragas			
	b. Impacto Pragas			
	c. Exist. Doenças			
	d. Impacto Doenças			
Fisiológico	e. Idade			
	f. Vigor	B M A	B M A	B M A
	g. Estágio de Produção			

Obs.: 1 - Verificar os códigos próprios nas instruções para preenchimento.
2 - Impostar os dados no Sistema SEAF, no endereço seaf.mda.gov.br.
3 - Manter em arquivo o original assinado.

Assinatura do Técnico

Assinatura do Produtor

Apêndice C: Laudo 2 (Plantio)



Ministério do Desenvolvimento Agrário
Secretaria da Agricultura Familiar

Seguro da Agricultura Familiar - SEAF
UNIDADES DE REFERÊNCIA **Laudo 2**

1- Entidade	Sigla / Nome			Unidade Local	
2- Técnico	a. CPF	b. Nome			c. CREA/UF
3- Produtor	a. CPF	b. Nome			c. Fone
4- Contrato	a. Banco	b. Código da Agência	c. Número Operação/Ano	d. Município da Agência	e. Safra
5- Lavoura Segurada	a. Cultura Principal	b. Cultura Consorciada		6 – Data Visita	

Seguro da Agricultura Familiar - SEAF **Laudo 2**
UNIDADES DE REFERÊNCIA Pós-Emergência/Transplantio*

ANEXO I

CONSORCIOS

		Área 1	Área 2	Área 3
11a- Cultura Consorciada	a. Cultura plantada			
	b. Cultivar Plantada			
	c. Ciclo Fenol. Cultivar			
	d. Semente			
	e. Área de Emergência (ha)			
	f. Emergência (%)			
	g. Stand (plantas/ha)			
	Produtividade h.Potencial Cultivar (kg/ha)			
	Produtividade i.Potencial Lavoura (kg/ha)			
	j. Data provista p/ Colheita			

		Área 1	Área 2	Área 3
11b- Cultura Consorciada	a. Cultura plantada			
	b. Cultivar Plantada			
	c. Ciclo Fenol. Cultivar			
	d. Semente			
	e. Área de Emergência (ha)			
	f. Emergência (%)			
	g. Stand (plantas/ha)			
	Produtividade h.Potencial Cultivar (kg/ha)			
	Produtividade i.Potencial Lavoura (kg/ha)			
	j. Data provista p/ Colheita			

		Área 1	Área 2	Área 3
11c- Cultura Consorciada	a. Cultura plantada			
	b. Cultivar Plantada			
	c. Ciclo Fenol. Cultivar			
	d. Semente			
	e. Área de Emergência (ha)			
	f. Emergência (%)			
	g. Stand (plantas/ha)			
	Produtividade h.Potencial Cultivar (kg/ha)			
	Produtividade i.Potencial Lavoura (kg/ha)			
	j. Data provista p/ Colheita			

		Área 1	Área 2	Área 3
11d- Cultura Consorciada	a. Cultura plantada			
	b. Cultivar Plantada			
	c. Ciclo Fenol. Cultivar			
	d. Semente			
	e. Área de Emergência (ha)			
	f. Emergência (%)			
	g. Stand (plantas/ha)			
	Produtividade h.Potencial Cultivar (kg/ha)			
	Produtividade i.Potencial Lavoura (kg/ha)			
	j. Data provista p/ Colheita			

_____

Assinatura do Técnico

_____

Assinatura do Produtor

Obs.: 1 - Verificar os códigos próprios nas instruções para preenchimento.
2 - Preencher em campo tendo em mãos o Laudo 1.
3 - Impostar os dados no Sistema SEAF, no endereço seaf.mda.gov.br.
4 - Manter em arquivo o original assinado.

Seguro da Agricultura Familiar - SEAF **Laudo 2**
UNIDADES DE REFERÊNCIA Pós-Emergência/Transplantio*

ANEXO II

ANÁLISE DE SOLO

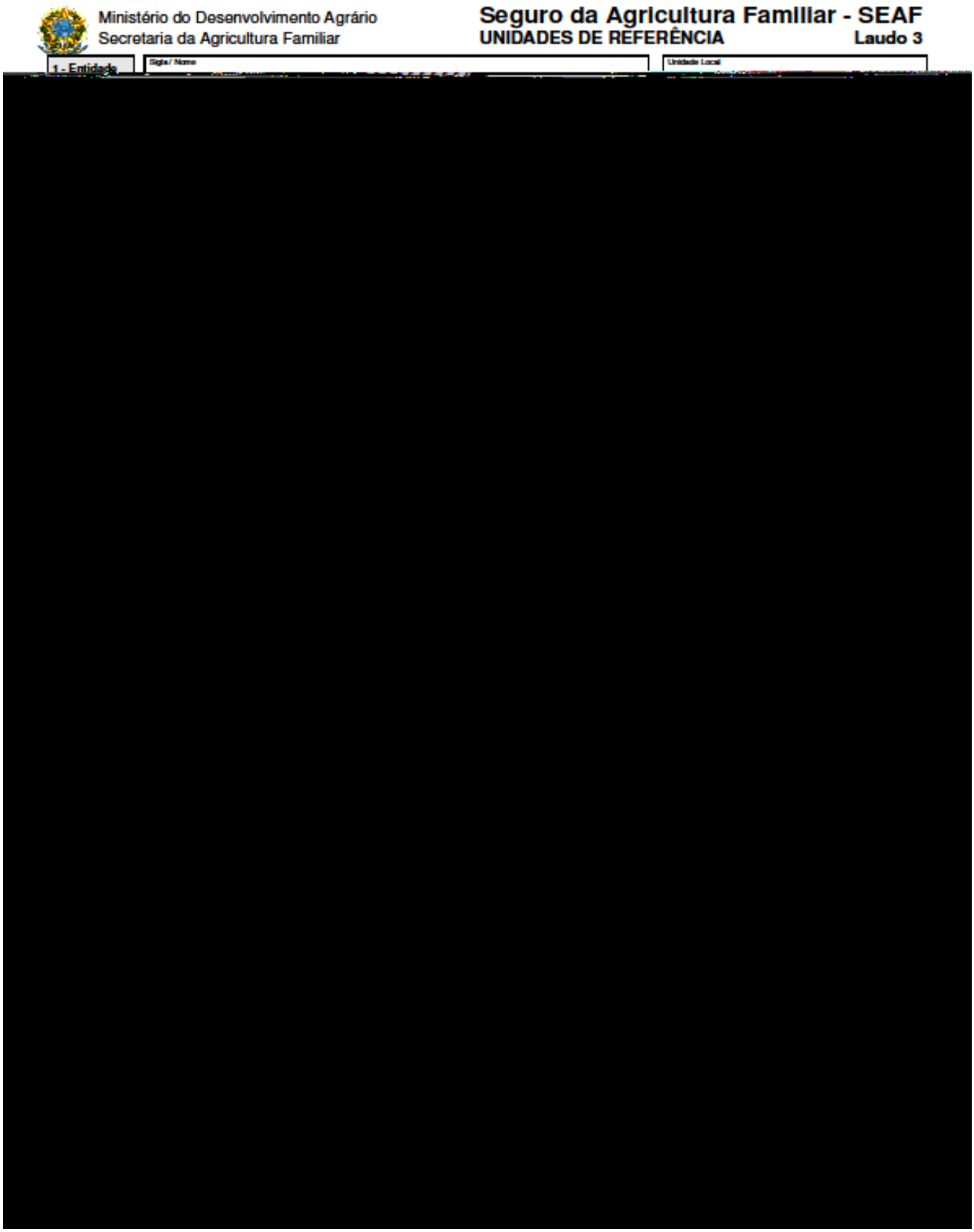
			Área 1	Área 2	Área 3
19 – Análise do Solo	a. Data da Análise				
	b. pH do Solo				
	c. Saturação de base (%)				
	d. CTC (meq/100ml)				
	e. H + Al (meq/100ml)				
	f. . Matéria Orgânica (g/kg)				
	g. Areia (g/kg = %/10)				
	h. Silte (g/kg = %/10)				
	i. Argila (g/kg = %/10)				
	MacroNutrientes	j.Ca (mmol/dm³)			
		l.Mg (mmol/dm³)			
		m. Na (mmol/dm³)			
		n. N (mmol/dm³)			
		o. P (mg/dm³)			
		p. K (mmol/dm³)			
	Micro Nutrientes	q. (mg/dm³)			
		r. Zn (mg/dm³)			
		s. Fe (mg/dm³)			
		t. Cu (mg/dm³)			
		u. Mn (mg/dm³)			
		v. Mo (mg/dm³)			
		x. Co (mg/dm³)			
		z.S (mg/dm³)			

_____
Assinatura do Técnico

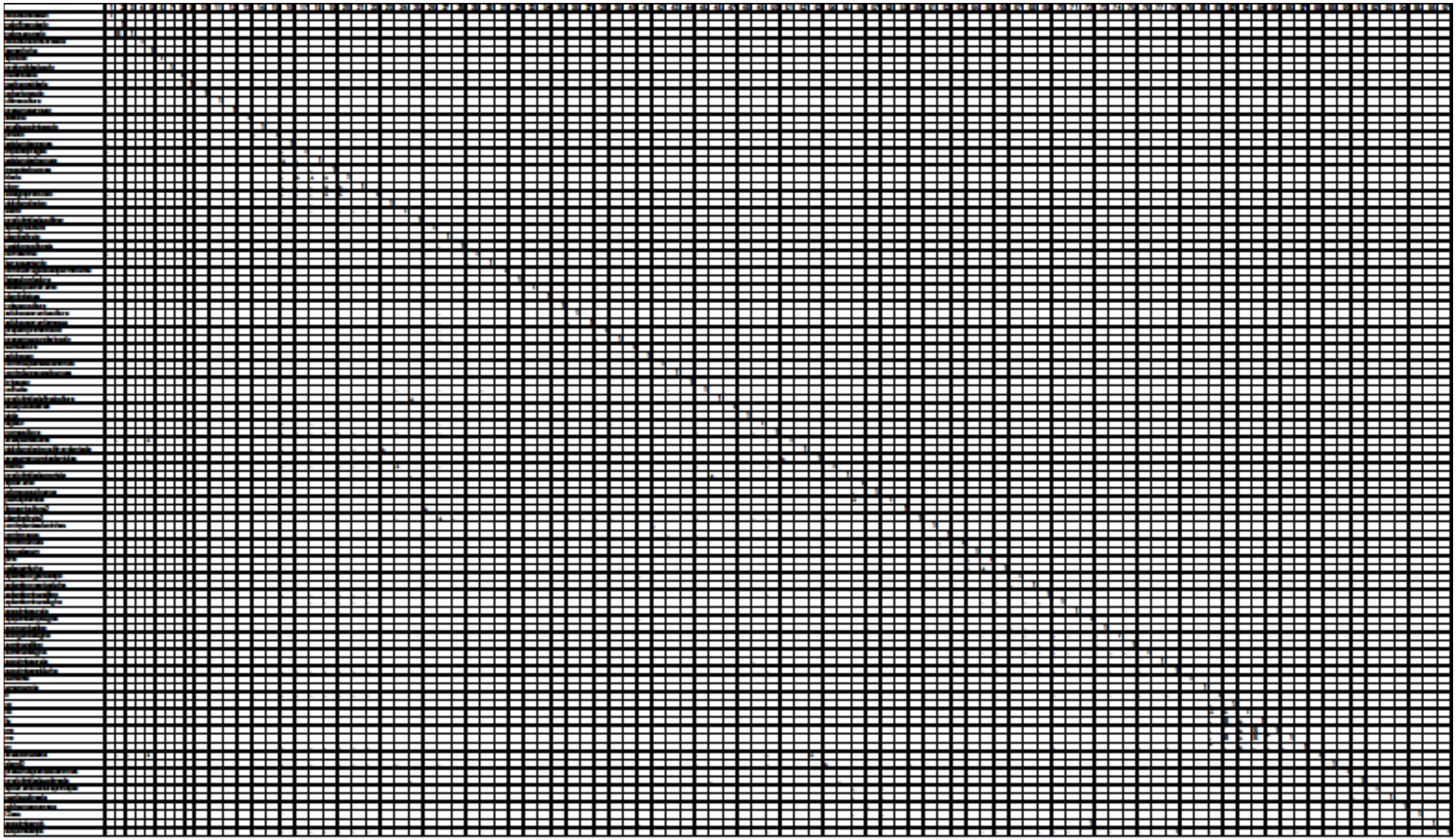
_____
Assinatura do Produtor

Obs.: 1 - Verificar os códigos próprios nas instruções para preenchimento.
2 - Preencher em campo tendo em mãos o Laudo 1.
3 - Impostar os dados no Sistema SEAF, no endereço seaf.mda.gov.br.
4 - Manter em arquivo o original assinado.

Apêndice D: Laudo 3 (Colheita)



Apêndice E: Correlação de Kendall entre as Variáveis



Apêndice F: Exemplo de Scripts Gerado em R

Funções Auxiliares

#cria os histogramas de var numéricas

```
save.histograma <- function (x) {

        ncy <- ncx <- ncol(x)
        colunas <- colnames(x)
      if (ncx == 0)
          stop("'x' is empty")
        for (i in seq_len(ncx)) {
                if(is.numeric(x[1, i]) == TRUE)
                {
                        print(colunas[i])
                        jpeg(file=paste("C:\\clientes\\unb\\mestrado\\IA2\\resultados\\hist_",
colunas[i], ".jpg"))
                        hist(x[,i], main = paste("Histograma de " , colunas[i]))
                        dev.off()
                }
        }
}
```

#cria todos os plots em relacao a alvo

```
save.plote <- function (x, alvo) {

        ncy <- ncx <- ncol(x)
        colunas <- colnames(x)
      if (ncx == 0)
          stop("'x' is empty")
        for (i in seq_len(ncx)) {
                        print(colunas[i])
                        jpeg(file=paste("C:\\clientes\\unb\\mestrado\\IA2\\resultados\\plot_",
colunas[i], ".jpg"))
                        plot(alvo ~ x[,i], x, main = paste("Plot de " , colunas[i]))
                        dev.off()
        }
}
```