FUNDAÇÃO GETULIO VARGAS
ESCOLA DE ECONOMIA DE SÃO PAULO

JOÃO CARLOS PACHECO JUNIOR

MODELOS PARA DETECÇÃO DE FRAUDES UTILIZANDO TÉCNICAS DE APRENDIZADO DE MÁQUINA

SÃO PAULO

2019

JOÃO CARLOS PACHECO JUNIOR

MODELOS PARA DETECÇÃO DE FRAUDES UTILIZANDO TÉCNICAS DE APRENDIZADO DE MÁQUINA

Dissertação apresentada ao Programa de Mestrado Profissional da Escola de Economia de São Paulo, da Fundação Getulio Vargas, como parte dos requisitos para a obtenção do título de Mestre em Economia.

Orientador: Prof. Dr. João Luiz Chela

SÃO PAULO

2019

Pacheco Junior, João Carlos.
Modelos para detecção de fraudes utilizando técnicas de aprendizado de máquina / João Carlos Pacheco Junior. - 2019.
102 f.

Orientador: João Luiz Chela.
Dissertação (mestrado profissional MPFE) – Fundação Getulio Vargas, Escola de Economia de São Paulo.

1. Aprendizado do computador. 2. Fraude. 3. Devedores e credores. 4. Sistemas auto-organizadores. I. Chela, João Luiz. II. Dissertação (mestrado profissional MPFE) – Escola de Economia de São Paulo. III. Fundação Getulio Vargas. IV. Título.

CDU 343.53

Ficha Catalográfica elaborada por: Raphael Figueiredo Xavier CRB SP-009987/O
Biblioteca Karl A. Boedecker da Fundação Getulio Vargas - SP

JOÃO CARLOS PACHECO JUNIOR

**MODELOS PARA DETECÇÃO DE FRAUDES
UTILIZANDO TÉCNICAS DE APRENDIZADO DE MÁQUINA**

Dissertação apresentada ao Programa de Mestrado Profissional da Escola de Economia de São Paulo, da Fundação Getulio Vargas, como parte dos requisitos para a obtenção do título de Mestre em Economia.

Data de Aprovação: ___/___/_____

Banca Examinadora:

Prof. Dr. João Luiz Chela
(Orientador)
Fundação Getulio Vargas

Prof. Dr. Ricardo Rochman
Fundação Getulio Vargas

Prof. Dr. Hellinton Hatsuo Takada
Convidado

AGRADECIMENTOS

Agradeço a Deus por sempre estar presente na minha vida, me dando força para superar todos os desafios.

Com alegria presto minha gratidão aos meus familiares, em especial a minha esposa Andressa, aos meus pais João Carlos e Edna, meus sogros Manoel e Lourdes, meus irmãos, cunhados, sobrinhos e a todos meus amigos.

À todos os colegas e amigos de curso que estiveram presente nesta minha caminhada, meu agradecimentos pelos momentos em que me auxiliaram a alcançar meus objetivos.

À todos os professores do curso que desempenharam papel fundamental na minha formação não só de forma acadêmica, mas também pessoal.

Aos amigos do Banco Votorantim que sempre me apoiaram nesta jornada.

Em especial agradeço ao Prof. João Luiz Chela, por sempre estar presente quando precisei no desenvolvimento deste trabalho, pela paciência e orientação.

“In God we trust,
all others must bring data.”
(W. Edwards Deming)

RESUMO

Devido à massificação da concessão do crédito no Brasil, proporcionada principalmente pelo desenvolvimento tecnológico, o combate a fraudes tornou-se imprescindível no âmbito das instituições financeiras, pois mesmo com baixa ocorrência têm apresentado uma tendência de crescimento, provocando assim um impacto negativo nos resultados das organizações. Neste contexto, os investimentos em técnicas mais sofisticadas para detecção de fraudes ocorrem com maior frequência, sendo que em muitos casos são utilizados métodos que aplicam técnicas de Aprendizado de Máquina, visando obter previsões mais precisas e confiáveis no combate aos eventos de fraude. Diante do exposto, este trabalho possui como objetivo propor modelos que utilizem técnicas de Aprendizado de Máquina em um banco de dados real, visando comparar os resultados obtidos com os de modelos tradicionalmente utilizados que aplicam técnicas de Regressão Logística. Adicionalmente o desafio do trabalho foi propor um classificador de *Random Forest* capaz de identificar 3 eventos distintos, que podem constituir uma fraude, ao mesmo tempo. Os resultados evidenciaram a viabilidade de se utilizar um único modelo frente a técnicas atuais onde múltiplos modelos são utilizados, um para cada evento, com uma baixa perda de desempenho, que pode ser compensada pela redução de complexidade na implantação do modelo.

Palavras-chaves: Aprendizado de Máquina, Detecção de Fraudes, Base de Dados Desbalanceadas, Classificador de Múltiplas Saídas.

ABSTRACT

Due to the massification of the credit concession in Brazil, mainly caused by recent technological development, fraud mitigation has become essential in financial institutions. Even with nowadays low occurrence rates, frauds have shown a significant increasing tendency for the future, causing, in this way, a negative impact on the organizations results. In this context, investments in more sophisticated techniques for detecting fraud has happen frequently, and in many cases, methods using Machine Learning techniques are been applied, in order to obtain more accurate and reliable predictions against fraud events. As a result of these context, this work aims to propose models and techniques that use Machine Learning in a real database, in order to compare the results obtained with traditional techniques that apply Logistic Regression techniques. Additionally, the challenge of this work was to propose a Random Forest classifier capable of identifying 3 distinct events at the same time, which may constitute a fraud. The results evidenced the viability of using a single model, as opposed to current techniques that employing multiple models, e.g. one model for each event, with a low loss of performance that can be compensated by the reduced complexity in the model implantation.

Keywords: Machine Learning, Fraud Detection, Imbalanced Datasets, Multi-Output Classifier.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AC	*Ada Boost Classifier*
BC	*Bagging Classifier*
GB	*Gradient Boosting Classifier*
KS	*Kolmogorov–Smirnov*
NB	*Naive Bayes*
FPD	*First Payment Default*
RF	*Random Forest*
RL	Regressão Logística
ROC	*Receiver operating characteristic*
SVM	*Support Vector Machine*

SUMÁRIO

1 INTRODUÇÃO

A avaliação de fraude é um processo complexo, pois ao contrário do crédito, onde é possível avaliar com precisão quais clientes pagaram ou não pagaram uma dívida, a identificação deste evento é controversa, pois nem sempre existe uma variável que identifique este evento.

Portanto, a identificação de fraude pode ocorrer no momento da concessão de um produto, onde avaliações de documentos e outros processos de validação capturam estes casos. Este é o evento mais comumente considerado ao desenvolver um modelo de fraude. Entretanto, se o processo atual já identificou este tipo de fraude, o modelo irá apenas ratificar o perfil encontrado. Dentro destas marcações, em alguns casos, podem existir falsos positivos, e não é possível sua confirmação, tendo em vista o cliente ter sido recusado na entrada.

Há um segundo momento onde as fraudes mais elaboradas, que não são detectadas pelo processo inicial entram no portfólio e possuem um comportamento que é em sua maioria confundido com inadimplência devido problemas de crédito. Este tipo de fraude não costuma ser considerado nos modelos antifraude, e seu perfil nem sempre é detectado pelos modelos de crédito, pois possuem comportamentos muito distintos e volume baixo.

Uma forma de considerar este público é utilizando clientes que foram *FPD* (First Payment Default - não pagaram nem a primeira parcela) e *Never Paid* (em 12 meses depois também não havia pago nem a primeira parcela nem outras). Há outros motivos além de fraude neste critério, mas na falta de um indicador mais próximo utiliza-se este.

Uma solução utilizada é a avaliação de escore de crédito alto (baixo risco de crédito) em conjunto com esta marcação. Estes casos podem ser observados também no momento da cobrança, onde as empresas tentam contato e recebem a informação de que houve uma fraude. Muitas vezes esta informação se perde e nunca é direcionada para a base de conhecimentos da área de fraudes.

Por fim há os casos onde se negativam os clientes, que significa inserir no sistema de algum órgão de proteção ao crédito a informação de que o consumidor não honrou alguma obrigação financeira, e por conta disso o cliente entra com uma ação cível contrária por danos morais, pois foram fraudados e tiveram algum problema de restrição no mercado (existem alguns outros motivos também para ações cíveis, mas a proporção de fraudes é relevante). Nestes, o percentual é baixo, mas os valores das ações são consideravelmente altos.

O perfil de cada um destes públicos é muito diferente, e se considerássemos um único modelo paramétrico que condensaria todas elas, este ficaria fraco. Hoje os modelos são desenvolvidos de forma separada, e a combinação entre eles é otimizada, seja através de matrizes ou de uma combinação de regras, políticas ou mesmo mistura de escores.

Portanto este trabalho busca identificar uma metodologia capaz de identificar todos os indícios de fraude anteriormente citados de uma única vez, sem necessidade de modelos diferentes, que seja bom para todos os tipos de fraude separadamente ou conjuntamente, visando assim ser uma importante ferramenta para auxiliar no combate a fraudes.

Este trabalho está dividido em 5 capítulos, sendo que o primeiro capítulo fornece uma apresentação inicial da motivação e objetivos propostos, o segundo capítulo exibe as principais referências disponíveis na literatura e o detalhamento dos conceitos utilizados. O terceiro capítulo delineia a metodologia utilizada, explanando sobre os procedimentos executados de forma sequencial. Em seguida segue o quarto capítulo com os resultados obtidos após a execução dos classificadores e uma avaliação completa das métricas utilizadas, por fim o sexto capítulo apresenta a conclusão final, demonstrando a contribuição obtida por meio deste estudo e as sugestões para trabalhos futuros.

2 REVISÃO TEÓRICA

Como passo inicial para este trabalho é importante apresentar uma definição formal de o que pode ser considerado fraude. No trabalho de Firozabadi, Tan e Lee (1998) encontramos uma definição geral de fraude, sendo que um evento pode ser considerado uma fraude quando os seguintes eventos ocorrem de forma conjunta:

- Uma pessoa ou uma organização realiza de forma intencionada uma representação falsa sobre um fato ou evento importante.
- A representação falsa é acreditada pela vítima (que é a pessoa ou organização a quem a representação foi feita).
- A vítima confia e age sobre a representação falsa.
- A vitima sofre perda financeira e/ou propriedade como resultado de confiar e agir sobre a representação falsa.

Assim ao analisarmos os eventos acima, podemos identificar vários momentos em que fraudes são aplicadas no dias atuais. E com a expansão da tecnologia, existe hoje o desenvolvimento de trabalhos utilizando metodologias de aprendizado de máquina, visando mitigar a ocorrência de fraudes em diversos setores. Deste modo é possível identificar bom suporte acadêmico para o desenvolvimento deste trabalho, como será apresentado a seguir.

O trabalho de Bolton e Hand (2002) demonstra a evolução que ocorre mundialmente na tecnologia e nos meios de comunicação, que contribuiu para o aumento considerável das fraudes. Além disso demonstra que fraude não é um evento contemporâneo, sendo que vem evoluindo com o passar do tempo, como podemos observar nos casos de lavagem de dinheiro. Entretanto estes avanços proporcionaram o surgimento de novos meios de fraude como o que vem ocorrendo nos meios de comunicação e a invasão de computadores.

Outro trabalho que desafia os modelos tradicionalmente aplicados com uma proposta de aprendizado de máquina aparece em Viaene et al. (2002) que aplica diferentes técnicas de detecção de fraude em seguros automotivos. Nesta abordagem ficou claro que modelos simples como Regressão Logística e *Lin-LS-SVM* obtiveram um ótimo desempenho. A Rede Bayesiana (suavizada) também obteve um resultado satisfatório, sendo que os algoritmos mais complexos como a classificação bayesiana de aprendizagem *MLP*, *RBF-LS-SVM* e *TAN Bayes*, adicionaram pouco ou nenhum

poder preditivo extra. Entretanto esta linearização do modelo ocorre, pois, o resultado consiste em um modelo que imita a avaliação de especialistas.

De forma complementar Phua et al. (2010) realiza uma análise profunda que compara e resume artigos técnicos e de revisão publicados sobre a detecção automatizada de fraudes nos últimos 10 anos, demonstrando que ela pode se beneficiar de áreas de pesquisa relacionadas. Especificamente, utilizando as metodologias de trabalhos com abordagens não supervisionadas (ex: contraterrorismo, sistemas reais de monitoramento e mineração de textos na aplicação da lei) além de abordagens semi-supervisionadas que empregam teoria dos jogos (ex: invasão de computadores e detecção de *spam*).

Muitas vezes apenas observar o passado para detectar novas ocorrências de fraude não é suficiente, assim Rosset et al. (1999) apresenta uma análise exploratória das possíveis regras que podem ser identificadas ao detectar fraudes. De tal modo que este trabalho traz uma grande contribuição no *design* e implementação de novos algoritmos para descoberta de regras para fraudes.

No Brasil existe uma vasta literatura na aplicação da metodologia de aprendizado de máquina com foco na detecção de fraudes. Existem vários trabalhos focados na detecção de fraudes em operações de cartões de crédito. Ramos (2015), Karcher (2009) e Moraes (2008) utilizam Redes Bayesiana para identificar a presença de fraudes em operações de cartões de crédito. Também foi identificado trabalho que utiliza Redes Bayesianas (RB) no gerenciamento e mensuração de riscos operacionais em Queiroz (2008).

Outro ponto importante levantado por Viaene et al. (2002) é que a escolha do algoritmo a ser utilizado não pode basear-se exclusivamente no poder preditivo. Deve-se levar em consideração questões como a velocidade de treinamento, ajuste, classificação e a capacidade de interpretação do modelo e dos resultados. Assim, antes de selecionar um algoritmo tem que se avaliar a disponibilidade de ferramentas, o conhecimento especializado, a experiência interna, as características do domínio, as características dos dados (conjuntos) e outras condições operacionais (comerciais).

As RB podem fornecer um método probabilístico robusto de raciocínio sob incerteza. Bobbio et al. (2001) foi um dos primeiros a apresentar o uso de RB em modelos reais, demonstrando a capacidade de predição na análise de sistemas críticos de segurança *Programmable Electronic Systems* (PES). Neste trabalho é realizada uma comparação entre duas metodologias, pois o método tradicional *Fault Tree* é comparado com resultado obtido por uma RB, demonstrando assim as possibilidades que esta metodologia tem a oferecer em vários campos de pesquisa.

Aprofundando na utilização de RB Friedman, Geiger e Goldszmidt (1997) apre-

senta que o classificador Bayesiano simples como *Naive Bayes* é capaz de concorrer com classificadores mais avançados como *C4.5* de *Ross Quilan*. Assim este trabalho também apresenta a aplicação de um classificador Bayesiano com maior sofisticação, chamado *Tree Augmented Naive Bayes* (TAN) este consegue manter a complexidade computacional encontrada no *Naive Bayes*, não prejudicando assim a sua implementação, fazendo com que seja ferramenta útil a ser utilizada em aprendizado de máquina.

Sharma e Panigrahi (2013) apresentam desenvolvimentos que vem ocorrendo na aplicação de metodologias de aprendizado de máquina no campo da contabilidade financeira. Devido ao grande aumento do número de fraudes e das falhas nos procedimentos de auditoria interna este campo tem crescido consideravelmente com o nome de Contabilidade Forense. O trabalho demonstra que atualmente são utilizados modelos *Logit*, *Stepwise Logistic*, *UTADIS* e *EGB2*, demonstrando assim a oportunidade para se utilizar diferentes classificadores com metodologia de aprendizado de máquina em trabalhos futuros.

Portanto ao observar os trabalhos anteriores à luz da detecção de fraude utilizando modelos de aprendizado de máquina, é possível identificar uma oportunidade de aplicação para o mercado brasileiro proporcionando assim complemento à literatura existente.

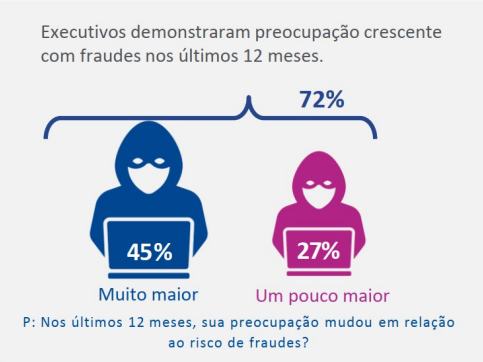
Agora serão apresentadas as técnicas para modelagem utilizadas atualmente no mercado e os modelos que utilizam aprendizagem de máquina, que aos poucos vem conquistando o seu espaço. Também serão apresentadas as técnicas de amostragem em base de dados desbalanceadas e as medidas de avaliação dos modelos que possibilitarão a comparação entre diferentes classificadores.

2.1 FRAUDES CORPORATIVAS

Os sistemas corporativos que visam facilitar e agilizar o acesso ao crédito do consumidor proporcionaram um grande avanço no aumento do acesso a produtos financeiros a uma maior parcela da população. Entretanto esta massificação da disponibilidade do crédito gerou adicionalmente um aumento na vulnerabilidade das empresas que fazem gestão destes produtos.

Assim nos dias atuais encontramos um cenário onde a quantidade de perdas decorrentes de eventos de fraudes vem aumentando anualmente no Brasil e no mundo. Conforme levantamento realizado pela consultoria Serasa Experian no ano de 2018 (EXPERIAN, 2018) 63% das empresas tiveram índices de perdas por fraude maiores ou iguais no últimos 12 meses e 72% das empresas citam que existe uma crescente preocupação em relação ao risco de fraudes para os próximos anos, como apresentado na Figura 1.

Figura 1: Preocupação dos Executivos Relacionada à Fraude.



Fonte: Experian Information Solutions, Inc (2018)

Outro ponto relevante apontado no relatório da Serasa Experian é que 54% das empresas são “pouco confiantes” em sua capacidade de detectar atividades de fraude, em comparação a um total de 40% de empresas que se auto designam como “muito confiantes”.

Um grande desafio para as empresas é a operação que é marcada como fraudulenta de forma incorreta, pois apesar de 67% das empresas considerarem que uma operação fraudulenta não recusada é mais cara para a empresa do que uma transação legítima recusada, 69% das empresas se preocupam com uma classificação incorreta (falso positivo), tendo em vista que uma má classificação pode incorrer na perda de novos clientes potenciais e de uma experiência comprometida para os clientes atuais.

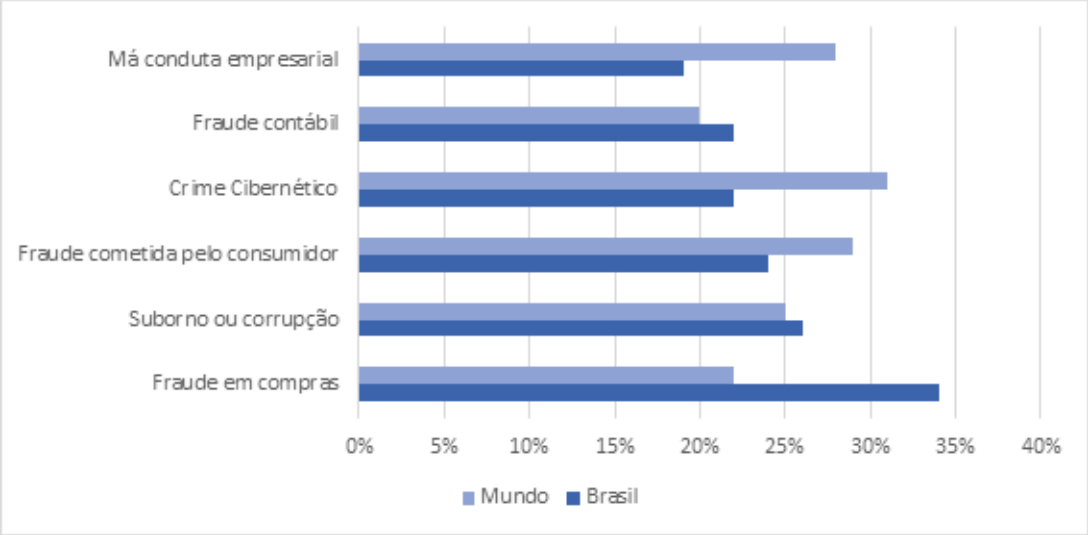
É possível identificar em relatório produzido pela consultoria PWC no ano de 2018 (LAVION, 2018) que grande parte dos crimes sofridos pelas empresas no Brasil está diretamente ligado ao relacionamento com os clientes, sendo que fraudes em compras e fraude cometida pelo consumidor afetam mais de 24% das empresas, conforme apresentado na Figura 2.

Assim, podemos concluir que o desenvolvimento de ferramentas que possibilitem maior assertividade na detecção de fraudes é de extrema importância no mercado, pois possibilita que as empresas reduzam perdas financeiras, além de incrementar a satisfação de seus clientes, ao evitar que eles sejam fraudados.

2.2 REGRESSÃO LOGÍSTICA

O termo regressão remonta o trabalho apresentado por Sir Francis Galton, no século XIX, que investigava a relação da altura dos pais com relação aos seus filhos. Durante os estudos ele percebeu que as alturas tendem a um valor mediano, ao qual ele chamou de regressão em direção a média, sendo esta a origem da palavra que empregamos em estatística como encontramos em Casella e Berger (2011).

Figura 2: Tipos de Crimes sofridos pelas Empresas.



Fonte: PWC (2018)

Portanto podemos descrever como regressão o estudo da relação entre um conjunto de variáveis que podemos chamar independentes ou explicativas (no exemplo anterior à altura dos pais) com a variável dependente ou resposta (no exemplo anterior à altura dos filhos). Esta regressão é construída através de um modelo, que busca prever a relação entre as variáveis dependentes e independentes, possibilitando assim prever qual será o valor da variável dependente por meio as variáveis independentes disponíveis.

Os modelos de regressão logística possuem objetivo igual os outros tipos de regressão, entretanto possui algumas características especificas citadas por Hosmer Jr, Lemeshow e Sturdivant (2013):

- As variáveis explicativas podem possuir tanto características quantitativas quanto qualitativas, sendo que no caso de características qualitativas se faz necessário o uso de indicador.
- A variável resposta será categórica.
- Os erros provenientes desta regressão não seguirão uma distribuição normal.

Portanto podemos verificar que a detecção de fraudes se encaixa perfeitamente neste contexto, tendo em vista possuir como variável dependente a possibilidade de haver ou não a fraude. Adicionalmente as variáveis do tipo independente possuem características tanto quantitativas quanto qualitativas pois devem descrever características da operação e do cliente.

2.2.1 FUNÇÃO RESPOSTA

O modelos de Regressão Logística é um tipo de modelo linear generalizado onde as variáveis respostas $Y_1, ..., Y_n$ são independentes e binárias com:

$$Y_i \sim Bernoulli(\pi_i) \tag{2.1}$$

sendo a família Bernoulli exponencial. Sabendo que a esperança de Y_i que é $EY_i = \pi_i = p(Y_i = 1)$, a Regressão Logística é descrita como uma relação de π_i com x_i conforme a seguir:

$$\log(\frac{\pi_i}{1-\pi_i}) = \alpha + \beta x_i \tag{2.2}$$

onde α é uma constante, que representa a interceptação da reta com o eixo vertical e o β representa a inclinação (coeficiente angular) em relação à variável x_i.

Interpretando a equação anterior temos que o lado esquerdo representa o log das chances de sucesso para Y_i. O modelo assume que este *logito* é uma função linear da preditora x_i. A função de probabilidade de Bernoulli pode ser escrita na forma de família exponencial:

$$\pi_i^{y_i^*}(1-\pi_i)^{1-y_i^*} = (1-\pi)e^{y_i \log(\frac{\pi_i}{1-\pi_i})} \tag{2.3}$$

Realizando uma ligação canônica, ao utilizar o parâmetro natural $\log(\frac{\pi_i}{1-\pi_i})$ podemos reescrever a equação da seguinte forma:

$$\pi_i = \frac{e^{\alpha+\beta x_i}}{1+e^{\alpha+\beta x_i}} \tag{2.4}$$

Fazendo π uma função de x_i:

$$\pi(x) = \frac{e^{\alpha+\beta x_i}}{1+e^{\alpha+\beta x_i}} \tag{2.5}$$

2.2.2 ESTIMAÇÃO DOS PARÂMETROS

Como no modelo com $Y_i \sim Bernoulli(\pi_i)$ não existe uma conexão direta entre Y_i e $\alpha+\beta x_i$ não é possível utilizar diretamente os mínimos quadrados. Assim o método de estimação a ser utilizado é a máxima verossimilhança, se consideramos $F_i = \pi(x_i)$ temos:

$$L(\alpha, \beta|y) = \prod_{i=1}^{n} \pi(x_i)^{y_i}(1-\pi(x_i))^{1-y_i} = \prod_{i=1}^{n} F_i^{y_i}(1-F_i)^{1-y_i} \tag{2.6}$$

Com o log da verossimilhança temos:

$$\log L(\alpha, \beta|y) = \sum_{i=1}^{n}\{\log(1 - F_i) + y_i \log(\frac{F_i}{1 - F_i})\} \tag{2.7}$$

Para testar a significância do modelo devemos avaliar a hipótese $H_0 : \beta = 0$, pois como é observado em uma regressão linear simples, esta hipótese declara que não existe relação entre a variável preditora e a variável resposta.

Assim a estatística do teste de Wald, $Z = \frac{\widehat{\beta}}{se(\widehat{\beta})}$, onde se é o erro padrão (*standard error*), possui aproximadamente uma distribuição normal padrão no caso de H_0 verdadeiro e a amostra seja grande o suficiente.

Assim, H_0 pode ser rejeitada se $|Z| \geq Z\frac{\alpha}{2}$, alternativamente a hipótese pode ser testada como log da estatística do Teste da Razão de Verossimilhança:

$$-2 \log \lambda(y^*) = 2[\log L(\widehat{\alpha}, \widehat{\beta}|y^*) - L(\widehat{a}_0, 0|y^*)] \tag{2.8}$$

onde $\widehat{\alpha}_0$ é o Estimador de Máxima Verossimilhança de α assumindo $\beta = 0$.

Com argumentos binomiais padrão, Casella e Berger (2011) demonstra que é possível mostrar que:

$$\widehat{\alpha}_0 = \sum_{i=1}^{n} \frac{y_i}{n} = \frac{\sum_{j=1}^{J} y_j^*}{\sum_{j=1}^{J} n_j} \tag{2.9}$$

Portanto podemos concluir que de acordo com H_0, $-2 \log \lambda$ possui uma distribuição aproximada X_1^2 e a hipótese H_0 pode ser rejeitada no nível α se $-2 \log \lambda \geq X_{1,\alpha}^2$.

2.3 APRENDIZADO DE MÁQUINA

O aprendizado de máquina é um novo campo da ciência que surgiu com o intuito de extrair valor/conhecimento tendo como origem grandes bases de dados, muitas vezes a motivação para utilizar estas técnicas é criar um algoritmo capaz de imitar ou até mesmo melhorar o desempenho alcançado por um ser humano.

Como apresentado por Bishop (2006), Trevor, Robert e JH (2009), Murphy (2012) e Barber (2012) o aprendizado de máquina desempenha papel fundamental em áreas como estatística, mineração de dados e inteligência artificial, além de ser cada vez mais utilizadas em outras disciplinas como áreas de engenharia.

Mitchell (2006) apresenta o uso de técnicas de aprendizado de máquina na área de economia, que se interessa em questões de mercado, como a forma em que estão distribuídos os conjuntos de indivíduos, com interesses próprios, que podem formar um sistema (mercado) que aprendem preços levando a alocações ótimas para o

maior bem comum. Os modelos matemáticos existentes para estas questões são diferentes dos utilizados por técnicas de aprendizado de máquina, apresentando assim um potencial de interação entre estas ciências para a evolução dos modelos e teorias.

O aprendizado de máquina possuí dois principais subcampos de aprendizado, sendo eles o aprendizado supervisionado e o não supervisionado. No aprendizado supervisionado, o foco está na previsão precisa, sendo que já existem dados prévios do padrão que se busca identificar, já no aprendizado não supervisionado os dados não são conhecidos sendo assim o objetivo a identificação de padrões nos dados.

Após uma breve explanação sobre os possíveis tipos de aprendizado no campo do aprendizado de máquina, serão apresentados os classificadores empregados neste trabalho.

2.3.1 APRENDIZADO SUPERVISIONADO

No aprendizado supervisionado como apresentado por Barber (2012) existe uma base de dados prévios com as características dos dados que denotaremos por x e sua classificação que denotaremos por y. Neste processo é criada uma base de treinamento, com uma amostra dos dados, onde o algoritmo irá identificar quais características presentes em x são suficientes para chegar à classificação de y.

Um exemplo simples que é bastante utilizado nos dias de hoje é a classificação de crédito que visa identificar a possibilidade de um cliente ser um mau pagador (*default*). Com todas as informações disponíveis sobre o cliente e também com a contratação de uma empresa externa que possui informações sobre os clientes, os bancos conseguem criar modelos que com as informações do passado, podem prever quais os clientes serão mau pagadores, evitando assim um aumento em sua taxa de clientes mau pagadores.

Assim a definição de aprendizado supervisionado é: dado um conjunto de dados $D = \{(x^n, y^n), n = 1, ..., N\}$ com N observações, a tarefa é aprender a relação entre a entrada x e a saída y te tal forma que quando ocorrer uma nova entrada x^* a saída prevista y^* possua alta precisão. O par (x^*, y^*) não está em D, mas é assumido como sendo gerado pelo mesmo processo que gerou D.

Ao utilizar o aprendizado supervisionado a base de dados disponível que será utilizada na modelagem é dividida em duas, que são as bases de treinamento e teste.

A base de treinamento possui um número superior de observações e será utilizada para treinar o modelo de aprendizado de máquina, para que ele possa identificar posteriormente os mesmos padrões em bases diferentes. Por outro lado, a base de teste é utilizada para verificar o poder de generalização do modelo, caso a eficácia seja menor que a base de treino, pode ter ocorrido *overffiting* onde o modelo só funciona

bem para a base de treinamento, devido a sua baixa capacidade de generalização. Isto ocorre muitas vezes devido ao uso excessivo de variáveis no modelo desenvolvido.

2.3.2 APRENDIZADO NÃO-SUPERVISIONADO

No aprendizado não-supervisionado não existe uma base de dados prévia que permite que os modelos sejam desenvolvidos com uma base de treinamento, portanto o modelo deverá ser executado conforme as informações são geradas. Esta abordagem visa identificar padrões em bases de dados, mas sem a existência de uma resposta prévia para a classificação que será identificada.

Este tipo de abordagem é utilizado, por exemplo, para identificar a ocorrência de fraudes em cartões de crédito, pois como muitas vezes as fraudes possuem comportamento atípicos e que ainda não ocorreram, este desvio de padrão pode ser identificado por um modelo de aprendizado não-supervisionado. Detectar eventos anômalos em processos industriais, monitoramento de motores e padrões de comportamento de compra inesperados em clientes, como o exemplo anterior, são tipos de modelos não-supervisionados classificados como detecção de anomalias, também chamados de detecção de “novidade”.

Como apresentado por Barber (2012), a definição de aprendizado não-supervisionado é: dado um conjunto de dados $D = \{(x^n, y^n), n = 1, ..., N\}$ na aprendizagem não-supervisionada, a tarefa é identificar uma descrição compacta que seja plausível dos dados (criar *clusters*). Assim deve-se criar uma função objetivo para quantificar a precisão da descrição obtida. Portanto neste procedimento não existe uma variável de predição especial, de modo que, em uma perspectiva probabilística, o que nos interessa é modelar uma distribuição $p(x)$ para os dados.

2.3.3 APRENDIZADO SEMI-SUPERVISIONADO

No aprendizado de máquina, é comum possuir uma pequena quantidade de dados rotulados frente a uma grande quantidade de dados não rotulados. Um exemplo deste caso é quando possuímos uma grande quantidade de fotos de plantas a serem classificadas, entretanto apenas um pequeno número das fotos foi rotulado por serem conhecidas.

No aprendizado semi-supervisionado, utiliza-se os dados não rotulados para tentar criar um classificador melhor do que o criado com base apenas nos dados rotulados.

Este tipo de problema é comum, pois criar dados sem rótulos é barato, como por exemplo tirar fotos, enquanto que por outro lado criar dados rotulados são mais custosos, pois necessitam da intervenção de seres humanos, e muitas vezes de espe-

cialistas encarecendo ou até mesmo inviabilizando uma abordagem com aprendizado totalmente supervisionado.

Por fim podemos concluir que o aprendizado semi-supervisionado é uma junção dos tipos de aprendizado citados anteriormente (supervisionado e não-supervisionado), pois o modelo pode identificar os casos conhecidos e agrupar os desconhecidos.

2.4 MÁQUINA DE VETORES DE SUPORTE

A máquina de vetores de suporte (SVM, *Support Vector Machine*) é um método de classificação que foi originalmente concebido por Cortes e Vapnik (1995), sendo por eles descrito como um procedimento construtivo universal de aprendizado, que é baseado na teoria de aprendizagem estatística (*statistical learning theory*). Consiste na classificação por meio da criação de uma superfície de decisão linear que pode possuir muitas dimensões, sendo que as propriedades especiais da superfície de decisão construída possibilitam uma alta capacidade de generalização deste classificador.

Assim podemos afirmar que o classificador SVM possui grande capacidade na detecção de padrões, sendo assim capazes de identificar operações fraudulentas de cartões de créditos, identificar dígitos escritos a mão, além de serem utilizados com sucesso em várias aplicações biológicas como apresentado em Noble (2006).

O classificador SVM possui dois casos específicos, um para os casos onde é possível realizar a separação linear dos dados, criando duas classes distintas e o outro caso quando a separação linear não é possível, sendo assim necessária a utilização de um penalizador como apresentado em Burges (1998).

2.4.1 CASO SEPARÁVEL LINEARMENTE

O caso mais simples ocorre quando os dados de treinamento possuem uma separação linear, possibilitando assim que o classificador identifique uma separação linear para os dados.

Assim ao classificarmos os dados de teste temos $\{x_i, y_i\}, i = 1, ..., l$ sendo que $y_i \in \{-1, 1\}$, $x_i \in R^d$ e l o número de observações. Supondo que exista um hiperplano que separe cada um dos pontos positivos dos negativos (hiperplano de separação). Os pontos x presentes no hiperplano satisfazem a equação $wx + b = 0$, onde w é normal ao hiperplano, $\frac{|b|}{||w||}$ é a distância perpendicular do hiperplano até a origem, e $||w||$ é a norma Euclidiana de w. Seja $d_+(d_-)$ a menor distância do hiperplano de separação para o mais próximo ponto, positivo ou negativo, definimos a "margem" do hiperplano de separação como $d_+ + d_-$.

Portanto para o caso linearmente separável, o algoritmo SVM simplesmente procura pelo hiperplano de separação com a maior margem. Para que seja possível formular é necessário supor que os dados de treinamento satisfaçam as seguintes restrições:

$$x_i w + b \geq +1 \text{ para } y_i = +1 \tag{2.10}$$

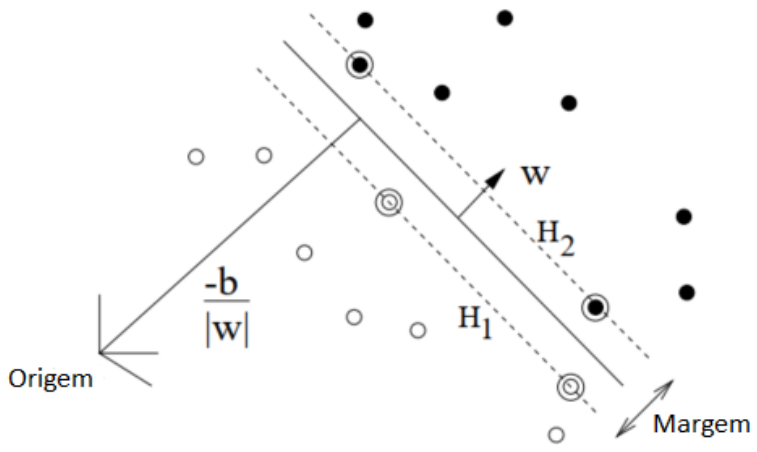
$$x_i w + b \leq -1 \text{ para } y_i = -1 \tag{2.11}$$

Que podem ser combinados na seguinte equação:

$$y_i(x_i w + b) - 1 \geq 0 \; \forall i \tag{2.12}$$

No caso apresentado na Figura 3 podemos verificar que os hiperplanos H_1 e H_2 são paralelos, possuindo assim a mesma normal, portanto para encontrar o par de hiperplanos com a margem máxima, temos que minimizar $||w||^2$, sujeito as restrições anteriormente apresentadas.

Figura 3: Hiperplano separado linearmente para o caso separável.



Fonte: Burges (1998)

Para facilitar a manipulação das restrições é aplicada uma formulação Lagrangeana do problema, assim são introduzidos os multiplicadores de Lagrange $\alpha_i, i = 1, ..., l$, para cada restrição de desigualdade, para chegar em:

$$L_p \equiv \frac{1}{2}||w||^2 - \sum_{i=1}^{l} \alpha_i y_i (x_i w + b) + \sum_{i=1}^{l} \alpha_i \tag{2.13}$$

Agora, deve-se minimizar L_p com relação à w, b e simultaneamente requer que as derivadas de L_p com relação a todos os α_i desapareçam, todas sujeitas a restrição $\alpha_i \geq 0$. De forma equivalente é possível resolver este problema maximizando L_p, sujeito as restrições do gradiente de L_p com relação a w e b desapareça, e sujeito também as restrições de $\alpha_i \geq 0$.

A formulação dual do problema é chamada dualidade de *Wolfe*, apresentado em Fletcher (1987). Assim exigindo que o gradiente de L_p em relação a w e b desapareça, temos as seguintes condições:

$$w = \sum_i \alpha_i y_i x_i \tag{2.14}$$

$$\sum_i \alpha_i y_i = 0 \tag{2.15}$$

Sendo que estas são restrições de igualdade na formulação dual, elas podem ser utilizadas na Equação 2.13, assim temos:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j \tag{2.16}$$

Agora obtemos equações diferentes de Lagrange com rótulos diferentes (P para Primal e D para Dual) sendo que possuem formulações diferentes, mas que surgem da mesma função objetivo, alterando as restrições e a solução que é encontrada minimizando L_P ou maximizando L_D.

Portanto podemos concluir que o treinamento de vetores de suporte para o caso linear separável consiste em maximizar o L_D em relação ao α_i sujeita as restrições apresentadas anteriormente e ao fato de α_i ser positivo.

2.4.2 CASO NÃO-SEPARÁVEL LINEARMENTE

O algoritmo SVM apresentado anteriormente para o caso separável não obtém resultados satisfatórios para os casos não separáveis. Assim é necessário relaxar as restrições iniciais:

$$x_i w + b \geq +1 \text{ para } y_i = +1 \tag{2.17}$$

$$x_i w + b \leq -1 \text{ para } y_i = -1 \tag{2.18}$$

Mas apenas quando necessário, ou seja, deve ser introduzido um custo adicional, que representa um aumento na função objetivo primordial. Isto pode ser feito

por meio da introdução de variáveis de folga positivas $\xi_i, i = 1, ..., l$ nas restrições, da seguinte forma:

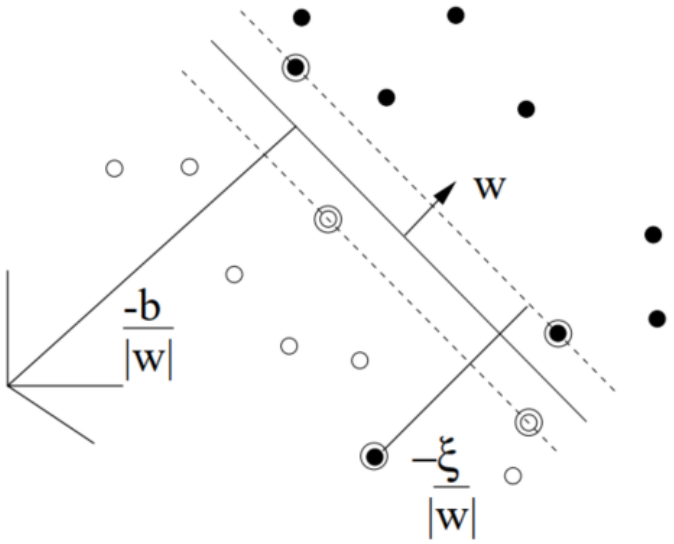
$$x_i w + b \geq +1 - \xi_i \text{ para } y_i = +1 \tag{2.19}$$

$$x_i w + b \leq -1 + \xi_i \text{ para } y_i = -1 \tag{2.20}$$

$$\xi_i \geq 0 \; \forall i \tag{2.21}$$

Portanto com este ajuste nas constantes, para que um erro ocorra, o correspondente ξ_i deve exceder a unidade, assim $\sum_i \xi_i$ é um limite superior no número de erros de treinamento. Assim uma maneira natural para atribuir um custo extra para erros é alterar a função objetivo a ser minimizada $\frac{||w||^2}{2}$ por $\frac{||w||^2}{2} + C(\sum_i \xi_i)^k$, onde C é um parâmetro a ser escolhido pelo usuário, onde um C maior representa atribuir uma penalidade maior aos erros. Este caso é representado pela Figura 4.

Figura 4: Hiperplano separado linearmente para o caso não-separável.



Fonte: Burges (1998)

Como podemos verificar trata-se de um problema de programação convexa para qualquer inteiro positivo k, para $k = 2$ e $k = 1$ também será um problema de computação quadrático, e a escolha de um $k = 1$ tem a vantagem adicional de que nem

os multiplicadores ξ_i, nem seus multiplicadores de Lagrange, aparecem no problema dual de *Wolfe*, fazendo com que o problema seja minimizar:

$$L_D \equiv \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j \tag{2.22}$$

Sujeito a:

$$0 \leq \alpha_i \leq C \tag{2.23}$$

$$\sum_i \alpha_i y_i = 0 \tag{2.24}$$

A solução é dada novamente por:

$$w = \sum_{i=1}^{N_S} \alpha_i y_i x_i \tag{2.25}$$

Onde N_S é o número de vetores de suporte. Assim a única diferença do hiperplano ideal é que α_i agora possui um limite superior de C. Por fim, o Lagrangeano primitivo será:

$$L = \frac{1}{2}||w||^2 + C \sum_i \xi_i - \sum_i \alpha_i \{y_i(x_i w + b) - 1 + \xi_i\} - \sum_i \mu_i \xi_i \tag{2.26}$$

Onde μ_i são os multiplicadores de Lagrange introduzidos para reforçar a positividade do ξ_i.

2.5 REDES BAYESIANAS

Conforme descrito por Friedman, Geiger e Goldszmidt (1997) as Redes Bayesianas são grafos acíclicos que possibilitam uma representação eficiente e efetiva da distribuição de probabilidade conjunta sobre um conjunto de variáveis aleatórias. Assim, cada um dos vértices do grafo representam uma variável aleatória e as suas respectivas arestas as correlações diretas entre as variáveis.

Podemos assim dizer que a rede codifica a regras de independência condicional, segundo a qual cada variável é independente de seus não-descendentes no grafo, segundo o estado dos seus pais. Essas independências possibilitam reduzir o número de parâmetros necessários para caracterizar uma distribuição de probabilidade facilitando o cálculo das probabilidades posteriores.

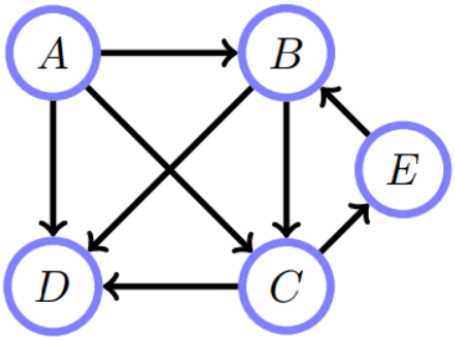
Por fim ao utilizar esta configuração de rede com a independência presente na rede, pode se determinar que a distribuição conjunta é atribuída unicamente pelas distribuições condicionais locais.

2.5.1 GRAFOS

Ao analisarmos um problema, muitas vezes podemos acreditar que um evento afeta outro e existem eventos que são independentes. Utilizar este tipo de informação pode produzir modelos que sejam melhores especificados e computacionalmente mais eficientes. Portanto para que este tipo de informação seja utilizado de forma eficiente utilizamos estruturas de grafos que nos permitem descrever como os objetos são vinculados e fornecem uma imagem conveniente da relação dos objetos (eventos).

Como apresentado em Barber (2012) podemos descrever grafos como a lista de adjacência $G = (V, E)$ que consiste de uma arranjo $(Adj.)$ de $|V|$ listas, um para cada vértice em V. Portanto para cada $u \in V$, a lista de adjacências $Adj[u]$ possui todos os vértices de v tais que as arestas$(u, v) \in E$. Um grafo direcionado G consiste em arestas direcionadas entre nós, como apresentado na Figura 5.

Figura 5: Grafo direcionado.



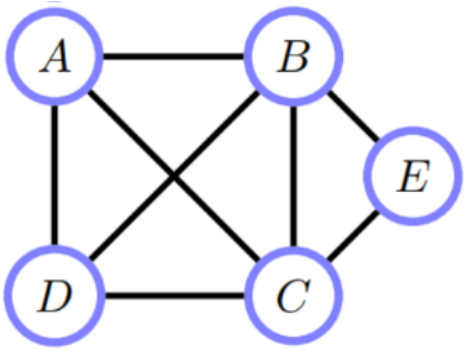
Fonte: Barber (2012)

O caminho $A \to B$ do nó A para o nó B é a sequência de nós que conectam A até B. Isso significa que o caminho é a forma $A_0, A_1, ..., A_{n-1}, A_n$ com $A_0 = A$ e $A_n = B$. Os grafos direcionados podem ser cíclicos, quando se inicia e termina no mesmo nó, caso contrário será um grafo acíclico.

Outro conceito muito importante em grafos direcionados é o DAG *Directed Acyclic Graph* que consistem em um grafo G com arestas direcionadas e com a característica que um caminho de nós, de um nó para outro ao longo da direção de cada aresta, nenhum caminho voltará a um nó anterior (característica marcante em árvores genealógicas).

Por fim, um grafo não direcionado G consiste em arestas que não possuem direção entre os nós como demonstrado na Figura 6.

Figura 6: Grafo não direcionado.



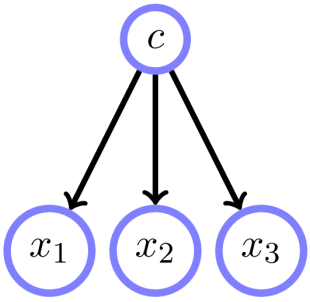
Fonte: Barber (2012)

O conceito de grafo é de suma importância para a aplicação de modelos de aprendizado de máquina de Redes Bayesianas.

2.5.2 NAIVE BAYES

Naive Bayes é um método de classificação popular que é considerado ingênuo (*Naive*), pois assume que os atributos são condicionalmente independentes, ou seja, a informação de um evento não possui efeito sobre nenhum outro evento. Entretanto, nas aplicações de Aprendizado de máquina é um dos métodos que possui ótimos resultados comparado com outras técnicas, conforme citado em Barber (2012).

Figura 7: Grafo não direcionado.



Fonte: Barber (2012)

Em Naive Bayes é construído um modelo conjunto para um vetor de atributos $D-dimensional\ x$ e o seu rótulo correspondente c e a rede de crenças é representada conforme Figura 7.

$$p(x,c) = p(c)\prod_{i=1}^{D} p(x_i|c) \tag{2.27}$$

Juntamente com a escolha adequada para cada distribuição condicional $p(x_i|c)$, pode-se utilizar a regra de Bayes para formar um classificador para um novo vetor de entrada x conforme equação a seguir:

$$p(c|x^*) = \frac{p(x^*|c)p(c)}{p(x^*)} = \frac{p(x^*|c)p(c)}{\sum_c p(x^*|c)p(c)} \tag{2.28}$$

Nos exemplos reais é comum considerar um domínio com apenas duas classes $dom(c) = \{0, 1\}$, que é o caso deste trabalho.

2.5.2.1 ATRIBUTOS BINÁRIOS

Considerando uma base de dados $\{(x^n, c^n), n = 1, ..., N\}$ de atributos binários, $x_i^n \in \{0, 1\}, i = 1, ..., D$ associados aos rótulos de classe c^n. O número de datapoints da classe $c = 0$ é denotado n_0 e o número vindos da classe $c = 1$ é denotado n_1. Para cada atributo das duas classes, é necessário estimar os valores $p(x_i = 1|c) \equiv \theta_i^c$. A outra probabilidade $p(x_i = 0|c)$ é dada pelo requerimento de normalização, $p(x_i = 0|c) = 1 - p(x_i = 1|c) = 1 - \theta_i^c$.

Com base na hipótese de independência condicional do classificador Naive Bayes, a probabilidade observada no vetor x pode ser escrita na seguinte forma compacta:

$$p(x|c) = \prod_{i=1}^{D} p(x_i|c) = \prod_{i=1}^{D} (\theta_i^c)^{x_i}(1 - \theta_i^c)^{1-x_i} \tag{2.29}$$

Na expressão anterior, x_i é 0 ou 1, portanto, cada termo i contribui com o fator θ_i^c se $x_i = 1$ ou com o fator $1 - \theta_i^c$ se $x_i = 0$. Juntamente com a suposição que os dados de treinamento gerados são independentes e identicamente distribuídos, o $\log$ da probabilidade dos atributos dos rótulos de classe é:

$$L = \sum_n \log p(x^n, c^n) = \sum_n \log p(c^n) \prod_i p(x_i^n|c^n) \tag{2.30}$$

$$= \{\sum_{i,n} x_i^n \log \theta_i^{c^n} + (1 - x_i^n) \log(1 - \theta_i^{c^n})\} + n_0 \log p(c = 0) + n_1 \log p(c = 1) \tag{2.31}$$

Que pode ser escrito mais explicitamente em termo dos parâmetros como:

$$L = \sum_{i,n} \{\mathbb{I}[x_i^n = 1, c^n = 0] \log \theta_i^0 + \mathbb{I}[x_i^n = 0, c^n = 0] \log(1 + \theta_i^0) + \mathbb{I}[x_i^n = 1, c^n = 1] \log \theta_i^1$$
$$+\mathbb{I}[x_i^n = 0, c^n = 1] \log(1 - \theta_i^1)\} + n_0 \log p(c = 0) = n_1 \log p(c = 1) \tag{2.32}$$

Assim é possível encontrar a ótima máximo verossimilhança para θ_i^c diferenciando e igualando a zero:

$$\theta_i^c = p(x = 1|c) = \frac{\sum_n \mathbb{I}[x_i^n = 1, c^n = c]}{\sum_n \mathbb{I}[x_i^n = 0, c^n = c] + \mathbb{I}[x_i^n = 1, c^n = c]} \tag{2.33}$$

Que pode ser expressado da seguinte forma:

$$= \frac{\text{número de vezes em que } x_i = 1 \text{ para a classe c}}{\text{número de dados para a classe c}} \tag{2.34}$$

De forma similar, otimizando a equação (2.30) com relação a $p(c)$ temos:

$$p(c) = \frac{\text{número de vezes em que a classe c ocorre}}{\text{número total de dados}} \tag{2.35}$$

2.5.2.2 VARIÁVEIS COM MULTI-ESTADOS

A extensão do método para multivariáveis na classe c é simples a partir do caso com atributos binários. Assim iremos apresentar o caso para dois ou mais variáveis.

Para a variável x_i com os seguintes domínios, $dom(x_i) = \{1, \ldots, S\}$ a probabilidade de observar um estado $x_i = s$ é dado por:

$$p(x_i = s|c) = \theta_s^i(c) \tag{2.36}$$

Com $\sum_s p(x_i = s|c) = 1$ a probabilidade condicional da classe que gera os dados com independência condicional do classificador $D = (x^n, c^n), n = 1, \ldots, N$ é:

$$\prod_{n=1}^{N} p(x^n|c^n) = \prod_{n=1}^{N} \prod_{i=1}^{D} \prod_{s=1}^{S} \prod_{c=1}^{C} \theta_i^s(c)^{\mathbb{I}[x_i^n = s, c^n = c]} \tag{2.37}$$

O efeito dos indicadores é que apenas os termos $\theta_s^i(c)$ sobrevivem, para os quais existem atributos i no estado s para a classe c. Podemos obter a classe condicional da verossimilhança logarítmica:

$$L(\theta) = \sum_{n=1}^{N} \sum_{i=1}^{D} \sum_{s=1}^{S} \sum_{c=1}^{C} \mathbb{I}[x_i^n = s]\mathbb{I}[c^n = c] \log \theta_s^i(c) \tag{2.38}$$

Que pode ser otimizada com relação ao parâmetro θ utilizando o multiplicador de Lagrange (um para cada um dos atributos i das classes c) para garantir assim a normalização. Assim obtemos o seguinte Lagrangiano:

$$\mathcal{L}(\theta, \lambda) = \sum_{n=1}^{N} \sum_{i=1}^{D} \sum_{s=1}^{S} \sum_{c=1}^{C} \mathbb{I}[x_i^n = s]\mathbb{I}[c^n = c] \log \theta_s^i(c) + \sum_{c=1}^{C} \sum_{i=1}^{D} \lambda_i^c (1 - \sum_{s=1}^{S} \theta_s^i(c)) \tag{2.39}$$

Para encontrar a função ótima é necessário diferenciar com relação a $\theta_s^i(c)$ e igualar a zero. Resolvendo assim a equação anterior é possível obter:

$$\sum_{n=1}^{N} \frac{\mathbb{I}[x_i^n = s]\mathbb{I}[c^n = c]}{\sum_{s',n'} \mathbb{I}[x_i^{n'} = s']\mathbb{I}[c^{n'} = c]} \tag{2.40}$$

A máximo verossimilhança para o parâmetro $p(x_i = s|c)$ é igual ao número relativo de vezes que o atributo i está no estado s para a classe c.

2.5.2.3 BAYESIAN NAIVE BAYES

Naive Bayes pode ser um método poderoso de classificação, mas pode ser excessivamente zeloso no caso de pequenas contagens. Se um único atributo i não tiver contagens para a classe c, então, independente dos outros atributos, o classificador dirá que x não pode ser da classe c. Isto ocorre, pois, o produto de 0 com qualquer outra coisa permanece 0. Para combater o efeito de excesso de confiança, pode-se utilizar o método *bayesiano* simples.

Dado uma base de dados $D = \{(x^n, c^n), n = 1, \ldots, N\}$ pode-se predizer a classe c para a nova entrada x utilizando:

$$p(c|x, D) \propto p(x, D, c) \propto p(x|D, c)p(c|D) \tag{2.41}$$

Por conveniência pode-se simplificar $p(c|D)$ utilizado a máximo verossimilhança:

$$p(c|D) = \frac{1}{N} \sum_n \mathbb{I}[c^n = c] \tag{2.42}$$

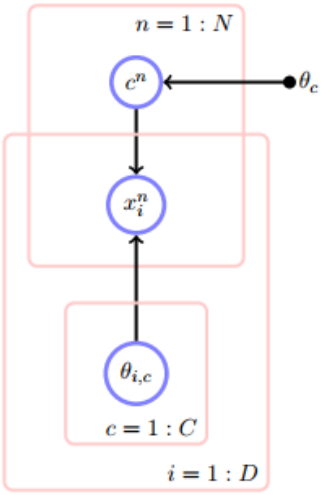
Entretanto, definir parâmetros de $p(x|D, c)$ utilizando o treinamento com máximo verossimilhança pode render previsões excessivamente confiantes no caso de dados esparsos. Assim, uma abordagem Bayesiana que lida com este tipo de dificuldade usa priorização nas probabilidades $p(x_i = s|c) \equiv \theta_s^i(c)$ para desencorajar valores extremos. Assim o modelo pode ser representado conforme a Figura 8.

2.6 MÉTODOS DE ENSEMBLE

Os métodos Ensemble possuem como principal objetivo combinar o poder preditivo de vários estimadores por meio de um algoritmo de aprendizado, visando aperfeiçoar a generalização e robustez quando comparado a um estimador utilizado sozinho.

Assim, é criado um conjunto de classificadores onde as decisões obtidas individualmente são combinadas, podendo ser por meio de votação ponderada ou não

Figura 8: Representação gráfica do modelo *Naive Bayes*.



Fonte: Barber (2012)

ponderada, assim obtendo um resultado final com a utilização de todos os resultados obtidos por cada classificador individualmente.

Como mencionado em Dietterich (2000) um algoritmo de aprendizado de máquina conjunto tem como característica marcante a possibilidade de gerar resultados mais precisos e robustos, quando exposto a base de dados com a presença de ruído, do que os classificadores que os compões utilizados individualmente.

Para que o classificador Ensemble possua precisão superior que qualquer um de seus membros, existe como condição necessária e suficiente que os classificadores sejam precisos, ou seja, tenha uma taxa de erro melhor do que uma adivinhação aleatória além de serem diversos, ou seja, que os classificadores possuam erros diferentes entre si ao classificar os dados.

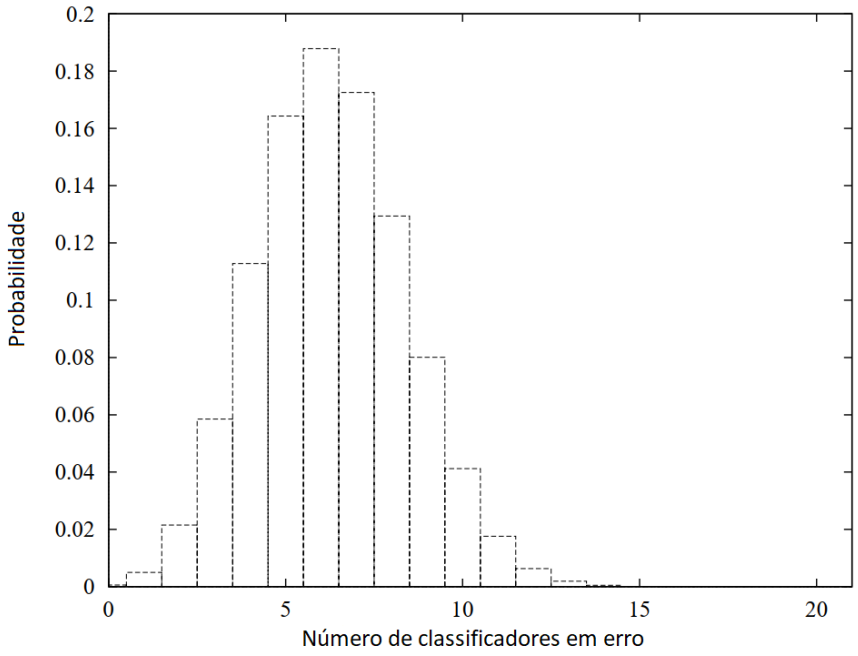
Ao avaliar um caso x, com a utilização de três classificadores idênticos $h_1(x)$, $h_2(x)$ e $h_3(x)$, quando o classificador $h_1(x)$ estiver errado os classificadores $h_2(x)$ e $h_3(x)$ também estarão. Entretanto caso os erros destes classificadores não estejam correlacionados, quando $h_1(x)$ estiver errado $h_2(x)$ e $h_3(x)$ podem estar corretos, fazendo com que o voto da maioria classifique x de forma correta.

Assim é possível concluir que se a taxa de erros de L hipóteses h_l forem iguais a $p \leq \frac{1}{2}$ e se os erros forem independentes, então a probabilidade de que o voto da maioria esteja errado será a área sob a distribuição binomial onde mais de $\frac{L}{2}$ hipóteses estejam erradas.

A Figura 9 apresenta um Ensemble simulado de 21 hipóteses, sendo que cada hipótese possui uma taxa de erro de $0,3$. A área abaixo da curva para 11 ou mais

hipóteses serem simultaneamente erradas é $0,026$, que é muito menor que o erro de uma hipótese individual.

Figura 9: Margem de erro em uma simulação de *Ensemble*.



Fonte: Dietterich (2000)

2.7 BOOTSTRAP AGGREGATING - BAGGING

Os algoritmos de *Bagging*, propostos por Breiman (1996) utilizam o método de *Ensembled*, pois gera várias versões de um preditor por meio de amostras *bootstrap* [1] dos dados, depois eles são agregados ou combinados por meio do voto para obter um preditor agregado.

Este procedimento possui a característica de melhorar o desempenho de preditores instáveis que são basicamente os preditores com alta variância.

Considerando uma amostra de treinamento $L = \{x_i, y_i\}_{(i=1)}^{N}$, e um preditor $\hat{d}_n(x)$, que foi construído com base na amostra L, o procedimento de *Bagging* pode ser definido por meio do seguinte algoritmo:

1. Por meio do uso de *boostsrap* deve ser criada uma amostra $L^* = \{x_i^*, y_i^*\}_{(i=1)}^{N}$, de acordo com a distribuição empírica de L.

[1] Técnica ou método de simulação, que objetiva a obtenção de intervalos de confiança para as estimativas dos parâmetros de interesse, por reamostragem do conjunto de dados original.

2. Computar o preditor por meio do uso de *booststrap* $\hat{d}_n^*(x)$, utilizando o mesmo procedimento utilizado anteriormente para criar o preditor original $\hat{d}_n(x)$, agora com a amostra L^*.

3. Por fim a classe preditiva construída por meio do classificador *bagging* é a mais votada pelos classificadores $\hat{d}_n^*(x)$.

Resumindo a implementação de *bagging* é realizada por meio da extração de K amostras de tamanho N, com reposição, de um conjunto de dados. Assim a cada amostra *bootstrap* k, uma dada observação do conjunto original pode não aparecer, ou aparecer mais de uma vez, sendo que um novo preditor é gerado em cada uma destas amostras. O preditor gerado no algoritmo *bagging* resulta na agregação destes K preditores.

Em Breiman (1996) é apresentado que a variância do preditor via *bagging* é menor ou igual a do preditor sem *bagging*, enquanto que o vício ao quadrado do preditor via *bagging* permanece próximo ao do preditor sem *bagging*. Por conta desta característica o preditor *bagging* funciona bem somente para preditores com alta variância e instáveis. Os preditores instáveis são aqueles que sofrem grandes variações nos preditores quando ocorrem pequenas perturbações nos dados de treinamento. Os métodos instáveis mais conhecidos são Redes Neurais, Árvore de Classificação e a regressão e seleção de variáveis em regressão linear via teste.

2.7.1 RANDOM FOREST

Árvores de decisão são forma de representar graficamente um processo sequencial de decisões. Uma árvore de decisão contem nós de decisões, cada um com ramificações para cada uma das decisões alternativas. Os nós de probabilidade (variáveis aleatórias) também são representados na árvore, onde a utilidade de cada ramificação é computada nas respectivas folhas. Assim, a utilidade esperada de qualquer decisão pode ser calculada com base no somatório ponderado de todas as ramificações da decisão, para cada uma das folhas do ramo.

Breiman (2001) propôs o classificador chamado *Random Forest* que é baseado na combinação de preditores de árvores, de tal forma que cada árvore depende dos valores de um vetor aleatório amostrado de forma independente e com a mesma distribuição para todas as árvores na floresta.

Portanto podemos definir que o classificador Random Forest consiste na coleção de classificadores estruturados em árvore $h(x, \Theta_k, k = 1, \dots)$ onde Θ_k são vetores aleatórios distribuídos de forma idêntica independente e cada árvore lança um voto voluntário para classe mais popular na entrada x.

A convergência de um classificador Random Forest pode ser demonstrado com o uso de um conjunto de classificadores $h_1(x), h_2(x), \ldots, h_k(x)$ e com o conjunto de treino sorteado da distribuição do vetor aleatório X, Y. A função de margem é definida por:

$$mg(X,Y) = av_k I(h_k(X) = Y) - \max_{j \neq Y} av_k I(h_k(X) = j) \tag{2.43}$$

onde I é a função do indicador que é igual a 1 quando $h_k(X) = Y$ e 0 em caso contrário.

A margem mede até que ponto a média (av) de número de votos em X, Y para a classe correta excede a média de votos para qualquer outra classe. Portanto quanto maior a margem, maior confiança na classificação. O erro de generalização do classificador é dado por:

$$PE^* = P_{X,Y}(mg(X,Y) < 0) \tag{2.44}$$

onde os subscritos X, Y indicam qual a probabilidade está acima do espaço X, Y.

Em florestas aleatórias, $h_k(X) = h(X, \Theta_k)$. Para muitas árvores, decorre da Lei dos Grandes Números e da estrutura da árvore que conforme o número de árvores cresce, com um certo grau de certeza todas as sequências $\Theta_1 \ldots PE^*$ converge para:

$$P_{X,Y}(P_\Theta(h(X,\Theta) = Y) - \max_{j \neq Y} P_\Theta(h(X,\Theta) = j) < 0) \tag{2.45}$$

Este resultado demonstra porque as florestas aleatórias não se adaptam à medida que mais árvores são adicionadas, mas produzem um valor limitante de erro de generalização.

2.8 BOOSTING

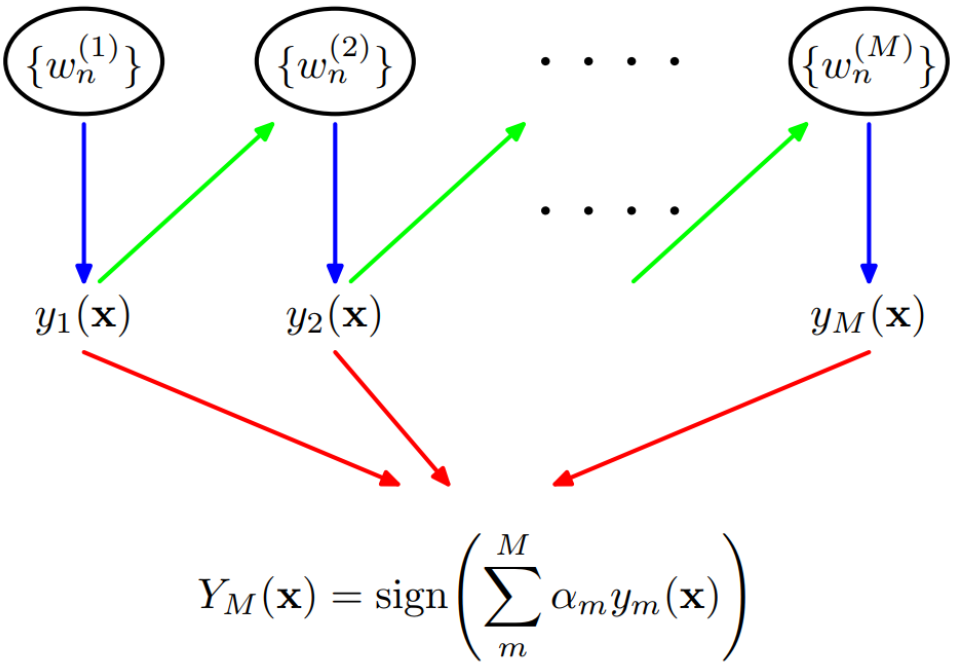
Como apresentado por Schapire (1990) o método de *Boosting* se refere a família de algoritmos que consegue converter preditores fracos em fortes preditores. Estes algoritmos surgiram a partir do problema teórico chamado de “problema de *boosting*”, que era exposto como: “Caso exista um método de classificação que seja pouco superior que uma classificação aleatória, para qualquer conjunto de dados X, este preditor será chamado de *weak learner*, ou preditor fraco. Assim a existência deste preditor fraco implica na existência de um preditor forte, que será chamado de *strong learne*r, com pequeno erro sobre o conjunto de dados X.

A principal diferença entre estes algoritmos e os métodos de *Bagging*, é que os preditores são treinados em sequência, e cada preditor é treinado usando uma forma

ponderada do conjunto de dados na qual o coeficiente de ponderação associado a cada ponto de dados depende do desempenho dos classificadores anteriores, como descrito em Bishop (2006). Em particular, os pontos que são classificados erroneamente pelos preditores base recebem um peso maior quando treinados pelo próximo classificador. Uma vez que todos os classificadores tenham sido treinados, suas previsões são combinadas por meio de um esquema de votação por maioria ponderada.

Conforme demonstrado na Figura 10, no algoritmo *Boosting* cada classificador de base $y_m(x)$ é treinado em uma forma ponderada do conjunto de treinamento, em que os pesos $w_n^{(m)}$ dependem do desempenho do classificador base anterior $y_{m-1}(x)$. Após todos os classificadores serem combinado é realizado o cálculo do classificador ponderado final $Y_M(x)$.

Figura 10: Esquema do classificador *Boosting*.



Fonte: Bishop (2006)

2.8.1 ADABOOST

O classificador *AdaBoost* é o mais conhecido dos algoritmos de *Boosting* proposto por Freund e Schapire (1997), que possui como principal objetivo encontrar a hipótese final com o menor erro na base de treinamento.

O algoritmo do classificador *AdaBoost* mantêm um conjunto de pesos sobre os dados de treinamento $w_n^{(1)}$. Durante a iteração m é calculada a distribuição J_m normalizando os pesos. Esta distribuição alimenta os preditores fracos, que geram a hipótese ε_m e que devem possuir erros menores com relação a distribuição. Utilizando

esta nova hipótese ε_m, o algoritmo de *boost* gera um novo vetor de pesos $w_n^{(m+1)}$, e o processo se repete. Após as M iterações, a hipótese final $Y_M(x)$ é gerada por meio da combinação dos resultados de M hipóteses fracas usando o voto majoritário ponderado. O detalhamento do funcionamento do algoritmo do classificador *AdaBoost* é apresentado a seguir:

1. Inicialize os coeficientes de ponderação $w_n^{(1)} = \frac{1}{N}$ para $n = 1, \ldots, N$.
2. Inicia a repetição, tendo como parâmetro de parada $m = 1, \ldots, M$:
   a) Ajuste o classificador $y_m(x)$ para o treinamento dos dados minimizando os pesos da função de erro:

$$J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n) \tag{2.46}$$

   onde $I(y_m(x_n) \neq t_n)$ é a função indicador que é igual a 1 quando $y_m(x_n) \neq t_n$ e 0 em caso contrário.

   b) Avalia as quantidades:

$$\varepsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(x_m) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}} \tag{2.47}$$

   este resultado é utilizado para avaliar:

$$\alpha_m = \ln\{\frac{1 - \varepsilon_m}{\varepsilon_m}\} \tag{2.48}$$

   c) Atualizar os coeficientes de ponderação:

$$w_n^{(m+1)} = w_n^{(m)} \exp\{\alpha_m I(y_m(x_n) \neq t_n)\} \tag{2.49}$$

3. Faça previsões usando o modelo final, que é dado por:

$$Y_m(x) = sign(\sum_{m=1}^{M} \alpha_m y_m(x)) \tag{2.50}$$

2.8.2 GRADIENT BOOSTING

O modelo de *Gradient Boosting* concebido por Friedman (2001) é um procedimento de *Ensemble* que combina várias árvores de decisão para criar um modelo mais poderoso. O que diverge estes modelos do *Random Forest*, é que o modelo de *Gradient Boosting* as árvores são construídas de forma serial, onde cada árvore busca corrigir o erro da árvore anterior.

Por padrão, os modelos de *Gradient Boosting* não geram árvores de forma randômica, sendo que é realizada uma forte poda nas árvores utilizadas. Assim possui

modelos com árvores rasas tendo profundidade de um a cinco, o que torna este tipo de modelo mais rápidos e com baixo consumo de memória, principalmente quando comparado com o classificador *Random Forest*

Como os modelos em Ensemble em geral, o principal objetivo dos modelos de *Gradient Boosting* é combinar muitos modelos simples, *weak leaners* que neste caso são as árvores rasas, sendo que cada árvore consegue realizar previsões de uma pequena parte dos dados, assim são geradas novas árvores iterativamente.

A seguir será apresentado o algoritmo genérico *Gradiente Tree-Boosting* para regressão. Para especificar o algoritmo se faz necessário inserir diferentes critérios de perda L conforme demonstrado por Trevor, Robert e JH (2009) e apresentado na Tabela 1.

Tabela 1: Funções de perda comumente usadas para Gradientes.

Configuração	Função de Perda	$-\partial L(y_i, f(x_i))/\partial f(x_i)$
Regressão	$\frac{1}{2}[y_i - f(x_i)]^2$	$y_i - f(x_i)$
Regressão	$\lvert y_i - f(x_i)\rvert$	$sign[y_i - f(x_i)]$
Regressão	Huber	$y_i - f(x_i)$ para $\lvert y_i - f(x_i)\rvert \leq \delta_m$ $\delta_m sign[y_i - f(x_i)]$ para $\lvert y_i - f(x_i)\rvert > \delta_m$ Quando $\delta_m = \alpha$th-quantile$\{\lvert y_i - f(x_i)\rvert\}$
Classificação	Desvio	kth componente: $I(y_i = \mathcal{G}_k) - p_k(x_i)$

Fonte: Trevor, Robert e JH (2009)

No primeiro passo do algoritmo inicializa-se a constante do modelo ótima, que é apenas uma árvore de nó de terminal mínimo. No passo seguinte são calculados os componentes do gradiente negativo, que se referem aos pseudo-resíduos generalizados e em seguida é utilizada a função de perda, conforme detalhamento a seguir:

1. Inicializar $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$.

2. Inicia o repetição, tendo como parâmetro de parada $m = 1, \ldots, M$:

   a) Para $i = 1, \ldots, N$ computar:

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_m=1} \tag{2.51}$$

   b) Ajustar uma árvore de regressão aos alvos $r_i m$ dando regiões terminais.

$$R_{im,j} = 1, 2. \ldots, J_m \tag{2.52}$$

   c) Para $j = 1, \ldots, J_m$ computar:

$$\gamma_{im} = \arg\min \sum_{x_i \in R_j m} L(y_i, f_{m-1}(x_i) + \gamma) \tag{2.53}$$

d) Atualizar $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{im} I(x \in R_{jm})$.

3. O resultado será $\hat{f}(x) = f_M(x)$.

2.9 BASES DE DADOS DESBALANCEADAS

Uma das características marcantes das bases de dados que possuem informações de fraude é o desbalanceamento. Isto ocorre pois em sua grande maioria os eventos de fraudes são eventos raros que ocorrem com baixa frequência. Esta particularidade causa um efeito negativo no processo de modelagem do evento, pois sem um olhar mais cuidadoso na base de dados o evento de fraude pode passar despercebido.

Neste trabalho os eventos que iremos modelar representam uma pequena fração da base sendo que a fraude na concessão ocorre em $3,75\%$ da base, o evento de operação nunca paga ocorrem em $4,33\%$ e por fim o evento de existência de ação contrária ocorre em apenas $3,8\%$ da base.

Caso um modelo seja executado nesta base dados de forma tradicional, o comum seria encontrar uma acurácia de aproximadamente 96%, pois o modelo iria ignorar o evento de fraude e marcar todas as operações como “boas”, sendo assim ineficientes para identificar este tipo de evento.

Assim, se faz necessária a utilização de abordagens específicas para este tipo de base de dados, utilizando abordagens sofisticadas de amostragem que permitem modelar de forma eficiente o evento de fraude. He e Garcia (2008) apresentam em seu trabalho os principais métodos utilizados para modelagem em base de dados desbalanceadas, sendo que neste trabalho foi utilizada a técnica de *Random Undersampling*.

Para formular os exemplos de amostragem é necessário definir como notação a base de treinamento S com m exemplos $(i.e., |S| = m)$, onde $S = \{(x_i, y_i)\}, i = 1, \ldots, m$, e $x_i \in X$ que é uma instância n-dimensional do espaço das variáveis $X = \{f_1, f_2, \ldots, f_n\}$ e $y_i \in Y = \{1, \ldots, C\}$ é um rótulo de identidade de classe associado à instância x_i, que também chamamos de variável(eis) resposta(s).

Como característica para as bases de dados desbalanceadas temos a existência de classes minoritárias e majoritárias, assim será representada a classe minoritária como $S_{min} \subset S$ e a classe majoritária $S_{maj} \subset S$, assim $S_{min} \cap S_{maj} = \{\emptyset\}$ e $S_{min} \cup S_{maj} = \{S\}$. Adicionalmente as amostras extraídas da base de treinamento S serão denominadas E.

2.9.1 RANDOM OVERSAMPLING E UNDERSAMPLING

O procedimento de *oversampling* aleatório tem como mecanismo criar uma amostra da classe minoritária maior que a disponível, ou seja, é criado um conjunto de exemplos selecionados de forma aleatória da classe minoritária S_{min}, que são inseridos na amostra E, posteriormente esta subamostragem é inserida na base de testes original S. Desta forma, o número total de exemplos em S_{min} é incrementado por $|E|$ e o saldo da distribuição das classes de S é ajustado, reduzindo assim o desbalanceamento. Este procedimento permite ajustar o grau de equilíbrio das classes para qualquer nível desejado.

Por outro lado, o procedimento de *undersampling* possui um mecanismo inverso, construindo uma base de testes menor do que a original, reduzindo informações da classe majoritária. Para que este processo seja possível, é realizada uma seleção randômica em S_{maj} e esta seleção é removida da base de testes S, realizando assim $|S| = |S_{min}| + |S_{maj}| - |E|$. De tal modo que de forma similar ao realizado no método anterior é possível também ajustar o grau de equilíbrio entre as classes.

2.10 SELEÇÃO DE VARIÁVEIS

Neste trabalho foram utilizadas duas técnicas para seleção de variáveis, pois a quantidade de variáveis disponibilizada é grande e um modelo contendo todas estas variáveis possui alta probabilidade de *overfitting*, que ocorre quando o modelo só possui boa performance na base de treinamento.

Portanto foi selecionado para os modelos lineares a técnica de *Stepwise*, já para os modelos que utilizam métodos de aprendizado de máquina utilizamos o procedimento de seleção sequencial usando o modelo de *Random Forest*.

A seguir serão apresentados maiores detalhamentos e embasamento teórico das metodologias escolhidas neste trabalho.

2.10.1 *STEPWISE*

O método *Stepwise* é o mais comumente utilizado para regressões lineares, ele possui como princípio a inclusão e remoção de variáveis do modelo, avaliando por meio de um teste F, desde que os erros tenham distribuição normal, a média de significância estatística do coeficiente associado à variável do modelo. Portanto a variável escolhida será a que causar maior mudança no logaritmo da verossimilhança em relação ao modelo que não possui a variável em questão.

O método de *Stepwise* segue o seguinte algoritmo, considerando a existência de p variáveis disponíveis:

1. No princípio é construído o modelo que possui apenas o intercepto, que é o β_0 da regressão logística, e o logaritmo da verossimilhança L_0 deste ajuste. Logo em seguida é realizado o ajuste dos p modelos que possuem apenas uma variável explicativa. Portanto $L_j^{(0)}$ é o log da verossimilhança do modelo que contem a variável x_j.

   O teste da Razão de Verossimilhança do modelo contendo x_j com relação ao modelo contendo apenas o intercepto será $G_j^{(0)} = 2-(L_0-L_j^{(0)})$ e o seu respectivo p-valor será calculado por $p_j^{(0)} = P[x_v^2 > G_j^{(0)}]$, onde $v = 1$ se x_j for contínuo e $v = k - 1$ caso x_j for categórico com k categorias.

   Considerando $p_{e_1}^{(0)}$ como o p-valor associado ao teste da variável x_1 e também, $p_{e_1}^{(0)} = min(p_j^{(0)})$, sendo assim o menor p-valor de todos os teste da Razão de Verossimilhança, caso $p_{e_1}^{(0)} < \alpha_e$ então a variável x_1 irá entrar no modelo. Neste caso consideramos α_e como o nível de significância para que a variável seja considerada no modelo, sendo que Lee e Koval (1997) demonstraram em seu trabalho que o valor recomendado para $\alpha_e = 0,15$ e para $\alpha_r = 0,20$.

2. O passo seguinte inicia-se com a variável selecionada no passo anterior que possui o menor p-valor estatisticamente significante. Assim é realizada a inclusão da variável x_2, entretanto a inclusão desta nova variável pode fazer com que a variável x_1 passe a não ser mais significativa, assim é realizado o teste de significância de uma das variáveis dado que a outra está no modelo.

   Para decidir se alguma variável é removida ou não é sempre selecionado o maior p-valor. O que garante que o algoritmo continue é que a inclusão da nova variável possua um p-valor inferior a α_e, caso contrário o processo para o modelo é explicado apenas pelas variáveis atuais.

3. Este passo consiste em um laço de repetição, sendo que uma nova variável é sempre incluída ou removida do modelo final, sempre realizando ajuste garantindo que com uma nova variável não ocorra perda de significância.

4. O final ocorre quando todas as variáveis foram utilizadas no modelo ou quando as variáveis utilizadas no modelo possuem p-valor inferior a α_r e todas as variáveis não incluídas no modelo possuem p-valor superior a α_e.

2.10.2 SELETOR DE VARIÁVEL SEQUENCIAL

Neste trabalho para os modelos que utilizam métodos aprendizado de máquina além do filtro inicial por ganho de informação, foi utilizado algoritmos de busca gulosos que visam reduzir o espaço de variáveis d-dimensional para um subespaço de variáveis k-dimensional, sendo $k < d$. Assim buscou-se selecionar automaticamente um subconjunto de variáveis mais relevantes para o modelo.

Basicamente o algoritmo consiste em adicionar e remover variáveis com base no desempenho do classificador até que um subconjunto com melhor performance seja alcançado. Para este trabalho o indicador de desempenho utilizado foi a curva *ROC*, que será detalhada mais a frente, garantindo um melhor conjunto de variáveis que classifica o evento de fraude, mas que ao mesmo tempo se preocupa com os falsos positivos.

No algoritmo apresentado em Ferri et al. (1994) que descreve o procedimento utilizado neste trabalho, são utilizadas todas as variáveis disponíveis que iremos chamar de $Y = \{y_1, y_2, \ldots, y_n\}$ e o resultado será $X_k = \{x_j | j = 1, 2, \ldots, k, x_j \in Y\}$ onde $k = (0, 1, 2, \ldots, d)$.

O algoritmo possui o seguinte funcionamento:

1. Inicializa $X_0 = \emptyset, k = 0$.
2. Enquanto $k \neq d$ são realizados os seguintes passos:
   a) $x^+ = \arg\max J(x^k + x)$, onde $x \in Y - x_k$
   b) $X_{k+1} = x_k + x^+$
   c) $k = k + 1$

A função J representa o processo de seleção do critério de escolha, neste trabalho é executado um modelo *Random Forest* sendo utilizado como critério de avaliação a curva *ROC*, após a execução deste procedimento, x^+ é a variável selecionada por maximizar o critério de escolha no subconjunto X_k.

Após a execução dos algoritmos obtemos uma tabela com todas as iterações, representando todas as variáveis do modelo, onde são apresentados os melhores modelos com uma variável até a quantidade total de variáveis. Em seguida é selecionado o resultado com a maior valor *ROC* e a menor quantidade de variáveis.

2.11 MÉTRICAS DE AVALIAÇÃO DE DESEMPENHO DO MODELO

Do mesmo modo que desenvolver modelos para bases de dados desbalanceadas exige métodos específicos para extração da base de treinamento, o mesmo ocorre para as métricas de avaliação dos modelos.

Como citado por He e Garcia (2008) tradicionalmente, a métrica mais utilizada para mensurar a efetividade de um modelo é a taxa de erro e a acurácia que calcula a precisão, a fração ou a contagem das previsões corretas conforme fórmula a seguir:

$$accuracy(y, \hat{y}) = \frac{1}{n_{amostras}} \sum_{i=0}^{n_{amostras-1}} 1(\hat{y}_i = y_i) \tag{2.54}$$

onde y é o valor real da variável resposta, $\hat{y}$ o valor calculado pelo modelo e $n_{amostras}$ o tamanho da amostra utilizada.

O grande problema em utilizar apenas métricas de acurácia é que ele pode gerar um resultado enganador em certas circunstâncias. Por exemplo, em casos em que o desbalanceamento seja de 5% para a classe minoritária, caso seja utilizado um classificador ingênuo, o modelo irá classificar todos os elementos como sendo da classe majoritária, obtendo assim um índice de acerto de 95%, desconsiderando por completo o evento da classe minoritária.

A representação da performance da classificação também pode ser formulada por meio da matriz de confusão que descreve os custos de se fazer atribuição das classes às amostras, portanto cada uma das linhas da matriz contém cada classe passível de ser atribuída às instâncias pelo classificador, enquanto cada coluna possui a classe à qual as amostras efetivamente pertencem na realidade.

Na Tabela 2 é apresentado um exemplo, com informações fictícias, da matriz de confusão com o exemplo de operações com a marcação de nunca pagas.

Tabela 2: Matriz de Confusão.

	Nunca Paga (Real)	Paga (Real)
Identificada como nunca Paga	100	20
Identificada como paga	30	100

Fonte: Elaboração Própria

Observando a matriz de exemplo, podemos verificar que o modelo acerta em 200 operações, sendo que 100 delas foram pagas e 100 não pagas. Entretanto existe erros no modelo sendo que a existência de 20 falso positivos, onde o modelo diz que existem 20 operações que nunca pagas, mas que foram pagas e a presença de 30 falso negativos, operações que nunca foram pagas e o modelo não conseguiu identificar.

Sendo assim a matriz de confusão fornece uma sensibilidade de como o modelo identifica as operações, não se concentrando apenas no acerto, tendo em vista que no caso de detecção de fraudes o falso positivo pode ser um problema que caracteriza a perda de negócio e deterioração do relacionamento com o cliente.

Outras métricas que são frequentemente adotadas na comunidade de pesquisa para fornecer avaliações abrangentes em base de dados desbalanceadas são: *precision*, *recall* e *F-mensure*.

A métrica de ***Precision*** fornece a informação de precisão, que é a relação:

$$Precision = \frac{tp}{(tp + fp)} \tag{2.55}$$

onde tp é o número de positivos verdadeiros e fp é o número de falsos positivos. Portanto podemos dizer que está métrica é intuitivamente a capacidade do classificador não rotular como positiva uma amostra que é negativa. O melhor valor para esta métrica é 1 e a pior 0. Uma breve avaliação sobre este indicador permite verificar que ele é não dependente da distribuição.

A métrica de ***Recall*** fornece informação de recordação, que é a relação:

$$Recall = \frac{tp}{(tp + fn)} \tag{2.56}$$

onde tp é o número de positivos verdadeiros e fn é o número de falsos negativos. Portanto podemos dizer que está métrica é intuitivamente a capacidade do classificado encontrar todas as amostras positivas. O melhor valor para esta métrica é 1 e a pior 0. Diferente do indicador de Precision este indicador possui dependência da distribuição.

A métrica de ***F-Mensure***, que também é conhecida como *F-score* ou *F1 score*, é a combinação das métricas anteriores de recordação e precisão e pode ser interpretada como uma média ponderada destas métricas, assim a métrica *F-Mersure* alcança o seu melhor valor em 1 e o pior em 0. Pode ser representado pela relação:

$$\textit{F-Mensure} = 2 * \frac{precision * recall}{precision + recall} \tag{2.57}$$

Embora estas métricas tenham melhorias com relação a acurácia, elas se mantêm não efetivas quando é necessário comparar a desempenho de diferentes classificadores aplicados em uma mesma amostra de dados, portanto existem técnicas adicionais como as apresentadas a seguir.

2.11.1 RECEIVER OPERATING CHARACTERISTICS (ROC) CURVES

Fawcett (2004) propôs a técnica de avaliação *ROC*, que faz uso da proporção de duas métricas de avaliação baseadas em coluna única, ou seja, a taxa de positivos reais (TP_{rate}) e a taxa dos falsos positivos (FP_{rate}), que são definidos da seguinte forma:

$$TP_{rate} = \frac{TP}{positivos}; FP_{rate} = \frac{FP}{negativos} \tag{2.58}$$

onde TP é o número de positivos verdadeiros, $positivos$ é a soma de positivos verdadeiros com falso negativos, FP é o número de falsos positivos e negativos a soma de falso positivos com negativos verdadeiros, o melhor detalhamento pode ser encontrado na tabela 3.

Tabela 3: Matriz de Confusão para cálculo da curva ROC.

		Classe dos Verdadeiros	
		P	**N**
Classe das hipóteses	**S**	Positivos Verdaderos	Falso Positivos
	N	Falso Negativo	Negativo Verdadeiro
Total das Colunas		**positivos**	**negativos**

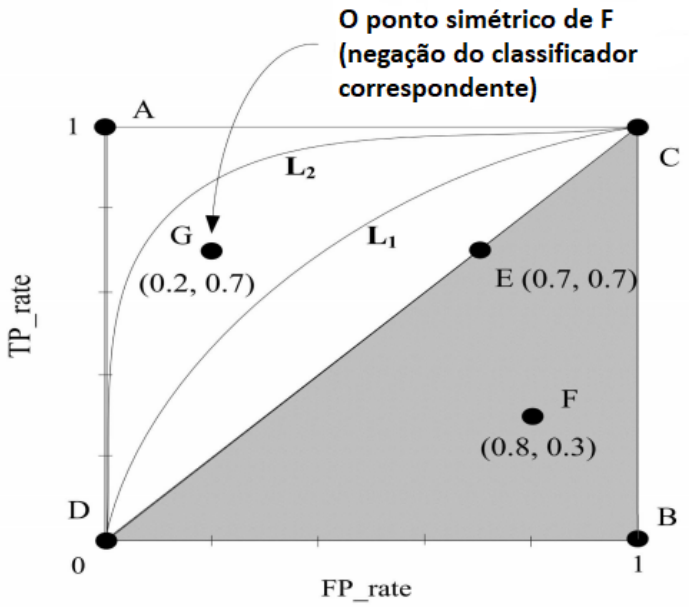
Fonte: Elaboração Própria

O gráfico da curva ROC é formado pela TP_{rate} sobre a taxa FP_{rate}, onde cada ponto no espaço *ROC* corresponde ao desempenho de um único classificador em uma determinada distribuição.

O ponto positivo em utilizar a curva *ROC* é a possibilidade de observar de forma visual as compensações relativas entre os benefícios (que são representados pelos verdadeiros positivos) e os respectivos custos (que são representados pelos falsos positivos) de classificação.

De um modo geral um classificador é melhor do que outro se o seu ponto correspondente no espaço *ROC* estiver mais próximo do ponto A, apresentado na Figura 11, do que o outro.

Figura 11: Representação da Curva ROC.



Fonte: Haibo e Garcia (2008)

Quando o classificador fica na linha diagonal, como o ponto E na Figura 11, significa que ele irá fornecer uma estimativa aleatória dos rótulos de classe (ou seja, um classificador aleatório, como jogar uma moeda não viciada). Portanto qualquer classificador que aparece no triângulo inferior direito do espaço do *ROC* tem um desempenho pior do que adivinhações aleatórias, como o classificador representado pela letra F.

2.11.2 TESTE KOLMOGOROV–SMIRNOV

O teste KS é um teste não paramétrico sobre a igualdade de distribuições de probabilidades contínuas que fornece um meio de avaliar quando um conjunto de observações é proveniente de uma específica distribuição contínua $F_0(x)$.

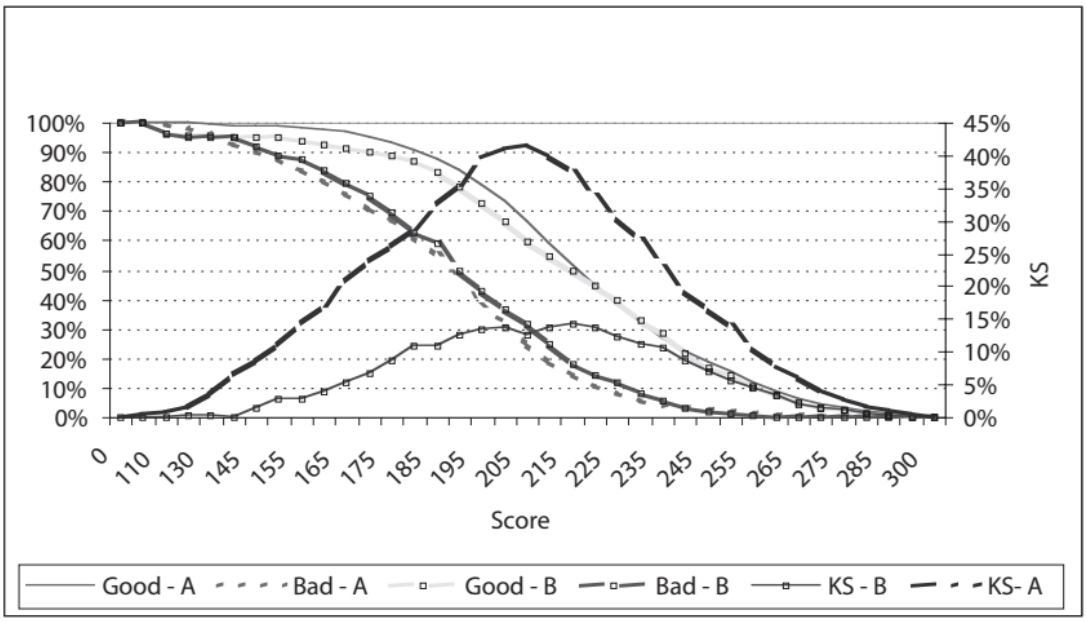
A estatística do teste pode ser descrita por:

$$D = \max_x |F(X) - F_n(X)| \tag{2.59}$$

onde $F_n(X)$ é a função de distribuição acumulada empírica dos dados e $F(X)$ é a função de distribuição acumulada assumida para os dados, portanto o valor D representa a maior distância entre as duas distribuições.

Como o exemplo apresentado em Siddiqi (2012) na Figura 12 é representada a utilização do teste de Kolmogorov-Smirnov em dois modelos de Score, que visa medir a qualidade creditícia de um cliente em uma instituição financeira.

Figura 12: Exemplo de aplicação do teste KS.



Fonte: Siddiqi (2006)

Os modelos são A e B e podemos verificar na figura que o modelo A possui um

valor de KS melhor pois alcança o desvio máximo de aproximadamente 41% enquanto o modelo B possui um valor de KS de apenas 15%.

A maior fragilidade no uso do teste Kolmogorov-Smirnov está no fato de a medida ocorrer em apenas um ponto, que é a maior distância entre as distribuições, o que pode não representar o ponto de corte esperado por não conter todo o intervalo de pontuação.

3 METODOLOGIA

O trabalho possui como objetivo identificar em um conjunto de operações quais são fraudulentas e quais são genuínas, entretanto neste trabalho o desafio é maior pois não existe apenas uma variável a se identificar, e sim 3 que podem de formas complementares indicar a ocorrência de fraude. Portanto por meio de uma base de dados que contenha todas as informações, com todos os tipos de operações, o objetivo é construir um modelo que seja capaz de identificar quais as operações possuem os indícios de fraude ao mesmo tempo.

O procedimento adotado neste trabalho possui 4 etapas, sendo elas o pré-processa-mento, construção das amostras, seleção das variáveis e aplicação efetiva dos modelos, sendo que todas as etapas foram executadas por meio da linguagem de programação *Python*, com exceção do procedimento de *Stepwise* que foi executado por meio do software estatístico *R*.

O procedimento de pré-processamento consiste em uma avaliação detalhada da base de dados onde é avaliada a qualidade das informações disponíveis. Nesta etapa é avaliado a presença de *missings*, que consistem em ausência de valor em determinada variável. Também é avaliada a presença de multicolinearidade entre as variáveis que pode acarretar em efeitos negativos nos modelos, baixa importância nas variáveis fazendo com que elas não auxiliem a identificar o evento de fraude e, por fim, a presença de valores únicos, que faz com que as variáveis possuam comportamento semelhante à de uma constante. Importante citar que estas etapas são executadas de forma sequencial, ou seja, uma após a outra.

A segunda etapa do trabalho é focada nas variáveis, sendo que após a execução da etapa anterior é realizado um processo de categorização das variáveis contínuas. Ou seja, as variáveis contínuas são transformadas em variáveis discretas por meio de análise bivariada.

O passo seguinte é construção das bases amostrais de treinamento e validação, além da seleção das variáveis que apresentem maior poder para identificar os eventos de interesse. Para os modelos lineares é utilizando o método de *Stepwise* para seleção de variáveis, por outro lado os modelos que utilizam aprendizado de máquina é utilizado procedimento de busca heurística utilizando modelo *Random Forest*, visando assim identificar a melhor combinação de variáveis utilizando com métrica de avaliação a curva *ROC*.

Por fim são aplicados os modelos propostos pelo trabalho, utilizando metodologia de regressão logística e outros aplicando métodos de aprendizado de máquina.

Após a execução dos modelos serão aplicados testes estatísticos visando identificar quais modelos possuem maior poder de predição dos eventos que compõe a fraude.

3.1 BASE DE DADOS

Os dados utilizados neste trabalho pertencem a empresa Serasa Experian, portanto para que fosse possível utilizar as informações neste trabalho foi necessário a assinatura de um acordo de não divulgação *NDA (Non Discloure Agreement)*. A primeira etapa do trabalho foi identificar as informações disponíveis para avaliar quais variáveis possuem maior possibilidade de compor o modelo proposto.

A base de dados utilizada neste trabalho contém operações de mercado de diversos produtos que possuem eventos de fraude, contendo 55.972 registros com 229 variáveis distribuídos em um período de 14 meses. Os clientes e os meses em que as operações estão distribuídas não foram revelados pela empresa que forneceu os dados, sendo assim criado uma variável chave com valores sequenciais.

A base de dados possui 3 variáveis respostas, sendo elas *Fraude_na_concessão* (disponível na base de dados como *CONCEITO_conces*) que é um Indicador de fraude encontrada no momento da avaliação do proponente, *Conta_nunca_paga* (disponível na base de dados como *CONCEITO_Never*) que é o indicador de conta nunca paga em 365 dias e por fim *Ocorrência_ação_cível* (disponível na base de dados como *CONCEITO_acao*) que é o indicador de ocorrência de ação cível após negativação. A Tabela 4 apresenta a contagem de ocorrência dos eventos que podem indicar fraude por variável resposta.

Tabela 4: Distribuição dos domínios das variáveis resposta.

Variável *Conta_nunca_paga*

	Operação normal	Operação nunca paga
Quantidade	53.549	2.423

Variável *Ocorrência_ação_cível*

	Sem ação contrária	Com ação contrária
Quantidade	53.800	2.172

Variável *Fraude_na_concessão*

	Não detectada na concessão	Detectada na concessão
Quantidade	53.873	2.099

Fonte: Elaboração Própria

Verificamos que existe o problema de desbalanceamento da base para cada uma das variáveis resposta utilizadas no processo. Para a variável resposta *Fraude_na_concessão* existe na base a ocorrência de apenas 2.099 casos de fraude. Para

a variável de parcelas nunca pagas *Conta_nunca_paga* identificamos na base disponibilizada a ocorrência de 2.423 operações que nunca foram pagas no período de 356 dias e, por fim, na variável resposta *Ocorrência_ação_cível* foram identificadas a ocorrência de 2.172 casos de ações contrárias abertas pelos clientes.

3.2 PRÉ PROCESSAMENTO

Nesta etapa do trabalho foram realizadas avaliações detalhadas na base de dados visando garantir a consistência dos dados que serão utilizadas nos modelos, podendo assim mitigar problemas devido a inconsistência/problemas dos dados.

3.2.1 VALORES AUSENTES

O primeiro passo foi avaliar a existência de ausência de informações (*missing value*) nas variáveis disponibilizadas. A ausência de informação pode ter sido gerada por problemas na geração das informações ou nos processos de transferência entre sistemas.

Tabela 5: Proporção de missing nas variáveis disponibilizadas.

Variável	**Fração de Missing**
EMP_PESS_ASSAL	100,00%
OBITOSTOT	100,00%
PESSO_OCUP	100,00%
BOLSA_FAMILIA	87,82%
IDADE	0,03%

Fonte: Elaboração Própria

Como é possível verificar na Tabela 5 existem 3 variáveis com ausência total de informação. Neste caso frente a inviabilidade de utilização destas variáveis, elas foram removidas da base de dados.

Para a variável *BOLSA_FAMILIA*, foi realizada avaliação de frequência das informações presentes. Tendo em vista a descrição da variável como *Indicador de bolsa família* a interpretação utilizada é que a ausência de informação significa a não utilização deste programa social, assim o *missing* foi substituído por zero.

Para a variável *IDADE*, existia a possibilidade de se criar um modelo para identificar os valores ausentes, entretanto neste trabalho optou-se por inserir o valor médio disponível, tendo em vista a baixa ocorrência deste evento $(0,03\%)$, sendo assim substituídos pelo valor 41 (quarenta e um) todos os casos de ausência de informação.

3.2.2 TRANSFORMAR VARIÁVEIS CONTÍNUAS EM DISCRETAS

Para utilizar os modelos lineares e diminuir a sua complexidade o ideal é transformar as variáveis contínuas em variáveis discretas. Assim neste trabalho foi utilizado como técnica a separação por quartis das variáveis contínuas.

Adicionalmente a base de dados possui como particularidade a existência de informações com valores negativos, que possuem uma justificativa específica para este domínio, como por exemplo a variável *VQt_ConsCRDTOD_10d*. Verificamos que ela representa a quantidade de consultas aos relatórios de crédito (de todos os segmentos) nos últimos 10 dias, entretanto quando valores negativos são utilizados, representam a ausência de consulta para o segmento ou a ausência de consulta nos últimos 5 anos, sendo impossível obter esta informação com os valores positivos. Os domínios desta variável são apresentados na Tabela 6.

Tabela 6: Proporção de missing nas variáveis disponibilizadas.

Valor	Descrição
>0	Quantidade de consultas;
0	Não possui consulta para o período indicado;
-1	Não possui consulta para o segmento indicado;
-2	Não possui consulta (de nenhum segmento) nos últimos 5 anos.

Fonte: Elaboração Própria

Portanto ao realizar o procedimento de transformar as variáveis contínuas em discretas foi realizado um tratamento específico para a quantidade de valores negativos e iguais a zero, visando separá-los no cálculo dos quartis. Assim, cada valor negativo foi considerado como uma classe exclusiva, não um intervalo, sendo que os intervalos tiveram início em zero.

Como exemplo apresentamos a variável *Qt_Divida_Serasa_venc_regularizada* disponível na base de dados com o código: *VQtRestrSemTELECOMInc1Ares*, que representa a quantidade de dívidas Serasa vencidas regularizadas, incluídas por empresas diferentes de Telefonia no último ano:

- REFIN - Dívidas vencidas e não pagas referentes ao CPF/CNPJ consultado, da qual são fornecidas pelas empresas participantes do convênio REFIN. Ex: Bancos e financeiras,
- PEFIN - Dívidas vencidas e não pagas referentes ao CPF/CNPJ consultado, da qual são fornecidas pelas empresas participantes do convênio PEFIN, empresas de diversos segmentos da economia. Ex: Varejo, indústria, serviços, etc.

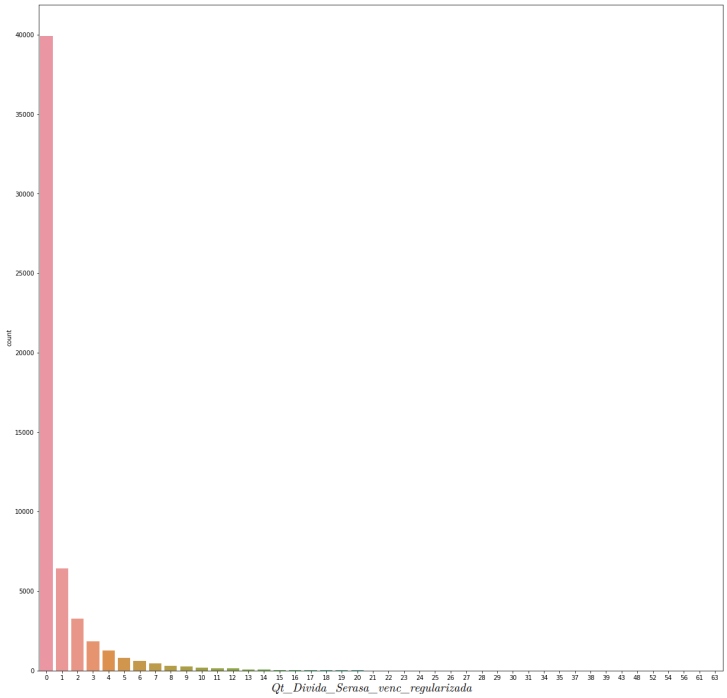
A distribuição do domínio da variável pode ser verificado na Figura 13. Adicionalmente podemos verificar as estatísticas descritivas da variável na Tabela 7.

Tabela 7: Estatísticas descritivas variável *Qt_Divida_Serasa_venc_regularizada*.

Variável: Qt_Divida_Serasa_venc_regularizada	
Média	0,9
Desvio Padrão	2,4
Mínimo	0
25%	0
50%	0
75%	1
Máximo	63

Fonte: Elaboração Própria

Figura 13: Frequência da variável *Qt_Divida_Serasa_venc_regularizada.*



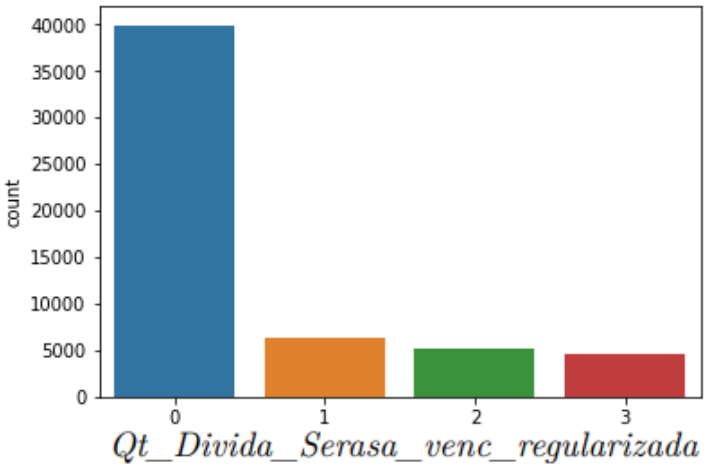
Fonte: Elaboração Própria

Para que fossem criados domínios com uma quantidade de domínios razoáveis neste trabalho optou-se por criar um máximo de 10 quartis, sendo eles de 10 em 10 $(10\%, 20\%, 30\%, ..., 90\%, 100\%)$. Para os quartis que não possuíam informações, eles foram agrupados.

Após a execução do procedimento para transformar a variável contínua em discreta, ela passou a possuir 4 domínios conforme Figura 14.

Por fim foi criada uma tabela contendo os domínios que devem ser considerados para as novas variáveis discretas que foram criadas neste processo. Na Tabela 8 é

Figura 14: Frequência da variável *Qt_Divida_Serasa_venc_regularizada* de forma discreta.



Fonte: Elaboração Própria

apresentado o detalhamento dos domínios gerados para a variável *Qt_Divida_Serasa-_venc_regularizada*.

Tabela 8: Domínio da variável *Qt_Divida_Serasa_venc_regularizada_disc*.

Domínio	Intervalo
0	]-1,0]
1	]0,1]
2	]1,3]
3	]3,64]

Fonte: Elaboração Própria

Desta forma quando a variável está no domínio 0 da variável construída, o seu valor será sempre igual a zero, entretanto quando o domínio for 3 a variável pode conter valores do intervalo entre 3 e 64.

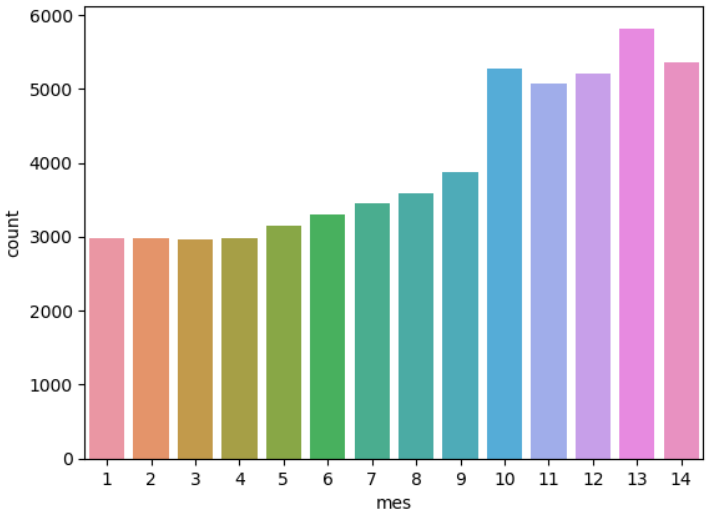
3.2.3 AMOSTRAGEM

O procedimento de amostragem foi parte muito importante deste trabalho, tendo em vista que o evento de fraude é de baixa frequência gerando assim bases de dados altamente desbalanceadas.

Antes de realizar a amostragem dos dados foram criadas as bases de teste e treinamento. A base de treinamento foi a utilizada para treinar o classificador, visando construir um modelo, por sua vez a base de teste é utilizada para identificar o poder de generalização do modelo, sendo que esta base não foi utilizada no desenvolvimento do modelo.

A base de dados disponibilizada possui a informação de “mês”, que neste caso é contínua não fornecendo detalhes exatos dos meses em que ocorreram, por questões de sigilo, como apresentado na Figura 15.

Figura 15: Distribuição da variável mês.



Fonte: Elaboração Própria

Assim, para a construção das bases de treinamento foram separadas todas as ocorrências dos meses marcados de 1 a 12, já para a base de teste foram selecionadas as ocorrências marcadas com os meses 13 e 14. Deste modo estamos utilizando os dados de um ano para treinar o modelo que visa identificar as ocorrências nos dois meses subsequentes.

O passo seguinte foi separar a base de treinamento em duas, sendo uma contendo a variável resposta e a outra contendo as possíveis variáveis que possuem poder de explicar o evento em questão. Como foram criados modelos diferentes para cada uma das variáveis respostas este procedimento foi realizado de forma apartada para cada uma delas.

Assim cada uma das 3 bases contendo a variável resposta possuem apenas uma das variáveis *Fraude_na_concessão*, *Conta_nunca_paga* e *Ocorrência_ação_cível*. Já para a construção da base com as variáveis explicativas, foram excluídas todas as possíveis variáveis resposta, além das variáveis auxiliares “mes” e “chave”, pois são características não relevantes para o modelo.

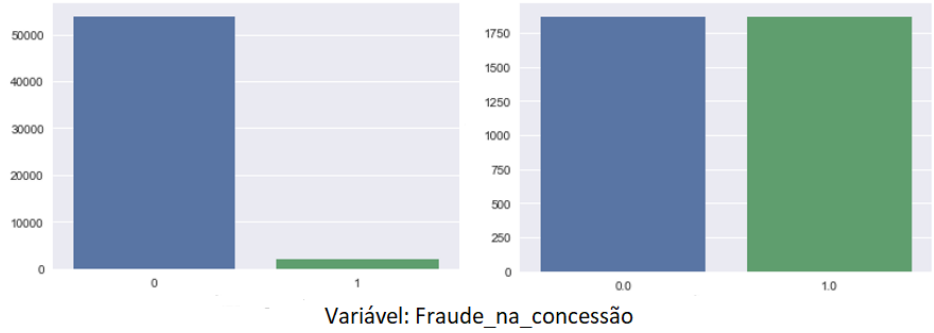
O passo seguinte foi importante devido ao alto desbalanceamento da base, pois as variáveis resposta representavam no máximo 4,5% de toda a base de dados, e criar modelos nesta situação não é uma boa alternativa, tendo em vista que o modelo pode simplesmente ignorar os casos de fraude tendo um bom percentual de acerto.

Portanto foi realizada uma amostragem da base de treinamento, visando deixar

o índice de desbalanceamento baixo, tornando a proporção de casos positivos iguais aos casos negativos.

Para tanto utilizamos o procedimento de *undersample* que seleciona todos os casos positivos, logo em seguida seleciona de forma randômica na base de teste a mesma quantidade de casos negativos, tornando assim a base balanceada (com a mesma quantidade de casos positivos e negativos, para este trabalho a variável resposta possui a mesma quantidade de zeros e uns). Por fim obtemos para cada uma das variáveis resposta uma amostra balanceada conforme apresentado na Figura 16.

Figura 16: Distribuição da variável *Fraude_na_concessão* antes e depois da amostragem.



Fonte: Elaboração Própria

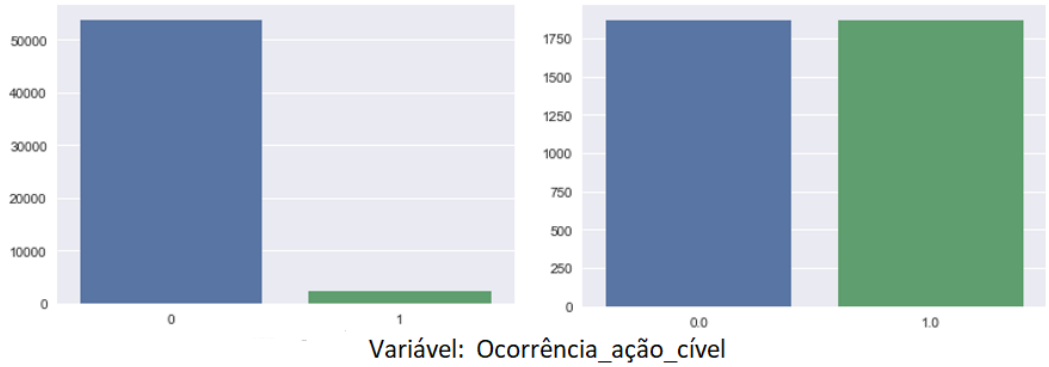
Figura 17: Distribuição da variável *Conta_nunca_paga* antes e depois da amostragem.

Fonte: Elaboração Própria

As bases possuem tamanhos diferentes, pois elas têm o dobro da quantidade de eventos que podem identificar uma fraude, para cada uma das variáveis resposta. Como exemplo a variável resposta *Fraude_na_concessão*, possui a ocorrência de 1.774 casos e a base de treinamento possui 3.548 observações. Este valor é menor que o total pois o restante foi separado para a base de teste.

Neste trabalho não foram utilizadas técnicas de *Cross Validation* pois o objetivo principal é comparar o desempenho de vários modelos sob as mesmas circunstâncias,

Figura 18: Distribuição da variável *Ocorrência_ação_cível* antes e depois da amostragem.



Fonte: Elaboração Própria

entretanto este procedimento é crucial quando se busca o melhor poder preditivo em um modelo que utilize técnicas de aprendizado de máquina.

3.2.4 MULTICOLINEARIDADE

Variáveis colineares são as variáveis que possuem alta correção entre si, ou seja, quando avaliadas em pares existe uma forte correlação entre elas. Nos modelos este comportamento gera uma redução no poder de generalização devido à alta variância e a menor interpretabilidade do modelo.

Assim aplicamos um método que identifica colinearidade acima de 98% entre as variáveis presentes na base de dados, de tal modo que dos pares de recursos com alta colinearidade apenas um é selecionado para remoção, sendo que apenas uma das variáveis precisa ser removida.

Para visualizar de forma adequada as correlações entre as variáveis, foi gerado um mapa de calor que identifica as variáveis que possuem ao menos uma correlação acima do limite estipulado de 98%.

Após a execução deste procedimento foram identificadas 49 variáveis que foram removidas da base de dados e estão disponíveis no Apêndice A.

3.2.5 BAIXA IMPORTÂNCIA

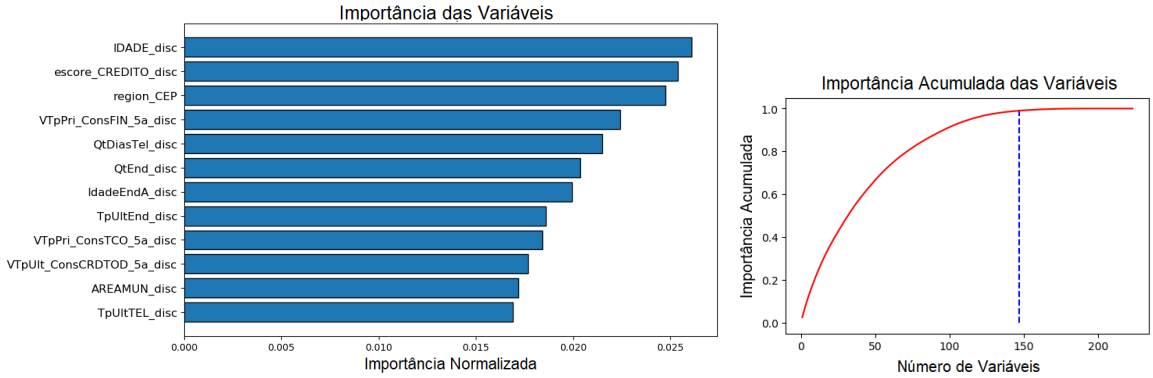
Os testes realizados anteriormente possuem caráter determinístico, sendo que dependem da base de dados como um todo sem a necessidade de um rótulo, ou seja, a variável resposta para serem executados. O procedimento adotado necessita das variáveis que iremos prever neste trabalho, buscando identificar quais variáveis possuem importância zero utilizando o modelo de aprendizado *Gradient Boosting*.

As variáveis que possuem zero importância não surtem efeito algum no modelo a ser desenvolvido, sendo assim desnecessárias para o modelo. Assim para aperfeiçoar o procedimento de seleção das variáveis elas foram retiradas da base de dados.

Como mencionado anteriormente este procedimento necessita de um rótulo para ser executado, e como neste trabalho possuímos três variáveis que devem ser previstas, este procedimento foi executado 3 vezes, um para cada uma das variáveis resposta disponíveis.

Para a variável *Fraude_na_concessão* foram identificadas 77 variáveis que não contribuem com a importância acumulada de 99%, sendo que 146 variáveis foram necessárias chegar a este nível, conforme gráficos da Figura 19.

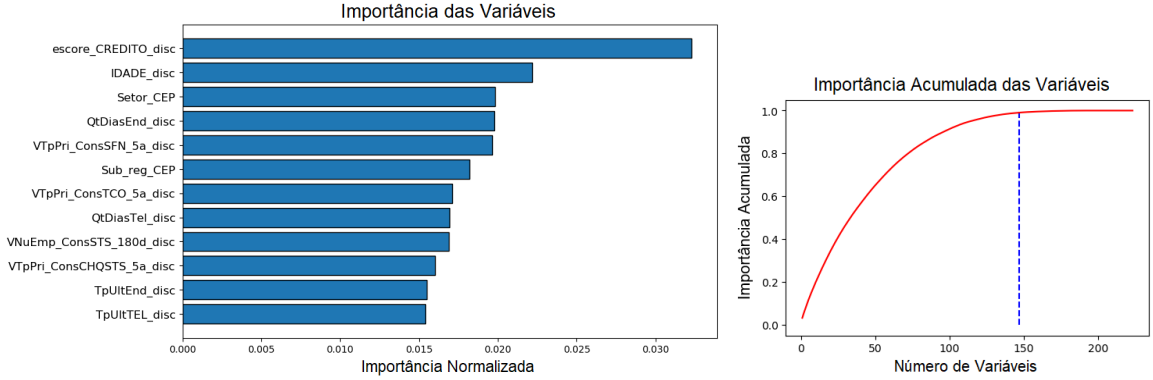
Figura 19: Avaliando importância das Variáveis - Rótulo *Fraude_na_concessão*.



Fonte: Elaboração Própria

Para a variável *Conta_nunca_paga* foram identificadas 77 variáveis que não contribuem com a importância acumulada de 99%, sendo que 146 variáveis foram necessárias chegar a este nível, conforme gráficos da Figura 20.

Figura 20: Avaliando importância das Variáveis - Rótulo *Conta_nunca_paga*

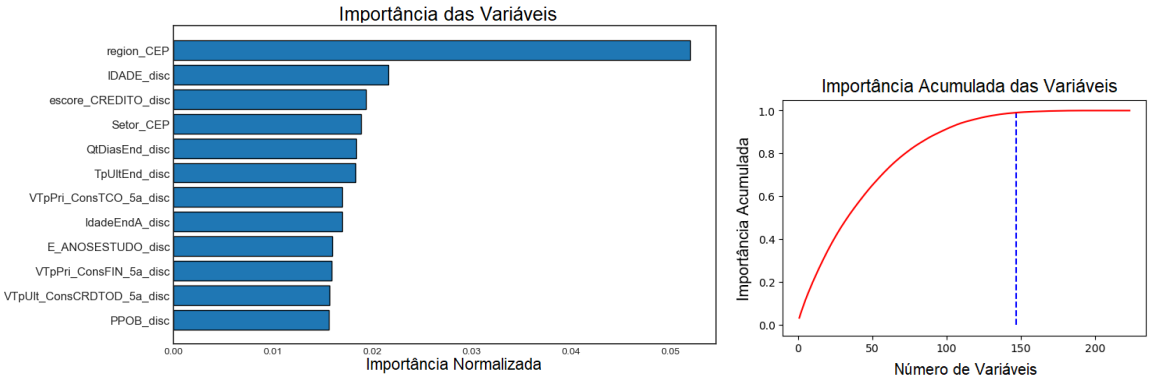


Fonte: Elaboração Própria

E por fim, para a variável *Ocorrência_ação_cível* foram identificadas 77 variáveis que não contribuem com a importância acumulada de 99%, sendo que 146

variáveis foram necessárias chegar a este nível, conforme gráficos da Figura 21.

Figura 21: Avaliando importância das Variáveis - Rótulo *Ocorrência_ação_cível*



Fonte: Elaboração Própria

3.2.6 VALORES ÚNICOS

Este é um teste básico que visa identifica a existência de alguma variável que possua um valor único, pois este tipo de variável possui o comportamento semelhante ao de uma constante, tendo assim variação zero, não sendo relevante para um modelo.

Foram identificadas 2 variáveis com estas características (*VQtRestrtvPagaAti030DPI*, *VQtRestrEmprAti030DPI*) que foram removidas da base de dados.

3.3 SELEÇÃO DE VARIÁVEIS

Nesta etapa foi realizada a seleção das variáveis que possuem maior poder para explicar a variável resposta presente na base de dados, assim selecionadas para compor o modelo.

Portanto nesta etapa foi realizada uma avaliação que garanta que os modelos não incorram no erro de utilizar poucas variáveis, gerando assim um *underfitting*, não sendo um bom preditor, mas também não utilize variáveis em excesso causando *overfitting*, perdendo assim o poder de generalização.

Outro ponto importante é a eficiência computacional que garante que o recurso tecnológico seja utilizado de forma otimizada, obtendo um ganho de desempenho e de tempo, que é crucial na aprovação de uma operação nos dias atuais.

Para os modelos lineares utilizamos a técnica de *Stepwise*, para selecionar as variáveis para o modelo. O teste foi realizado de forma independente para cada uma das variáveis respostas, visando criar 3 modelos diferentes.

Para a variável resposta *Ocorrência_ação_cível* foram selecionadas 44 variáveis para fazerem parte do modelo final, já para a variável resposta *Fraude_na_concessão* por sua vez teve 40 variáveis selecionadas e por fim a variável resposta *Conta_nunca_paga* teve 32 variáveis selecionadas. O detalhe das variáveis selecionadas está disponível no Apêndice C.1.

De forma similar ao apresentado anteriormente para o procedimento Stepwise, como serão criados 3 modelos distintos para cada uma das variáveis resposta, foram selecionados 3 conjuntos de variáveis para cada uma das variáveis resposta, por meio do método de seleção de variável sequencial.

Este método selecionou uma quantidade inferior de variáveis para a variável resposta *Ocorrência_ação_cível* com 27 variáveis e para a variável resposta *Conta_nunca_paga* com 25 variáveis. Por outro lado a variável resposta *Fraude_na_concessão* obteve uma quantidade maior de variáveis selecionadas com um total de 44 variáveis. O detalhe das variáveis selecionadas está disponível no Apêndice C.2.

3.4 MODELAGEM

Os modelos utilizados neste trabalho são implementados por meio de "softwares livres" que não necessitam de licença comercial para serem utilizados. Assim foi utilizada a linguagem de programação *Python* com o projeto *scikit-learn*, desenvolvido por Pedregosa et al. (2011), que possui a implementação dos principais modelos que utilizam técnicas de aprendizado de máquina.

Adicionalmente também foram necessárias bibliotecas Python que possuem funcionalidades que facilitam o trabalho de implementação de modelos utilizando aprendizado de máquina, sendo eles *NumPy* [1], *SciPy* [2], *Matplotlib* [3] e *Pandas* [4].

Assim, todos os modelos foram executados 3 vezes de forma separada para cada uma das variáveis resposta disponíveis neste trabalho que são *Fraude_na_concessão* para o caso de fraude detectada na entrada da proposta, *Conta_nunca_paga* para cliente que nunca pagou uma parcela e *Ocorrência_ação_cível* quando existe a identificação de ação contrária. Neste primeiro momento foram desenvolvidos 7 modelos distintos, utilizando diferentes classificadores, para cada uma das variáveis resposta, buscando identificar o que possui melhor desempenho.

[1] *NumPy* é o pacote matemático para a linguagem *Python* que permite trabalhar com arranjos, vetores e matrizes de N dimensões.

[2] *SciPy* é uma biblioteca desenvolvida para trabalhar com *arrays NumPy*, e fornece muitas rotinas amigáveis e bem eficientes como rotinas para integração numérica e otimização.

[3] *Matplotlib* é uma biblioteca de visualização de dados para a linguagem *Python*

[4] *Pandas* é uma biblioteca para a linguagem *Python* para manipulação e análise de dados. Em particular, oferece estruturas de dados e operações para manipular tabelas numéricas e séries temporais.

O segundo passo do trabalho foi criar um modelo capaz de predizer todas as variáveis respostas ao mesmo tempo. De tal modo como técnica selecionada neste trabalho, foi utilizado o algoritmo *Randon Forest* que possui suporte para classificação *multi-output*, ou seja, com múltiplas saídas.

Para realizar um contraponto no método de *multi-output* nativo do classificador *Ensenble Randon Forest* foi utilizado o procedimento implantado pelo projeto *scikit-learn MultiOutputClassifier*, que fornece um meio de utilização de qualquer classificador para uma classificação de múltiplas variáveis respostas. A estratégia consiste em encaixar um classificador por alvo, permitindo assim várias classificações de variável de destino.

Assim, o procedimento *MultiOutputClassifier*, permite fazer com que qualquer estimador individual possa estimar uma série de funções alvo $(f_1, f_2, f_3, \ldots, f_n)$. Para tanto são treinadas em uma matriz única com preditor X todas as variáveis resposta y^*, possibilitando assim no final alcançar uma série de respostas $(y_1, y_2, y_3, \ldots, y_n)$.

Tendo em vista que os modelos utilizados são os implementados pelo projeto *scikit-learn* são apresentados no Apêndice B os parâmetros utilizados para cada um dos modelos, visando demonstrar as principais características que possibilitaram chegar aos mesmos resultados deste trabalho.

Adicionalmente como métrica de avaliação dos modelos, foi realizada a comparação de tempo de processamento, para o treinamento do modelos e para utilização do classificador como preditor. Para tanto o equipamento utilizado foi um Notebook com processador *Intel*® *Core*™ i7-75000U possuindo 16 Gigabytes de memória RAM.

4 RESULTADOS

Inicialmente foram desenvolvidos diferentes modelos para cada uma das variáveis respostas disponíveis na base de dados deste trabalho. Posteriormente foi proposto um modelo utilizando a metodologia de aprendizado de máquina que incorpore as três variáveis ao mesmo tempo, fazendo com que não seja necessária a criação de diversos modelos, e por fim utilizando todos os classificadores mencionados neste trabalho, foram criados modelos que misturam o resultado dos classificadores por meio do uso de uma matriz. Os resultados encontrados em cada uma destas etapas serão apresentados abaixo.

4.1 CLASSIFICADORES COM SAÍDA ÚNICA

Nesta etapa do trabalho foram desenvolvidos modelos separados para cara uma das variáveis respostas disponíveis para se identificar um evento de fraude. Para a variável resposta *Fraude_na_concessão* foram obtidos os resultados apresentados na Tabela 9 após a execução dos classificadores RF, NB, SVM, RL, BC, AC e GB.

Tabela 9: Resultados para variável *Fraude_na_concessão*.

	RF	**NB**	**SVM**	**RL**	**BC**	**AC**	**GB**
recall:	74,46%	68,31%	76,92%	76,92%	73,85%	76,00%	75,38%
precision:	6,48%	7,30%	6,74%	6,43%	6,79%	6,46%	6,40%
f-mensure:	11,92%	13,19%	12,40%	11,87%	12,43%	11,90%	11,80%
accuracy:	67,98%	73,85%	68,38%	66,76%	69,73%	67,27%	67,23%
ROC :	79,31%	75,66%	78,74%	78,70%	79,24%	79,64%	79,63%
KS:	30,53%	24,30%	30,27%	31,90%	28,74%	31,33%	31,34%
Tempo Treino:	0,264 s	0,002 s	10,740 s	0,197 s	0,862 s	0,470 s	0,130 s
Tempo Predição:	0,111 s	0,012 s	0,802 s	0,004 s	0,253 s	0,112 s	0,009 s

Fonte: Elaboração Própria

É possível verificar que o classificador de *Boosting* **AB** alcançou o melhor resultado para identificar os casos de fraude na concessão, tendo como parâmetro o indicador de Curva ROC. Entretanto quando comparado conjuntamente ao tempo de processamento o classificador **GB** apresenta-se como uma melhor escolha, tendo em vista possuir um ROC 0,01 pontos menor e um tempo de treinamento e predição consideravelmente menor.

Ao avaliar todos os resultados conjuntamente é possível verificar que todos os modelos possuem um resultado similar, com pequenas variações quando considerada

a curva ROC, sendo que o classificador que possui o resultado inferior, que descola do demais, é o classificador NB.

Entretanto o resultado do classificador NB é esperado, devido a sua simplicidade. Contudo esta simplicidade permite alcançar um resultado satisfatório, com baixo consumo computacional para execução, com apenas 0,002 segundos de processamento para o treinamento. Esta característica é um ganho de eficiência a ser considerado no momento de selecionar um classificador, quando o tempo de treinamento é um diferencial relevante.

Para verificar o poder de generalização do classificador observamos inicialmente a matriz de confusão para a base de treinamento, que está disponível na Tabela 10, com relação a matriz de confusão para a base de teste, disponível na Tabela 11. A grande diferença perceptível ao avaliar as duas matrizes é a questão do desbalanceamento, tendo em vista que foi balanceada apenas a base de treinamento. Com relação a generalização podemos verificar que o modelo possui uma acurácia de 76% na base de treinamento e de 67,23% na base de teste, o que representa um poder de generalização razoável com perda de aproximadamente 9 pontos percentuais.

Tabela 10: Matriz de confusão da variável *Fraude_na_concessão* para base de treinamento, classificador GB.

	Não é Fraude	**Fraude constatada**
Identificada como não Fraude	1393	381
Identificada como Fraude	459	1315

Fonte: Elaboração Própria

Tabela 11: Matriz de confusão da variável *Fraude_na_concessão* para base de teste, classificador GB.

	Não é Fraude	**Fraude constatada**
Identificada como não Fraude	7265	3581
Identificada como Fraude	80	245

Fonte: Elaboração Própria

Analisando os resultados das outras métricas, podemos identificar resultados importantes do modelo com o melhor desempenho. Por exemplo, o indicador de *recall* demonstra a precisão do modelo em identificar os casos de fraude de forma correta, ou seja, como demonstrado na matriz de confusão na Tabela 11 verificamos que 245 casos foram marcados como fraude de forma correta, enquanto 80 casos foram marcados como fraude de forma incorreta, o que representa uma taxa de acerto de 75%, como apresentado no indicador de *recall*.

Como efeito colateral este modelo apresenta uma perda de acurácia (*accuracy*), pois como podemos verificar na matriz de confusão, na Tabela 11, ele consegue

acertar 67,23% dos casos, ou seja, este classificador marca 3.578 operações como fraude de forma incorreta e 80 que não são fraude também de forma incorreta. Esta métrica não pode ser avaliada sozinha em casos de bases desbalanceadas, como a deste trabalho, pois ele acaba ficando cego com relação a classe minoritária.

Por outro lado, a métrica de Curva *ROC* consegue fornecer como informação, qual modelo possui um melhor desempenho geral, levando em consideração os acertos e erros tanto de falso positivo quanto de falso negativo.

O indicador de *precision* possui um valor baixo devido ao alto desbalanceamento da base de teste, sendo que neste trabalho foi realizado o balanceado, por meio do método de *Random Undersampling*, apenas da base de treinamento. Para o indicador de *precision* ser calculado foi considerado o valor de falsos positivos 3.578, que é consideravelmente maior frente a quantidade de fraudes identificadas que é 247. Como o calculo é a razão de fraudes identificadas pela soma dos falsos positivo com as fraudes identificadas, o resultado é de 6,46%.

De forma similar o indicador de *f-mensure* é prejudicado pelo desbalanceamento da base de teste, sendo que este indicador é composto pela combinação das métricas de *precision* e *recall*.

Por fim podemos verificar que o indicador de *KS* segue comportamento semelhante ao da curva *ROC*, para este caso, entretanto ao avaliar apenas este indicador o classificador de *benchmark* obteve resultado superior.

Como neste trabalho possuímos 3 variáveis respostas a serem avaliadas, foram executados todos os 7 classificadores citados anteriormente. Seguindo para a próxima variável resposta *Conta_nunca_paga*, identificamos os resultados presentes na Tabela 12 após a execução dos classificadores.

Tabela 12: Resultados para variável resposta *Conta_nunca_paga*.

	RF	**NB**	**SVM**	**RL**	**BC**	**AC**	**GB**
recall:	65,90%	55,76%	65,67%	63,82%	61,75%	63,82%	64,52%
precision:	6,22%	5,84%	6,27%	6,38%	5,95%	6,38%	6,07%
f-mensure:	11,37%	10,57%	11,45%	11,60%	10,86%	11,60%	11,09%
accuracy:	60,08%	63,36%	60,54%	62,21%	60,60%	62,21%	59,81%
ROC :	68,67%	64,67%	67,43%	67,49%	66,16%	67,35%	67,20%
KS:	37,28%	33,20%	36,79%	34,97%	36,42%	34,97%	37,44%
Tempo Treino:	0,377 s	0,009 s	14,390 s	0,378 s	0,669 s	0,386 s	0,114 s
Tempo Predição:	0,208 s	0,011 s	1,369 s	0,647 s	0,279 s	0,116 s	0,006 s

Fonte: Elaboração Própria

É possível verificar que a variável resposta que representa o cliente que nunca pagou uma parcela, foi a que encontramos maior dificuldade em identificar um classifi-

cador com bom poder preditivo. Desta forma, podemos observar este comportamento por meio dos resultados da curva ROC. Neste caso o classificador que obteve melhor resultado foi o **RF** com um resultado de ROC igual a 68,37%. O tempo de processamento para o classificador RF também foi satisfatório sendo de 0,377 segundos para treinamento e 0,208 segundos para predição. O classificador GB possui um tempo inferior de processamento mas como a diferença de ROC é superior a um ponto percentual, neste trabalho será considerado como melhor classificador RF.

Apesar do baixo resultado encontrado ao avaliar a curva ROC, podemos verificar que estes modelos obtiveram resultados superiores aos obtidos pelo modelo de *benchmark* (RL), entretanto a superioridade é marginal, demonstrando assim o poder preditivo desta técnica comumente utilizada.

Adicionalmente para a variável *Conta_nunca_paga* identificamos que o resultado de *KS*, para o classificador RF, foi superior que o encontrado pelo classificador de *benchmark*, demonstrando assim a superioridade dos modelos de aprendizado de máquina para este caso.

Por fim, para a variável resposta *Ocorrência_ação_cível* foram obtidos os resultados presentes na Tabela 13 após a execução dos classificadores.

Tabela 13: Resultados para variável resposta *Ocorrência_ação_cível*.

	RF	**NB**	**SVM**	**RL**	**BC**	**AC**	**GB**
recall:	77,74%	67,77%	72,76%	73,09%	76,08%	76,08%	75,75%
precision:	8,57%	7,91%	7,59%	7,54%	7,93%	8,70%	8,80%
f-mensure:	15,44%	14,17%	13,75%	13,67%	14,36%	15,61%	15,76%
accuracy:	77,06%	77,88%	75,41%	75,12%	75,54%	77,84%	78,18%
ROC:	**85,37%**	80,60%	82,07%	82,07%	85,17%	84,16%	84,26%
KS:	21,74%	20,38%	23,12%	23,43%	23,17%	20,88%	20,51%
Tempo Treino:	0,267 s	0,011 s	14,577 s	0,159 s	0,649 s	0,398 s	0,109 s
Tempo Predição:	0,109 s	0,018 s	0,950 s	0,004 s	0,152 s	0,096 s	0,007 s

Fonte: Elaboração Própria

A variável resposta que representa a existência de ação cível foi a que possibilitou identificar um classificador com maior poder preditivo, neste caso sendo o classificador **RF** o selecionado. É interessante verificar que o classificador que utiliza técnicas de aprendizado de máquina obteve um resultado superior ao encontrado ao utilizar a técnica *benchmark* (RL) com um valor de *ROC* de 85,37% contra 82,07%.

Outro ponto relevante ao analisar os resultados presentes na Tabela 13 foram os indicadores de *recall* e *accuracy*, demonstrando que o classificador de *Random Forest* consegue classificar corretamente o evento de fraude em 77% e o possui também a mesma probabilidade de classificar corretamente uma operação que não é fraude.

4.2 CLASSIFICADORES COM SAÍDAS MÚLTIPLAS

O passo seguinte do trabalho foi utilizar o classificador RF, entretanto utilizando múltiplas variáveis respostas, ou seja, neste caso não são gerados classificadores independentes para cada uma das variáveis respostas, mas sim um classificador capaz de considerar todas as variáveis conjuntamente. O resultado obtido com esta abordagem é apresentado na Tabela 14.

Tabela 14: Resultado classificador *Randon Forest* com múltiplas variáveis respostas.

	Concessão	**Never**	**Ação**
recall:	52,00%	14,10%	54,80%
precision:	10,40%	7,00%	15,20%
f-mensure:	17,30%	9,30%	23,80%
accuracy:	85,50%	89,40%	90,50%
ROC:	78,20%	63,10%	84,20%
Tempo Treino:	2,366 s		
Tempo Predição:	0,209 s		

Fonte: Elaboração Própria

Ao analisar os resultados deste classificador conjunto com relação aos resultados obtidos anteriormente, verificamos que os indicadores de *ROC* são inferiores aos obtidos individualmente, sendo que para a variável resposta *Fraude_na_concessão* a diferença é de 1,44% e para a variável *Ocorrência_ação_cível* de 1,77%, sendo assim diferenças marginais. A variável resposta em que a diferença foi grande foi apenas para *Conta_nunca_paga*, onde a diferença foi de 5,57%.

Para uma análise mais detalhada para a execução deste modelo geramos as matrizes de confusão, para a base de teste, de cada uma das variáveis resposta conforme apresentado nas Tabelas 15, 16 e 17.

Tabela 15: Matriz de confusão para a variável resposta *Fraude_na_concessão*.

	Não é Fraude	**Fraude constatada**
Identificada como não Fraude	9387	1459
Identificada como Fraude	156	169

Fonte: Elaboração Própria

Adicionalmente ao verificar o tempo de treinamento, para o classificador com múltiplas variáveis respostas, o tempo é superior ao encontrado de forma separada, sendo de 2,366 segundos. Entretanto o tempo de predição é baixo sendo de 0,209 segundos, que pode ser considerado atrativo, visto que se considerarmos os resultados dos modelos anteriores de forma conjunta, o tempo de predição seria de 0,326 segundos (considerando 0,109 com classificador RF para variável *Ocorrência_ação_cível*,

Tabela 16: Matriz de confusão para a variável resposta *Conta_nunca_paga*.

	Operação Normal	**Nunca paga**
Identificada como Normal	9922	815
Identificada como Nunca Paga	373	61

Fonte: Elaboração Própria

Tabela 17: Matriz de confusão para a variável resposta *Ocorrência_ação_cível*.

	Sem Ação	**Com Ação**
Identificada Sem Ação	9948	922
Identificada Ação	136	165

Fonte: Elaboração Própria

0,208 com classificador RF para variável *Conta_nunca_paga* e 0,009 para o classificador GB para variável *Fraude_na_concessão*).

4.3 MATRIZ DE CLASSIFICADORES

O passo final foi utilizar a junção dos classificadores em uma única matriz para executar o modelo para as 3 variáveis resposta ao mesmo tempo. Para facilitar a comparação entre os resultados serão apresentados os resultados por variável resposta, entretanto as 3 variáveis respostas foram executadas conjuntamente para cada um dos modelos apresentados.

Ao avaliar o resultado da utilização dos modelos de forma conjunta, apresentados nas Tabelas 18, 19 e 20, verificamos que os classificadores que são construídos como uma junção de outros classificadores foram os que obtiveram melhor resultado. com classificador AC para a variável *Fraude_na_concessão* tendo 80,9% de ROC e BC para a variável *Ocorrência_ação_cível* tendo 86% de ROC. Por outro lado para a variável *Conta_nunca_paga* o classificador que obteve melhor resultado foi o *benchmark* (RL).

Tabela 18: Resultados para variável resposta *Fraude_na_concessão* para os modelos *MultiOutputClassifier*.

	RL	**SVM**	**NB**	**BC**	**AC**	**GB**
recall:	21,2%	12,0%	86,2%	15,7%	17,2%	19,4%
precision:	24,8%	32,5%	3,9%	32,9%	26,7%	26,8%
f-mensure:	22,9%	17,5%	7,5%	21,3%	20,9%	22,5%
accuracy:	95,8%	96,7%	37,9%	96,6%	96,2%	96,1%
ROC:	79,6%	76,1%	75,3%	78,6%	**80,9%**	80,3%

Fonte: Elaboração Própria

Tabela 19: Resultados para variável resposta *Conta_nunca_paga* para os modelos *MultiOutputClassifier*.

	RL	**SVM**	**NB**	**BC**	**AC**	**GB**
recall:	0,7%	40,6%	40,8%	0,7%	0,2%	0,5%
precision:	13,0%	6,4%	6,3%	10,0%	5,3%	13,3%
f-mensure:	1,3%	11,0%	10,9%	1,3%	0,4%	0,9%
accuracy:	96,0%	74,5%	74,1%	95,9%	96,0%	96,0%
ROC:	**67,3%**	65,2%	65,2%	64,7%	66,8%	67,1%

Fonte: Elaboração Própria

Tabela 20: Resultados para variável resposta *Ocorrência_ação_cível* para os modelos *MultiOutputClassifier*.

	RL	**SVM**	**NB**	**BC**	**AC**	**GB**
recall:	17,3%	64,5%	64,5%	25,6%	24,6%	26,6%
precision:	29,1%	7,2%	7,2%	39,3%	32,7%	35,1%
f-mensure:	21,7%	13,0%	12,9%	31,0%	28,1%	30,2%
accuracy:	96,6%	76,7%	76,6%	96,9%	96,6%	96,7%
ROC:	82,3%	77,8%	77,8%	**86,0%**	85,6%	85,6%

Fonte: Elaboração Própria

Adicionalmente para uma melhor avaliação dos classificadores processados de forma conjunta para todas as variáveis respostas com uso de matriz, realizamos a avaliação de tempo de processamento para o treinamento e predição, conforme apresentado na Tabela 21.

Tabela 21: Tempo de processamento dos modelos *MultiOutputClassifier*.

	RL	**SVM**	**NB**	**BC**	**AC**	**GB**
Tempo Treino:	4,301 s	1000,622 s	0,104 s	15,894 s	6,789 s	2,547 s
Tempo Predição:	0,009 s	51,111 s	0,113 s	1,976 s	0,587 s	0,070 s

Fonte: Elaboração Própria

Ao realizar a análise dos resultados, o classificador que obtém os melhores ROCs é o classificador de AC. Entretanto este resultado possui valor muito próximo ao encontrado no classificador GB, mas com um tempo de processamento significativamente maior. Assim foi selecionado como melhor classificador para o método com o uso de matrizes o classificador **GB**, sendo que ele utiliza 2,547 segundos para treinamento e apenas 0,07 segundos para predição.

Outro ponto muito importante identificado nos modelos desenvolvidos, por classificadores agrupados em uma matriz, para todas as variáveis respostas, é o ganho de acurácia, sendo que nestes modelos as taxas são superiores a 96%, o que garante modelos com poucos falsos positivos em seus resultados.

4.4 COMPARANDO OS RESULTADOS

Neste passo serão analisadas todas as técnicas utilizadas neste trabalho de forma conjunta, visando identificar a que possui resultado superior.

Ao avaliar todos os resultados a resposta não pode ser direta pois existe um "perde-e-ganha" quando avaliamos todos os resultados conjuntamente, sendo que cada uma das metodologias possui as suas vantagens e desvantagens.

Os modelos executados de forma individual possuem forte capacidade de identificar os eventos de indícios de fraudes, comportamento que podemos verificar por meio da métrica de *recall* nas Tabelas 22, 23 e 24. Por outro lado os modelos executados para múltiplas variáveis respostas possuem a maior capacidade de não marcar falsos positivos, ou seja, classificar um cliente que não é fraudador como sendo um cliente fraudador, como podemos observar por meio da métrica de *accuracy*. O método com múltiplas saídas é o que fica em um meio termo, sendo assim uma interessante escolha para aplicações que visam detectar fraudes.

Para a variável resposta *Fraude_na_concessão* é nítido o comportamento destacado anteriormente, sendo que o método RF com múltiplas saídas possuí um acerto de 52% das fraudes com uma acurácia de 85,5%. Os outros dois métodos ficam em extremos opostos sendo que o de saída única possui um bom *recall* enquanto a matriz de classificadores AC possuí uma ótima acurácia 80,3%, como podemos verificar na Tabela 22.

Tabela 22: Resultados consolidados para variável *Fraude_na_concessão*.

	Saída Única GB	**Múltiplas Saídas RF**	**Matriz de classificadores AC**
recall:	75,38%	52,00%	19,4%
precision:	6,40%	10,40%	26,8%
f-mensure:	11,80%	17,30%	22,5%
accuracy:	67,23%	85,50%	96,1%
ROC:	79,63%	78,20%	80,3%
Tempo Treino:	0,130 s	2,366 s	2,547 s
Tempo Predição:	0,009 s	0,209 s	0,070 s

Fonte: Elaboração Própria

Por sua vez a variável *Conta_nunca_paga* é a que gera maior dificuldade de predição. Ao executar o modelo de forma simples, ou seja, com saída única RF, encontramos um bom resultado de *recall* (65,90%) mas que para tanto sacrifica e muito a acurácia (60,80%), marcando muitos casos como falsos positivos, como podemos observar na Tabela 23. Os modelos com múltiplas variáveis respostas para este caso possuem uma péssima capacidade de identificar os casos em que nenhuma parcela

foi paga, entretanto mantendo uma ótima acurácia.

Tabela 23: Resultados consolidados para variável *Conta_nunca_paga*.

	Saída Única RF	**Múltiplas Saídas RF**	**Matriz de classificadores AC**
recall:	65,90%	14,10%	0,5%
precision:	6,22%	7,00%	13,3%
f-mensure:	11,37%	9,30%	0,9%
accuracy:	60,08%	89,40%	96,0%
ROC:	68,67%	63,10%	67,1%
Tempo Treino:	0,377 s	2,366 s	2,547 s
Tempo Predição:	0,208 s	0,209 s	0,070 s

Fonte: Elaboração Própria

Finalizando, ao avaliar a variável *Ocorrência_ação_cível*, o comportamento é muito semelhante ao encontrado na variável *Fraude_na_concessão*, com o modelo com múltiplas saídas RF sendo um bom meio termo por conseguir identificar 54,8% dos casos em que ocorrem ação cível, mantendo uma ótima acurácia de 90,5%, como pode ser observado na Tabela 24.

Tabela 24: Resultados consolidados para variável *Ocorrência_ação_cível*.

	Saída Única RF	**Múltiplas Saídas RF**	**Matriz de classificadores AC**
recall:	77,74%	54,80%	26,6%
precision:	8,57%	15,20%	35,1%
f-mensure:	15,44%	23,80%	30,2%
accuracy:	77,06%	90,50%	96,7%
ROC:	85,37%	84,20%	85,6%
Tempo Treino:	0,267 s	2,366 s	2,547 s
Tempo Predição:	0,109 s	0,209 s	0,070 s

Fonte: Elaboração Própria

Assim, após a execução de todos os modelos conseguimos verificar que a utilização de modelos com múltiplas variáveis respostas RF é completamente plausível, sendo assim um classificador com boa capacidade de identificar os casos de fraudes, sem prejudicar em excesso a acurácia do modelo. A exceção ficou por conta da variável *Conta_nunca_paga*, onde apenas o classificador utilizado de forma única possui resultado razoável, mas sacrificando a acurácia, enquanto que os classificadores com múltiplas saídas conseguirem manter altos índices de acurácia.

4.5 AVALIANDO AS VARIÁVEIS SELECIONADAS PARA OS MODELOS

Nesta etapa do trabalho, serão avaliadas as variáveis selecionadas pelo método de seleção sequencial, com uso do classificador RF, para os modelos que utilizam técnicas de aprendizado de máquina, sendo que o detalhe das variáveis está disponível no Apêndice C.2.

A única variável que esteve presente no treinamento de todos os modelos, independente da variável resposta foi a que identifica a quantidade de responsáveis pelos domicílio particular. Entretanto, devido às características da base de dados disponibilizada, todos os modelos possuem relação direta com variáveis relacionadas a consulta e inclusão de restritivos de crédito. Foram selecionadas variáveis que identificam os tipos de instituições que realizaram a consulta (ex: financeira, seguradora) bem como o tipo de produto (ex: cartão, empréstimo pessoal).

Para a variável resposta *Fraude_na_concessão*, os modelos adicionalmente consideraram o escore de crédito do cliente e duas variáveis relacionadas a fraudes criadas pela empresa que disponibilizou a base de dados (Serasa Experian). Foram selecionadas também variáveis relacionadas a cidade em que reside o cliente como CEP, existência de iluminação pública, energia elétrica nos domicílios, estabelecimentos de saúde, expectativa de vida, domicílios com rendimentos de até 2 salários mínimos e índice de desenvolvimento humano. Adicionalmente foram consideradas variáveis relacionadas ao tempo do endereço do cliente e indicativo da residencia do cliente, se está em uma capital e se faz também faz parte de um edifício.

Para a variável relacionada a fraude na concessão também foi utilizado dados de dividas vencidas que ficaram negativadas ao mesmo tempo, além de verificar se o cliente possui participação em alguma empresa de pequeno porte.

Para a variável resposta *Conta_nunca_paga* foi utilizado o escore de crédito e o indicador de bolsa família, além de informações do local onde o cliente reside como classe social, quantidade de estabelecimentos de saúde e índice de desenvolvimento humano. Adicionalmente foi verificada a participação em microempresas e a idade em dias da principal empresa em quem o cliente participa.

E por fim para a variável resposta *Ocorrência_ação_cível* também foram verificadas informações de onde o cliente reside, como a existência de iluminação pública e proporção de vulneráveis a pobreza. Além disso informações relacionadas a CEP e o tempo de atualização do endereço e informação se o cliente reside em um edifício em uma capital. Como informações adicionais foi avaliado se o cliente participa de uma empresa, e em caso positivo a quanto tempo, além do salário médio mensal.

Assim é possível verificar que para as variáveis *Fraude_na_concessão* e *Conta_nunca_paga* foram selecionadas variáveis que analisam de forma mais consistente

a capacidade creditícia do cliente, enquanto que para a variável *Ocorrência_ação_cível* são utilizadas informações mais gerais do cliente.

Outro ponto relevante ao avaliar as bases de dados, foi a percepção de que devido a ausência de informações especificadas do produto e da operação que o cliente está realizando, a avaliação da variável *Conta_nunca_paga* ficou comprometida, sendo assim, caso estas informações estivessem disponíveis, provavelmente o poder de predição do modelo para esta variável resposta seria melhor.

5 CONCLUSÃO

Como proposto, o objetivo deste trabalho era verificar a viabilidade de utilizar técnicas de aprendizado de máquina na construção de um modelo capaz de predizer com eficiência e bom desempenho 3 variáveis resposta ao mesmo tempo.

Inicialmente foi realizado todo um trabalho na base de dados, realizando um pré-processamento que visava garantir que a base disponibilizada estivesse adequada para a construção dos modelos. Adicionalmente a este processo ocorreu a parte mais importante do trabalho que foi a seleção das variáveis que iriam compor os modelos.

Para identificar o modelo que melhor iria classificar os dados foi realizada a construção de modelos distintos para verificar o melhor resultado que poderia ser obtido, utilizando técnicas diferentes para que fosse possível avaliar a que melhor se adaptava aos dados disponibilizados.

Neste passo chegamos em um ponto interessante pois verificamos que os classificadores possuíam resultados muito semelhantes entre si. Para que fosse possível identificar o melhor utilizamos o indicador de curva ROC que demonstrou que olhando os modelos de forma individual o modelo de Randon Forest possuí o desempenho superior.

Na construção dos modelos de forma conjunta para verificar qual iria obter melhor desempenho foi utilizado o classificador RF puro que já possui suporte para este tipo de modelagem utilizando diversas variáveis resposta em um mesmo modelo.

Adicionalmente foram criados modelos que utilizam classificadores de forma conjunta, por meio da utilização de matrizes, o que possibilitou a avaliação de um método adicional para identificação de fraudes.

Com a execução de todos os modelos anteriormente citados, foi possível verificar que existe uma diferença significativa entre os resultados obtidos por cada um deles, sendo que os classificadores utilizados de forma únicas possuem bom poder de identificar o evento que pode ser uma fraude, mas tendo como efeito colateral uma alta marcação de falsos positivos. Por outro lados os modelos que executam classificadores conjuntos possuem uma boa acurácia mais com dificuldade em identificar o evento que pode representar uma fraude.

Estes resultados demonstram que os algoritmos de aprendizado de máquina representam atualmente recursos importantes para a modelagem estatística na detecção de fraudes, pois o modelo RF que possui em sua implementação o recurso de se utilizar múltiplas variáveis respostas, obteve o melhor resultado comparativo, de-

monstrando assim o seu poder em construir modelos com previsões robustas mesmo em situações onde necessita-se trabalhar com múltiplas variáveis respostas.

Assim, este trabalho proporcionou uma demonstração com dados reais da criação de modelos que podem prever mais de uma variável resposta ao mesmo tempo, com resultados satisfatórios, proporcionando uma alternativa ao uso de modelos individuais que carecem de combinação de matrizes, regras, políticas ou o desenvolvimento de mistura de modelos.

Em trabalhos futuros sugerimos que seja expendido maior esforço nas etapas de pré-processamento, especificamente na transformação das variáveis contínuas em discretas e principalmente na factibilidade de criar novas variáveis a partir da combinação das existentes. Apesar destas etapas não terem sido exploradas exaustivamente neste trabalho, a literatura disponível demonstra que o aprimoramento destas etapas podem acrescentar poder preditivo aos classificadores utilizados.

REFERÊNCIA

BARBER, D. *Bayesian reasoning and machine learning*. [S.l.]: Cambridge University Press, 2012. Citado 5 vezes nas páginas 22, 23, 24, 30 e 31.

BISHOP, B. Cm: Pattern recognition and machine learning. *Journal of Electronic Imaging*, Springer, v. 16, n. 4, p. 140–155, 2006. Citado 2 vezes nas páginas 22 e 39.

BOBBIO, A. et al. Improving the analysis of dependable systems by mapping fault trees into bayesian networks. *Reliability Engineering & System Safety*, Elsevier, v. 71, n. 3, p. 249–260, 2001. Citado na página 17.

BOLTON, R. J.; HAND, D. J. Statistical fraud detection: A review. *Statistical science*, JSTOR, p. 235–249, 2002. Citado na página 16.

BREIMAN, L. Bagging predictors. *Machine learning*, Springer, v. 24, n. 2, p. 123–140, 1996. Citado 2 vezes nas páginas 36 e 37.

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

BURGES, C. J. A tutorial on support vector machines for pattern recognition. *Data mining & knowledge discovery*, Springer, v. 2, n. 2, p. 121–167, 1998. Citado na página 25.

CASELLA, G.; BERGER, R. L. Inferência estatística-tradução da 2ª edição norte americana. *Centage Learning*, 2011. Citado 2 vezes nas páginas 19 e 22.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, n. 3, p. 273–297, 1995. Citado na página 25.

DIETTERICH, T. G. Ensemble methods in machine learning. In: SPRINGER. *International workshop on multiple classifier systems*. [S.l.], 2000. p. 1–15. Citado na página 35.

EXPERIAN, S. *Relatório Global de Fraude e Identificação*. [S.l.], 2018. Citado na página 18.

FAWCETT, T. Roc graphs: Notes and practical considerations for researchers. *Machine learning*, v. 31, n. 1, p. 1–38, 2004. Citado na página 47.

FERRI, F. et al. Comparative study of techniques for large-scale feature selection. In: *Machine Intelligence and Pattern Recognition*. [S.l.]: Elsevier, 1994. v. 16, p. 403–413. Citado na página 45.

FIROZABADI, B. S.; TAN, Y.-H.; LEE, R. M. Formal definitions of fraud. *Norms, logics and information systems-new studies in Deontic logic and computer science*, p. 275–288, 1998. Citado na página 16.

FLETCHER, R. Practical methods of optimization john wiley & sons. *New York*, v. 80, 1987. Citado na página 27.

FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, Elsevier, v. 55, n. 1, p. 119–139, 1997. Citado na página 39.

FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, JSTOR, p. 1189–1232, 2001. Citado na página 40.

FRIEDMAN, N.; GEIGER, D.; GOLDSZMIDT, M. Bayesian network classifiers. *Machine learning*, Springer, v. 29, n. 2-3, p. 131–163, 1997. Citado 2 vezes nas páginas 17 e 29.

HE, H.; GARCIA, E. A. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, n. 9, p. 1263–1284, 2008. Citado 2 vezes nas páginas 42 e 45.

Hosmer Jr, D. W.; LEMESHOW, S.; STURDIVANT, R. X. *Applied logistic regression*. [S.l.]: John Wiley & Sons, 2013. Citado na página 20.

KARCHER, C. *Redes Bayesianas aplicadas à análise do risco de crédito.* Dissertação (Mestrado) — Universidade de São Paulo, 2009. Citado na página 17.

LAVION, D. *Pulling fraud out of the shadows - Global Economic Crime and Fraud Survey*. [S.l.], 2018. Citado na página 19.

LEE, K. I.; KOVAL, J. J. Determination of the best significance level in forward stepwise logistic regression. *Communications in Statistics-Simulation and Computation*, Taylor and Francis, v. 26, n. 2, p. 559–575, 1997. Citado na página 44.

MITCHELL, T. M. *The discipline of machine learning*. [S.l.]: Carnegie Mellon University, School of Computer Science, Machine Learning Department Pittsburgh, PA, 2006. Citado na página 22.

MORAES, D. d. Modelagem de fraude em cartão de crédito. Universidade Federal de São Carlos, 2008. Citado na página 17.

MURPHY, K. P. *Machine learning: a probabilistic perspective*. [S.l.]: MIT press, 2012. Citado na página 22.

NOBLE, W. S. What is a support vector machine? *Nature biotechnology*, Nature Publishing Group, v. 24, n. 12, p. 1565, 2006. Citado na página 25.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, v. 12, n. Oct, p. 2825–2830, 2011. Citado na página 62.

PHUA, C. et al. A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*, 2010. Citado na página 17.

QUEIROZ, C. D. N. *Redes Bayesianas no gerenciamento e mensuração de riscos operacionais.* Dissertação (Mestrado) — Universidade de São Paulo, 2008. Citado na página 17.

RAMOS, J. E. *Redes bayesianas aplicadas a- modelagem de fraudes em carta-o de cre-dito*. Dissertação (Mestrado) — Fundação Getulio Vargas, 2015. Citado na página 17.

ROSSET, S. et al. Discovery of fraud rules for telecommunications—challenges and solutions. In: ACM. *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.], 1999. p. 409–413. Citado na página 17.

SCHAPIRE, R. E. The strength of weak learnability. *Machine learning*, Springer, v. 5, n. 2, p. 197–227, 1990. Citado na página 38.

SHARMA, A.; PANIGRAHI, P. K. A review of financial accounting fraud detection based on data mining techniques. *arXiv preprint arXiv:1309.3944*, 2013. Citado na página 18.

SIDDIQI, N. *Credit risk scorecards: developing and implementing intelligent credit scoring*. [S.l.]: John Wiley and Sons, 2012. Citado na página 49.

TREVOR, H.; ROBERT, T.; JH, F. *The elements of statistical learning: data mining, inference, and prediction*. [S.l.]: New York, NY: Springer, 2009. Citado 2 vezes nas páginas 22 e 41.

VIAENE, S. et al. A comparison of state-of-the-art classification techniques for expert automobile insurance claim fraud detection. *Journal of Risk and Insurance*, Wiley Online Library, v. 69, n. 3, p. 373–421, 2002. Citado 2 vezes nas páginas 16 e 17.

APÊNDICES

APÊNDICE A – VARIÁVEIS EXCLUÍDAS POR MULTICOLINEARIDADE

Na tabela a seguir são apresentadas as variáveis excluídas devido a Multicolinearidade:

EQtConsSF_disc	EQtConsCheq30d_disc
EQtPgtoFaixa1Ult_12m_disc	EVQtAcoes_disc
EPctMaxPartPF_disc	EQtEmpConsCheq30d_disc
EQtConsPdRec15U90d_disc	EVQtAcoesAti_disc
EVQtMaxRestrAti_disc	EQTBANCOS_disc
EQtCons180d_disc	ECoMaxInadimpUlt_00m_disc
EVQtRestrAti_disc	EVQtCCFsAti030DPI_disc
EQtPgtoVistaUlt_12m_disc	EVFlRestrAti1A_disc
EQtConsCred360d_disc	VTpPri_ConsTOD_5a_disc
EVQtCCFs_disc	VQt_ConsALI_360d_disc
EQtConsCheq360d_disc	VQt_ConsCCP_360d_disc
EQtConsOrigSER180d_disc	VQt_ConsCCP_180d_disc
EQtConsSFOrigTEL_disc	VNuEmp_ConsSEG_360d_disc
EQtFornec_00m_06m_disc	VNuEmp_ConsSEG_180d_disc
EQtRecheqOcor12M_disc	VQt_ConsSEG_10d_disc
EQtFornecDifUlt_02m_disc	VQt_ConsCCP_30d_disc
EVQtCCFsAti_disc	VQt_ConsALI_30d_disc
EQtRecheqSusS12M_disc	VQt_ConsASS_10d_disc
EQtRecheqOcor06M_disc	VQt_ConsCCP_10d_disc
EVQtRefinsResU12m_disc	VQt_ConsALI_10d_disc
EQtEmpConsCheq360d_disc	PEA_disc
EQtConsOrigSFN180d_disc	VFlPriRestrAti_disc
EMaxQtConsEmp60d_disc	VIndTodasRestrCoPart_disc
EQtfamilia_disc	VQtRestrAti_disc
EVQtCredoresAti_disc	

APÊNDICE B – PARÂMETROS DOS MODELOS

Como os modelos utilizados são os implementados pelo projeto *scikit-learn* seguem os parâmetros utilizados para cada um dos modelos, visando demonstrar as principais características que possibilitaram chegar aos resultados deste trabalho.

B.1 REGRESSÃO LOGÍSTICA

Para o modelo de Regressão Logística *(LogisticRegression)* foram utilizados os seguintes parâmetros:

C=1.0, representa o parâmetro de penalidade *C* para o termo de erro.

fit_intercept=*True*, especifica que uma constante deve ser adicionada à função de decisão.

intercept_scaling= 1, representa que *x* se torna um recurso sintético.

multi_class= *ovr*, nesta opção um problema binário é adequado para cada rótulo.

penalty= *l1*, representa a norma de penalização utilizada.

solver = *liblinear*, representa o algoritimo utilizado na otimização do problema, neste trabalho foi selecionado *liblinear*.

tol = 0.0001, representa a tolerância ao critério de parada.

B.2 SVM

Para o modelo SVM *(svm.SVC)* foram utilizados os seguintes parâmetros:

C=1.0, representa o parâmetro de penalidade *C* para o termo de erro.

decision_function_shape = *ovr*, representa a função de decisão utilizada, no trabalho em questão será “um-contra-restante”.

kernel = *linear*, especifica o tipo de kernel a ser utilizado no algoritmo, no caso deste trabalho o selecionado foi o linear.

max_iter = -1, representa o limite máximo de iterações no solver, neste caso como foi inserido -1 não há limite.

shrinking = *True*, representa a utilização da heurística de encolhimento.

tol = 0.0001, representa a tolerância ao critério de parada.

B.3 RANDOM FOREST

Para o modelo *Random Forest* foram utilizados os seguintes parâmetros:

bootstrap = *True*, significa que as amostras de *bootstrap* foram usadas ao construir as árvores.

class_weight = *None*, representa os pesos associados a classes, como está marcado como nenhum, o peso será igual a um.

criterion = *gini*, representa a função para medir a qualidade de uma divisão. Os critérios possíveis neste parâmetro é “gini” para impureza e “entropy” para ganho de informação.

max_depth = *None*, representa a máxima profundidade da árvore. Como está marcado como nenhum, os nós se expandem até que todas as folhas fiquem puras ou até que todas as folhas contenham o mínimo de casos.

max_features = *auto*, representa o número de recursos a serem considerados ao procurar a melhor divisão, como está marcado como auto o máximo é igual a raiz quadrada da quantidade de variáveis.

max_leaf_nodes = *None*, significa que o número de folhas por nó é ilimitado.

min_impurity_decrease = 0.0, um nó será dividido caso a divisão induzir a uma diminuição da impureza maior ou igual a 0. A diminuição da impureza ponderada é calculada por meio da equação:

$$\frac{N\_t}{N} * (impurity - \frac{N\_t\_R}{N\_t} * right\_impurity - \frac{N\_t\_L}{N\_t} * left\_impurity)$$

Onde N é o número total de amostras, N_t é o número total de amostras no nó atual, N_t_L é o número de amostras no filho da esquerda, N_t_R é o número de amostras no filho da direita. Assim N, N_t, N_t_L e N_t_R referem-se à soma ponderada, quando o peso das amostras for utilizado.

min_samples_leaf = 10, representa o número mínimo de amostras necessárias em um nó de folha. Um ponto de divisão em qualquer profundidade só será considerado se deixar ao menos 10 amostra de treinamento em cada uma das ramificações da esquerda e da direita. Em casos de regressão este parâmetro pode ter o poder de suavizar o resultado.

min_samples_split = 2, representa o mínimo de amostras necessárias para dividir um nó interno.

n_estimators = 100, representa o número de árvores que serão utilizadas na floresta.

B.4 NAIVE BAYES

Para o modelo *Naive Bayes* o método utilizado é o *Gaussiano* e neste trabalho não foram atualizados os parâmetros como por exemplo o *prior*.

B.5 BAGGING

Para o modelo de *Bagging* foram utilizados os seguintes parâmetros:

base_estimator = *None*, representa o estimador base para ser utilizado em subconjuntos aleatórios do conjunto de dados, neste caso como não há especificação de um o utilizado é uma árvore de decisão.

bootstrap = *True*, significa que as amostras são desenhadas sem repetição.

max_features = 0.5, representa o percentual de variáveis utilizadas a partir da base de treinamento para treinar cada estimador de base.

max_samples = 0.5, representa o percentual de amostras utilizadas a partir da base de treinamento para treinar cada estimador de base.

n_estimators = 100, representa o número de estimadores base utilizados no método Ensemble.

B.6 ADA BOOST

Para o modelo de *Ada Boost* foram utilizados os seguintes parâmetros:

base_estimator = *None*, representa o estimador de base a partir do qual o conjunto Ensemble é construído, neste caso é utilizada uma árvore de decisão.

learning_rate = 0.4, representa a taxa de aprendizado, que reduz a contribuição de cada classificador pela taxa de aprendizado. Existe um *trade off* entre este parâmetro e o número de estimadores.

n_estimators = 100, representa o número máximo de estimadores em que o boosting é finalizado. No caso em que o ajuste perfeito é atingido precocemente o aprendizado é interrompido.

B.7 GRADIENT BOSSTING

Para o modelo de *Gradient Boosting* foram utilizados os seguintes parâmetros:

criterion = *friedman_mse*, representa o método utilizado para avaliar a qualidade de uma divisão, neste trabalho é utilizado o erro quadrático médio com pontuação de melhoria de Friedman, que geralmente fornece uma melhor aproximação.

learning_rate = 0.4, representa a taxa de aprendizado, que reduz a contribuição de cada classificador pela taxa de aprendizado. Existe um *trade off* entre este parâmetro e o número de estimadores.

loss = *deviance*, representa a função de perda a ser otimizada, neste caso estamos utilizando uma referência ao desvio, que se assemelha a regressão logística.

max_depth = 1, representa a profundidade máxima dos estimadores de regressão individuais, assim a profundidade limita o número de nós da árvore.

max_features = *None*, representa o número de variáveis a serem consideradas ao procurar uma melhor divisão, neste caso como está marcando *None* significa que o máximo de variáveis a serem consideradas e a quantidade de variáveis disponíveis na base de treinamento.

max_leaf_nodes = *None*, significa que o número de folhas por nó é ilimitado.

min_impurity_decrease = 0.0, um nó será dividido caso a divisão induzir a uma diminuição da impureza maior ou igual a 0. A diminuição da impureza ponderada é calculada por meio da equação:

$$\frac{N\_t}{N} * (impurity - \frac{N\_t\_R}{N\_t} * right\_impurity - \frac{N\_t\_L}{N\_t} * left\_impurity)$$

Onde N é o número total de amostras, N_t é o número total de amostras no nó atual, N_t_L é o número de amostras no filho da esquerda, N_t_R é o número de amostras no filho da direita. Assim N, N_t, N_t_L e N_t_R referem-se à soma ponderada, quando o peso das amostras for utilizado.

min_samples_leaf = 10, representa o número mínimo de amostras necessárias em um nó de folha. Um ponto de divisão em qualquer profundidade só será considerado se deixar ao menos 10 amostras de treinamento em cada uma das ramificações da esquerda e da direita. Em casos de regressão este parâmetro pode ter o poder de suavizar o resultado.

min_samples_split = 2, representa o mínimo de amostras necessárias para dividir um nó interno.

n_estimators = 100, representa o número de estágios de esforço a serem executados. O aumento de gradiente é bastante robusto para o ajuste excessivo, assim o ideal é que este número seja grande.

presort = *auto*, representa o processo de pré-ordenar os dados, visando acelerar a descoberta das melhores divisões no ajuste. Assim ao inserir o método automático é realizada uma pré-classificação dos dados densos em uma classificação normal de dados esparsos.

APÊNDICE C – VARIÁVEIS SELECIONADAS

C.1 MÉTODO STEPWISE

Para a variável resposta *Fraude_na_concessão* foram selecionadas por meio do método Stepwise as seguintes variáveis:

model	Beta	Std. Error	Std. Beta	t	Sig	lower	upper
(Intercept)	0,156	0,059		2.643	0,008	0,04	0,271
IndFrdTelU1A_disc	0,098	0,013	0,181	7.728	0	0,073	0,123
QtMaxInc360dAposCons_disc	-0,019	0,005	-0,101	-3.592	0	-0,029	-0,008
VQt_ConsTOD_10d_disc	0,135	0,029	0,206	4.575	0	0,077	0,192
PercDomicilios2SM_disc	-0,021	0,008	-0,073	-2.660	0,008	-0,036	-0,005
escore_CREDITO_disc	0,019	0,004	0,106	4.330	0	0,01	0,027
VQt_ConsTCO_180d_disc	0,021	0,009	0,047	2.386	0,017	0,004	0,039
VQt_ConsBCO_10d_disc	-0,032	0,03	-0,036	-1.099	0,272	-0,09	0,025
VTpPri_ConsFIN_5a_disc	-0,011	0,003	-0,063	-3.940	0	-0,017	-0,006
QtDiasTel_disc	0,015	0,004	0,088	3.430	0,001	0,006	0,023
TpUltTEL_disc	-0,015	0,004	-0,079	-3.516	0	-0,023	-0,007
QtEnd_disc	-0,027	0,005	-0,128	-5.733	0	-0,037	-0,018
QtDiasEnd_disc	0,007	0,004	0,044	1.950	0,051	0	0,015
VTpUlt_ConsCRDTOD_5a_disc	-0,006	0,003	-0,033	-1.750	0,08	-0,012	0,001
E_ANOSESTUDO_disc	-0,012	0,003	-0,071	-3.787	0	-0,019	-0,006
IndFrdEndU1A_disc	0,053	0,013	0,069	3.989	0	0,027	0,079
IndPartPP	0,152	0,057	0,039	2.690	0,007	0,041	0,263
VTdRel_ConsSTS_P02dU30d_disc	0,068	0,019	0,091	3.532	0	0,03	0,105
VNuEmp_ConsCHQSTS_360d_disc	-0,036	0,01	-0,057	-3.576	0	-0,055	-0,016
VNuAnos_ConsSEG_5a_disc	0,043	0,012	0,134	3.536	0	0,019	0,067
VNuEmp_ConsSEG_10d_disc	-0,176	0,032	-0,193	-5.485	0	-0,239	-0,113
VQt_ConsTCO_10d_disc	0,062	0,022	0,06	2.866	0,004	0,02	0,105
VQt_ConsCRDSTS_180d_disc	0,026	0,007	0,09	3.659	0	0,012	0,04
ESPVIDA_disc	-0,017	0,004	-0,099	-4.069	0	-0,025	-0,009
VQtRestrCTRes_disc	-0,021	0,011	-0,034	-1.956	0,051	-0,042	0
QtInclEndU1A_disc	-0,036	0,015	-0,042	-2.415	0,016	-0,065	-0,007
VQt_ConsSFN_10d_disc	-0,047	0,022	-0,068	-2.108	0,035	-0,09	-0,003
VQt_ConsVES_360d_disc	-0,026	0,011	-0,035	-2.329	0,02	-0,048	-0,004
TmpMedBaixConsCredor_disc	0,008	0,003	0,036	2.474	0,013	0,002	0,014
QtConsLNO_disc	0,009	0,004	0,05	2.485	0,013	0,002	0,016
IDADE_disc	0,007	0,003	0,037	2.121	0,034	0,001	0,013
QtNuTel_disc	0,015	0,007	0,055	2.197	0,028	0,002	0,029
IdadeEndA_disc	0,007	0,003	0,035	2.248	0,025	0,001	0,013
FlAptoInt	0,085	0,042	0,03	2.055	0,04	0,004	0,167
VQtRestrLDep_disc	-0,038	0,019	-0,031	-2.004	0,045	-0,076	-0,001
AREAMUN_disc	0,007	0,003	0,04	2.632	0,009	0,002	0,012
IDHM2010_disc	0,018	0,005	0,108	3.991	0	0,009	0,027
VNuMaxEmp_ConsSTS_360d_disc	-0,013	0,007	-0,037	-1.857	0,063	-0,027	0,001
VQt_ConsCRDTOD_10d_disc	-0,044	0,026	-0,064	-1.692	0,091	-0,095	0,007
VQt_ConsBCO_30d_disc	-0,039	0,022	-0,056	-1.740	0,082	-0,083	0,005
VTpPri_ConsSEG_5a_disc	0,025	0,014	0,079	1.720	0,086	-0,003	0,053
D_ComEnergia_disc	-0,018	0,01	-0,047	-1.714	0,087	-0,038	0,003

Para a variável resposta *Conta_nunca_paga* foram selecionadas por meio do

método *Stepwise* as seguintes variáveis:

model	Beta	Std. Error	Std. Beta	t	Sig	lower	upper
(Intercept)	0,642	0,049		13.042	0	0,546	0,739
escore_CREDITO_disc	-0,026	0,005	-0,145	-5.614	0	-0,035	-0,017
ClasseSocial_disc	-0,03	0,009	-0,071	-3.295	0,001	-0,047	-0,012
QtMaxInc360dAposCons_disc	-0,038	0,011	-0,218	-3.342	0,001	-0,061	-0,016
VTdRel_ConsSTS_P02dU30d_disc	0,048	0,015	0,064	3.099	0,002	0,018	0,078
VQt_ConsBCO_10d_disc	-0,091	0,019	-0,106	-4.706	0	-0,129	-0,053
BOLSA_FAMILIA	0,028	0,007	0,058	3.779	0	0,014	0,043
VNuEmp_ConsSTS_180d_disc	0,028	0,007	0,095	3.902	0	0,014	0,042
VTpPri_ConsCHQSTS_5a_disc	-0,014	0,005	-0,05	-2.836	0,005	-0,023	-0,004
TmpMedBaixConsLNO_disc	-0,008	0,003	-0,044	-2.357	0,018	-0,015	-0,001
VTpPri_ConsCOB_5a_disc	0,067	0,026	0,042	2.552	0,011	0,016	0,119
VTpPri_ConsSFN_5a_disc	-0,014	0,004	-0,085	-3.200	0,001	-0,023	-0,005
QtMaxInc180dAposCons_disc	0,03	0,014	0,148	2.188	0,029	0,003	0,057
Setor_CEP	-0,003	0,002	-0,019	-1.207	0,228	-0,008	0,002
VTpPri_ConsBCO_5a_disc	0,01	0,005	0,063	2.165	0,03	0,001	0,02
VTpPri_ConsSEG_5a_disc	-0,019	0,007	-0,052	-2.759	0,006	-0,032	-0,005
PercDomicSemRenda_disc	-0,014	0,006	-0,049	-2.273	0,023	-0,026	-0,002
VTpPri_ConsCFL_5a_disc	0,015	0,006	0,043	2.490	0,013	0,003	0,027
VTpPri_ConsIND_5a_disc	-0,029	0,013	-0,036	-2.202	0,028	-0,055	-0,003
PRENTRAB_disc	-0,008	0,003	-0,049	-2.855	0,004	-0,013	-0,002
VQt_ConsMED_10d_disc	0,083	0,031	0,086	2.723	0,006	0,023	0,143
VTpPri_ConsMED_5a_disc	-0,031	0,015	-0,065	-2.113	0,035	-0,06	-0,002
VQtRestrExc1A_disc	0,019	0,009	0,052	2.109	0,035	0,001	0,037
QtInc180dAposUltCons_disc	0,028	0,014	0,042	1.944	0,052	0	0,056
VTpPri_ConsCCP_5a_disc	-0,047	0,03	-0,027	-1.602	0,109	-0,105	0,011
IDADE_disc	-0,005	0,003	-0,031	-1.803	0,071	-0,011	0
IndFrdTelU1A_disc	-0,037	0,011	-0,062	-3.270	0,001	-0,06	-0,015
PESOURB_disc	0,045	0,011	0,274	4.073	0	0,023	0,067
ESAUDETOT_disc	-0,04	0,011	-0,242	-3.668	0	-0,061	-0,018
TpUltTEL_disc	-0,008	0,003	-0,041	-2.155	0,031	-0,014	-0,001
VNuMaxEmp_ConsSTS_360d_disc	0,013	0,007	0,038	1.903	0,057	0	0,026
QtInclEndU30d	0,079	0,032	0,067	2.451	0,014	0,016	0,143
IndFrdEndU1A_disc	-0,029	0,018	-0,044	-1.656	0,098	-0,064	0,005

Para a variável resposta *Ocorrência_ação_cível* foram selecionadas por meio do método *Stepwise* as seguintes variáveis:

model	Beta	Std. Error	Std. Beta	t	Sig	lower	upper
(Intercept)	0,241	0,054		4.446	0	0,135	0,348
VQtRestrSemTELECOMInc1ARes_disc	0,059	0,013	0,136	4.430	0	0,033	0,084
PPOB_disc	0,02	0,005	0,124	3.943	0	0,01	0,03
PercDomicilios2SM_disc	-0,028	0,007	-0,093	-3.955	0	-0,042	-0,014
FECTOT_disc	-0,011	0,004	-0,063	-2.424	0,015	-0,019	-0,002
VNuEmp_ConsSTS_10d_disc	0,096	0,026	0,13	3.763	0	0,046	0,146
IDADE_disc	0,01	0,003	0,058	3.591	0	0,005	0,016
TmpMedBaixConsCredor_disc	-0,01	0,003	-0,055	-3.202	0,001	-0,017	-0,004
VNuEmp_ConsCHQSTS_180d_disc	0,042	0,012	0,057	3.463	0,001	0,018	0,066
TmpMedBaixConsLNO_disc	-0,018	0,003	-0,106	-5.153	0	-0,025	-0,011
VQtCredoresExc_disc	0,03	0,01	0,101	3.141	0,002	0,011	0,049
TmpMedConsIncl_disc	0,012	0,004	0,06	3.028	0,002	0,004	0,02
VTpUlt_ConsCRDTOD_5a_disc	-0,006	0,003	-0,034	-1.871	0,061	-0,012	0
IndFrdTelU1A_disc	-0,016	0,012	-0,026	-1.264	0,206	-0,04	0,009
VQt_ConsALN_180d_disc	0,023	0,009	0,043	2.456	0,014	0,005	0,042
PRENTRAB_disc	-0,008	0,003	-0,048	-2.602	0,009	-0,014	-0,002
VQt_ConsTCO_360d_disc	0,019	0,005	0,057	3.571	0	0,009	0,03

model	Beta	Std. Error	Std. Beta	t	Sig	lower	upper
VQtRestrFIRes_disc	-0,019	0,008	-0,058	-2.594	0,01	-0,034	-0,005
VTpPri_ConsVES_5a_disc	-0,022	0,008	-0,047	-2.797	0,005	-0,037	-0,007
VTpPri_ConsCTV_5a_disc	0,016	0,006	0,05	2.936	0,003	0,005	0,027
VQt_ConsMED_360d_disc	-0,026	0,011	-0,038	-2.390	0,017	-0,046	-0,005
VQtRestrLDepAti_disc	0,251	0,089	0,043	2.816	0,005	0,076	0,427
ClasseSocial_disc	-0,021	0,01	-0,05	-2.125	0,034	-0,04	-0,002
MORT1_disc	0,015	0,004	0,091	3.345	0,001	0,006	0,024
E_ANOSESTUDO_disc	-0,009	0,003	-0,054	-2.950	0,003	-0,015	-0,003
region_CEP	0,005	0,003	0,028	1.749	0,08	-0,001	0,011
VTpPri_ConsTCO_5a_disc	-0,005	0,003	-0,027	-1.637	0,102	-0,01	0,001
VQtRestrLDep_disc	-0,031	0,016	-0,032	-1.903	0,057	-0,063	0,001
AREAMUN_disc	-0,008	0,003	-0,046	-2.418	0,016	-0,014	-0,002
ESAUDETOT_disc	0,008	0,004	0,045	2.097	0,036	0,001	0,015
QtDiasTel_disc	-0,009	0,004	-0,054	-2.527	0,012	-0,017	-0,002
TpUltTEL_disc	0,007	0,004	0,04	1.815	0,07	-0,001	0,016
VQt_ConsCRDSTS_10d_disc	-0,054	0,025	-0,069	-2.106	0,035	-0,104	-0,004
VQt_ConsCRDSTS_360d_disc	0,011	0,005	0,049	2.414	0,016	0,002	0,021
VQt_ConsBCO_10d_disc	-0,051	0,021	-0,054	-2.398	0,017	-0,092	-0,009
VTpPri_ConsIND_5a_disc	-0,021	0,012	-0,027	-1.718	0,086	-0,044	0,003
QtCons360dAposUltsBaixa_disc	0,019	0,01	0,043	1.912	0,056	0	0,038
QtMaxInc30dAposCons_disc	-0,026	0,011	-0,083	-2.459	0,014	-0,047	-0,005
VQtRestrSemTELECOMInc_disc	0,017	0,008	0,074	1.989	0,047	0	0,033
VQtRestrExc180d_disc	0,021	0,012	0,047	1.717	0,086	-0,003	0,044
VTpPri_ConsBCO_5a_disc	0,006	0,003	0,033	1.676	0,094	-0,001	0,012

C.2 MÉTODO SEQUENCIAL DE SELEÇÃO DE VARIÁVEL

Para as variáveis resposta *Fraude_na_concessão*, *Conta_nunca_paga* e *Ocorrência_ação_cível* que foram consideradas nos modelos foram identificadas por meio do método sequencial de seleção de variáveis:

Fraude_na_concessão	Conta_nunca_paga	Ocorrência_ação_cível
BOLSA_FAMILIA	BOLSA_FAMILIA	FlAptoCap
FlAptoCap	QtInclEndU30d	QtInclEndU30d
VIndRestrReinc	IndPartME	VFlRestrAti180d
VFlRestrAti180d	EDiasFund_disc	region_CEP
IndPartPP	TmpMedConsIncl_disc	Sub_reg_CEP
region_CEP	QtInc180dAposUltCons_disc	QtMaxCons360dAposBaixa_disc
Setor_CEP	QtInc30dAposUltCons_disc	EDiasFund_disc
TmpMedBaixCons_disc	VTpPri_ConsMED_5a_disc	QtMaxInc360dAposCons_disc
TmpMedConsIncl_disc	VTpPri_ConsIND_5a_disc	VTpPri_ConsBCO_5a_disc
QtMaxInc180dAposCons_disc	VTpPri_ConsCCP_5a_disc	VTpPri_ConsFIN_5a_disc
QtInc30dAposUltCons_disc	VQtPdRec_ConsSTS_P02dU12d_disc	VTpPri_ConsVES_5a_disc
VTpPri_ConsCTC_5a_disc	VNuMaxEmp_ConsSTS_360d_disc	VTpPri_ConsIND_5a_disc
VTpPri_ConsCFL_5a_disc	VNuEmp_ConsSTS_180d_disc	VQt_ConsTOD_360d_disc
VTpUlt_ConsCRDTOD_5a_disc	VQt_ConsSFN_10d_disc	VNuEmp_ConsSTS_180d_disc
VTpPri_ConsCCP_5a_disc	VNuEmp_ConsSTS_10d_disc	VQt_ConsTOD_10d_disc
VQt_ConsTOD_180d_disc	VQt_ConsBCO_30d_disc	VQt_ConsCRDTOD_10d_disc
VTdRel_ConsSTS_P02dU30d_disc	VQt_ConsBCO_10d_disc	PercResponsaveis_disc
VNuEmp_ConsSTS_360d_disc	VQt_ConsCTC_10d_disc	ComIluminacaoPublica_disc
VQt_ConsCTC_360d_disc	VQt_ConsFIN_10d_disc	TpUltEnd_disc
VQt_ConsSEG_180d_disc	VQt_ConsMED_10d_disc	PPOB_disc
VQtMax_ConsSTS_P10dU5a_disc	PercResponsaveis_disc	EMP_SALMED_disc
VQt_ConsSFN_10d_disc	VTpPriRestr_disc	VQtRestrLDepAti_disc
VQt_ConsTCO_10d_disc	IDHM2010_disc	QtNuTel_disc

Fraude_na_concessão	**Conta_nunca_paga**	**Ocorrência_ação_cível**
VQt_ConsFIN_10d_disc	ESAUDETOT_disc	VQtRestrLDep_disc
VNuMaxEmp_ConsSTS_10d_disc	ClasseSocial_disc	VQtRestrSemTELECOMInc1ARes_disc
VQt_ConsVAR_10d_disc	VQtCredoresExc_disc	VQtRestrVl100Ati_disc
VNuAnos_ConsSEG_5a_disc	VQtRestrLDep_disc	VQtRestrExc1A_disc
PercResponsaveis_disc	VQtRestrSemTELECOMInc_disc	
PercDomicilios2SM_disc	escore_CREDITO_disc	
ComIluminacaoPublica_disc		
TpUltTEL_disc		
D_ComEnergia_disc		
ESPVIDA_disc		
IDHM2010_disc		
ESAUDETOT_disc		
IdadeEndA_disc		
IndFrdTelU1A_disc		
IndFrdEndU1A_disc		
VQtRestrFIAti_disc		
VQtRestrtvPaga_disc		
VQtRestrCTRes_disc		
VQtRestrEmprRes_disc		
VQtMaxRestrAti_disc		
escore_CREDITO_disc		

ANEXOS

ANEXO A – VARIÁVEIS DA BASE DISPONIBILIZADAS

CÓDIGO	TIPO	DESCRIÇÃO
Fraude_na_concessão	Num	Indicador de fraude encontrada no momento da avaliação do proponente
Conta_nunca_paga	Num	Indicador de conta nunca paga em 365 dias
Ocorrência_ação_cível	Num	Indicador de ocorrência de ação cível após negativação
chave	Num	Chave identificadora
mes	Num	Safra de referência
escore_CREDITO	Num	Escore de credito (quanto maior menor probabilidade de inadimplência)
AREAMUN	Num	Área da unidade territorial
BOLSA_FAMILIA	Char	Indicador de bolsa familia
CEPCOMPLETO	Num	cep
ClasseSocial	Num	Código de classe social a partir da renda média domiciliar do setor
ComIluminacaoPublica	Num	Domicílios particulares permanentes – Existe iluminação pública
D_ComEnergia	Num	Quantidade de domicílios particulares permanentes com energia elétrica
ECoMaxInadimpUlt_00m	Num	Máxima faixa de atraso no mês de referência (em pagamentos) da principal empresa onde o cpf participa
EDiasFund	Num	Idade (em dias) da principal empresa onde o cpf participa
EMP_PESS_ASSAL	Num	Pessoal ocupado assalariado
EMP_SALMED	Num	Salário médio mensal
EMaxQtConsEmp60d	Num	Quantidade máxima de consultas realizada pela mesma empresa nos últimos 60 dias da principal empresa onde o cpf participa
EPctMaxPartPF	Num	Percentual da maior participação societária dentre os sócios PF da principal empresa onde o cpf participa
EQTBANCOS	Num	Quantidade de bancos diferentes para um mesmo titular da principal empresa onde o cpf participa
EQtCons180d	Num	Quantidade de consultas realizadas nos últimos 180 dias da principal empresa onde o cpf participa
EQtConsCheq30d	Num	Quantidade de consultas realizadas a cheque nos últimos 30 dias da principal empresa onde o cpf participa
EQtConsCheq360d	Num	Quantidade de consultas realizadas a cheque nos últimos 360 dias da principal empresa onde o cpf participa
EQtConsCred360d	Num	Quantidade de consultas realizadas a crédito nos últimos 360 dias da principal empresa onde o cpf participa
EQtConsOrigSER180d	Num	Quantidade de consultas realizadas por empresas do segmento de serviços nos últimos 180 dias da principal empresa onde o cpf participa

CÓDIGO	TIPO	DESCRIÇÃO
EQtConsOrigSFN180d	Num	Quantidade de consultas realizadas por empresas do Sistema Financeiro Nacional nos últimos 180 dias da principal empresa onde o cpf participa
EQtConsPdRec15U90d	Num	Quantidade de consultas realizadas nos últimos 90 dias ponderadas por periodos de 15 dias. da principal empresa onde o cpf participa
EQtConsPdRec30U360d	Num	Quantidade de consultas realizadas nos últimos 360 dias ponderadas por periodos de 30 dias. da principal empresa onde o cpf participa
EQtConsSF	Num	Quantidade de consultas realizadas (incluindo consultas de telecomunicações e seguradoras, e consultas excluídas) nos últimos 5 anos da principal empresa onde o cpf participa
EQtConsSFOrigTEL	Num	Quantidade de consultas realizadas por empresas de telefonia e serviços públicos (incluindo consultas de telecomunicações e seguradoras, e consultas excluídas) nos últimos 5 anos da principal empresa onde o cpf participa
EQtEmpConsCheq30d	Num	Quantidade de empresas distintas que realizaram alguma consulta a cheque nos últimos 30 dias da principal empresa onde o cpf participa
EQtEmpConsCheq360d	Num	Quantidade de empresas distintas que realizaram alguma consulta a cheque nos últimos 360 dias da principal empresa onde o cpf participa
EQtFornecDifUlt_02m	Num	Quantidade de fornecedores distintos em que ocorreram relacionamentos comerciais nos últimos 2 meses da principal empresa onde o cpf participa
EQtFornec_00m_06m	Num	Quantidade de fornecedores no período de 0 a 6 meses da principal empresa onde o cpf participa
EQtPgtoFaixa1Ult_12m	Num	Quantidade total dos pagamentos com atraso entre 1 e 7 dias nos últimos 12 meses da principal empresa onde o cpf participa
EQtPgtoVistaUlt_12m	Num	Quantidade total dos pagamentos a vista nos últimos 12 meses da principal empresa onde o cpf participa
EQtRecheqOcor06M	Num	Quantidade de ocorrências de cheques roubados, sustados, extraviados ou cancelados nos últimos 6 meses da principal empresa onde o cpf participa
EQtRecheqOcor12M	Num	Quantidade de ocorrências de cheques roubados, sustados, extraviados ou cancelados nos últimos 12 meses da principal empresa onde o cpf participa
EQtRecheqSusS12M	Num	Quantidade de ocorrências de cheques nos últimos 12 meses do tipo SUSTADO da principal empresa onde o cpf participa
EQtfamilia	Num	Número de solicitações de cheque sustado. da principal empresa onde o cpf participa
ESAUDETOT	Num	Quantidade total de Estabelecimentos de Saúde

CÓDIGO	TIPO	DESCRIÇÃO
ESPVIDA	Num	Esperança de vida ao nascer
ETpEntrUltSoc	Num	Tempo de entrada do último sócio no quadro societário da principal empresa onde o cpf participa
ETpMedCons360d	Num	Tempo médio entre as consultas nos últimos 360 dias da principal empresa onde o cpf participa
ETpPriConsCheq	Num	Tempo desde a primeira consulta realizada a cheque nos últimos 5 anos da principal empresa onde o cpf participa
ETpPriConsCred	Num	Tempo desde a primeira consulta realizada a crédito nos últimos 5 anos da principal empresa onde o cpf participa
ETpPriConsOrigBCO	Num	Tempo desde a primeira consulta realizada por empresas do segmento de bancos nos últimos 5 anos da principal empresa onde o cpf participa
EVFlRestrAti1A	Num	Flag de restrição ativa 1 ano antes da data de referência da principal empresa onde o cpf participa
EVQtAcoes	Num	Quantidade de ações incluídas nos últimos 5 anos da principal empresa onde o cpf participa
EVQtAcoesAti	Num	Quantidade de Ações que estão ativas da principal empresa onde o cpf participa
EVQtCCFs	Num	Quantidade de cheques sem fundos (cheques sem fundos) da principal empresa onde o cpf participa
EVQtCCFsAti	Num	Quantidade de cheques sem fundos ativos da principal empresa onde o cpf participa
EVQtCCFsAti030DPI	Num	Quantidade de cheques sem fundos que estão ativos a menos de 31 dias da principal empresa onde o cpf participa
EVQtCredoresAti	Num	Quantidade de diferentes credores onde possui restrição ativa da principal empresa onde o cpf participa
EVQtMaxRestrAti	Num	Quantidade máxima de restrições que ficaram ativas ao mesmo tempo da principal empresa onde o cpf participa
EVQtRefinsResU12m	Num	Quantidade de pagamentos de refin realizados nos últimos 12 meses da data de referência da principal empresa onde o cpf participa
EVQtRestrAti	Num	Quantidade de restrições ativas. da principal empresa onde o cpf participa
E_ANOSESTUDO	Num	Expectativa de anos de estudo
FECTOT	Num	Taxa de fecundidade total
FlAptoCap	Num	Indicativo de que o cpf possui um endereço na capital e que faz parte de um edifício.
FlAptoInt	Num	Indicativo de que o cpf possui um endereço no interior e que faz parte de um edifício.

CÓDIGO	TIPO	DESCRIÇÃO
IDADE	Num	IDADE
IDHM2010	Num	Índice de Desenvolvimento Humano Municipal - 2010
IdadeEndA	Num	Idade estimada em anos do endereço. (considerando apenas os registros sem inconsistência)
IndFrdEndU1A	Num	Índice de fraude = Quantidade de NSU inseridos no último mês / Quantidade de NSU inseridos no último ano (não considerando o último mês).
IndFrdTelU1A	Num	Índice de fraude = Quantidade de telefones inseridos no último mês / Quantidade de telefones inseridos nos último ano (não considerando o último mês).
IndPartME	Num	Indicativo se o cpf possui alguma participação em micro empresas (empresas com faturamento anual de 60 mil até 360 mil)
IndPartMEI	Num	Indicativo de que o cpf possui uma micro empresa individual (empresa com faturamento anual de até 60mil)
IndPartPP	Num	Indicativo se o cpf possui alguma participação em empresas de pequeno porte (empresas com faturamento anual de 360mil até 4,2MM)
MORT1	Num	Mortalidade infantil
OBITOSTOT	Num	Óbitos Total
PEA	Num	População economicamente ativa
PESOURB	Num	População residente na área urbana
PESS_OCUP	Num	Pessoal ocupado total
PPOB	Num	Proporção de vulneráveis à pobreza
PRENTRAB	Num	Percentual da renda proveniente de rendimentos do trabalho
PercDomicSemRenda	Num	Percentual de domicílios sem renda
PercDomicilios2SM	Num	Percentual de domicílios com rendimentos de até 2 SM
PercResponsaveis	Num	Responsáveis pelos domicílios particulares
QtCons180dAposUltsBaixa	Num	Quantidade de consultas, considerando o período de 180 dias após a ultima baixa de restritivo.
QtCons30dAposUltsBaixa	Num	Quantidade de consultas, considerando o período de 30 dias após a ultima baixa de restritivo.
QtCons360dAposUltsBaixa	Num	Quantidade de consultas, considerando o período de 360 dias após a ultima baixa de restritivo.
QtConsLNO	Num	Quantidade de consultas após baixa de restritivo, onde ficou sem restritivos (nome limpo)
QtDiasEnd	Num	Tempo desde o primeiro registro de endereço até a data de refêrencia
QtDiasTel	Num	Tempo desde o primeiro registro de telefone até a data de refêrencia
QtEnd	Num	Quantidade de NSU cadastrados (considerando todo o histórico da base)
QtInc180dAposUltCons	Num	Quantidade de restritivos incluidos, considerando o período de 180 dias após a ultima consulta.

CÓDIGO	TIPO	DESCRIÇÃO
QtInc30dAposUltCons	Num	Quantidade de restritivos incluidos, considerando o período de 30 dias após a ultima consulta.
QtInc360dAposUltCons	Num	Quantidade de restritivos incluidos, considerando o período de 360 dias após a ultima consulta.
QtInclEndU1A	Num	Quantidade de NSU inseridos no último ano
QtInclEndU30d	Num	Quantidade de NSU inseridos no últimos mês
QtMaxCons180dAposBaixa	Num	Quantidade máxima de consultas, considerando um período de 180 dias após uma baixa de restritivo.
QtMaxCons30dAposBaixa	Num	Quantidade máxima de consultas, considerando um período de 30 dias após uma baixa de restritivo.
QtMaxCons360dAposBaixa	Num	Quantidade máxima de consultas, considerando um período de 360 dias após uma baixa de restritivo.
QtMaxInc180dAposCons	Num	Quantidade máxima de restritivos incluidos, considerando um período de 180 dias após uma consulta.
QtMaxInc30dAposCons	Num	Quantidade máxima de restritivos incluidos, considerando um período de 30 dias após uma consulta.
QtMaxInc360dAposCons	Num	Quantidade máxima de restritivos incluidos, considerando um período de 360 dias após uma consulta.
QtNuTel	Num	Quantidade de telefones cadastrados (considerando todo o historico da base)
RDPC	Num	Renda per capita média
TmpMedBaixCons	Num	Tempo médio entre uma baixa de restritivo e a primeira consulta
TmpMedBaixConsCredor	Num	Tempo médio entre uma baixa de restritivo e a primeira consulta de mesmo credor
TmpMedBaixConsLNO	Num	Tempo médio entre uma baixa de restritivo, onde ficou sem restritivos (nome limpo) e a primeira consulta
TmpMedConsIncl	Num	Tempo médio entre uma consulta e a primeira inclusão
TpUltEnd	Num	Tempo desde o último registro de endereço até a data de refêrencia
TpUltTEL	Num	Tempo desde o último registro de telefone até a data de refêrencia
VFlPriRestrAti	Num	Indicativo de que possui a primeira dívida vencida negativada - últimos 5 anos.-1: não possui dívida vencida (negativada ou regularizada) na data de referencia; 0: possui dívida vencida regularizada e/ou dívida vencida negativada na data de referencia; 1: possui apenas uma dívida vencida negativada na data de referencia.
VFlRestrAti180d	Num	Indicativo de que possui dívida vencida negativada ou que tenha sido regularizada nos últimos 180 dias.

CÓDIGO	TIPO	DESCRIÇÃO
VFlRestrAti1A	Num	Indicativo de que possui dívida vencida negativada ou que tenha sido regularizada no último ano.
VFlRestrRes	Num	Indicativo de que possui dívida vencida regularizada na data de referencia.0: não possui dívida vencida regularizada na data de referencia; 1: possui dívida vencida regularizada na data de referencia.
VFlSemAtiComRes	Num	Indicativo de que não possui dívida vencida negativada mas possui dívida vencida regularizada na data de referencia. 0: não possui nenhuma dívida vencida e/ou não possui dívida vencida negativada na data de referencia; 1: possui apenas dívida vencidas regularizadas na data de referencia.
VFlSemRestr	Num	Indicativo de que não possuiu nenhuma dívida vencida (negativada ou regularizada) nos últimos 5 anos 0: possuiu, no mínimo, uma dívida vencida (negativada ou regularizada) nos últimos 5 anos; 1: não possuiu nenhuma dívida vencida (negativada ou regularizada) nos últimos 5 anos
VIndRestrReinc	Num	Indicador de ocorrência reincidente ativa ou resolvida.
VIndTodasRestrCoPart	Num	Indicativo de que todas as dívidas vencidas (negativadas ou regularizadas) são de "co-participação"
VInd_ConsTOD_5a	Num	Indicativo de consultas (de todos os segmentos) nos últimos 5 anos
VNuAnos_ConsSEG_5a	Num	Número de anos de consultas de Seguradoras nos últimos 5 anos
VNuEmp_ConsCHQSTS_10d	Num	Número de empresas que realizaram consultas à cheques (exceto Telecoms e Seguradoras) nos últimos 10 dias
VNuEmp_ConsCHQSTS_180d	Num	Número de empresas que realizaram consultas à cheques (exceto Telecoms e Seguradoras) nos últimos 180 dias
VNuEmp_ConsCHQSTS_360d	Num	Número de empresas que realizaram consultas à cheques (exceto Telecoms e Seguradoras) nos últimos 360 dias
VNuEmp_ConsSEG_10d	Num	Número de empresas que realizaram consultas de Seguradoras nos últimos 10 dias
VNuEmp_ConsSEG_180d	Num	Número de empresas que realizaram consultas de Seguradoras nos últimos 180 dias
VNuEmp_ConsSEG_360d	Num	Número de empresas que realizaram consultas de Seguradoras nos últimos 360 dias
VNuEmp_ConsSTS_10d	Num	Número de empresas que realizaram consultas (exceto de Telecoms e Seguradoras) nos últimos 10 dias
VNuEmp_ConsSTS_180d	Num	Número de empresas que realizaram consultas (exceto de Telecoms e Seguradoras) nos últimos 180 dias
VNuEmp_ConsSTS_360d	Num	Número de empresas que realizaram consultas (exceto de Telecoms e Seguradoras) nos últimos 360 dias

CÓDIGO	TIPO	DESCRIÇÃO
VNuMaxEmp_ConsSTS_10d	Num	Número máximo de consultas que uma mesma empresa fez ao CPF/CNPJ (exceto de Telecoms e Seguradoras) nos últimos 10 dias
VNuMaxEmp_ConsSTS_180d	Num	Número máximo de consultas que uma mesma empresa fez ao CPF/CNPJ (exceto de Telecoms e Seguradoras) nos últimos 180 dias
VNuMaxEmp_ConsSTS_360d	Num	Número máximo de consultas que uma mesma empresa fez ao CPF/CNPJ (exceto de Telecoms e Seguradoras) nos últimos 360 dias
VQtCredoresAti	Num	Número de diferentes empresas que incluíram dívidas Serasa (REFIN e PEFIN) vencidas negativadas
VQtCredoresExc	Num	Número de empresas diferentes que incluíram dívidas Serasa (REFIN e PEFIN) vencidas regularizadas
VQtMaxRestrAti	Num	Quantidade máxima de dívidas vencidas (negativadas ou regularizadas) que ficaram negativadas ao mesmo tempo
VQtMax_ConsSTS_P10dU5a	Num	Quantidade máxima de consultas (exceto de Telecoms e Seguradoras) em periodos de 10 dias nos últimos 5 anos
VQtPdRec_ConsSTS_P02dU12d	Num	Quantidade ponderada pela recência de consultas (exceto de Telecoms e Seguradoras) em períodos de 2 dias nos últimos 12 dias
VQtRestrAti	Num	Quantidade de dívidas vencidas negativadas
VQtRestrAti030DPI	Num	Quantidade de dívidas vencidas negativadas incluídas nos últimos 30 dias
VQtRestrAtiAtraso180d	Num	Quantidade de dívidas vencidas negativadas com atraso superior a 180 dias (vencidas a mais de 180 dias)
VQtRestrCTAti	Num	Quantidade de restrições ativas por motivo de cartão de crédito
VQtRestrCTAti030DPI	Num	Quantidade de restrições ativas por motivo de cartão de crédito nos últimos 30 dias
VQtRestrCTRes	Num	Quantidade de restrições por motivo de cartão de crédito que tiveram pagamento realizado.
VQtRestrEmprAti	Num	Quantidade de restrições ativas por motivo de empréstimo.
VQtRestrEmprAti030DPI	Num	Quantidade de restrições ativas por motivo de empréstimo nos últimos 30 dias
VQtRestrEmprRes	Num	Quantidade de restrições por motivo de empréstimo que tiveram pagamento realizado.
VQtRestrExc180d	Num	Quantidade de dívidas vencidas regularizadas nos últimos 180 dias.
VQtRestrExc1A	Num	Quantidade de dívidas vencidas regularizadas no último ano.
VQtRestrFIAti	Num	Quantidade de restrições ativas por motivo de financiamento

CÓDIGO	TIPO	DESCRIÇÃO
VQtRestrFIAti030DPI	Num	Quantidade de restrições ativas por motivo de financiamento nos últimos 30 dias
VQtRestrFIRes	Num	Quantidade de restrições por motivo de financiamento que tiveram pagamento realizado.
VQtRestrLDep	Num	Quantidade total de Restrições de Lojas de Departamento - Periodo Total 5 Anos
VQtRestrLDepAti	Num	Quantidade de Restrições Ativas de Lojas de Departamento
VQtRestrLDepAti030DPI	Num	Quantidade de Restrições Ativas de Lojas de Departamento: incluídas nos últimos 30 dias
VQtRestrSemTELECOMInc	Num	Quantidade de dívidas Serasa (REFIN e PEFIN) vencidas (negativadas ou regularizadas) incluídas por empresas diferentes de Telefonia
VQtRestrSemTELECOMInc1ARes	Num	Quantidade de dívidas Serasa (REFIN e PEFIN) vencidas regularizadas, incluídas por empresas diferentes de Telefonia no último ano.
VQtRestrVl100Ati	Num	Quantidade de restrições ativas com valor igual ou superior a R$ 100,00
VQtRestrtvPaga	Num	Quantidade total de Restrições de Tv Paga - Periodo Total 5 Anos
VQtRestrtvPagaAti	Num	Quantidade de Restrições Ativas de Tv Paga
VQtRestrtvPagaAti030DPI	Num	Quantidade de Restrições Ativas de Tv Paga: incluídas a mais de 30 dias
VQt_ConsALI_10d	Num	Quantidade de consultas de Alimentação / Supermercados nos últimos 10 dias
VQt_ConsALI_30d	Num	Quantidade de consultas de Alimentação / Supermercados nos últimos 30 dias
VQt_ConsALI_360d	Num	Quantidade de consultas de Alimentação / Supermercados nos últimos 360 dias
VQt_ConsALN_180d	Num	Quantidade de consultas de confirmação de dados cadastrais nos últimos 180 dias.
VQt_ConsALN_90d	Num	Quantidade de consultas de confirmação de dados cadastrais nos últimos 90 dias.
VQt_ConsASS_10d	Num	Quantidade de consultas de CDL's, Associações e Distribuidores de Informação nos últimos 10 dias
VQt_ConsASS_180d	Num	Quantidade de consultas de CDL's, Associações e Distribuidores de Informação nos últimos 180 dias
VQt_ConsASS_30d	Num	Quantidade de consultas de CDL's, Associações e Distribuidores de Informação nos últimos 30 dias
VQt_ConsASS_360d	Num	Quantidade de consultas de CDL's, Associações e Distribuidores de Informação nos últimos 360 dias

CÓDIGO	TIPO	DESCRIÇÃO
VQt_ConsBCO_10d	Num	Quantidade de consultas de Bancos Múltiplos nos últimos 10 dias
VQt_ConsBCO_30d	Num	Quantidade de consultas de Bancos Múltiplos nos últimos 30 dias
VQt_ConsCCP_10d	Num	Quantidade de consultas de Cooperativas de Crédito (inclusive Rural) nos últimos 10 dias
VQt_ConsCCP_180d	Num	Quantidade de consultas de Cooperativas de Crédito(inclusive Rural) nos últimos 180 dias
VQt_ConsCCP_30d	Num	Quantidade de consultas de Cooperativas de Crédito (inclusive Rural) nos últimos 30 dias
VQt_ConsCCP_360d	Num	Quantidade de consultas de Cooperativas de Crédito (inclusive Rural) nos últimos 360 dias
VQt_ConsCRDSTS_10d	Num	Quantidade de consultas aos relatórios de crédito (exceto Telecoms e Seguradoras) nos últimos 10 dias
VQt_ConsCRDSTS_180d	Num	Quantidade de consultas aos relatórios de crédito (exceto Telecoms e Seguradoras) nos últimos 180 dias
VQt_ConsCRDSTS_360d	Num	Quantidade de consultas aos relatórios de crédito (exceto Telecoms e Seguradoras) nos últimos 360 dias
VQt_ConsCRDTOD_10d	Num	Quantidade de consultas aos relatórios de crédito (de todos os segmentos) nos últimos 10 dias. >0: Quantidade deconsultas; 0: Não possui consulta para o periodo indicado;-1: Não possui consulta para o segmento indicado;-2: Não possui consulta (de nenhum segmento) nos últimos 5 anos.
VQt_ConsCRDTOD_180d	Num	Quantidade de consultas aos relatórios de crédito (de todos os segmentos) nos últimos 180 dias. >0: Quantidade de consultas; 0: Não possui consulta para o periodo indicado;-1: Não possui consulta para o segmento indicado;-2: Não possui consulta (de nenhum segmento) nos últimos 5 anos.
VQt_ConsCRDTOD_360d	Num	Quantidade de consultas aos relatórios de crédito (de todos os segmentos) nos últimos 360 dias. >0: Quantidade de consultas; 0: Não possui consulta para o periodo indicado;-1: Não possui consulta para o segmento indicado;-2: Não possui consulta (de nenhum segmento) nos últimos 5 anos.

CÓDIGO	TIPO	DESCRIÇÃO
VQt_ConsCTC_10d	Num	Quantidade de consultas de Adm. de Cartões de Crédito (inclusive os fornecidos pelo Varejo) nos últimos 10 dias
VQt_ConsCTC_180d	Num	Quantidade de consultas de Adm. de Cartões de Crédito (inclusive os fornecidos pelo Varejo) nos últimos 180 dias
VQt_ConsCTC_360d	Num	Quantidade de consultas de Adm. de Cartões de Crédito (inclusive os fornecidos pelo Varejo) nos últimos 360 dias
VQt_ConsFIN_10d	Num	Quantidade de consultas de Financeiras nos últimos 10 dias
VQt_ConsFIN_180d	Num	Quantidade de consultas de Financeiras nos últimos 180 dias
VQt_ConsFIN_360d	Num	Quantidade de consultas de Financeiras nos últimos 360 dias
VQt_ConsMED_10d	Num	Quantidade de consultas de Comércio de Móveis, Eletrodomésticos e Lojas de Departamento nos últimos 10 dias
VQt_ConsMED_180d	Num	Quantidade de consultas de Comércio de Móveis, Eletrodomésticos e Lojas de Departamento nos últimos 180 dias
VQt_ConsMED_360d	Num	Quantidade de consultas de Comércio de Móveis, Eletrodomésticos e Lojas de Departamento nos últimos 360 dias
VQt_ConsSEG_10d	Num	Quantidade de consultas de Seguradoras nos últimos 10 dias
VQt_ConsSEG_180d	Num	Quantidade de consultas de Seguradoras nos últimos 180 dias
VQt_ConsSEG_360d	Num	Quantidade de consultas de Seguradoras nos últimos 360 dias
VQt_ConsSFN_10d	Num	Quantidade de consultas do setor financeiro nos últimos 10 dias
VQt_ConsSFN_180d	Num	Quantidade de consultas do setor financeiro nos últimos 180 dias
VQt_ConsSFN_360d	Num	Quantidade de consultas do setor financeiro nos últimos 360 dias
VQt_ConsTCO_10d	Num	Quantidade de consultas de Telecoms (inclusive fornecedoras de sinal de tv a cabo e internet) nos últimos 10 dias
VQt_ConsTCO_180d	Num	Quantidade de consultas de Telecoms (inclusive fornecedoras de sinal de tv a cabo e internet) nos últimos 180 dias
VQt_ConsTCO_360d	Num	Quantidade de consultas de Telecoms (inclusive fornecedoras de sinal de tv a cabo e internet) nos últimos 360 dias
VQt_ConsTOD_10d	Num	Quantidade de consultas (de todos os segmentos) nos últimos 10 dias
VQt_ConsTOD_180d	Num	Quantidade de consultas (de todos os segmentos) nos últimos 180 dias
VQt_ConsTOD_360d	Num	Quantidade de consultas (de todos os segmentos) nos últimos 360 dias
VQt_ConsVAR_10d	Num	Quantidade de consultas do setor varejista nos últimos 10 dias
VQt_ConsVAR_180d	Num	Quantidade de consultas do setor varejista nos últimos 180 dias

CÓDIGO	TIPO	DESCRIÇÃO
VQt_ConsVAR_360d	Num	Quantidade de consultas do setor varejista nos últimos 360 dias
VQt_ConsVES_10d	Num	Quantidade de consultas de comércio de vestuário nos últimos 10 dias
VQt_ConsVES_180d	Num	Quantidade de consultas de comércio de vestuário nos últimos 180 dias
VQt_ConsVES_360d	Num	Quantidade de consultas de comércio de vestuário nos últimos 360 dias
VTdRel_ConsSTS_P02dU30d	Num	Tendência relativa de consultas (exceto de Telecoms e Seguradoras) em períodos de 2 dias nos últimos 30 dias
VTpMed_ConsSTS_10d	Num	Tempo médio entre consultas (exceto de Telecoms e Seguradoras) nos últimos 10 dias
VTpPriRestr	Num	Tempo desde a primeira inclusão de dívida vencida (negativada ou regularizada)
VTpPriRestrIncAti	Num	Tempo desde a primeira inclusão de dívida vencida negativada
VTpPri_ConsALI_5a	Num	Tempo desde a primeira consulta de Alimentação / Supermercados nos últimos 5 anos
VTpPri_ConsASS_5a	Num	Tempo desde a primeira consulta de CDL's, Associações e Distribuidores de Informação nos últimos 5 anos
VTpPri_ConsBCO_5a	Num	Tempo desde a primeira consulta de Bancos Múltiplos nos últimos 5 anos
VTpPri_ConsCCP_5a	Num	Tempo desde a primeira consulta de Cooperativas de Crédito (inclusive Rural) nos últimos 5 anos
VTpPri_ConsCFL_5a	Num	Tempo desde a primeira consulta para comércio e financiamento de veículos leves (carros e motos) nos últimos 5 anos
VTpPri_ConsCHQSTS_5a	Num	Tempo desde a primeira consulta à cheques (exceto Telecoms e Seguradoras) nos últimos 5 anos
VTpPri_ConsCOB_5a	Num	Tempo desde a primeira consulta para cobrança, factoring e fomento Mercantil nos últimos 5 anos
VTpPri_ConsCRDSTS_5a	Num	Tempo desde a primeira consulta aos relatórios de crédito (exceto Telecoms e Seguradoras) nos últimos 5 anos
VTpPri_ConsCRDTOD_5a	Num	Tempo desde a primeira consulta aos relatórios de crédito (todos os segmentos) nos últimos 5 anos.>0: Tempo desde a primeira consulta; 0: a primeira consulta foi realizada na mesma data da referência; -1: Não possui consulta para o segmento indicado; -2: Não possui consulta (de nenhum segmento) nos últimos 5 anos.
VTpPri_ConsCTC_5a	Num	Tempo desde a primeira consulta de Adm. de Cartões de Crédito (inclusive os fornecidos pelo Varejo) nos últimos 5 anos
VTpPri_ConsCTV_5a	Num	Tempo desde a primeira consulta de Adm. de Cartões de Crédito exclusivamente fornecidos pelo Varejo nos últimos 5 anos
VTpPri_ConsFIN_5a	Num	Tempo desde a primeira consulta de Financeiras nos últimos 5 anos

CÓDIGO	**TIPO**	**DESCRIÇÃO**
VTpPri_ConsIND_5a	Num	Tempo desde a primeira consulta de Indústrias nos últimos 5 anos
VTpPri_ConsMED_5a	Num	Tempo desde a primeira consulta de Comércio de Móveis, Eletrodomésticos e Lojas de Departamento nos últimos 5 anos
VTpPri_ConsSEG_5a	Num	Tempo desde a primeira consulta de Seguradoras nos últimos 5 anos
VTpPri_ConsSFN_5a	Num	Tempo desde a primeira consulta do setor financeiro nos últimos 5 anos
VTpPri_ConsSSG_5a	Num	Tempo desde a primeira consulta (exceto de Seguradoras) nos últimos 5 anos
VTpPri_ConsSTS_5a	Num	Tempo desde a primeira consulta (exceto de Telecoms e Seguradoras) nos últimos 5 anos
VTpPri_ConsTCO_5a	Num	Tempo desde a primeira consulta de Telecoms (inclusive fornecedoras de sinal de tv a cabo e internet) nos últimos 5 anos
VTpPri_ConsTOD_5a	Num	Tempo desde a primeira consulta (de todos os segmentos) nos últimos 5 anos
VTpPri_ConsVAR_5a	Num	Tempo desde a primeira consulta do setor varejista nos últimos 5 anos
VTpPri_ConsVES_5a	Num	Tempo desde a primeira consulta de comércio de vestuário nos últimos 5 anos
VTpUlt_ConsCRDTOD_5a	Num	Tempo desde a última consulta aos relatórios de crédito (todos os segmentos) nos últimos 5 anos.