Juelline Shelci Silva

Gerenciamento Integrado de Riscos: Modelos de Predição de Risco de Crédito em *Machine Learning* para a Identificação de Ativos Problemáticos em uma Instituição Financeira – Segmento Habitacional PF

Brasil

2022

Juelline Shelci Silva

Gerenciamento Integrado de Riscos: Modelos de Predição de Risco de Crédito em *Machine Learning* para a Identificação de Ativos Problemáticos em uma Instituição Financeira – Segmento Habitacional PF

Dissertação apresentada ao Curso de Mestrado Profissional em Economia, Universidade de Brasília, como requisito parcial para a obtenção do título de Mestre em Economia

Universidade de Brasília - UnB

Faculdade de Administração Contabilidade e Economia - FACE

Departamento de Economia - ECO

Mestrado em Economia

Orientador: Prof. Daniel Oliveira Cajueiro, PhD

Brasil

2022

Juelline Shelci Silva
Gerenciamento Integrado de Riscos:
Modelos de Predição de Risco de Crédito em
Machine Learning para a Identificação
de Ativos Problemáticos em uma Instituição Financeira –
Segmento Habitacional PF/ Juelline Shelci Silva. – Brasil, 2022-
72p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Daniel Oliveira Cajueiro, PhD

Dissertação (Mestrado) – Universidade de Brasília - UnB
Faculdade de Administração Contabilidade e Economia - FACE
Departamento de Economia - ECO
Mestrado em Economia, 2022.

1. Palavra-chave1. 2. Palavra-chave2. 3. Palavra-chave3. II. Universidade de Brasília. III. Faculdade de Administração, Contabilidade e Economia - FACE. IV. Departamento de Economia

Juelline Shelci Silva

Gerenciamento Integrado de Riscos: Modelos de Predição de Risco de Crédito em *Machine Learning* para a Identificação de Ativos Problemáticos em uma Instituição Financeira – Segmento Habitacional PF

Dissertação apresentada ao Curso de Mestrado Profissional em Economia, Universidade de Brasília, como requisito parcial para a obtenção do título de Mestre em Economia

Trabalho aprovado. Brasil, 14 de Junho de 2022:

Prof. Daniel Oliveira Cajueiro, PhD
Orientador

Professor
Convidado 1

Professor
Convidado 2

Brasil
2022

Dedico esse trabalho à minha mãe, Dona Juraci,
que desde o início da minha caminhada me ensinou
a acreditar que os nossos sonhos nos guiam
e que eles são possíveis.

Agradecimentos

Agradeço a Deus pela oportunidade do aprendizado, aos professores, que compartilharam suas experiências com generosidade, especialmente ao meu professor e orientador, Daniel Oliveira Cajueiro, por toda dedicação e empatia, também ao professor Herbert Kimura.

Obrigada aos colegas de estudos, dentre os quais destaco Almeida Barroso, Lemonier Lima e Camila Pinto, que demonstraram a forma da união e apoio mútuo durante todo o curso.

Aos amigos e colegas de profissão, que me apoiaram compreenderam a necessidade do tempo de dedicação, muito obrigada.

"Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12, 2)

Resumo

A previsão de risco de *default* para clientes de uma instituição financeira, ou seja, apurar se deixará de cumprir com suas obrigações financeiras, é de extrema importância e pode causar um grande impacto nos resultados da Instituição. Nesse sentido, buscamos com nosso estudo supervisionado a identificação de modelos adequados para subsidiar as ações a serem implementadas pelos bancos com a definição da aderência dos modelos mediante variáveis previsoras que acreditamos ter potencial para discriminar bons e maus pagadores. Recorremos à técnica de Regressão Logística (*Logit*) e aos modelos de *Machine Learning*, *Random Forest* e *Gradient Boosting*, com o objetivo de prever quando um contrato habitacional pode se tornar Ativo Problemático (AP), para subsidiar decisões para enfrentamento de cenários extremos, em atendimento às exigências do órgão regulador. Para validação dos modelos, utilizamos, a Curva ROC, além de *Precision* e *Recall*, bem como critério de erro quadrático médio (RMSE). Os modelos apresentaram resultados muito próximos, acreditamos que em função dos algoritmos refletirem a homogeneidade da carteira avaliada, tanto a nível de perfil dos clientes como a nível de comportamento histórico observado nos contratos. Ao final, o modelo *Gradient Boosting* apresentou melhor capacidade para a predição esperada, em termos de acurácia e eficiência quanto à previsão e sensibilidade, o que nos permitiu concluir que os modelos em *Machine Learning* podem ser utilizados para essa finalidade

Palavras-chave: Aprendizagem de Máquina, Risco de Crédito, Ativo Problemático, *Machine Learning*, *Random Forest*, *Gradient Boosting*.

Abstract

The default risk forecast for clients of a financial institution, that is, to determine whether it will no longer comply with its financial obligations, is extremely important and could have a major impact on the institution's results. In this sense, we sought with our supervised study the identification of appropriate models to support the actions to be implemented by banks with the definition of adherence of models through forecasting variables that we believe have the potential to discriminate good and bad payers. We used the Logistic Regression technique (Logit) and the machine learning, random forest and gradient boosting models, in order to predict when a housing contract can become problematic asset (PA), to support decisions to cope with extreme scenarios, in compliance with the requirements of the regulatory body. For validation of the models, we used the ROC Curve, in addition to Precision and Recall, as well as mean quadratic error criterion (RMSE). The models presented very close results, we believe that as the algorithms reflect the homogeneity of the portfolio evaluated, both at the level of the profile of the clients and in the level of historical behavior observed in the contracts. In the end, the Gradient Boosting model presented better capacity for the expected prediction, in terms of accuracy and efficiency in terms of prediction and sensitivity, which allowed us to conclude that the models in Machine Learning could be used for this purpose

Keywords: *Credit Risk*, Problem Asset, *Machine Learning*, *Random Forest*, *Gradient Boosting*.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AP	Ativo Problemático
BACEN	Banco Central do Brasil
CMN	Conselho Monetário Nacional
ES	Entidades Supervisionadas
GPS	Guia de Práticas de Supervisão
IF	Instituições Financeiras
LGPD	Lei Geral de Proteção de Dados
LOGIT	Modelo de Regressão Logística
ML	*Machine Learnning*
REF	Relatório de Estabilidade Financeira
RF	Modelo *Randon Forest*
RL	Modelo de Regressão Logística
SBPE	Sistema Brasileiro de Poupança e Empréstimo
SFI	Sistema de Financiamento Imobiliário
XG	Modelo *Gradient Boosting*

Sumário

1 Introdução

O processo de gerenciamento de risco de crédito em Instituições Financeiras (IF) vem passando por uma revisão ao longo dos últimos anos. Nesse contexto, diversas novas técnicas de mensuração de risco de crédito e tomadores têm sido desenvolvidas e implementadas por grandes Bancos.

Leo, Sharma e Maddulety (2019) constatam que a aplicação de tecnologias em evolução e análises avançadas permitem o surgimento de novos produtos e técnicas de gerenciamento de riscos. Assim, *Machine Learning*, identificada como uma das tecnologias com implicações importantes para o gerenciamento de riscos, pode permitir a construção de modelos de risco mais precisos, que identificam padrões complexos e não lineares dentro de grandes conjuntos de dados. Os autores ressaltam que o poder preditivo desses modelos pode crescer com cada bit de informação adicionada, aumentando assim o poder preditivo ao longo do tempo.

Pesquisas consideráveis (LIEBERGEN et al., 2017; WALSH; VOLINI, 2017; BLANCK et al., 2020; HELBEKKMO et al., 2013) tanto na academia quanto na indústria, se concentram no desenvolvimento da gestão bancária e de riscos e nos desafios atuais e emergentes. Em conjunto, observamos uma influência crescente do aprendizado de máquina em aplicações de negócios, com muitas soluções já implementadas e muitas outras sendo exploradas.

O objetivo desta pesquisa é identificar um modelo de classificação capaz de avaliar o risco de crédito de clientes pessoa física, especificamente na carteira de segmento habitacional de um banco do segmento S1, permitindo a adoção de estruturas adequadas de gerenciamento de riscos, por meio de modelos e sistemas que possibilitem a mensuração e avaliação dos riscos relevantes incorridos para manutenção da sua solvência, liquidez, rentabilidade e estrutura de capital dos bancos. Diversas pesquisas foram realizadas utilizando algoritmos de Regressão Logística, *Random Forest* e/ou *Gradient Boosting* para fins de gestão do crédito (HAMORI SHIGEYUKI E KAWAI, ; KERAMATI; YOUSEFI, 2011; ZHOU; WANG, 2012; BARBOZA; KIMURA; ALTMAN, 2017; ADDO; GUEGAN; HASSANI, 2018).

Outros autores utilizaram algoritmos de *Machine Learning* para desenvolver modelos de previsão da inadimplência e, portanto, de eventual *default*. Vieira (2016) compara os resultados do *Bootstrap Aggregating (Bagging)*, *Random Forest* e Adaptive Boosting (AdaBoost) para predição do bom e do mau pagador também de uma carteira habitacional inserida no Programa Minha Casa Minha Vida, utilizando variáveis predominantemente relacionadas ao tomador do crédito. Destacamos utilizamos no nosso estudo, predominantemente, dados de desempenho dos contratos, diferente do referido trabalho.

Em nosso estudo buscamos comparar modelos de *Machine Learning*, a partir da aplicação de técnicas de Regressão Logística, *Random Forest* e *Gradient Boosting*como forma de avaliar os resultados da carteira habitacional pessoal física, na predição de um contrato se tornar Ativo Problemático, conforme definições dispostas na Res. n° CMN 4557/17. Diferentemente do estudo de Vieira (2016), utilizamos exclusivamente variáveis relacionadas ao desempenho dos contratos concedidos por meio da linha de crédito vinculada ao Sistema Brasileiro de Poupança e Empréstimo (SBPE).

Investigamos as variáveis mais relevantes para a predição, compararmos os resultados obtidos dos modelos por meio das métricas de acurácia, precisão, sensibilidade, curva ROC, MAE e MSE, e apresentamos sugestões direcionadas ao tratamento dos respectivos clientes, com base nos padrões identificados, com o objetivo de contribuir com melhorias na estratégia de acompanhamento do cliente, com adoção de medidas negociais para mitigação do risco de crédito, previamente à efetiva marcação do contrato como ativo problemático.

Verificamos que, comparando técnicas de Regressão Logística com os modelos de *Machine Learning*, *Random Forest* e *Gradient Boosting*, este último possibilita melhor previsão da ocorrência de eventos de *default* com antecedência, para tomada de medidas negociais (pausa, prorrogação, alongamento do prazo, entre outros) ou para renegociação do contrato antes de um contrato de tornar AP, alcançando relevante índice de acerto.

A partir dessa constatação, utilizamos algoritmos para otimização do modelo selecionado, como validação cruzada e outros artifícios como *Hold-out*, *K-fold* e *Hiperparâmetros*, com o objetivo de maximizar o desempenho e otimização dos modelos.

Dado o exposto, o trabalho está dividido da seguinte maneira. No capítulo 02, trazemos os argumentos de literatura em reforço à importância do adequado gerenciamento de riscos inerentes ao acompanhamento da carteira de crédito, bem como o cenário de regulação às IF em resposta às incertezas do mercado, em âmbito mundial. No capítulo 03, falamos sobre o escopo da carteira selecionada, segmento habitacional pessoa física para uma IF do segmento S1, bem como o tratamento realizado na base de dados real. Também falamos sobre as técnicas aplicadas para transformação das variáveis, sobre balanceamento e normalização da base. Levantamos a aplicabilidade e o uso de cada um dos modelos propostos, bem como as metodologias para regularização, validação e aferição da acurácia dos modelos referenciados, quais sejam: Regressão Logística *(Logit)*, *Random Forest* e *Gradient Boosting*. No capítulo 04, apresentamos o estudo comparativo dos resultados obtidos, considerando a particularidade da carteira de crédito abordada, investigamos a seleção das variáveis preditivas mais relevantes na apuração dos resultados e apresentamos sugestões direcionadas ao tratamento dos respectivos clientes. Por fim, o capítulo 05 apresenta a conclusão do nosso estudo, bem como sugestões para trabalhos futuros.

2 Fundamentação Teórica e Legal

2.1 O Crédito e o Risco de Crédito

O conceito de crédito pode ser analisado sob diversas perspectivas. Para uma IF, crédito refere-se, principalmente, à atividade de colocar um valor à disposição de um tomador de recursos sob a forma de um empréstimo ou financiamento, mediante compromisso de pagamento em uma data futura. O crédito geralmente envolve a expectativa do recebimento de um valor em um certo período de tempo. Nesse sentido, (SILVA, 2004) afirma que o risco de crédito é a chance de que essa expectativa não se cumpra. De forma mais específica, o risco de crédito pode ser entendido como a possibilidade de o credor incorrer em perdas, em razão de as obrigações assumidas pelo tomador não serem liquidadas nas condições pactuadas (HU; SU, 2022).

Cada instituição financeira adota seu próprio conceito de evento de *default*, que está normalmente relacionado ao atraso no pagamento de um compromisso assumido pelo tomador. O *default* pode ser apurado de forma subjetiva ou utilizando metodologia quantitativa. Como exemplo, os modelos de *Credit Scoring* são adequados para análises massificadas que definem scores utilizando técnicas de análise estatística, atribuindo-se pesos a variáveis que caracterizam o solicitante e a operação (SICSÚ, 2010). Entretanto, esperamos com o nosso estudo, em contrapartida ao método tradicional de escoragem e utilizando algoritmos para a tomada de decisões no âmbito financeiro com foco em maior efetividade e eficiência nas classificações, garantir maior acurácia sobre o risco de eventual inadimplência, além da redução de demanda de tempo e trabalho.

Existem inúmeros trabalhos que reforçam a motivação do nosso estudo, como Bessis (2011) que define o risco de crédito pelas perdas geradas por um evento de *default* do tomador ou pela deterioração da sua qualidade de crédito. Há diversas situações que podem caracterizar um evento de *default* de um tomador. O autor cita como exemplo o atraso no pagamento de uma obrigação, o descumprimento de uma cláusula contratual restritiva (*covenant*), o início de um procedimento legal como a concordata e a falência ou, ainda, a inadimplência de natureza econômica, que ocorre quando o valor econômico dos ativos da empresa se reduz a um nível inferior ao das suas dívidas, indicando que os fluxos de caixa esperados não são suficientes para liquidar as obrigações assumidas. Com o nosso trabalho, identificamos as variáveis de risco de crédito mais relevantes, considerando o universo e as técnicas aplicadas.

Fitzpatrick e Mues (2016) utilizam dados de empréstimos hipotecários para identificar se técnicas de *Machine Learning* apresentam maior capacidade de prever um *default*

frente a técnicas já consolidadas, como Regressão Logística. Os autores apresentam como vantagens da comparação o impacto que pequena melhora na predição pode gerar na carteira de crédito, na estratégia de preços da instituição e em outras tomadas de decisão estratégicas, bem como na alocação de capital mais apropriada.

O estudo de Trivedi (2020) tratou do problema de pontuação de crédito em dados, fazendo um comparativo entre classificadores de *Machine Learning*, e concluiu que a combinação de *Random Forest* e *Chi Square* é considerada o melhor par entre todos para a construção de modelos de pontuação de crédito.

O processo de gerenciamento de risco de crédito em instituições financeiras também foi abordado no estudo de Brito e Neto (2008), bem como a necessidade de novas técnicas de mensuração de risco de crédito e tomadores por grandes Bancos. Os autores desenvolveram um modelo de classificação de risco para avaliar o risco de crédito de empresas no mercado brasileiro utilizando a técnica estatística de Regressão Logística e, como resultado, o modelo classificou corretamente 90% das empresas da amostra.

Ressaltamos que a deterioração da qualidade de crédito do tomador não resulta em uma perda imediata para a IF, mas sim no incremento da probabilidade de que um evento de *default* venha a ocorrer. Nos sistemas de classificação de risco, as alterações na qualidade de crédito dos tomadores dão origem às chamadas migrações de risco. (BRITO; NETO, 2008).

2.2 Cenário e Regulação

O cenário atual de incertezas associadas às consequências econômico-financeiras decorrentes da pandemia COVID-19, bem como o elevado potencial de agravamento quanto a essas consequências, reforçam a importância de uma regulação precisa para manter a estabilidade do mercado bancário mediante crises financeiras recorrentes em todo o mundo. Para Brito (2022), por mais eficaz que tenha sido o isolamento social, economicamente, impacta fortemente a vida das pessoas e a continuidade das empresas. Esses são fatores que potencializam a inadimplência e o risco de crédito.

Por se tratar de tema recente e pela ausência de outros estudos acerca do tema Ativos Problemáticos (AP), trazemos fundamentos especialmente originados dos próprios órgãos de regulação (CMN) e de supervisão (BACEN).

Em termos de supervisão, citamos os aspectos regulatórios que determinam sobre o tema e trazem a previsão conceitual, de gerenciamento e implementação das regras para marcação de um ativo como problemático, por meio da Resolução CMN nº 4.557/17, norma principiológica que registra em seu escopo que as Instituições Financeiras - IF por ela orientadas. Esta dispõe que as IF devem implementar estrutura de gerenciamento

de riscos compatível com o modelo de negócio, com a natureza das operações e com a complexidade dos produtos, serviços, atividades e processos da Instituição, e, inclusive, apresenta as características gerais de exposições identificadas como AP.

Portanto, conforme a norma, caracteriza-se nesse universo o crédito apresenta atraso superior a 90 dias e/ou que demonstra indicativos de que a respectiva obrigação não será integralmente honrada sem que seja necessário recurso a garantias ou a colaterais e, no nosso estudo, será identificado a partir da verificação de cliente bom ou cliente mau (potencial Ativo Problemático).

Ressaltamos as medidas regulatórias adotas pelo CMN para enfrentamento aos impactos adversos do COVID-19 no setor econômico, a partir da publicação da Resolução CMN nº 4.782/20 e Resolução CMN nº 4.856/20, estabelecendo critérios temporários para a caracterização das reestruturações de operações de crédito realizadas até 30/09/2020 e até 31/12/2020, respectivamente, para fins de gerenciamento de risco de crédito. Tais ações dispensavam, por tempo determinado, as instituições financeiras de marcar essas exposições como AP, buscando facilitar os trâmites de renegociação de dívidas de pessoas físicas e jurídicas que apresentaram ao longo do período boa capacidade de pagamento. Por esse motivo, e pelos impactos na economia a nível mundial, selecionamos base de estudo com referência posterior às referidas resoluções.

O tema Ativo Problemático tem sido objeto de monitoramento e controle recorrente pelo BACEN conforme tem sido publicado através do Relatório de Estabilidade Financeira – REF, que apresenta panorama da evolução recente e perspectivas para a estabilidade financeira no Brasil, com foco nos principais riscos e na resiliência do Sistema Financeiro Nacional (SFN). O relatório é importante indicador das práticas adotadas no mercado quanto ao gerenciamento de riscos e, portanto, deve também balizar o tratamento da carteira de AP.

O Guia de Práticas de Supervisão (GPS) publicado para a Gestão do Risco de Crédito, apresenta recomendações e expectativas do regulador às Entidades Supervisionadas (ES) pelos diversos prismas do risco de crédito, dentre os quais pode-se destacar para o escopo do presente estudo aqueles que versam sobre a Gestão do Risco de Inadimplência e traz, inclusive, critérios para alteração da condição de ativo problemático (desmarcação).

3 ESTUDO EMPÍRICO

3.1 Método e Design de Pesquisa

Nosso estudo foi realizado em Python e busca identificar modelos de predição que possam ser gerados por algoritmos, a partir de dados reais de uma carteira de crédito, para classificação dos contratos vigentes e, se for aplicado, sem necessidade de outras fundamentações teóricas para inclusão das variáveis identificadas como potenciais.

Assim, escolhemos os modelos *Gradient Boosting*, *Random Forest (Cross Validation)* e Regressão Logística para o estudo por considerarmos acessíveis, consolidados, populares e extremamente testados para diversos fins, inclusive como algoritmos de escolha para muitas equipes vencedoras de competições de *Machine Learning*. Como exemplos, foram utilizados por Vieira (2016) para *Random Forest* e Árvore de Decisão, Brito e Neto (2008) para Regressão.

Segundo Addo, Guegan e Hassani (2018), previsões de risco de crédito, monitoramento, confiabilidade do modelo e processamento eficaz de empréstimos são fundamentais para a tomada de decisão e transparência. Foram construídos classificadores binários baseados em modelos de *Machine* e *Deep Learning* em dados reais para prever a probabilidade de inadimplência de empréstimos, concluindo-se que os modelos baseados em árvore são mais estáveis que os modelos baseados em redes neurais artificiais multicamadas.

Utilizamos o Colaboratory ou "Colab" para tratamento dos dados, geração de gráficos, além da aplicação e avaliação dos modelos em *Python*, por meio de códigos disponíveis nas bibliotecas de linguagem de software de código aberto que fornecem framework específicos, quais sejam Pandas, *Numpy* e *scikit-learn*, *XGBoost*.

3.2 Base de Dados

Recorremos a uma base de dados reais contemplando informações de operações de crédito contratadas em uma IF brasileira, no âmbito do segmento imobiliário pessoa, física que obtiveram crédito anteriormente aprovado e contratado relativo a financiamento junto ao Sistema Brasileiro de Poupança e Empréstimo. O SBPE utiliza os rendimentos da caderneta de poupança para emprestar recursos financeiros para compra de compra imóveis por meio de um financiamento, relevante diferença em relação maior programa de financiamento imobiliário do País, Casa Verde e Amarela, que é mais procurado quando a pessoa tem uma renda que não se enquadra no SBPE e adquire vantagens do programa do Governo Federal. Além disso, ele também permite um financiamento de longo prazo para

a maior parte do valor da moradia (80%).

Nossos dados foram coletados com referência em JUL/2021 e apuraram o universo não caracterizado como ativo problemático naquele mês e que, observada a performance do contrato durante o período de 6 meses, configurou como ativo problemático em DEZ/21. Preocupamo-nos com a seleção de dados após 31/12/2020, quando estavam superados os efeitos imediatos pós-pandemia, inclusive as medidas dispostas nas resoluções temporárias do CMN, citadas anteriormente. Apesar de se tratar de modelo de risco de crédito, dado o foco do presente estudo, não consideramos situações conjunturais do mercado de crédito ou especificidades individuais das variáveis, que foram ponderadas pelos modelos de forma associada.

3.2.0.1 Seleção de Amostras

Partimos das informações extraídas da base de dados para uma linha de crédito específica, tanto adimplentes quanto inadimplentes, que contou com 3.980.673 observações. Diante do tamanho da base e para avaliar a eficiência dos modelos em número limitado de observações, consideramos uma amostra aleatória simples de 15% desse público, restando 592.788 contratos após a qualificação do banco (*Missing Value*, *Outliers*, outros). Entendemos o tamanho da base como adequada para o estudo, pois não há prejuízo em desenvolver modelo de risco com softwares estatísticos usuais a partir da amostra de 100 mil observações (SICSÚ, 2010) uma vez que, ainda que seja necessário em estudo posterior trabalhar com toda a base de dados, em geral, ela representa apenas uma amostra do mercado-alvo do credor.

Com relação ao tratamento de *Outliers*, não houve necessidade de ajustes. Como exemplo, mantivemos os valores elevados identificados para os prazos, por exemplo, de até 12.000 dias, uma vez que a base se trata de contratos essencialmente de longo prazo, cuja modalidade prevê vencimento em até 35 anos. Também não foram identificados valores discrepantes em variáveis que relacionam valores (saldo devedor, provisionamento), que se mostraram coerentes com os valores contratados, nem inconsistência de datas ou outros índices, caracterizando, ao nosso ver, o reflexo na eficiência dos controles que abrangem os referidos registros.

3.3 Tratamento das Variáveis

Para definição do universo de contratos, calculamos como variável dependente o atributo “Ativo Problemático”; caracterizando-os como “maus pagadores” caso o contrato se enquadre nesse critério, enquanto os demais clientes foram caracterizados como “bons pagadores”. A amostra quantificada acima considera a restrição dos “maus pagadores” da

base utilizada no estudo, por já se encontrarem em situação deteriorada, convergindo com o objetivo do nosso estudo, essencialmente relacionado ao risco de crédito.

A criação das demais variáveis, as variáveis independentes, foi realizada para construir cenários e comparar qual algoritmo se comporta melhor para a predição de AP no intervalo. O critério de bom e mau pagador é necessário para investigar a acuidade classificatória dos modelos e a segregação dos diferentes tipos de tomadores. Foram eliminadas da base inicial, objeto de análise deste estudo, os contratos liquidados e os contratos cancelados por não se tratar de exposições vigentes.

Lembramos que eventuais variações no cenário econômico do país e estratégias políticas de incentivo ao financiamento imobiliário, não foram objeto desse estudo e, portanto, não foram avaliados.

a) Variáveis Independentes

Tabela 1 – Variáveis Independentes

Variáveis do contrato	**Descrição**	**Métrica**	**Tipo**
Histórico renegociação	Identifica a recorrência de renegociações	Numérica	Informada
Renegociação COVID	Identifica de foi renegociado após COVID-19	Binária	Calculada
Prazo do contrato	Prazo total na data da contratção	Em dias	Informado
Prazo remanescente	Prazo restante para liquidação do contrato	Em dias	Informado
Índice - Remanescente	Prazo restante frente ao prazo total	Numérica	Calculada
Saldo Devedor	Valor remanescente do contrato	Categórica	Informado
Índice - Saldo Devedor	Valor remanescente do contrato – evolução mensal	Numérica	Calculada
Dívida Vencida	Valor em atraso do contrato no início do período (M0)	Categórica	Informado
Maturidade	Valor em atraso do contrato frente ao valor total	Numérica	Calculada
Garantia Real	Existência de garantia real vinculada ao contrato	Binária	Informado
Atraso inicial	Atraso em no início do período (M0)	Categórica	Informado

Atributos de entrada verificados pelos algoritmos

Foram consideradas 79 variáveis nos modelos, bem como as performances dos contratos durante o período de 06 meses de observação, sendo M0 o mês inicial, e M1, M2, M3, M4, M5 e M6 os meses subsequentes. Por se tratar de modelo de risco de crédito,

desconsideramos variáveis que demostrassem “atraso”, “adimplência” e “meses não pagos”, uma vez que atraso é componente direto para marcação de um ativo problemático e, portanto, altamente correlacionado à variável dependente.

Em seguida, utilizando o artifício da *Permutation Feature Importance*, inicialmente utilizado para identificação das variáveis mais relevantes, durante a análise exploratória identificamos a necessidade de exclusão de variáveis independentes, que reduziram de um total de 127 para 79.

Variáveis de risco de crédito são utilizadas, predominantemente relacionadas às características do tomador. No nosso trabalho, foram utilizadas características do contrato, com o objetivo de apurar o comportamento deste no tempo, por meio de EVER30, recorrentemente utilizadas para apurar índices de inadimplência para créditos massificados (BOHN,).

b) Variável Dependente

No contexto de treinamento dos nossos modelos de ML, identificamos os nomes dos atributos nos dados de entrada que contém as respostas "corretas", enquanto algoritmos frente ao atributo de destino, e usamos essas previsões pelos modelos treinados para descobrir padrões nos dados de entrada e gerar esses modelos.

Consideramos como variável dependente, *target*, a possibilidade de marcação de um contrato como AP, de um determinado nicho de clientes do Segmento Habitacional Pessoa Física, conforme definições dispostas na Res. CMN 4557/17, no intuito de subsidiar as instituições financeiras nas decisões para enfrentamento aos cenários extremos, em atendimento às exigências do órgão regulador, permitindo a adoção de estruturas adequadas de gerenciamento de riscos, além de modelos que possibilitem a identificação, mensuração e avaliação dos riscos relevantes incorridos na carteira.

Conforme premissas descritivas na resolução supracitada, o ativo de crédito é considerado Ativo Problemático quando é verificado que a respectiva obrigação está em atraso há mais de noventa dias e/ou há indicativos de que esta não será integralmente honrada sem que seja necessário recorrer a garantias ou a colaterais. Por esse motivo, e pela ausência de estudos específicos relacionados ao tema, utilizamos em nosso trabalho a similaridade entre esse conceito e o conceito de *default*.

Para fins de definição da performance dessa variável, foi utilizado método EVER, para identificação marcação dos contratos como Ativo Problemático mês a mês em determinada base, no início e ao final de um período de 06 meses (binários 0 ou 1), dentre os 592.788 contratos selecionados aleatoriamente. Dessa forma, consideramos como universo desse estudo os contratos que não estavam em *default* no início do período (apenas binário 0) e aqueles que foram ou não marcados nos meses subsequentes, com binários 0 ou 1 para “bons clientes” e “maus clientes”.

Tabela 2 – Variável Dependente - Target

Ativo Problemático	**M0**	**M1**	**M2**	**M3**	**M4**	**M5**	**M6**
Não	0	0	0	0	0	0	0
Sim	-	1	1	1	1	1	1

Atributo Ativo Problemático como resttrição de "maus pagadores"

Durante as pesquisas, não identificamos outros autores que utilizaram métrica idêntica como variável dependente. Assim, reforçamos o tratamento análogo no nosso estudo às vaiáveis de risco de crédito que diferenciam “bons clientes” de “maus clientes” como fizeram Vieira (2016) Barboza, Kimura e Altman (2017) Brito e Neto (2008) Petropoulos et al. (2020)

3.3.1 Engenharia de *Features*

No intuito de extrair o máximo potencial dos dados, utilizamos o artifício de *Feature Engineering* no nosso modelo preditivo, não apenas para selecionar boas *features*, mas também abrangendo a transformação matemática e criação de novas *features*. Assim, para tratamento de *Missing Values*, a exemplo das lacunas identificadas nas ausências de informações para identificação de contratos renegociados, criamos variáveis *dummies* aceitando que a ausência de dados retornava ausência de ocorrências de renegociações para aqueles contratos. Ainda, os dados sobre datas passaram por transformações gerando outras variáveis. Por fim, criamos variáveis associadas aos dados com o objetivo de demonstrar a evolução destes parâmetros como, por exemplo, calculando o percentual de amortização mês a mês para utilização dessa evolução, tanto para a “Amortização” e “Provisão”. Também calculamos o incremento de amortização de provisionamento comparando o início e fim do período estudado. Para cada variáveis categóricas, criamos variáveis *dummy.c*.

3.3.2 Normalização das *Features*

Previamente à aplicação dos modelos, utilizamos a funcionalidade *“Standard Scaler”* abaixo transcrita, com o objetivo de padronizar os valores das variáveis numéricas, tornando-as mais manejáveis para nossos modelos, calculando a média e o desvio padrão no conjunto de treinamento, de modo a poder reaplicar posteriormente a mesma transformação no conjunto de teste. A variação da unidade significa dividir todos os valores pelo desvio padrão. Como resultado obtemos uma distribuição com um desvio padrão igual a 1. A variância é igual a 1 também, porque variância = desvio padrão ao quadrado. E 1 ao quadrado = 1. Por fim, ele torna a média da distribuição aproximadamente 0.

Tabela 3 – Normalização

Split
X = df1.iloc[:, df1.columns.isin(['y'])]
y = df1.iloc[:, df1.columns.isin(['y'])]
X = StandardScaler().fit$_transform(X)$

3.3.3 Balanceamento da Base

Em modelos de *Machine Learning*, problemas de previsão com dados desbalanceados são comuns e ocorrem quando não há uma proporção equilibrada de observações entre as possíveis classes da variável resposta (Referência). De fato, essa falha foi identificada na nossa base de estudo, considerando que em torno de 1% (6.159 registros) foram marcados como AP, nossa variável dependente (*target*).

Para resolver isso, foi necessário realizar multiclassificação dos dados desbalanceados. Optamos por não utilizar métodos em termos de algoritmo, uma vez que modificam os parâmetros e hiperparâmetros do modelo de forma a compensar disparidades, podendo gerar falsos negativos (ou seja, penalizar o modelo se identificar um produto defeituoso como não defeituoso) e indução de tendências. (incluir Referência).

Então, aplicamos balanceamento de dados por *undersample*, reduzindo o volume das classes majoritárias e buscando um equilíbrio numérico em relação ao restante dos dados, especificamente, adequando as classes nas proporções entre "maus pagadores" e "bons pagadores" de 1/1, 3/1, 5/1, 7/1 e 10/1. Ao final, observamos que o *Near Miss* mostrou-se eficiente para balancear os dados, agrupando em algoritmos por subamostragem, observando a distribuição de classes e eliminando aleatoriamente amostras da classe maior. Observamos melhor resultado F1 na amostragem de proporção 5/1 para o *Random Forest*, 1/1 para a Regressão Logística, e 1/1 para o *Gradient Boosting*, conforme demonstrado nos nossos resultados, disposto nos resultados.

Para otimizar nossos resultados, aplicamos o *Troubleshooting imbalanced* com o objetivo de, Além de fracionar as amostras nas proporções acima, assegurarmos a separação na qual o modelo utiliza, necessariamente, 1 amostra de "bons pagadores" para 1 amostra de "maus pagadores". Esse algoritmo foi utilizado para todos os cenários de amostragem acima citados. Para isso, utilizamos o algoritmo:

Tabela 4 – Balanceamento

Troubleshooting imbalanced
$\text{tt}_y = df1[df1['y'] > 0].shape[0]$
$\text{print}(\text{tt}_y)$
$\text{amostra1} = \text{pd.concat}([\text{df1}[\text{df1}['\text{y}']==1].\text{sample}(\text{tt}_y), df1[df1['y']! = 1].sample(tt_y)])$
$\text{amostra3} = \text{pd.concat}([\text{df1}[\text{df1}['\text{y}']==1].\text{sample}(\text{tt}_y), df1[df1['y']! = 1].sample(tt_y * 3)])$
$\text{amostra5} = \text{pd.concat}([\text{df1}[\text{df1}['\text{y}']==1].\text{sample}(\text{tt}_y), df1[df1['y']! = 1].sample(tt_y * 5)])$
$\text{amostra7} = \text{pd.concat}([\text{df1}[\text{df1}['\text{y}']==1].\text{sample}(\text{tt}_y), df1[df1['y']! = 1].sample(tt_y * 7)])$
$\text{amostra10} = \text{pd.concat}([\text{df1}[\text{df1}['\text{y}']==1].\text{sample}(\text{tt}_y), df1[df1['y']! = 1].sample(tt_y * 10)])$

3.3.4 Análise Exploratória

Nosso estudo conta com 592.788 registros que, necessariamente, não eram AP na base inicial e perfaziam uma exposição total de R$ 83.523,37 milhõoes. Destes, observamos um montante de R$ 212,86 milh]oes provisionados, representando uma perda esperada média de 0,25% para a carteira, ou seja, o valor de perda para o produto, projetado para determinado período e calculado com base no histórico de pagamento.

Os contratos apresentaram prazo total entre 458 dias (15 meses) a 12.889 dias (420 meses), coerentes com os prazos máximos exercidos pelas IF para linhas de financiamento habitacional, tendo ocorrência de atrasos entre 0 a 90 dias.

Figura 1 – Evolução Ativos Problemáticos no Período

O gráfico acima demonstra a marcação de AP no tempo, que vai de 0 (em M0) a 6159 (em M6) mostra a tendência de evolução das marcações no período, ainda sob efeitos da crise econômico-financeira sistêmica, decorrente da pandemia COVID-19. Destacamos que um contrato marcado como AP nos meses observados não será desmarcado, portanto, a redução observada entre M3 e M4 foi considerada como contratos liquidados no período.

Reconhecemos que variáveis não financeiras, como indicadores macroeconômicos, decisões estratégicas das IF e decisões das famílias, embora possam influenciar na recuperabilidade do crédito, não estão incorporados às variáveis testadas nos modelos, em que pese alguns autores tenham estudado a utilidade de modelos que incluem informações não financeiras complementares (VEGANZONES; SÉVERIN; CHLIBI, 2021; CIAMPI, 2015)

3.4 Métodos em Machine Leaning

3.4.1 Regressão Logística

Os modelos de regressão logística determinam a importância relativa dos coeficientes na classificação dos devedores em duas classes distintas com base em seu risco de crédito (ou

seja, bons ou maus devedores). Para explicar as não linearidades e relaxar a suposição de normalidade, uma função de verossimilhança sigmóide é normalmente usada (KAMSTRA; KENNEDY; SUAN, 2001).

Dessa forma o *Logit* compreende à relação entre variáveis independentes *Xi* e uma variável dependente *Y*, representando a presença ou ausência de uma característica. Lemeshow, Sturdivant e Jr (2013) descreve o comportamento matemático de *Y* em função dos valores de *Xi* utilizando o método de estimação da máxima verossimilhança, ou seja, estima-se uma relação linear entre variáveis e a probabilidade de pertencer a um ou outro grupo, no caso, adimplente ou inadimplente.

Sun e Lei (2021) utilizaram modelo de regressão logística para construção de modelos para o estudo de alerta financeiro antecipado e alertaram que esse modelo não é particularmente ideal devido ao seu processamento mais aproximado no cálculo, mas uma saída dada a limitação do escopo de aplicação limitado às pré-condições estritas, por exemplo, para previsão do modelo multivariado. Optamos por utilizar o modelo no estudo comparativo ainda que as nossas variáveis são sejam, predominantemente, binárias.

3.4.2 *Random Forest*

Random Forest caracteriza-se como uma combinação de preditores de árvores de tal forma que cada árvore depende dos valores de um vetor aleatório amostrado independentemente e com a mesma distribuição para todas as árvores na floresta. O erro de generalização para florestas converge para um limite à medida que o número de árvores na floresta torna-se grande. O erro de generalização de uma floresta de classificadores de árvores depende da força das árvores individuais na floresta e da correlação entre eles (BREIMAN, 2001).

O modelo foi utilizado em estudos para previsão de insolvências bancárias, em uma amostra de IF sediadas nos Estados Unidos, cujos resultados empíricos indicam que o método de *Random Forest* (RF) apresenta um desempenho preditivo superior, fora da amostra e fora do tempo (PETROPOULOS et al., 2020).

O modelo utilizado por Addo, Guegan e Hassani (2018) é parecido com os nossos, uma vez que focava em questões relacionadas ao uso dos algoritmos para resolver ou atingir um objetivo, estudando estabilidade desses modelos em relação a uma escolha.

3.4.3 *Gradient Boosting*

Gradient Boosting Decision Tree (GBDT) é um algoritmo de *Machine Learning* popular e possui algumas implementações eficazes, como XGBoost e pGBRT. Segundo Ke et al. (2017), embora muitas otimizações de engenharia tenham sido adotadas nessas implementações, a eficiência e a escalabilidade ainda são insatisfatórias quando a dimensão

do recurso é alta e o tamanho dos dados é grande. Uma das principais razões é que, para cada recurso, eles precisam varrer todas as instâncias de dados para estimar o ganho de informações de todos os pontos de divisão possíveis, o que consome muito tempo.

Addo, Guegan e Hassani (2018) concluiu, em seu estudo, que algoritmos baseados em árvore de decisão têm alto desempenho na classificação binária de problemas em comparação com os modelos de *Deep Leanning* considerados naquela análise.

3.5 Validação dos Modelos

3.5.1 *Permutation Feature Importance*

Para fins de explicação global independente dos nossos modelos, utilizamos a técnica *Permutation Feature Importance*, no intuito de obtermos insights sobre os comportamentos. Ele estima e classifica a importância do recurso com base no impacto que cada recurso tem nas previsões do modelo de aprendizado de máquina treinado, medindo o valor preditivo de um recurso para qualquer estimador, classificador ou regressor. Ele faz isso avaliando como o erro de previsão aumenta quando um recurso não está disponível. Para evitar realmente remover recursos e treinar novamente o estimador para cada recurso, o algoritmo embaralha aleatoriamente os valores dos recursos, adicionando ruído ao recurso. Em seguida, o erro de previsão do novo conjunto de dados é comparado com o erro de previsão do conjunto de dados original. Se o modelo depender muito da coluna que está sendo embaralhada para prever com precisão a variável de destino, essa reordenação aleatória causará previsões menos precisas. Se o modelo não depender do recurso para suas previsões, o erro de previsão permanecerá inalterado.

Importante trabalho (PETROPOULOS et al., 2020) analisou individualmente a relevância das variáveis, em relação aos métodos de *Random Forest* e Redes Neurais e indicou que a importância de um indicador em relação a outro é puramente modelado, ou seja, quaisquer conclusões estão fortemente relacionadas com a sofisticação do modelo adjacente utilizado para prever falhas.

3.5.2 *Cross Validation*

Utilizamos a validação cruzada para avaliar a capacidade de generalização dos nossos modelos, a partir do conjunto de dados, objeto do estudo. Esta técnica é amplamente empregada em problemas onde o objetivo da modelagem é a predição. Essa técnica possui diferentes métodos de particionamento, dentre os quais utilizamos o *k-fold*.

O método de validação cruzada denominado *k-fold* consiste em dividir o conjunto total de dados em k subconjuntos mutuamente exclusivos do mesmo tamanho e, a partir daí, um subconjunto é utilizado para teste e os k-1 restantes são utilizados para estimação

dos parâmetros, fazendo-se o cálculo da acurácia do modelo. Este processo é realizado k vezes alternando de forma circular o subconjunto de teste.

O estudo de Triba et al. (2015), analisou como resultado da maneira como o software seleciona os membros dos subconjuntos de calibração e validação, uma simples permutação de linhas do conjunto de dados pode, em vários casos, levar a conclusões contraditórias sobre a significância dos modelos quando uma validação cruzada *k-fold* é usado.

No nosso estudo utilizamos, para fins de particionamento e execução do método, um *K-fold* com k = 10 e, ao final das 10 iterações, calculamos a acurácia sobre os erros encontrados, obtendo assim uma medida mais confiável sobre a capacidade do modelo de representar o processo gerador dos dados.

Ainda, utilizamos o método *Hold-out*, que consiste em dividir o conjunto total de dados em dois subconjuntos mutuamente exclusivos, um para treinamento (estimação dos parâmetros) e outro para teste (validação). Uma proporção muito comum é considerar 2/3 dos dados para treinamento e o 1/3 restante para teste. Após o particionamento, a estimação do modelo é realizada e, posteriormente, os dados de teste são aplicados e o erro de predição calculado. Esta abordagem é indicada quando está disponível uma grande quantidade de dados. Caso o conjunto total de dados seja pequeno, o erro calculado na predição pode sofrer muita variação.

3.6 Avaliação dos Modelos

Outros autores utilizaram métodos diversos para validar o poder discriminante de um modelo supervisionado de predição), dentre os quais observamos maior incidência da escolha (VIEIRA, 2016; PETROPOULOS et al., 2020; BRITO; NETO, 2008) pelo ROC, índice relacionado à curva AUROC, cujos componentes são a especificidade (probabilidade de que estes irão corretamente classificar os não *default*) e sensibilidade (probabilidade de classificar corretamente o *default*).

Para avaliarmos os desempenhos dos modelos, utilizamos, para validação dos modelos, a Curva ROC, além de *Precision*, *Recall*, bem como critérios de erro quadrático médio (RMSE), além da Matriz de Confusão para essas apresentações. Tratam-se de medidas de desempenho amplamente utilizadas, de forma combinada, para comparar o desempenho dos modelos e apresentação de resultados, a exemplo de Addo, Guegan e Hassani (2018) PM Addo, D Guegan, B Hassani - Risks, 2018 e Guegan et al. 2018 em seus estudos.

3.6.1 Matriz de Confusão

Matriz de confusão é uma matriz que traz a informação de todos os acertos e erros do modelo ao prever as classes. Trata-se de uma matriz quadrada em que se compara os verdadeiros valores de uma classificação com os valores preditos através de algum modelo, apurando: Verdadeiro Negativo (VN), Falso Positivo (FP), Falso Negativo (FN) e Verdadeiro Positivo (VP). Sua diagonal é composta pelos acertos do modelo e os demais valores são os erros cometidos.

No nosso estudo, para detecção do *default*, bom desempenho significa uma alta taxa de detecção (verdadeiros positivos), ou seja, quantos ativos foram detectados corretamente, com uma baixa taxa de falsos positivos, isto é, com que frequência um caso de não AP é falsamente detectado como AP.

3.6.2 Acurácia

Acurácia é a métrica mais simples e representa o número de previsões corretas do modelo. Ótima métrica para utilizar quando os dados estão balanceados, dando uma visão geral do quanto ohttps://pt.overleaf.com/project/6168ecd1dfea57369a5b4b12 modelo está identificando as classes corretamente. A Acurácia é medida por: (VP + VN) / (VP + VN + FP + FN).

3.6.3 *Precision*

Precision ou precisão, também conhecida como Valor Preditivo Positivo (VPP), é a métrica que traz a informação da quantidade de observações classificadas como positiva (1) que realmente são positiva. Ou seja, entre todas as observações identificadas como positivas (1), quantas foram identificadas corretamente. A Precisão é medida por: (VP) / (VP + FP).

3.6.4 *Recall*

Recall ou Sensibilidade é a proporção dos Verdadeiros Positivos entre todas as observações que realmente são positivas no seu conjunto de dados. Ou seja, entre todas as observações que são positivas quantas o modelo conseguiu identificar como positiva. Representa a capacidade de um modelo em prever a classe positiva. A Sensibilidade é medida por: (VP) / (VP + VN).

3.6.5 *F1 - Score*

A pontuação F1 de média macro é usada para avaliar o desempenho preditivo de modelos de ML multiclasse, média harmônica entre o *Recall* e a precisão (*Precision*).

Utilizada quando temos classes desbalanceada, que é o caso da base do nosso estudo. O F-Score é medido por: (2* Precisão * Sensibilidade) / (2* Precisão + Sensibilidade).

3.6.6 *Receiver Operating Characteristic* – ROC

A curva ROC (*Receiver Operating Characteristic*) constitui uma técnica bastante útil para validar modelos de risco de crédito e está baseada nos conceitos da sensitividade e da especificidade. A sensitividade é a proporção de acerto na previsão da ocorrência de um evento nos casos em que ele de fato ocorreu. A especificidade é proporção de acerto na previsão da não ocorrência de um evento nos casos em que ele de fato não ocorreu (BRITO; NETO, 2008). Assim, para a construção da Curva ROC, são calculadas a sensitividade e a especificidade para todas as observações da amostra, considerando diferentes pontos de corte do modelo. A curva é obtida registrando em um gráfico "sensitividade" x "1 – especificidade" para os diversos pontos de corte. A área sob a curva mede a capacidade de discriminação do modelo.

No nosso estudo, utilizamos AUC (*"area under the ROC curve"*), uma maneira de resumir a curva ROC em um único valor e agregando todos os limiares da ROC, calculando a "área sob a curva".

A métrica de acurácia (AUROC) foi empregada por **??**) na subamostra de validação para selecionar a parametrização mais eficiente dos modelos Logit, LDA, SVM, NN, tomando um intervalo de valores para "m"(subamostras aleatórias de feições) e para o número de árvores a serem geradas e também para avaliar o modelo de Regressão Logística para classificação de risco de crédito de grandes empresas brasileiras (BRITO; NETO, 2008).

3.6.7 MAE e MSE

No intuito de ajudar a verificar a precisão de erro no final da saída do script, utilizamos MAE (Erro Absoluto Médio) e MSE (Erro quadrático médio), sendo a primeira uma métrica intuitiva pois, faz a verificação em termos médios e sem direção, ou seja, livre de quaisquer valores negativos.

O MSE eleva ao quadrado os erros para que uma diferença de 2 se torne 4, uma diferença de 3 se torne 9, sendo que O algoritmo então continua a somá-los e a média deles. Já o RMSE, é a raiz quadrada do erro quadrático médio (RMSE). Trata-se de uma métrica usada para avaliar o desempenho preditivo de modelos de ML de regressão (PETROPOULOS et al., 2020).

Kamstra, Kennedy e Suan (2001) utilizaram a métrica MSE em modelo de árvore de decisão, para selecionar a parametrização mais eficiente. No modelo otimizado, o erro

MSE foi verificado à medida que o número de árvores aumentava, uma vez que, em gráfico específico, à medida que o número de árvores se aproxima do corte, o MSE se achatava.

Guegan et al. 2018 comparou, em seu estudo, a regra de ajuste e decisão do *Elastic Net* com base na área sob a curva (AUC), bem como critérios de erro quadrático médio (RMSE) e identificou que a comparação entre os resultados obtidos como critérios AUC e os critérios RMSE indicaram que um critério único não é suficiente para conclusão comparativa.

4 Resultados e Discussão

4.1 *Feature Importance*

Para avaliar a qualidade de um modelo de classificação, diversos testes e medidas podem ser utilizados. O primeiro passo é avaliar a significância das variáveis explicativas incluídas no modelo, o que foi realizado por *Feature Importance*, selecionou variáveis mais relevantes, as quais destacamos as tiveram maior poder de determinação da *target*. Destacamos o campo que determina que o contrato foi renegociado, pela última vez, no exercício 2021 e, portanto, indica que este sofreu impacto econômico-financeira ocasionado pela pandemia COVID-19.

Em seguida, foi determinante a varável que traz a quantidade de renegociações para o contrato, ficando demonstrado que a medida negocial, apesar de melhorar a condição econômico-financeira do contrato (por meio da postergação do pagamento ou pela redução do atraso, por exemplo), traz solução apenas paliativa e temporária.

O indicador "Maturidade" foi apurado no nosso estudo em intervalo de 0 a 2,16 e indica a relação entre o valor da dívida vencida em comparação ao saldo devedor do contrato. Calculado em cada base do período estudado.

A variável "Dívida Vencida" da contraparte foi gerada no marco inicial da base estudada e confirmou-se como indicativo de que o contrato se tornará AP, uma vez que o perfil ou a condição do cliente transmite-se à recuperabilidade a nível de contrato.

Destacamos que foram desconsideradas, no nosso estudo, as variáveis relacionadas a atraso/inadimplência, dada a forte correlação com o modelo preditivo de risco de crédito, uma vez que o atraso é a principal evidência desse *default*.

Em caráter excepcional, mantivemos a variável que informa a quantidade de dias em atraso no início do período observado, que configurou como a 5ª em termos de relevância ao final da Feature Importance. Justificamos pelo fato de que identificamos que parte significativa dos contratos retornou à condição inicial de adimplência e ponderamos, ainda, que uma parte considerável de contratos que apresentam atraso até 15 dias representam lapso temporal para reconhecimento sistêmico do efetivo pagamento (risco operacional ou não risco de crédito).

Para a variável "Base de Cálculo", consideramos o percentual de evolução em relação ao mês anterior e observamos que a mesma se mostrou variável relevante, uma vez que aumenta pela ausência de amortização e consequente aumento de encargos.

Inicialmente, para a comparação entre os modelos, havíamos utilizado a variável

"Índice de Provisão", que se refere à perda esperada para o contrato e reflete os níveis de perda da operação, da carteira de crédito e solidez financeira da IF, observado os critérios de aprovisionamento, conforme definido pelos órgãos reguladores.

Em que pese a apuração ser realizada na ocasião da contratação da operação e considerar aspectos de riscos como perfil, cadastro, endividamento, entre outras, optamos por excluí-la uma vez que pode ser impactada no curso do contrato por eventos de atraso, que guarda forte relação com nossa *target*. Para conhecimento do nosso leitor, preliminarmente à exclusão da variável, segue tabela com os resultados comparativos das validações dos modelos de *Machine Learning*, que se mostraram próximos a 100%, levantando a possibilidade de *Overfitting*:

	RF	LR	XG
Acurácia	99.75%	90.96%	99.9%
AUC	99.75%	90.96%	99.9%
F1	99.75%	90.62%	99.9%
Recall	99.86%	87.32%	99.92%
Precision	99.64%	94.18%	99.89%
MSE	0.0	0.09	0.0
MAE	0.0	0.09	0.0

4.2 Resultados das Validações dos Modelos

Para demonstrar a performance dos modelos diante dos critérios utilizados para a variável dependente, *default* ou AP (considerados similares nesse estudo), dividimos a base em amostras conforme proporções a seguir (1/1, 3/1, 5/1, 7/1, 10/1 e Total).

Para o universo "total", tendo em vista limitação operacional, foi considerada uma amostra de 5% do total de observações:

Tabela 5 – Amostra 1/1

Índice	RF	LR	XG
Acurácia	84.81%	78.65%	84.04%
AUC	84.81%	78.65%	84.04%
F1	83.94%	77.62%	83.06%
Recall	79.4%	74.06%	78.27%
Precision	89.04%	81.54%	88.48%
MSE	0.15	0.21	0.16
MAE	0.15	0.21	0.16

Resultados das Validações dos Modelos

Tabela 6 – Amostra 3/1

Índice	RF	LR	XG
Acurácia	98.27%	93.64%	98.38%
AUC	98.52%	89.06%	98.63%
F1	96.63%	86.26%	96.83%
Recall	99.0%	79.89%	99.13%
Precision	94.37%	93.74%	94.63%
MSE	0.02	0.06	0.02
MAE	0.02	0.06	0.02

Resultados das Validações dos Modelos

Tabela 7 – Amostra 5/1

Índice	RF	LR	XG
Acurácia	98.4%	94.42%	98.42%
AUC	98.5%	86.0%	98.54%
F1	95.37%	81.41%	95.41%
Recall	98.65%	73.38%	98.74%
Precision	92.31%	91.42%	92.3%
MSE	0.02	0.06	0.02
MAE	0.02	0.06	0.02

Resultados das Validações dos Modelos

Tabela 8 – Amostra 7/1

Índice	RF	LR	XG
Acurácia	98.35%	95.24%	98.4%
AUC	98.24%	84.28%	98.44%
F1	93.71%	78.53%	93.9%
Recall	98.08%	69.68%	98.49%
Precision	89.71%	89.96%	89.72%
MSE	0.02	0.05	0.02
MAE	0.02	0.05	0.02

Resultados das Validações dos Modelos

Tabela 9 – Amostra 10/1

Índice	RF	LR	XG
Acurácia	98.37%	95.98%	98.39%
AUC	97.44%	82.14%	97.89%
F1	91.47%	74.68%	91.67%
Recall	96.32%	65.21%	97.28%
Precision	87.1%	87.37%	86.68%
MSE	0.02	0.04	0.02
MAE	0.02	0.04	0.02

Resultados das Validações dos Modelos

Tabela 10 – Amostra Total

Índice	RF	LR	XG
Acurácia	99.47%	99.17%	99.45%
AUC	76.26%	68.42%	77.19%
F1	65.29%	45.83%	65.16%
Recall	52.61%	37.09%	54.5%
Precision	86.05%	59.96%	80.99%
MSE	0.01	0.01	0.01
MAE	0.01	0.01	0.01

Resultados das Validações dos Modelos

No cenário acima, verificamos que o modelo *Gradient Boosting* apresentou melhor resultados em 5 dos 6 cenários, com maior ou menor partição, inclusive na amostragem total. Também apresentou resultados melhores em todas as validações, dentre as quais ressaltamos o *F1-Score*. Em contrapartida, a Regressão logística apresentou sempre o pior resultado.

Ressaltamos que a nossa base inicial é relevantemente desbalanceada, em proporção próxima a 1% para a variável dependente, sendo que os resultados satisfatórios nos levam a prever que a técnica *undersampling* utilizada para balanceamento por *Near Miss* é mais adequada para esse tipo de modelo. Apesar de termos utilizados apenas *Near Miss*, (ARIK, 2022) destaca o *Smote* como *oversampling*, como o fator mais crítico para aumentar a precisão do seu modelo, sendo que o método *Smote* foi utilizado no seu estudo para aumentar o número de classes minoritárias amostrais, enquanto o método *Near-Miss* foi usado para reduzir o viés, subdimensionando as classes majoritárias

Também entendemos que o algoritmo *Troubleshooting Imbalanced* contribuiu com resultado satisfatório quando associado ao *Near Miss*, no esforço em dividir as amostras entre "bons pagadores"ou "maus pagadores".

No intuito de verificar a precisão de erro no final da saída do script, utilizamos indicadores MAE (Erro Absoluto Médio) e MSE (Erro Quadrático Médio) e observamos erros em níveis aceitáveis, em maior recorrência para a Regressão Logística, em todas as amostras fracionadas.

Ao testarmos o modelo com a própria amostra utilizada para a estimação dos seus parâmetros, concluímos que o seu desempenho é bom quando, na realidade, ele pode funcionar bem apenas para essas observações. Assim, para avaliar se o modelo mantém o seu poder preditivo para outras amostras provindas da mesma população, são necessários testes para a sua validação. Segundo Jr, Lemeshow e Sturdivant (2013), a validação do modelo é especialmente importante quando ele é usado com a finalidade de previsão de resultados.

Assim, após a definição de modelo mais adequado, aplicamos validação cruzada associada à técnica *Hold-Out*, na proporção 80/20. Consideraremos a seguir os resultados referentes à Amostra Total, conforme Tabela 10 deste estudo.

Para assegurar esse poder preditivo dos modelos, utilizamos a validação cruzada, particularmente, o método de particionamento *K-Fold*, utilizando k = 5 e, ao testarmos a estimação dos seus parâmetros nosso melhor modelo, o *Gradient Boosting* classificou corretamente 99% dos registros da amostra de teste, em termos de acurácia.

Seguindo nosso foco na performance, rodamos o algoritmo *Cross Predict* com *Hold-Out*, ou seja, utilizando os 20% restantes da nossa base, agora para teste, considerando *K-fold* com k=5, quando chegamos ao resultado F1 = 67,41%, conforme Tabela 11.

Tabela 11 – Resultados Cross Predict

F1 Score	**67.41573033707866**
P (Precision)	78.94736842105263
R (Recall)	58.82352941176471
A (Acurácia)	79.34379529211856

Resultados para o modelo selecionado: Gradient Boosting

Em complemento, testamos também o desempenho do nosso modelo, ainda por meio do *Cross Predict*, agora utilizando o algoritmo *K-Fold* (k =5) e identificamos uma leve piora nos resultados, com redução do F1, de 67,41% para 64,40%, conforme Tabela 12.

Tabela 12 – Resultados Cross Predict: k = 5

F1 Score	**64.40677966101694**
P (Precision)	79.58115183246073
R (Recall)	54.092526690391466
A (Acurácia)	76.9799277893378

Resultados para o modelo selecionado: Gradient Boosting

Com objetivo de maximizar a performance do nosso modelo selecionado, utilizamos ainda a aplicação de hiperparâmetros, ou seja, parâmetros de alto impacto que controlam o referido processo de aprendizado. Uma vez que são ajustáveis, desempenham um papel no alcance do máximo desempenho do modelo em um tempo razoável. Assim, para nosso melhor modelo, após essa recalibragem, observamos melhora relevante no indicador F1, principal referência considerada no nosso estudo, que passou para 98%.

No nosso estudo utilizamos o RandomizedSearchCV um módulo do *Scikit Learn* no qual, ao contrário do GridSearchCV, nem todos os valores de parâmetro são testados, mas um número fixo de configurações de parâmetro é amostrado das distribuições especificadas. O número de configurações de parâmetro que são tentadas é dado por n_iter e é altamente recomendável usar distribuições contínuas para parâmetros contínuos.

Tabela 13 – Hiperparâmetro

Hiperparâmetro Randomized Search CV
$\text{clf}_g v = RandomizedSearchCV(clf, param, scoring =' f1'$, cv=kfold, refit=True, verbose=1, $\text{n}_i ter = 20)$
$\text{clf}_g v.fit(X_t est, y_t est)$

Assim, diante dos resultados gerados pelo modelo, propomos aplicar o modelo *Gradient Boosting* a novos conjuntos de dados e realizar as previsões de marcações de AP, permitindo proativamente aos bancos a tomada de decisões estratégicas referentes a determinada carteira de crédito, com foco na solvência e liquidez da instituição.

5 Considerações Finais

O objetivo deste estudo supervisionado foi identificar modelos de classificação de risco de crédito para clientes pessoas físicas em instituições financeiras que atuam no Brasil, comparando técnicas de Regressão Logística aos modelos de *Machine Learning*, *Random Forest* e *Gradient Boosting*. Por meio de índices e variáveis explicativas, o modelo proposto possibilita a previsão da ocorrência de eventos de *default* com antecedência, para tomada de medidas antes de um contrato de tornar Ativo Problemático, alcançando significativo índice de acerto. Os modelos utilizaram variáveis previsoras que acreditamos ter potencial para discriminar bons e maus clientes, tendo o *Gradient Boosting* apresentado melhor desempenho preditivo, com 99,45% de acerto inicial, quando verificada a acurácia, e de 65,83% quanto ao índice *F1-Score*.

Após a aplicação de algoritmos para otimização, foi possível potencializar esses resultados, que chegaram a 98% para o índice *F1-Score* quando considerado o cenário de aplicação de 5% da Amostra Total.

Destacamos que tais resultados favoráveis foram obtidos, exclusivamente, a partir de dados dos contratos vigentes, ou seja, partir de informações internas dos bancos, criadas para fins de controle e gerenciamento das exposições, portanto, acessíveis e cuja qualidade está associada à eficiência dos respectivos controles.

Assim, há indicativo de eficiência dos modelos sem utilização dos dados pessoais dos clientes, cujas informações possuem limitações relacionadas à defasagem e/ou restrições legais decorrentes da Lei Geral de Proteção de Dados (LGPD). Considerando que não se tratava do objetivo do nosso trabalho, recomendamos que sejam realizados tais estudos futuros, voltados para o acompanhamento da situação de créditos já vigentes e, portanto, com base na performance histórica do contrato, que poderão trazer ganhos relevantes aos bancos em termos de eficiência e segurança da informação.

Outro ponto observado, relacionado à necessidade mínima de ajustes nos dados da base real utilizada (*Missing Value*, *Outliers*, inconsistências) refletem, ao nosso ver, a eficiência dos controles que abrangem os referidos registros por meio dos sistemas corporativos. Complementamos tal percepção com fato de que a base de estudo contempla um produto específico, que sugere previsão de perfil para os respectivos tomadores do crédito, otimizando o nível de acerto. Considerando que não foram realizados estudos para mensurar essa correlação, recomenda-se a realização de estudos posteriores nesse sentido.

Pode-se inferir que as aplicações proporcionariam redução no tempo de desenvolvimento de modelos e, consequentemente, no acionamento dos clientes e na aplicação das medidas negociais com foco na mitigação do risco de *default*.

Além da tempestividade, possível redução de custos para as instituições na aplicação, uma vez que a metodologias utilizadas se encontram acessíveis, consolidados e populares e extremamente testados para diversos fins.

Por fim, ressalte-se que os modelos derivados empiricamente, *Random Forest* e *Gradient Boosting* desenvolvidos em *Machine Learning*, apresentaram expressivo nível de acerto nas classificações, verificados de forma recorrente neste estudo e em inúmeros trabalhos desenvolvidos, sugerindo que o evento de risco de crédito pode ser previsto com sucesso, uma vez que foi possível gerar evidências e aprendizados, para obter conclusões.

Referências

ADDO, P. M.; GUEGAN, D.; HASSANI, B. Credit risk analysis using machine and deep learning models. *Risks*, Multidisciplinary Digital Publishing Institute, v. 6, n. 2, p. 38, 2018. Citado 5 vezes nas páginas 23, 29, 37, 38 e 39.

ARIK, A. O. *A robust Gradient boosting model based on SMOTE and NEAR MISS methods for intrusion detection in imbalanced data sets*. Dissertação (Mestrado) — Işık Üniversitesi, 2022. Citado na página 46.

BARBOZA, F.; KIMURA, H.; ALTMAN, E. Machine learning models and bankruptcy prediction. *Expert Systems with Applications*, Elsevier, v. 83, p. 405–417, 2017. Citado 2 vezes nas páginas 23 e 33.

BESSIS, J. *Risk management in banking*. [S.l.]: John Wiley & Sons, 2011. Citado na página 25.

BLANCK, H. L. et al. Capital de risco e startups: modelo de suporte na tomada de decisão com aprendizado de máquina. 2020. Citado na página 23.

BOHN, D. D. Estudo da aplicabilidade de créditos massificados como ferramenta de apoio na análise e concessão de crédito para a gestão de resultados na cooperativa sicredi união rs. Citado na página 32.

BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001. Citado na página 37.

BRITO, G. A. S.; NETO, A. A. Modelo de classificação de risco de crédito de empresas. *Revista Contabilidade & Finanças*, SciELO Brasil, v. 19, n. 46, p. 18–29, 2008. Citado 5 vezes nas páginas 26, 29, 33, 39 e 41.

BRITO, J. T. O agravamento do risco de crédito devido aos problemas econômicos e sociais da covid-19. Universidade Federal de São Paulo, 2022. Citado na página 26.

CIAMPI, F. Corporate governance characteristics and default prediction modeling for small enterprises. an empirical analysis of italian firms. *Journal of Business Research*, Elsevier, v. 68, n. 5, p. 1012–1025, 2015. Citado na página 36.

FITZPATRICK, T.; MUES, C. An empirical comparison of classification algorithms for mortgage default prediction: evidence from a distressed mortgage market. *European Journal of Operational Research*, v. 249, n. 2, p. 427–439, 2016. ISSN 0377-2217. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0377221715008383>. Citado na página 25.

HAMORI SHIGEYUKI E KAWAI, M. e. K. T. e. M. Y. e. W. C. *Ensemble learning ou deep learning? Aplicação para análise de risco padrão*. [S.l.]: Instituto Multidisciplinar de Publicação Digital. Citado na página 23.

HELBEKKMO, H. et al. Enterprise risk management—shaping the risk revolution. *New York: McKinsey & Co., Available online: www. rmahq. org (accessed on 18 June 2018)*, 2013. Citado na página 23.

JR, D. W. H.; LEMESHOW, S.; STURDIVANT, R. X. *Applied logistic regression*. [S.l.]: John Wiley & Sons, 2013. v. 398. Citado na página 46.

KAMSTRA, M.; KENNEDY, P.; SUAN, T.-K. Combining bond rating forecasts using logit. *Financial Review*, Wiley Online Library, v. 36, n. 2, p. 75–96, 2001. Citado 2 vezes nas páginas 37 e 41.

KE, G. et al. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, v. 30, 2017. Citado na página 37.

KERAMATI, A.; YOUSEFI, N. A proposed classification of data mining techniques in credit scoring. In: *the Proceeding of 2011 International Conference of Industrial Engineering and Operations Management, Kuala Lumpur, Malaysia, Jurnal*. [S.l.: s.n.], 2011. p. 22–4. Citado na página 23.

LEMESHOW, S.; STURDIVANT, R. X.; JR, D. W. H. *Applied logistic regression*. [S.l.]: John Wiley & Sons, 2013. Citado na página 37.

LEO, M.; SHARMA, S.; MADDULETY, K. Machine learning in banking risk management: A literature review. *Risks*, Multidisciplinary Digital Publishing Institute, v. 7, n. 1, p. 29, 2019. Citado na página 23.

LIEBERGEN, B. V. et al. Machine learning: a revolution in risk management and compliance? *Journal of Financial Transformation*, Capco Institute, v. 45, p. 60–67, 2017. Citado na página 23.

PETROPOULOS, A. et al. Predicting bank insolvencies using machine learning techniques. *International Journal of Forecasting*, Elsevier, v. 36, n. 3, p. 1092–1113, 2020. Citado 5 vezes nas páginas 33, 37, 38, 39 e 41.

SICSÚ, A. L. *Credit Scoring: desenvolvimento, implantação, acompanhamento*. [S.l.]: Blucher, 2010. Citado 2 vezes nas páginas 25 e 30.

SILVA, F. G. d. A. Risco de crédito bancário e informação assimétrica: teoria e evidência. 2004. Citado na página 25.

SUN, X.; LEI, Y. Research on financial early warning of mining listed companies based on bp neural network model. *Resources Policy*, Elsevier, v. 73, p. 102223, 2021. Citado na página 37.

TRIBA, M. N. et al. Pls/opls models in metabolomics: the impact of permutation of dataset rows on the k-fold cross-validation quality parameters. *Molecular BioSystems*, Royal Society of Chemistry, v. 11, n. 1, p. 13–19, 2015. Citado na página 39.

TRIVEDI, S. K. A study on credit scoring modeling with different feature selection and machine learning approaches. *Technology in Society*, Elsevier, v. 63, p. 101413, 2020. Citado na página 26.

VEGANZONES, D.; SÉVERIN, E.; CHLIBI, S. Influence of earnings management on forecasting corporate failure. *International Journal of Forecasting*, Elsevier, 2021. Citado na página 36.

VIEIRA, J. R. d. C. Predição do bom e do mau pagador no programa minha casa, minha vida. 2016. Citado 5 vezes nas páginas 23, 24, 29, 33 e 39.

WALSH, B.; VOLINI, E. Rewriting the rules for the digital age: 2017 deloitte global human capital trends. Deloitte University Press, 2017. Citado na página 23.

ZHOU, L.; WANG, H. Loan default prediction on large imbalanced data using random forests. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, v. 10, n. 6, p. 1519–1525, 2012. Citado na página 23.

Apêndices

APÊNDICE A – Códigos Phyton

Lib

```
import pandas as pd
import numpy as np

from sklearn.preprocessing import StandardScaler
from fast_ml.model_development import train_valid_test_split

import xgboost
from sklearn.model_selection import train_test_split, cross_val_score,
TimeSeriesSplit, cross_val_predict, KFold, StratifiedKFold,
GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
from sklearn.feature_selection import SelectKBest, f_classif, chi2

from sklearn.metrics import accuracy_score, recall_score, confusion_matrix,
f1_score, precision_score, precision_recall_curve, auc, roc_curve,
classification_report,
precision_recall_fscore_support, mean_absolute_error,
roc_auc_score,mean_squared_error

# Plots
# ================================================================================
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
plt.rcParams['lines.linewidth'] = 1.5
%matplotlib inline

#visualização de dados
import seaborn as sns
sns.set_style("darkgrid")

#Loading Data
df1 = pd.read_csv('drive/MyDrive/Colab Notebooks/Trabalho/
```

```
Base_Inicial_Tratamento_Ju.csv'
                ,sep=';'
                # ,error_bad_lines=False
                ,low_memory=False
                ,decimal=','
                #  ,parse_dates=['dDateTime']
                #  ,nrows=10000
                #,dtype={'AM_REFERENCIA':int, 'PC_MES_1_BASE_CALCULO':float,
                'PC_MES_2_BASE_CALCULO': float}
                 )

from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount,
call drive.mount("/content/drive", force_remount=True).

df = df1[:]

df1[['IC_MES_1_APR','IC_MES_1_APR_INADIMPLENTE','IC_MES_1_APR_RATING_E_PIOR',
'IC_MES_1_APR_RESTRUTURADO','IC_MES_1_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_1_APR_RESTRUTURADO_INADIMPLENTE','IC_MES_1_GARANTIA_REAL',
'IC_MES_2_APR','IC_MES_2_APR_INADIMPLENTE','IC_MES_2_APR_RATING_E_PIOR',
'IC_MES_2_APR_RESTRUTURADO','IC_MES_2_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_2_APR_RESTRUTURADO_INADIMPLENTE','IC_MES_2_GARANTIA_REAL',
'IC_MES_3_APR','IC_MES_3_APR_INADIMPLENTE','IC_MES_3_APR_RATING_E_PIOR',
'IC_MES_3_APR_RESTRUTURADO','IC_MES_3_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_3_APR_RESTRUTURADO_INADIMPLENTE','IC_MES_3_GARANTIA_REAL',
'IC_MES_4_APR','IC_MES_4_APR_INADIMPLENTE','IC_MES_4_APR_RATING_E_PIOR',
'IC_MES_4_APR_RESTRUTURADO','IC_MES_4_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_4_APR_RESTRUTURADO_INADIMPLENTE',
'IC_MES_4_GARANTIA_REAL',
'IC_MES_5_APR','IC_MES_5_APR_INADIMPLENTE','IC_MES_5_APR_RATING_E_PIOR',
'IC_MES_5_APR_RESTRUTURADO','IC_MES_5_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_5_APR_RESTRUTURADO_INADIMPLENTE','IC_MES_5_GARANTIA_REAL',
'IC_MES_6_APR','IC_MES_6_APR_INADIMPLENTE','IC_MES_6_APR_RATING_E_PIOR',
'IC_MES_6_APR_RESTRUTURADO','IC_MES_6_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_6_APR_RESTRUTURADO_INADIMPLENTE','IC_MES_6_GARANTIA_REAL',]] = df1[[
'IC_MES_1_APR','IC_MES_1_APR_INADIMPLENTE','IC_MES_1_APR_RATING_E_PIOR',
```

```
'IC_MES_1_APR_RESTRUTURADO','IC_MES_1_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_1_APR_RESTRUTURADO_INADIMPLENTE','IC_MES_1_GARANTIA_REAL',
'IC_MES_2_APR','IC_MES_2_APR_INADIMPLENTE','IC_MES_2_APR_RATING_E_PIOR',
'IC_MES_2_APR_RESTRUTURADO','IC_MES_2_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_2_APR_RESTRUTURADO_INADIMPLENTE','IC_MES_2_GARANTIA_REAL',
'IC_MES_3_APR','IC_MES_3_APR_INADIMPLENTE','IC_MES_3_APR_RATING_E_PIOR',
'IC_MES_3_APR_RESTRUTURADO','IC_MES_3_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_3_APR_RESTRUTURADO_INADIMPLENTE','IC_MES_3_GARANTIA_REAL',
'IC_MES_4_APR','IC_MES_4_APR_INADIMPLENTE','IC_MES_4_APR_RATING_E_PIOR',
'IC_MES_4_APR_RESTRUTURADO','IC_MES_4_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_4_APR_RESTRUTURADO_INADIMPLENTE','IC_MES_4_GARANTIA_REAL',
'IC_MES_5_APR','IC_MES_5_APR_INADIMPLENTE','IC_MES_5_APR_RATING_E_PIOR',
'IC_MES_5_APR_RESTRUTURADO','IC_MES_5_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_5_APR_RESTRUTURADO_INADIMPLENTE','IC_MES_5_GARANTIA_REAL',
'IC_MES_6_APR','IC_MES_6_APR_INADIMPLENTE','IC_MES_6_APR_RATING_E_PIOR',
'IC_MES_6_APR_RESTRUTURADO','IC_MES_6_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_6_APR_RESTRUTURADO_INADIMPLENTE','IC_MES_6_GARANTIA_REAL',]]*1

#EDA
df1.shape
(597101, 127)

df1.describe()
8 rows × 58 columns

print(df1['IC_MES_1_APR'].value_counts())
print((df1['IC_MES_1_APR'].value_counts()
[1]/df1['IC_MES_1_APR'].value_counts()[0])*100,'%')

0    592788
1      3703
Name: IC_MES_1_APR, dtype: int64
0.6246752633319164 %

#Feature Engineering
# df1['y'] = df1[['IC_MES_1_APR','IC_MES_1_APR_INADIMPLENTE',
'IC_MES_1_APR_RATING_E_PIOR',
'IC_MES_1_APR_RESTRUTURADO',
'IC_MES_1_APR_RECUPERACAO_JUDICIAL_FALENCIA',
```

```
'IC_MES_1_APR_RESTRUTURADO_INADIMPLENTE',
#                  'IC_MES_2_APR','IC_MES_2_APR_INADIMPLENTE',
'IC_MES_2_APR_RATING_E_PIOR',
'IC_MES_2_APR_RESTRUTURADO',
'IC_MES_2_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_2_APR_RESTRUTURADO_INADIMPLENTE',
#                  'IC_MES_3_APR','IC_MES_3_APR_INADIMPLENTE',
'IC_MES_3_APR_RATING_E_PIOR',
'IC_MES_3_APR_RESTRUTURADO',
'IC_MES_3_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_3_APR_RESTRUTURADO_INADIMPLENTE',
#                  'IC_MES_4_APR','IC_MES_4_APR_INADIMPLENTE',
'IC_MES_4_APR_RATING_E_PIOR',
'IC_MES_4_APR_RESTRUTURADO',
'IC_MES_4_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_4_APR_RESTRUTURADO_INADIMPLENTE',
#                  'IC_MES_5_APR','IC_MES_5_APR_INADIMPLENTE',
'IC_MES_5_APR_RATING_E_PIOR',
'IC_MES_5_APR_RESTRUTURADO',
'IC_MES_5_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_5_APR_RESTRUTURADO_INADIMPLENTE',
#                  'IC_MES_6_APR','IC_MES_6_APR_INADIMPLENTE',
'IC_MES_6_APR_RATING_E_PIOR',
'IC_MES_6_APR_RESTRUTURADO',
'IC_MES_6_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_6_APR_RESTRUTURADO_INADIMPLENTE']].sum()
df1.rename(columns={'IC_MES_1_APR':'y'},inplace=True)
df1.fillna(0, inplace=True)
# df1.drop(columns=['DT_MES_1_MAIOR_RENEGOCIACAO_DIA',
'DT_MES_2_MAIOR_RENEGOCIACAO_DIA',
'DT_MES_3_MAIOR_RENEGOCIACAO_DIA',
'DT_MES_4_MAIOR_RENEGOCIACAO_DIA',
'DT_MES_5_MAIOR_RENEGOCIACAO_DIA',
'DT_MES_6_MAIOR_RENEGOCIACAO_DIA'],
inplace=True)

df1['PC_RATING_AVALIACAO_INICIAL'] =
pd.to_numeric(df1['PC_RATING_AVALIACAO_INICIAL']
.str.replace(',','.').str.replace('%',''), errors='ignore')
```

```
df1['DT_MES_1_RENEGOCIACAO'] =
df1['DT_MES_1_RENEGOCIACAO'].apply(lambda x: 0 if x == 0 else 1)
# df1['DT_MES_1_RENEGOCIACAO_DIA'] = df1['DT_MES_1_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_1_RENEGOCIACAO_MES'] = df1['DT_MES_1_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[3:5]))
# df1['DT_MES_1_RENEGOCIACAO_ANO'] = df1['DT_MES_1_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[6:10]))
# df1['DT_MES_1_ULTIMA_RENEGOCIACAO_DIA'] =
df1['DT_MES_1_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_1_ULTIMA_RENEGOCIACAO_MES'] =
df1['DT_MES_1_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[3:5]))
df1['DT_MES_1_ULTIMA_RENEGOCIACAO_COVID'] =
df1['DT_MES_1_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else 0 if int(str(x)[6:10])!=2021 else 1)
# df1['DT_MES_1_MAIOR_RENEGOCIACAO_DIA'] =
df1['DT_MES_1_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_1_MAIOR_RENEGOCIACAO_MES'] =
df1['DT_MES_1_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[3:5]))
# df1['DT_MES_1_MAIOR_RENEGOCIACAO_ANO'] =
df1['DT_MES_1_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[6:10]))

df1['DT_MES_2_RENEGOCIACAO'] =
df1['DT_MES_2_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else 1)
# df1['DT_MES_2_RENEGOCIACAO_DIA'] =
df1['DT_MES_2_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_2_RENEGOCIACAO_MES'] = df1['DT_MES_2_RENEGOCIACAO']

# df1['DT_MES_2_RENEGOCIACAO_ANO'] = df1['DT_MES_2_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[6:10]))
# df1['DT_MES_2_ULTIMA_RENEGOCIACAO_DIA'] =
```

```
df1['DT_MES_2_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_2_ULTIMA_RENEGOCIACAO_MES'] =
df1['DT_MES_2_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[3:5]))
df1['DT_MES_2_ULTIMA_RENEGOCIACAO_COVID'] =
df1['DT_MES_2_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else 0 if int(str(x)[6:10])!=2021 else 1)
# df1['DT_MES_2_MAIOR_RENEGOCIACAO_DIA'] =
df1['DT_MES_2_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_2_MAIOR_RENEGOCIACAO_MES']=
df1['DT_MES_2_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[3:5]))
# df1['DT_MES_2_MAIOR_RENEGOCIACAO_ANO'] =
df1['DT_MES_2_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[6:10]))

df1['DT_MES_3_RENEGOCIACAO'] =
df1['DT_MES_3_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else 1)
# df1['DT_MES_3_RENEGOCIACAO_DIA'] =
df1['DT_MES_3_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_3_RENEGOCIACAO_MES'] =
df1['DT_MES_3_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[3:5]))
# df1['DT_MES_3_RENEGOCIACAO_ANO'] =
df1['DT_MES_3_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[6:10]))
# df1['DT_MES_3_ULTIMA_RENEGOCIACAO_DIA'] =
df1['DT_MES_3_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_3_ULTIMA_RENEGOCIACAO_MES'] =
df1['DT_MES_3_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[3:5]))
df1['DT_MES_3_ULTIMA_RENEGOCIACAO_COVID'] =
df1['DT_MES_3_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0
```

```
else 0 if int(str(x)[6:10])!=2021 else 1)
# df1['DT_MES_3_MAIOR_RENEGOCIACAO_DIA']  =
df1['DT_MES_3_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_3_MAIOR_RENEGOCIACAO_MES']  =
df1['DT_MES_3_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[3:5]))
# df1['DT_MES_3_MAIOR_RENEGOCIACAO_ANO']  =
df1['DT_MES_3_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[6:10]))

df1['DT_MES_4_RENEGOCIACAO'] = df1['DT_MES_4_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else 1)
# df1['DT_MES_4_RENEGOCIACAO_DIA'] =
df1['DT_MES_4_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_4_RENEGOCIACAO_MES'] =
df1['DT_MES_4_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[3:5]))
# df1['DT_MES_4_RENEGOCIACAO_ANO']        = df1['DT_MES_4_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[6:10]))
# df1['DT_MES_4_ULTIMA_RENEGOCIACAO_DIA'] =
df1['DT_MES_4_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_4_ULTIMA_RENEGOCIACAO_MES'] =
df1['DT_MES_4_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[3:5]))
df1['DT_MES_4_ULTIMA_RENEGOCIACAO_COVID'] =
df1['DT_MES_4_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else 0 if int(str(x)[6:10])


!=2021 else 1)
# df1['DT_MES_4_MAIOR_RENEGOCIACAO_DIA']  =
df1['DT_MES_4_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_4_MAIOR_RENEGOCIACAO_MES']  =
df1['DT_MES_4_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[3:5]))
```

```
# df1['DT_MES_4_MAIOR_RENEGOCIACAO_ANO']  =
df1['DT_MES_4_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[6:10]))

df1['DT_MES_5_RENEGOCIACAO'] = df1['DT_MES_5_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else 1)
# df1['DT_MES_5_RENEGOCIACAO_DIA']         =
df1['DT_MES_5_RENEGOCIACAO'].apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_5_RENEGOCIACAO_MES']         =
df1['DT_MES_5_RENEGOCIACAO'].apply(lambda x: 0 if x == 0 else int(str(x)[3:5]))
# df1['DT_MES_5_RENEGOCIACAO_ANO']         =
df1['DT_MES_5_RENEGOCIACAO'].apply(lambda x: 0 if x == 0 else int(str(x)[6:10]))
# df1['DT_MES_5_ULTIMA_RENEGOCIACAO_DIA'] =
,df1['DT_MES_5_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_5_ULTIMA_RENEGOCIACAO_MES'] =
df1['DT_MES_5_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[3:5]))
df1['DT_MES_5_ULTIMA_RENEGOCIACAO_COVID'] =
df1['DT_MES_5_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else 0
if int(str(x)[6:10])!=2021 else 1)
# df1['DT_MES_5_MAIOR_RENEGOCIACAO_DIA']  =
df1['DT_MES_5_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_5_MAIOR_RENEGOCIACAO_MES']  =
df1['DT_MES_5_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[3:5]))
# df1['DT_MES_5_MAIOR_RENEGOCIACAO_ANO']  =
df1['DT_MES_5_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[6:10]))

df1['DT_MES_6_RENEGOCIACAO'] = df1['DT_MES_6_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else 1)
# df1['DT_MES_6_RENEGOCIACAO_DIA']         =
df1['DT_MES_6_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_6_RENEGOCIACAO_MES']         =
df1['DT_MES_6_RENEGOCIACAO']
```

```
.apply(lambda x: 0 if x == 0 else int(str(x)[3:5]))
# df1['DT_MES_6_RENEGOCIACAO_ANO']         =
df1['DT_MES_6_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[6:10]))
# df1['DT_MES_6_ULTIMA_RENEGOCIACAO_DIA'] =
df1['DT_MES_6_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_6_ULTIMA_RENEGOCIACAO_MES'] =
df1['DT_MES_6_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[3:5]))
df1['DT_MES_6_ULTIMA_RENEGOCIACAO_COVID'] =
df1['DT_MES_6_ULTIMA_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else 0
if int(str(x)[6:10])!=2021 else 1)
# df1['DT_MES_6_MAIOR_RENEGOCIACAO_DIA']  =
df1['DT_MES_6_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[:2]))
# df1['DT_MES_6_MAIOR_RENEGOCIACAO_MES']  =
df1['DT_MES_6_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[3:5]))
# df1['DT_MES_6_MAIOR_RENEGOCIACAO_ANO']  =
df1['DT_MES_6_MAIOR_RENEGOCIACAO']
.apply(lambda x: 0 if x == 0 else int(str(x)[6:10]))

df1.drop(columns=
['DT_MES_1_ULTIMA_RENEGOCIACAO', 'DT_MES_1_MAIOR_RENEGOCIACAO',
'DT_MES_2_ULTIMA_RENEGOCIACAO', 'DT_MES_2_MAIOR_RENEGOCIACAO',
'DT_MES_3_ULTIMA_RENEGOCIACAO', 'DT_MES_3_MAIOR_RENEGOCIACAO',
'DT_MES_4_ULTIMA_RENEGOCIACAO', 'DT_MES_4_MAIOR_RENEGOCIACAO',
'DT_MES_5_ULTIMA_RENEGOCIACAO', 'DT_MES_5_MAIOR_RENEGOCIACAO',
'DT_MES_6_ULTIMA_RENEGOCIACAO', 'DT_MES_6_MAIOR_RENEGOCIACAO',
'CO_MES_1_ATIVO_PROBLEMATICO','CO_MES_2_ATIVO_PROBLEMATICO',
'CO_MES_3_ATIVO_PROBLEMATICO','CO_MES_4_ATIVO_PROBLEMATICO',
'CO_MES_5_ATIVO_PROBLEMATICO','CO_MES_6_ATIVO_PROBLEMATICO',
'IC_MES_1_APR_INADIMPLENTE','IC_MES_1_APR_RATING_E_PIOR',
'IC_MES_1_APR_RESTRUTURADO','IC_MES_1_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_1_APR_RESTRUTURADO_INADIMPLENTE',
'IC_MES_2_APR','IC_MES_2_APR_INADIMPLENTE',
'IC_MES_2_APR_RATING_E_PIOR','IC_MES_2_APR_RESTRUTURADO',
```

```
'IC_MES_2_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_2_APR_RESTRUTURADO_INADIMPLENTE',
'IC_MES_3_APR','IC_MES_3_APR_INADIMPLENTE',
'IC_MES_3_APR_RATING_E_PIOR','IC_MES_3_APR_RESTRUTURADO',
'IC_MES_3_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_3_APR_RESTRUTURADO_INADIMPLENTE',
'IC_MES_4_APR','IC_MES_4_APR_INADIMPLENTE',
'IC_MES_4_APR_RATING_E_PIOR','IC_MES_4_APR_RESTRUTURADO',
'IC_MES_4_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_4_APR_RESTRUTURADO_INADIMPLENTE',
    'IC_MES_5_APR','IC_MES_5_APR_INADIMPLENTE'
    'IC_MES_5_APR_RATING_E_PIOR','IC_MES_5_APR_RESTRUTURADO',
    'IC_MES_5_APR_RECUPERACAO_JUDICIAL_FALENCIA',
    'IC_MES_5_APR_RESTRUTURADO_INADIMPLENTE',
'IC_MES_6_APR','IC_MES_6_APR_INADIMPLENTE',
'IC_MES_6_APR_RATING_E_PIOR','IC_MES_6_APR_RESTRUTURADO',
'IC_MES_6_APR_RECUPERACAO_JUDICIAL_FALENCIA',
'IC_MES_6_APR_RESTRUTURADO_INADIMPLENTE',
    'NU_MES_1_MESES_NAO_PAGO','NU_MES_2_MESES_NAO_PAGO',
    'NU_MES_3_MESES_NAO_PAGO','NU_MES_4_MESES_NAO_PAGO',
    'NU_MES_5_MESES_NAO_PAGO','NU_MES_6_MESES_NAO_PAGO'
                ], inplace=True)
df1.drop(columns=['NU_CONTRATO_SIAPC', 'NU_CPF_CNPJ'], inplace=True)

df1['ANO'] = df1['AM_REFERENCIA'].apply(lambda x: int(float(str(x)[:4])))
df1['MES'] = df1['AM_REFERENCIA'].apply(lambda x: 0 if (str(x)[4:6]).strip()
=='' else int(str(x)[4:6]))

df1.drop(columns=['AM_REFERENCIA'], inplace=True)

df1['PC_MES_1_BASE_CALCULO'] =
pd.to_numeric(df1['PC_MES_1_BASE_CALCULO'].str.replace(',','.'), errors='coerce')
df1['PC_MES_2_BASE_CALCULO'] =
pd.to_numeric(df1['PC_MES_2_BASE_CALCULO'].str.replace(',','.'), errors='coerce')
df1['PC_MES_3_BASE_CALCULO'] =
pd.to_numeric(df1['PC_MES_3_BASE_CALCULO'].str.replace(',','.'), errors='coerce')
df1['PC_MES_4_BASE_CALCULO'] =
pd.to_numeric(df1['PC_MES_4_BASE_CALCULO'].str.replace(',','.'),errors='coerce')
df1['PC_MES_5_BASE_CALCULO'] =
```

```
pd.to_numeric(df1['PC_MES_5_BASE_CALCULO'].str.replace(',','.'), errors='coerce')
df1['PC_MES_6_BASE_CALCULO'] =
pd.to_numeric(df1['PC_MES_6_BASE_CALCULO'].str.replace(',','.'), errors='coerce')

df1.dropna(inplace=True)


#Correlation
df1.shape
(593542, 73)

df1.corr()
73 rows × 73 columns


#Split
X = df1.iloc[:, ~df1.columns.isin(['y'])]
y = df1.iloc[:, df1.columns.isin(['y'])]

X = StandardScaler().fit_transform(X)


#Troubleshooting imbalanced
tt_y = df1[df1['y']>0].shape[0]
print(tt_y)

amostra1 = pd.concat([df1[df1['y']==1].sample(tt_y),
df1[df1['y']!=1].sample(tt_y)])
amostra3 = pd.concat([df1[df1['y']==1].sample(tt_y),
df1[df1['y']!=1].sample(tt_y*3)])
amostra5 = pd.concat([df1[df1['y']==1].sample(tt_y),
df1[df1['y']!=1].sample(tt_y*5)])
amostra7 = pd.concat([df1[df1['y']==1].sample(tt_y),
df1[df1['y']!=1].sample(tt_y*7)])
amostra10 = pd.concat([df1[df1['y']==1].sample(tt_y),
df1[df1['y']!=1].sample(tt_y*10)])


#Function
```

```
def evaluate_model_cross(cv, X, y, tp):
  if(tp == 'RF'):
    clf = RandomForestClassifier()
  elif(tp=='LR'):
    clf = LogisticRegression()
  elif(tp=='XG'):
    clf = xgboost.XGBClassifier()

  y_pred = cross_val_predict(clf, X, y, cv=cv, n_jobs=-1)
  acc = round(accuracy_score(y,y_pred) *100,2)
  auc = round(roc_auc_score(y, y_pred)*100,2)
  mse = round(mean_squared_error(y,y_pred),2)
  mae = round(mean_absolute_error(y,y_pred),2)
  f1  = round(f1_score(y,y_pred)*100,2)
  rec = round(recall_score(y,y_pred)*100,2)
  pre = round(precision_score(y,y_pred)*100,2)

  df = pd.DataFrame({tp: [str(acc) + '%',str(auc)+ '%', str(f1) +'%',
  str(rec) +'%', str(pre) +'%',str(mse) +'', str(mae) +'']},
                    index = ['Acurácia','AUC',
                    'F1','Recall','Precision','MSE','MAE'])
  return df


#1/1
dfPredRF = evaluate_model_cross(10, amostra1.iloc[:,
~amostra1.columns.isin(['y'])].values,amostra1['y'].values,'RF')
dfPredLR = evaluate_model_cross(10, amostra1.iloc[:,
~amostra1.columns.isin(['y'])].values,amostra1['y'].values,'LR')
dfPredXG = evaluate_model_cross(10, amostra1.iloc[:,
~amostra1.columns.isin(['y'])].values,amostra1['y'].values,'XG')

dfPredRF.join([dfPredLR, dfPredXG])
dfPredRF.join([dfPredLR, dfPredXG]).to_latex('Tabela1-1.tex')


# 3/1
dfPredRF = evaluate_model_cross(10, amostra3.iloc[:,
~amostra3.columns.isin(['y'])].values,amostra3['y'].values,'RF')
```

```
dfPredLR = evaluate_model_cross(10, amostra3.iloc[:,
~amostra3.columns.isin(['y'])].values,amostra3['y'].values,'LR')
dfPredXG = evaluate_model_cross(10, amostra3.iloc[:,
~amostra3.columns.isin(['y'])].values,amostra3['y'].values,'XG')

dfPredRF.join([dfPredLR, dfPredXG])
dfPredRF.join([dfPredLR, dfPredXG]).to_latex('Tabela3-1.tex')

#5/1
dfPredRF = evaluate_model_cross(10, amostra5.iloc[:,
~amostra5.columns.isin(['y'])].values,amostra5['y'].values,'RF')
dfPredLR = evaluate_model_cross(10, amostra5.iloc[:,
~amostra5.columns.isin(['y'])].values,amostra5['y'].values,'LR')
dfPredXG = evaluate_model_cross(10, amostra5.iloc[:,
~amostra5.columns.isin(['y'])].values,amostra5['y'].values,'XG')

dfPredRF.join([dfPredLR, dfPredXG])
dfPredRF.join([dfPredLR, dfPredXG]).to_latex('Tabela5-1.tex')


#7/1
dfPredRF = evaluate_model_cross(10, amostra7.iloc[:,
~amostra7.columns.isin(['y'])].values,amostra7['y'].values,'RF')
dfPredLR = evaluate_model_cross(10, amostra7.iloc[:,
~amostra7.columns.isin(['y'])].values,amostra7['y'].values,'LR')
dfPredXG = evaluate_model_cross(10, amostra7.iloc[:,
~amostra7.columns.isin(['y'])].values,amostra7['y'].values,'XG')

dfPredRF.join([dfPredLR, dfPredXG])
dfPredRF.join([dfPredLR, dfPredXG]).to_latex('Tabela7-1.tex'


#10/1
dfPredRF = evaluate_model_cross(10, amostra10.iloc[:,
~amostra10.columns.isin(['y'])].values,amostra10['y'].values,'RF')
dfPredLR = evaluate_model_cross(10, amostra10.iloc[:,
~amostra10.columns.isin(['y'])].values,amostra10['y'].values,'LR')
dfPredXG = evaluate_model_cross(10, amostra10.iloc[:,
~amostra10.columns.isin(['y'])].values,amostra10['y'].values,'XG'
```

```
dfPredRF.join([dfPredLR, dfPredXG])
dfPredRF.join([dfPredLR, dfPredXG]).to_latex('Tabela10-1.tex')


#Total
df2 = df1.sample(frac=0.15, random_state=1)
X = df2.iloc[:, ~df2.columns.isin(['y'])]
y = df2.iloc[:, df2.columns.isin(['y'])]
X = StandardScaler().fit_transform(X)

dfPredRF = evaluate_model_cross(10, X,y,'RF')
dfPredLR = evaluate_model_cross(10, X,y,'LR')
dfPredXG = evaluate_model_cross(10, X,y,'XG')

dfPredRF.join([dfPredLR, dfPredXG])

dfPredRF.join([dfPredLR, dfPredXG]).to_latex('TabelaTotal.tex')

print(df2['y'].value_counts())
print((df2['y'].value_counts()[1]/df2['y']
.value_counts()[0])*100,'%')

0    88476
1      555
Name: y, dtype: int64
0.6272887562728875 %

#Selected Model
X_train, X_test,y_train, y_test = train_test_split(X, y.values,
test_size = 0.20, random_state=1)

clf = xgboost.XGBClassifier()
clf.fit(X_train, y_train)
# clf.fit(X_train, y_train,eval_set=[(X_test, y_test)])

XGBClassifier()

kfold = KFold(n_splits=2, shuffle=False)
```

```
kf_cv_scores = cross_val_score(clf, X_train, y_train, cv=kfold)
print("K-fold CV average score: %.2f" % kf_cv_scores.mean())

K-fold CV average score: 1.00

y_pred = clf.predict(X_test)

print("f ", f1_score(y_test, y_pred)*100)
print("p ", precision_score(y_test, y_pred)*100)
print("r ", recall_score(y_test, y_pred)*100)
print("a ", roc_auc_score(y_test, y_pred)*100)

f  98.68995633187774
p  99.12280701754386
r  98.26086956521739
a  99.12760864650198

y_pred = cross_val_predict(clf, X, y, cv=kfold, n_jobs=-1)

print("f ", f1_score(y, y_pred)*100)
print("p ", precision_score(y, y_pred)*100)
print("r ", recall_score(y, y_pred)*100)
print("a ", roc_auc_score(y, y_pred)*100)

f  98.8235294117647
p  99.27272727272727
r  98.37837837837839
a  99.18692868916659

filterValue = 0
dfFI = pd.DataFrame(clf.feature_importances_, columns=['Importance'])
.join(df1.columns[df1.columns != 'y'].to_frame().reset_index())
.drop(columns=[0])
.sort_values(by='Importance', ascending=True)
dfFI[dfFI.Importance > filterValue].plot.barh(x='index', figsize=(20,10))
plt.show()
print('Variáveis Relevantes: ', dfFI[dfFI.Importance > filterValue].shape[0])
```

```
cv_outer = KFold(n_splits=10, shuffle=True, random_state=1)
outer_results = list()
y_ = y.values
for train_ix, test_ix in cv_outer.split(X):
  X_train, X_test = X[train_ix, :], X[test_ix, :]
  y_train, y_test = y_[train_ix], y_[test_ix]

  cv_inner = KFold(n_splits=3, shuffle=True, random_state=1)
  model = xgboost.XGBClassifier(random_state=1)

  space = dict()
  space['n_estimators'] = [10, 100, 500]
  space['max_features'] = [2, 4, 6]

  search = GridSearchCV(model, space, scoring='accuracy', cv=cv_inner, refit=True)
  result = search.fit(X_train, y_train)
  best_model = result.best_estimator_
  yhat = best_model.predict(X_test)
  acc = accuracy_score(y_test, yhat)
  outer_results.append(acc)
  print('>acc=%.3f, est=%.3f, cfg=%s' % (acc, result.best_score_,
  result.best_params_))
print('Accuracy: %.3f (%.3f)' % (np.mean(outer_results), np.std(outer_results)))

>acc=1.000, est=1.000, cfg={'max_features': 2, 'n_estimators': 500}
>acc=1.000, est=1.000, cfg={'max_features': 2, 'n_estimators': 100}
>acc=1.000, est=1.000, cfg={'max_features': 2, 'n_estimators': 500}
>acc=1.000, est=1.000, cfg={'max_features': 2, 'n_estimators': 500}
>acc=1.000, est=1.000, cfg={'max_features': 2, 'n_estimators': 100}
```