Rodolfo Hauret Spolador

Aplicação do método de *Gradient Boosting*

Niterói - RJ, Brasil

10 de maio de 2021

Universidade Federal Fluminense

Rodolfo Hauret Spolador

Aplicação do método de *Gradient Boosting*

Trabalho de Conclusão de Curso

Monografia apresentada para obtenção do grau de Bacharel em Estatística pela Universidade Federal Fluminense.

Orientador(a): Prof. Dr. Karina Yuriko Yaginuma

Niterói - RJ, Brasil

10 de maio de 2021

Universidade Federal Fluminense

Rodolfo Hauret Spolador

Aplicação do método de *Gradient Boosting*

Monografia de Projeto Final de Graduação sob o título *\Aplicacao do metodo de Gradient Boosting"*, defendida por Rodolfo Hauret Spolador e aprovada em 10 de maio de 2021, na cidade de Niterói, no Estado do Rio de Janeiro, pela banca examinadora constituída pelos professores:

Profa. Dra. Karina Yuriko Yaginuma
Departamento de Estatística – UFF

Prof. Dr. Douglas Rodrigues Pinto
Departamento de Estatística – UFF

Profa. Dra. Jessica Quintanilha Kubrusly
Departamento de Estatística – UFF

Niterói, 10 de maio de 2021

Ficha catalográfica automática - SDC/BIME
Gerada com informações fornecidas pelo autor

S762a Spolador, Rodolfo Hauret
Aplicação do método de Gradient Boosting / Rodolfo Hauret Spolador ; Karina Yuriko Yaginuma, orientadora. Niterói, 2021.
63 f. : il.

Trabalho de Conclusão de Curso (Graduação em Estatística)-Universidade Federal Fluminense, Instituto de Matemática e Estatística, Niterói, 2021.

1. Aprendizado de Máquina. 2. Gradient Boosting. 3. Regressão Logística. 4. Classificação. 5. Produção intelectual. I. Yaginuma, Karina Yuriko, orientadora. II. Universidade Federal Fluminense. Instituto de Matemática e Estatística. III. Título.

CDD -

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

Resumo

Devido ao aumento exponencial da quantidade de dados, os custos mais baixos de processamento computacional e uma maior acessibilidade no armazenamento de dados, as técnicas de aprendizado de máquinas tornaram-se mais atrativas. O aprendizado de máquina é um método de análise de dados que automatiza o desenvolvimento de modelos e permite a criação de modelos preditores, que auxiliam na tomada de decisões, reduzindo assim possíveis riscos. Os modelos de previsão de aprendizado de máquinas podem utilizar de regressões, árvores de classificação, entre outros. Neste trabalho é estudado o modelo supervisionado de *Gradient Boosting* que é baseado em árvores de classificação, ele constrói o modelo em etapas, como outros métodos de boosting, e os generaliza, permitindo a otimização de uma função de perda diferenciável arbitrária. Este método e o método de Regressão Logística serão aplicados em um conjunto de dados rotulados, a fim de compará-los. Os resultados obtidos foram diferentes em ambos os métodos, nos dados de treino o modelo de Gradient Boosting apresentou maiores valores de AUC do que o modelo de Regressão Logística, entretanto este padrão não se manteve na base de teste. O melhor modelo de *Gradient Boosting* ajustado apresentou uma acurácia de 0.5685, este modelo apresentou métricas de sensibilidade (0.652) e especificidade (0.4063) não muito discrepantes, indicando que ele acerta bem ambas as características, enquanto que o melhor modelo de Regressão Logística foi o modelo com uma acurácia de 0.6054, sensibilidade (0.81) e especificidade (0.2084). Apesar do modelo de Regressão Logística apresentar maior acurácia, considerou-se que o Gradient Boosting apresentou melhor desempenho, visto que ele acertou as duas características da variável resposta de forma mais consistente.

Palavras-chave: Aprendizado de Máquina. *Gradient Boosting*. Regressão Logística. Classificação.

Dedicatória

Dedico este trabalho aos meus pais, que batalharam muito para me permitir chegar aqui. Dedico também aos professores e amigos que estiveram ao meu lado durante a minha graduação.

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

Realizar previsões é muito importante para o ser humano, pois elas auxiliam nas tomadas de decisão. Com o aumento dos dados coletados, e a entrada na era do *Big Data*, tornou-se inviável a análise de dados de forma manual. Graças às novas tecnologias computacionais, a análise dessa grande massa de dados foi viabilizada, com isso foi criado o *Machine Learning*.

O Aprendizado de Máquina (em inglês, *Machine Learning*) é um método de análise de dados que automatiza a construção de modelos analíticos. É um ramo da inteligência artificial baseado na ideia de que sistemas podem aprender com dados, identificar padrões e tomar decisões com o mínimo de intervenção humana. Em 1959, Arthur Samuel definiu aprendizado de máquina como o "campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados"(SIMON, 2013).

Existem diversos algorítmos de Aprendizado de Máquina, tais como, Regressão Logística, Árvores de Decisão, *Gradient Boosting* (GB), *Suport Vector Machine* (SVM), entre outros. Alguns desses métodos são sofisticados, mas o mais importante a saber sobre eles não é o que os torna tão sofisticados, e sim decidir qual método se adapta melhor às nossas necessidades usando dados de teste.

Neste trabalho abordou-se o estudo e a compreensão do método de *Gradient Boosting*, pois é um método recente e que tem ganhado mais espaço entre os usuários de Aprendizado de Máquina.

1.1 Motivação

Este trabalho foi motivado devido a crescente utilização do Aprendizado de Máquina. Ele permite produzir, de maneira rápida e automatica, modelos capazes de analisar uma enorme quantidade de dados complexos, e entregar resultados mais rápidos e precisos – mesmo em grande escala.

Prever ou classificar determinado acontecimento é muito importante para diversos setores, pois aumentam as chances de identificar boas oportunidades e de evitar riscos. Um exemplo é no setor financeiro que necessita classificar os cliente que precisam de crédito como potenciais pagadores ou não, na hora de um empréstimo de crédito. Diante disso fica clara a necessidade do estudo do método de Gradient Boosting, que é um método de regressão e classificação.

Este método combina estimadores fracos para fazer a classificação. O método se baseia na criação de Árvores de Decisão para realizar a classificação, e os erros cometidos por cada Árvore é utilizado para melhorar a previsão da Árvore seguinte.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é estudar o Método de *Gradient Boosting* e realizar uma aplicação deste método em um banco de dados. Além disso, comparar o Método de *Gradient Boosting* para classificação com o Método de Regressão Logística.

1.2.2 Objetivo Específico

Como objetivo específico destaca-se a utilização do método de *Gradient Boosting* para classificar um certo indivíduo como bom ou mau pagador, a fim de permitir a decisão do empréstimo de crédito.

2 Materiais e Métodos

O aprendizado de máquina visa determinar um modelo para um conjunto de dados em questão. Há duas maneiras de se realizar isso, ou pelo aprendizado supervisionado ou pelo aprendizado não supervisionado.

O aprendizado não supervisiondado visa agrupar dados não rotulados ao encontrar determinado padrão. Um exemplo é a sugestão de uma nova música no aplicativo Spotify, que utiliza o método não supervisionado para sugerir novas músicas com base nas músicas já ouvidas pelo usuário do aplicativo.

O aprendizado supervisionado, diferente do aprendizado não supervisionado, trabalha com um conjunto de dados já rotulados. Seu objetivo é aprender com esses dados já rotulados, e com base neles, ser capaz de rotular dados do mesmo tipo que ainda não foram rotulados. Um exemplo é através do histórico escolar dos alunos estimar a nota que será tirada no Exame Nacional do Ensino Médio(ENEM).

Para a utilização do método de aprendizado de máquinas supervisionado é necessário dividir a mostra em dois tipos de amostras. A primeira é chamada de Amostra Treino, ela é utilizada para construir o modelo de aprendizado de máquina. A segunda amostra é chamada de Amostra Teste, ela é utilizada para estimar a taxa de erro do modelo. Quanto melhor o modelo classifica ou prediz a Amostra Teste melhor é o modelo para este caso.

O objetivo de utilizar Amostra Treino e Amostra Teste é, criar um modelo que se ajuste bem em qualquer base de dados. Se um modelo for testado com os dados de treino (os mesmos dados em que foi construído), entãp seus resultados não podem ser generalizados, pois não se sabe qual será o comportamento do modelo em dados nunca vistos. Segundo Afendras, G. (2018) é recomendado utilizar 70% dos dados como dados de treino e 30% como dados de teste.

Se os valores da variável de interesse, são valores discretos finitos ou ainda categóricos, então tem-se um problema de classificação e o algoritmo para resolver esse problema é chamado de Classificador. Se os valores da variável de interesse são valores contínuos ou

discretos infinitos, então tem-se um problema de regressão e o algoritmo será chamado de Regressor.

Ao trabalhar com aprendizado de máquina deve-se atentar para o sobreajuste do modelo. Um modelo apresenta sobreajuste (*Over tting*) quando ele se ajusta muito bem a um conjunto de dados anteriormente observado e, como consequência, se mostra ineficaz para prever novas amostras. Um exemplo claro de sobreajuste é representado na Figura (1), em que no primeiro momento são apresentados todos os dados, em seguida são apresentado os dados de treino (Pontos Azuis) e o ajuste do modelo que, como pode-se observar, passa por cima dos pontos indicando um ótimo ajuste, entretanto, no último gráfico, apresentam-se os dados de teste (Pontos Verdes) e percebe-se que o modelo não possui boas medições.

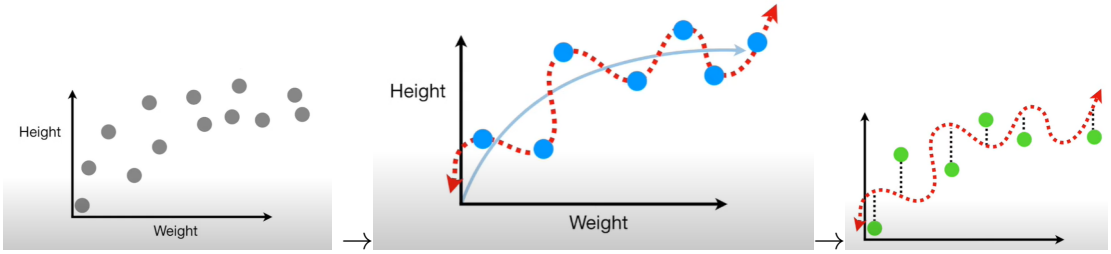


Figura 1: Exemplo de sobreajuste (Fonte:Adaptado de (STARMER, 2018))

Existem diversos métodos que podem ser utilizados no aprendizado de máquina, métodos baseados em Árvores, *Support Vector Machine*, Modelos Lineares Generalizados. Os métodos baseados em Árvore apresentam alta precisão e estabilidade, mapeiam bem relações não lineares e podem ser adaptados para resolver problemas de classificação ou regressão. O *Support Vector Machine* por sua vez é um classificador que divide os dados por um hiperplano de separação cujo objetivo é maximizar a margem do classificador linear. Os Modelos Lineares Generalizados modelam o comportamento médio de uma variável de interesse por meio da combinação de covariáveis explicativas.

Neste texto será trabalhado o método de *Gradient Boosting* que é um método baseado em árvore. Este método será comparado ao método de Regressão Logística que pertence a classe dos modelos lineares generalizados. Os conceitos explicados nas subseções abaixo foram retirados de (JAMES et al., 2014), (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

2.1 Avaliação dos Métodos

Na aprendizagem de máquina é muito importante ser capaz de avaliar o método utilizado, pois é necessário saber o quanto pode-se confiar nas predições e classificações realizadas por ele. Desta maneira, serão apresentados nesta seção algumas medidas de qualidade do método.

2.1.1 Erro dentro da amostra(*In Sample Error*)

É a taxa de erro presente no conjunto de dados usado para criar seu preditor. Na literatura às vezes é chamado de erro de resubstituição. Em outras palavras, é quando o algoritmo de previsão se ajusta um pouco ao que foi coletado em um conjunto de dados específico. E assim, quando utiliza-se um novo conjunto de dados, a precisão do algorítimo diminuirá um pouco.

2.1.2 Erro fora da amostra(*Out of Sample Error*)

É a taxa de erro presente em um novo conjunto de dados. Na literatura às vezes é chamado de erro de generalização. A ideia é que uma vez que se coleta uma amostra de dados e se constrói um modelo para ela, pode-se querer testá-lo em uma nova amostra, por exemplo uma amostra coletada por outro indivíduo ou em um horário diferente. Pode-se então analisar o quão bem o algoritmo executará a predição nesse novo conjunto de dados. Este erro é utilizado para obter uma estimativa para a taxa de erro do classificador/regressor

2.1.3 Validação Cruzada(*Cross Validation*)

A Validação Cruzada é uma ferramenta que permite comparar diferentes métodos de aprendizado de máquina ou parâmetros para o método escolhido e, desta forma, avaliar qual apresenta melhor performance.

A idéia central da validação cruzada é o particionamento do conjunto de dados em subconjuntos mutuamente exclusivos, e em seguida, utilizar alguns destes subconjuntos para a estimação dos parâmetros do modelo, sendo os subconjuntos restantes utilizados na validação do modelo. A Validação Cruzada funciona através da repetição dos seguintes passos:

1. Separar os dados em conjunto de treino e conjunto de teste;
2. Treinar um modelo no conjunto de treino;
3. Avaliar no conjunto de teste.

Outra vantagem da utilização da Validação Cruzada é que o método leva em consideração diversas divisões possíveis para a base de dados (em Dados de Treino e Dados de Teste) usando uma de cada vez e tirando a média do resultado no final. A repetição do primeiro passo da Validação Cruzada é considerada uma reamostragem.

2.1.4 Métodos de Reamostragem

2.1.4.1 Método K-Fold

Este método consiste em dividir os dados em K pedaços iguais. Em seguida um pedaço é utilizado como Dados de Teste e o restante como Dados de Treino. Este processo é realizado K vezes, de modo que a cada repetição um novo pedaço seja utilizado para o teste. Para avaliar o erro é retirada a média de todos os erros de todas as replicações das amostras teste.

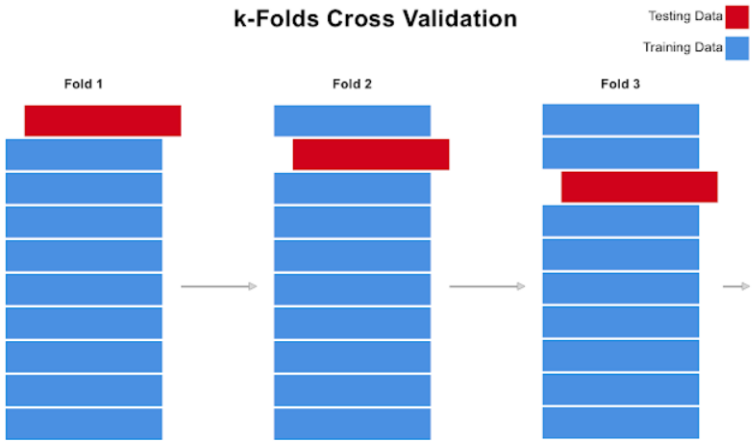


Figura 2: Representação do método K-Fold. (Fonte:(MACDONALD, 2017))

Quanto maior o K menor é o viés do modelo, porém maior é sua variância. Em outras palavras, será obtida uma estimativa muito precisa entre os valores previstos e os valores verdadeiros, porém altamente variável. Em contra partida quanto menor o K escolhido, maior é o viés e menor a variância. Ou seja, não necessariamente será obtida uma boa estimativa, mas ela será menos variável.

2.1.4.2 Bootstrap

Este método consiste em gerar novos dados de uma população através de uma amostragem repetida do conjunto de dados original com repetição. Desta forma, o novo banco de dados pode apresentar elementos repetidos(Figura 3). O propósito de utilizar esta técnica de reamostragem é que ela reduz a variância. Segundo Bradley Efron, criador do método de Bootstrap, é recomendado que sejam realizadas de 50 a 200 amostras por bootstrap para gerar estimativas confiáveis (EFRON; TIBSHIRANI, 1994).

Figura 3: Representação do método Bootstrap (Fonte:(RASCHKA, 2020))

2.2 Pré-Processamento

O objetivo de pré-processar os dados é melhorar/otimizar os resultados do preditor. Para isso, o pré-processamento realiza alterações nas variáveis ou remove as que não auxiliam na predição. Isto é, o pré-processamento é o processo de preparação, organização e estruturação dos dados.

As técnicas de pré-processamento que removem variáveis que não auxiliam na predição são Variância quase zero, Alta correlação. O objetivo da Variância quase zero é remover variáveis com um único valor, ou uma frequência muito alta de um único valor. O coeficiente de correlação é uma medida que resume a relação entre duas variáveis, remover as variáveis que apresentam uma alta correlação diminui a complexidade do modelo.

As técnicas de pré-processamento que realizam alterações nas variáveis são a padronização e a normalização. O objetivo da padronização é evitar que o algoritmo fique enviesado para as variáveis com maior ordem de grandeza, para isso efetua-se a transformação das variáveis de modo que apresentem média 0 e desvio padrão 1. O cálculo da

padronização é realizado para cada observação X_i e é a diferença entre o valor observado e a média amostral da variável dividido pelo desvio padrão amostral da mesma (Equação 2.1).

$$X_i^{(padr)} = \frac{X_i - \overline{X}}{\sqrt{\widehat{Var}(X)}} \tag{2.1}$$

A normalização dos dados visa alterar os valores das variáveis numéricas para uma escala comum sem distorcer as diferenças nos intervalos dos valores. Um método de normalização muito utilizado é o método de Box-Cox por ser mais simples e eficiente computacionalmente, entretanto só é aplicável a dados positivos. A transformação é dada pela Equação 2.2, onde o parâmetro λ, que é o parâmetro de transformação, é estimado pelo método de máxima verossimilhança

$$X_i^{(box)}(\lambda) = \frac{X_i^{\lambda} - 1}{\hat{\lambda}} \tag{2.2}$$

Vale ressaltar a transformação Yeo-Johnson que é semelhante a de Box-Cox, porém ela é aplicável em dados nulos e negativos.

O pré-processamento dos dados também lida com o problema de dados faltantes (NA's), em geral, o mais recomendável é descartar esses dados, porém é possível tentar estimar os valores falantes com base em indivíduos que possuam características parecidas, entretanto este método não será abordado.

2.3 Métodos de avaliação de modelos

Após o ajustar um modelo, é necessário verificar o quão bom é o modelo para predição, ou seja, é necessário calcular medidas que mostrem a precisão do modelo. Modelos de regressão e de classificação apresentam medidas de avaliação diferentes, e elas serão tratadas nas subseções a seguir.

2.3.1 Modelos de Regressão

Modelos de regressão são utilizados para estimar um valor para variável resposta do tipo quantitativa. O erro cometido pelo modelo é calculado através da distância do valor estimado ($\hat{y}_i$) para o valor real (y_i). As medidas de erro mais conhecidas são o Erro Quadrático Médio, ou MSE (*Mean Squared Error*), definido pela Equação 2.3; Erro Médio

Absoluto, ou MAE (*Mean Absolute Error*), definido pela Equação 2.4; e Raiz Quadrada do Erro Médio Absoluto, ou RMSE (*Root Mean Squared Error*), definido pela Equação 2.5. As métricas MSE, MAE e RMSE devem sempre apresentar valores não negativos. Quanto menor o valor dessas métricas, melhor o modelo será.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.3}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{2.4}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{2.5}$$

2.3.2 Modelos de Classificação

Modelos de classificação visam classificar os dados dentro de classes, desta forma, só existem duas possíveis conclusões para a classificação, o modelo classificou corretamente ou não. Uma maneira de resumir os resultados de um modelo de classificação é construir uma matriz de confusão (Figura 4). As linha representam a predição do método de aprendizagem de maquina, e as colunas representam o verdadeiro valor. A diagonal da matriz confusão representam os acertos de predição do modelo, e as demais posições são os erros.

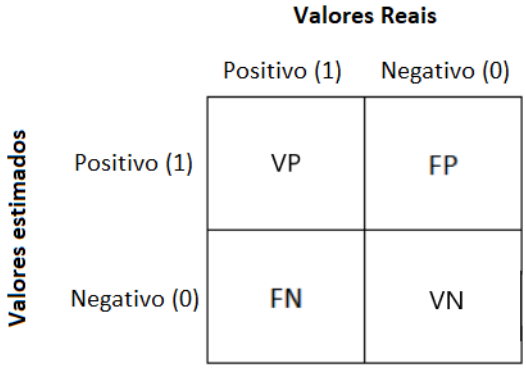


Figura 4: Representação de uma Matriz de Confusão
Fonte:Adaptado de (NARKHEDE, 2018)

Onde:

- Verdadeiros Positivos(VP) são os elementos que possuem a característica de interesse e foram corretamente identificados;
- Falsos Positivos(FP) são os elementos que possuem a característica de interesse e não foram corretamente identificados;
- Verdadeiros Negativos(VN) são os elementos que não possuem a característica de interesse e foram corretamente identificados;
- Falsos Negativos(FN) são os elementos que não possuem a característica de interesse e não foram corretamente identificados.

Parar verificar o desempenho da classificação de modelos de aprendizado de máquina são utilizadas algumas medidas como a Acurácia, a Sensibilidade e a Especificidade. Sensibilidade corresponde a proporção de acertos na classificação da característica principal calculado através da Equação 2.6.

$$\text{Sensibilidade} = \frac{VP}{VP + FN} \tag{2.6}$$

Especificidade corresponde a proporção de acertos na classificação da característica secundária calculado através da equação 2.7.

$$\text{Especificidade} = \frac{VN}{FP + VN} \tag{2.7}$$

E a Acurácia corresponde a proporção de acertos totais na classificação dos indivíduos, e é calculada da seguinte forma

$$\text{Acurácia} = \frac{VP + VN}{VP + FP + VN + FN} \tag{2.8}$$

2.4 Árvores de Decisão

Árvores de decisão são modelos de aprendizado supervisionado que podem ser aplicadas tanto em problemas de regressão quanto em problemas de classificação. Uma árvore de decisão, em geral, visa estratificar ou segmentar o espaço preditivo em várias regiões

simples. Através de uma pergunta ela utiliza as variáveis de cada indivíduo para criar uma regra de separação, que posteriormente será utilizada para rotular novas amostras. É importante ressaltar que no processo de construção de uma árvore de decisão a separação dos dados deve envolver apenas duas respostas: “Sim” ou “Não”.

As árvores de decisão seguem a seguinte estrutura: o topo da árvore é chamado de Raíz, variáveis subsequentes são chamadas de Nós, por último têm-se as Folhas. A Raíz tem apenas setas saindo dela, os Nós têm setas apontando para ele e saindo dele, as Folhas têm apenas setas apontadas para elas, como pode-se observar na Figura 5. Árvores que não apresentam Nós são chamadas de Tocos.

Quando tem-se um problema de regressão a árvore é chamada de Árvore de Regressão, em geral, em uma árvore de regressão as folhas representam valores numéricos. Quando tem-se um problema de classsificação a árvore é chamada de Árvore de Classificação e as folhas representam como o elemento é classificado.

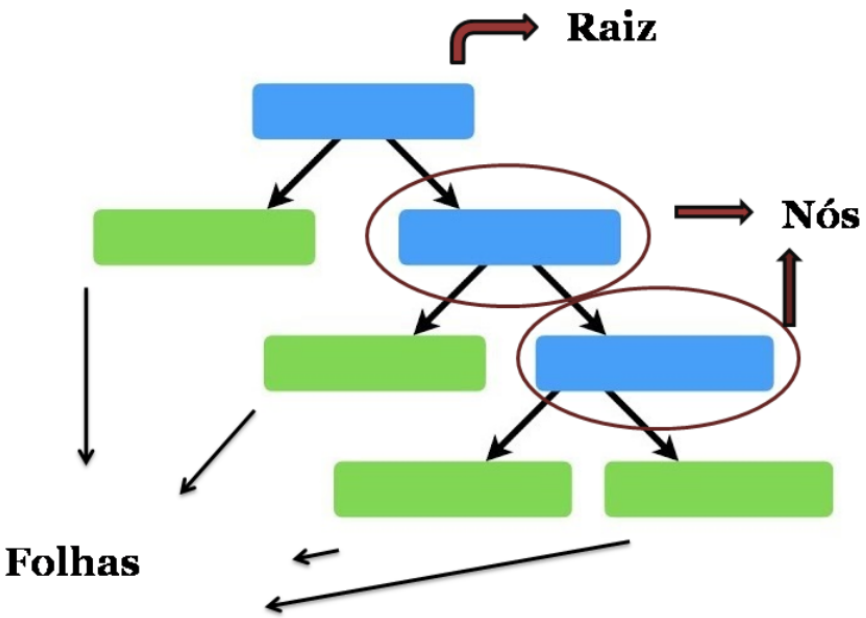


Figura 5: Representação de uma árvore de decisão.

Após o pré-processamento dos dados, é necessário decidir qual é a ordem das variáveis como, a variável que começa a Árvore ou as variáveis presentes nos Nós, para isso é necessário calcular o nível de impureza de todas as variáveis. Os erros de classificação e predição cometidos pelo modelo são impurezas, e esses erros são utilizados para calcular os índices de impureza. A variável que apresenta o menor índice de impureza é a que melhor separa os dados. O método para calcular o nível de impureza é diferente para

problemas de regressão e problemas de classificação.

Com o objetivo de auxiliar na compreesão da construção de uma árvore de decisão será utilizado um exemplo com poucas observações e variáveis. Pesquisadores visam estimar o peso da próxima geração de ratos de laboratório, para isso são coletados algumas informações da geração atual como, comprimento(centímetros), idade(semanas), peso(gramas), e o IMC apresentados na Tabela 1

Tabela 1: Informações dos ratos. (Fonte:Autor)

	Comprimento	Sexo	Ração	Idade	Peso	IMC
1	7	M	A	4	39	Obeso
2	8	M	B	6	43	Obeso
3	8	F	B	3	40	Normal
4	10	M	A	4	40	Normal
5	8	F	B	5	38	Normal
6	9	M	A	5	42	Normal

Em problemas de regressão a construção da árvore começa ao se definir o ponto de corte dos dados, isto é, o valor que irá segmentar o espaço. Ao se definir um ponto de corte os dados são separados em regiões, o processo de divisão de regiões continua até que a regra de parada seja aplicada, criando então K regiões. Para cada região $(R_1, \ldots, R_k)$ é estimada uma constante c_k, que corresponde a uma estimativa para y_i, apresentado na Equação (2.10).

$$F(x) = \sum_{k=1}^{K} c_k I(x \in R_k). \tag{2.9}$$

Com o objetivo de encontrar o melhor c_k utiliza-se o critério de encontrar o $F(x)$ que minimiza a soma dos quadrados dos resíduos, obtém-se que $\hat{c_k}$ é a média dos y_i presentes na região R_k. Isto é, para cada segmentação do espaço, o valor estimado do peso dos ratos é a média das observações presentes neste espaço.

$$\hat{c_k} = \sum_{x_i \in R_k} \frac{y_i}{n_k} \tag{2.10}$$

O ponto de corte é definido então pelo valor que minimiza a soma dos quadrados dos resíduos (SQR), o calculo é realizado para todas as variáveis e aquela que apresentar menor resíduo é escolhida como raíz da árvore. Observa-se na Equação (2.11), a equação que calcula a soma dos quadrados dos resíduos.

$$SQR = \sum_{i=1}^{N} (y_i - \hat{c_k})^2 \tag{2.11}$$

Em suma, os dados são divididos em dois grupos através do ponto de corte com o intuito de encontrar a separação que reduz a soma dos quadrados dos resíduos, como pode ser visto na Figura 6. Seguindo o exemplo, a variável comprimento seria dividida 3 vezes, comprimento menor que 8 centímetros, menor que 9 centímetros e menor que 10 centímetros. Para cada caso seria calculado a SQR e aquele que apresenta o menor valor é definido como ponto de corte dos dados. Este processo é aplicado para todas as variáveis.

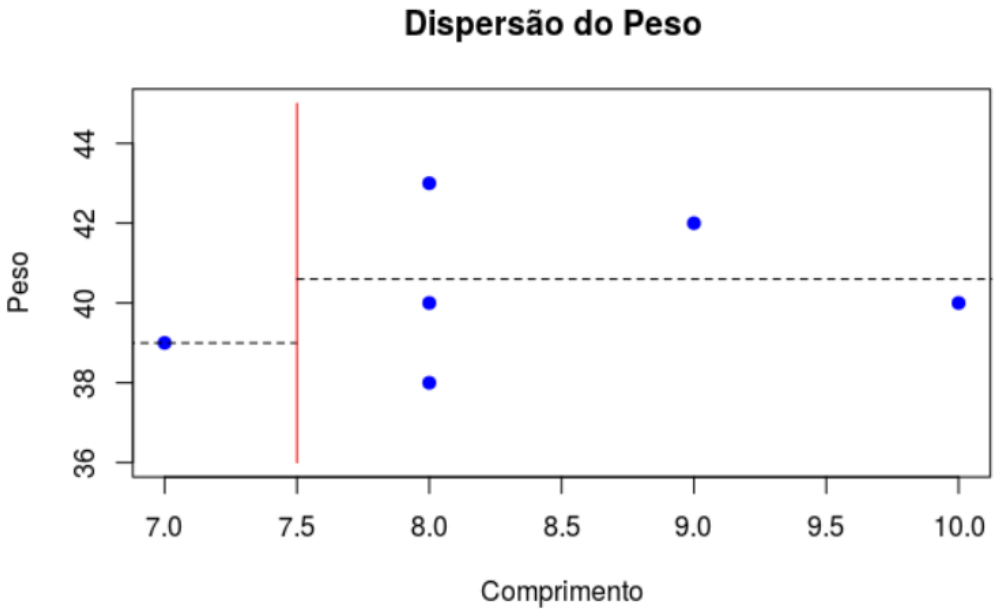


Figura 6: Segmentando os dados. (Fonte:Autor)

Em seguida repete-se o processo de encontrar um novo ponto de corte, entretanto, cada ponto de corte agora dividirá os grupos já criados. Isto é, dividirá os nós subsequentes. O processo é o mesmo da criação da raíz, e os critérios de parada são o número mínimo de observações na folha, ou o tamanho da árvore. Quando o número de elementos em cada folha é inferior a um valor pré-determinado ou,quando o limite pré-determinado de nós é alcançado o processo de criação da árvore para.

Após as segmentações serem realizadas e os indivíduos alocados em suas folhas correspondentes é necessário calcular o valor que a folha representa. O valor da folha é a média dos valores da variável de interesse presentes na folha (Figura 7). Para realizar a estimação de um novo indivíduo basta passar suas variáveis pela árvore e sua estimativa será a folha que ele for alocado.

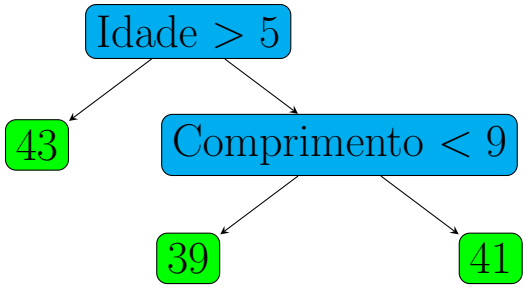


Figura 7: Árvore Final. (Fonte:Autor)

Em problemas de classificação a construção da árvore de decisão é semelhante a de regressão, entretanto os valores da variável de interesse não são numéricos com isso, não é possível utilizar o método de SQR. A escolha da variável que será a raíz da árvore é através do calculo do nível de impureza das variáveis, aquela com menor impureza é a que melhor separa os dados. A impureza dos dados pode ser calculada através do Índice de Gini, da Entropia Cruzada ou Erro de Classificação. O Índice de Gini é obtido da seguinte maneira:

$$\text{Índice de Gini} = 1 - {p_S}^2 - {p_N}^2 \tag{2.12}$$

onde p_S é a proporção de "Sim"na resposta e p_N é a proporção de "Não". O Índice de Gini consiste em um número entre 0 e 1/2, onde 0 representa uma separação perfeita dos dados, isto é, o modelo não errou em nenhuma classificação e 1/2 representa uma classificação ou predição que não apresentou nenhum acerto. A Entropia Cruzada e o Erro de Classificação não serão abordados.

2.5 *Gradient Boosting*

No aprendizado de máquina, *boosting* é um meta-algoritmo utilizado para reduzir o viés e a variação no aprendizado supervisionado. Os algoritmos de Boosting visam melhorar o poder de previsão utilizando uma sequência de modelos fracos, cada um compensando os pontos fracos de seus antecessores. O método de Boosting consiste em, iterativamente, utilizar estimadores fracos para realizar estimações e em seguida adicionar esta estimativa de forma ponderada a uma estimação gerada por um estimador forte.

O método de *Gradient Boosting* utiliza árvores de decisão para realizar suas estimações, já que, sozinhas, as árvores apresentam estimações fracas. As árvores criadas neste método têm entre 8 e 32 folhas, dependendo do tamanho da base de dados. O

método também necessita de uma função de perda diferenciável para avaliar o quão boa é a estimativa. Seu pseudo algoritmo é apresentado logo abaixo.

O *Gradient Boosting* é uma técnica de aprendizado de máquina supervisionada para problemas de regressão e classificação. Através da combinação de estimadores fracos o método visa produzir uma estimação mais precisa(forte). A resolução de problemas de regressão e classificação apresentam o mesmo pseudo algoritmo, porém as funções perda utilizadas em cada caso são diferentes portanto serão tratados separadamente.

No contexto de um algoritmo de otimização, a função usada para buscar o valor ótimo é referida como a função objetivo. Pode-se buscar maximizar ou minimizar a função objetivo, o que significa que procura-se uma solução candidata que tenha a pontuação mais alta ou mais baixa, respectivamente. Quando o foco é em minimizar esta função a chamamos de função de perda (GOODFELLOW et al., 2016).

Pseudo Algoritmo *Gradient Boosting*

1 - Inicializar $F_0(x) = argmin_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$

2 - Para m = 1 até M:

(a) Para $i = 1, 2, \ldots, N$ calcule

$$r_{im} = -\left[\frac{\delta L(y_i, F(x_i))}{\delta F(x_i)}\right]_{f=f_{m-1}}$$

(b) Ajustar uma árvore de regressão para os resíduos r_{im} determinando regiões terminais $R_{jm}, j = 1, 2, \ldots, J_M$

(c) Para $j = 1, 2, \ldots, J_M$ calcular

$$\gamma_{j,m} = argmin_\gamma \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma)$$

(d) Atualizar $F_m(x) = F_{m-1}(x) + \nu \sum \gamma_{jm} I(x \in R_{jm})$

3 - Retornar $\hat{F}(x) = F_M(x)$

2.5.1 *Gradient Boosting* para Regressão

2.5.1.1 Contexto Geral

Para facilitar a compreensão do método ele será explicado com um exemplo, os dados do exemplo estão presentes na Figura 1. O primeiro passo ao se utilizar o método de *Gradient Boosting* para regressão é definir uma função perda diferenciável para avaliar o quão boa é a estimativa. A função perda mais utilizada para uma única observação é a mostrada na Equação 2.13, onde, y_i é o valor observado e $F(x_i)$ é a previsão.

$$L(y_i, F(x_i)) = \frac{1}{2} \times (y_i – F(x_i))^2 \tag{2.13}$$

Definida a função perda, é necessário então calcular uma estimativa inicial, denotada como $F_0(x)$. Para se determinar a estimativa precisa-se encontrar o valor que minimiza a função perda, $F_0(x)$ calculada pela Equação (2.14), onde, y_i é referente ao i-ésimo valor observado e γ é o valor predito.

$$F_0(x) = argmin_\gamma \sum_{i=1}^{n} L(y_i, \gamma). \tag{2.14}$$

O valor obtido para a estimativa inicial é a média dos valores da variável de interesse, logo, no exemplo a estimativa inicial é a média dos pesos dos ratos que é 40.3, este valor, pertence a primeira árvore. O passo seguinte é construir a árvore de decisão subsequente baseada nos erros cometidos pela primeira árvore. Os erros cometidos pela árvore anterior é a diferença entre o valor observado da variável de interesse e o valor predito da mesma variável. Esta diferença, chamada de pseudo resíduo, é armazenada em uma nova coluna da base de dados visto na Tabela 2.

Tabela 2: Resíduo dos ratos. (Fonte:Autor)

	Comprimento	Sexo	Ração	Idade	Peso	IMC	Resíduos
1	7.00	M	A	4.00	39.00	Obeso	-1.333
2	8.00	M	B	6.00	43.00	Obeso	2.667
3	8.00	F	B	3.00	40.00	Normal	-0.333
4	10.00	M	A	4.00	40.00	Normal	-0.333
5	8.00	F	B	5.00	38.00	Normal	-2.333
6	9.00	M	A	5.00	42.00	Normal	-1.667

Calculados os erros, a árvore será construída utilizando as demais variáveis. Esta árvore não visa estimar valores para a variável de interesse, mas sim estimar os resíduos. Como o número de folhas é restrito, é comum ter menos folhas do que resíduos, então torna-se necessário calcular a média dos resíduos nas folhas que apresentarem mais de um resíduo. A árvore construida para o exemplo pode ser observada na Figura 8.

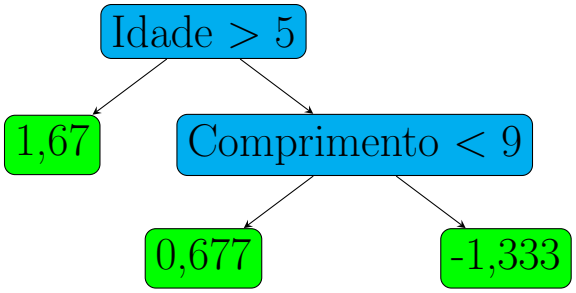


Figura 8: Árvore dos resíduos. (Fonte:Autor)

Criada a nova árvore, para realizar a nova estimação, será utilizada a primeira árvore (folha que apresenta a média) e os dados aplicados na segunda árvore. A nova estimação será a soma da primeira árvore com a folha da segunda árvore, gerando assim uma estimação (Figura 9). Entretanto esta estimação não é ideal pois apresenta um baixo viés mas, provavelmente uma alta variância. A maneira de contornar este problema é utilizando um coeficiente de aprendizado que irá escalar a contribuição da nova árvore. O coeficiente de aprendizado é um valor entre 0 e 1, quanto mais próximo de zero menor é o efeito de cada árvore na predição (FRIEDMAN, 2001). Logo o novo valor estimado será a primeria árvore somado ao valor da segunda árvore multiplicado por 0.1 que é o coeficiente de aprendizado.

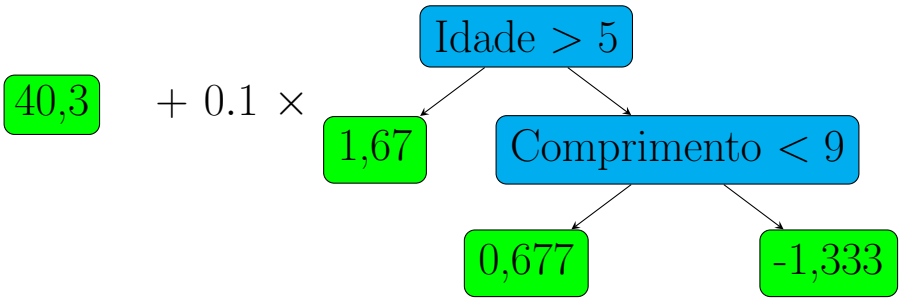


Figura 9: Construção da árvore dos resíduos. Fonte:Autor)

Para estimar o valor da variável basta passar o indivíduo pela árvore e somar a primeira estimativa, isto é, seguindo o exemplo dos ratos, para estimar o peso de um rato que apresenta uma idade de 7 semanas, e um comprimento de 9 centímetros seria obtido que o peso desse rato é 40,3 + 0,0677 que é igual a 40,3677. Entretanto, como uma única árvore não gera uma boa estimação é necessário realizar o processo de criação de árvores repetidas vezes. Cada nova árvore aprende com a árvore anterior, pois na criação da árvore seguinte é utilizado um novo valor para os resíduos, obtidos através da diferença entre valor observado da variável de interesse e o valor da nova estimativa.

A partir de agora os passos a cima serão repetidos, ou seja, serão calculados os novos pseudo resíduos com os novos valores estimados e os valores observados. Pode-se observar que os pseudo resíduos diminuem conforme aumenta o número de árvores criadas. Esta cadeia de árvores é gerada até atingir o máximo de árvores pré-definido ou a adição de uma nova árvore não reduz os resíduos de forma significante.

2.5.1.2 Algoritmo

Dada a explicação anterior, agora será apresentado o algoritmo e seus cálculos serão abertos. O input do algoritmo do Gradient Boosting é a função perda aprensentada na Equação 2.13 e os dados de treino. Estes dados apresentam-se da seguinte forma:

Amostra treino $(x_i, y_i)_{i=1}^{n}$, onde:

- n é referente ao número de observações;
- x_i é referente as variáveis utilizadas para predição da i-ésima linha;
- y_i é referente ao i-ésimo valor observado da variável a ser predita.

O primeiro passo do algoritmo é calcular uma constante para iniciar a iteração, esta constante é calculada através da Equação 2.14.

$$\begin{aligned} F_0(x) &= argmin_\gamma \sum_{i=1}^{n} L(y_i, \gamma) \\ &= argmin_\gamma \sum_{i=1}^{n} \frac{1}{2} \times (y_i - \gamma)^2 \\ &= argmin_\gamma \frac{1}{2}((y_1 - \gamma)^2 + (y_2 - \gamma)^2 + \ldots + (y_n - \gamma)^2) \end{aligned} \tag{2.15}$$

Para se encontra o γ que minimiza a soma das funções de perda, deriva-se $\sum L(y_i, \gamma)$ em relação a γ na Equação (2.15).

$$\frac{\partial}{\partial \gamma}(\frac{1}{2}((y_1 - \gamma)^2 + \ldots + (y_n - \gamma)^2)) = -((y_1 - \gamma) + (y_2 - \gamma) + \ldots + (y_n - \gamma)) \tag{2.16}$$

Em seguida iguala-se a derivada da Equação (2.16) a zero

$$
\begin{aligned}
-((y_1-\gamma)+(y_2-\gamma)+\ldots+(y_n-\gamma)) &= 0 \\
&\Rightarrow \quad n\times\gamma = y_1+y_2+\ldots y_n \\
&\Rightarrow \quad \gamma = \frac{\sum_i^n y_i}{n} \qquad (2.17)
\end{aligned}
$$

A constante γ é a média das observações da variável de interesse. O valor da constante será um toco, e representa uma estimativa inicial para a variável. Este toco é considerado a primeira árvore de decisão.

O passo seguinte é criar o loop no qual todas as árvores de decisão serão construídas, o loop criará M árvores. M refere-se ao total de árvores a serem criadas enquanto m refere-se m-ésima árvore criada. A primeira ação dentro do loop é calcular os resíduos gerados pela primeira estimativa. Calcula-se os resíduos através da Equação (2.18), onde, r_{im} é o i-ésimo resíduo da m-ésima árvore.

$$
r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}, i = 1,\ldots,n \qquad (2.18)
$$

Ao analisar a Equação(2.18), pode-se observar que é a derivada da função perda da Equação (2.13)

$$
\begin{aligned}
r_{im} &= -\left[\frac{\delta L(y_i, F(x_i))}{\delta F(x_i)}\right] \\
&= \frac{\partial}{\partial F(x_i)}\frac{1}{2}(y_i - F(x_i))^2 \\
&= -[-\frac{2}{2}(y_i - F(x_i))] \\
&= (y_i - F_{m-1}(x_i)) \qquad (2.19)
\end{aligned}
$$

O índice $F(x) = F_{m-1}(x)$ na Equação (2.18) refere-se ao Predito pela árvore criada no passo anterior (m-1), logo, quando m=1 têm-se que os resíduos são calculados através da diferença entre o valor observado (y_i) e o valor predito ($F_0(x)$) que é o valor da constante inicial, como pode-se observar na Equação (2.20)

$$
r_{i1} = (y_i - F_0(x)). \qquad (2.20)
$$

Para $m \geq 1$, o algoritmo calcula os n resíduos através da $(m-1)$-ésima árvore. Em

seguida é modelada uma árvore de regressão para os valores de r_{im} criando as regiões finais R_{jm} para $j = 1,...,J_m$. As regiões finais são as folhas das árvores e o índice j refere-se a j-ésima folha da m-ésima árvore. Como as árvores podem ser diferentes J_m refere-se ao total de folhas na m-ésima árvore.

Determinadas as regiões o algoritmo irá calcular as saídas das folhas e, como mais de um resíduo pode cair na mesma folha não fica claro qual é o valor da saída, tornando-se necessário então calcular $\gamma_{j,m}$

$$\begin{aligned}
\gamma_{j,m} &= argmin_\gamma \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma) \\
&= argmin_\gamma \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma) \\
&= argmin_\gamma \sum_{x_i \in R_{jm}} \frac{1}{2}(y_i - (F_{m-1}(x_i) + \gamma))^2 \qquad (2.21)
\end{aligned}$$

A saída de cada folha é o valor de γ que minimiza a Equação (2.21). Observa-se que esta minimização é a mesma realizada na Equação (2.14), a diferença é que agora leva-se em consideração a predição realizada anteriormente, enquanto antes não havia uma previsão prévia. A segunda diferença está no intervalo do somatório, a equação (2.14) considera todos os indivíduos da amostra, enquanto (2.21) considera apenas os indivíduos presentes nas regiões finais, isto é, somam-se os indivíduos que foram colocados na mesma folha.

$$\begin{aligned}
\gamma_{j,m} &\Rightarrow \frac{\delta}{\delta\gamma} \sum_{x_i \in R_{jm}} \frac{1}{2}(y_i - (F_{m-1}(x_i) + \gamma))^2 = 0 \\
&\Rightarrow \sum_{x_i \in R_{jm}} -\frac{2}{2}(y_i - (F_{m-1}(x_i) + \gamma)) = 0 \\
&\Rightarrow \sum_{x_i \in R_{jm}} -(y_i - F_{m-1}(x_i)) - \sum_{x_i \in R_{jm}} \gamma = 0 \\
&\Rightarrow \gamma \times (\mathrm{n}_{x_i \in R_{jm}}) = - \sum_{x_i \in R_{jm}} (y_i - F_{m-1}(x_i)) \qquad (2.22)
\end{aligned}$$

Ao se observar a Equação (2.22), que demonstra como são calculados os γ's, percebe-se que ao se calcular o mínimo através da derivada têm-se que o somatório de γ é equivalente a γ vezes o número de elementos na folha, logo, $\gamma_{j,m}$ pode ser calculado como a média dos resíduos presentes na mesma folha.

Calculados os valores para as folhas, o loop utiliza esta nova árvore para atualizar a predição de cada indivíduo. A atualização é realizada através da Equação 2.23.

$$F_m(x) = F_{m-1}(x) + \nu \sum \gamma_{jm} I(x \in R_{jm}) \tag{2.23}$$

A nova previsão $F_m(x)$ é baseada na previsão anterior $F_{m-1}(x)$ e na árvore recém criada. O somatório indica que os valores de saída (γ_{jm}'s) para todas as folhas $R_{j,m}$ sejam somados apenas para o caso que um indivíduo caia em mais de uma folha. E a letra grega ν (nu) é o coeficiente de aprendizado que vai escalonar a árvore. Ao fim do loop o algoritmo retorna $F_M(x)$ que é a função preditora após M iterações. Esta função é utilizada para fazer a predição de novas entradas.

2.5.2 *Gradient Boosting* para Classificação

2.5.2.1 Contexto Geral

Diferente do método de *Gradient Boosting* para Regressão, que trabalha com variáveis respostas quantitativas, o método de classificação trabalha com variáveis respostas qualitativas. As árvores ajustadas no modelo retornam como resposta valroes em função do log(chances), para permitir a classificação dos indivíduos, estes valores são transformados em probabilidade. O método assemelha-se a regressão logística que necessita definir um ponto de corte de probabilidade para classificar o indivíduo,o ponto de corte escolhido é aquele que maximiza as medidas de sensibilidade (Equação 2.6) e especificidade (Equação 2.7).

Para facilitar a compreensão do método será utilizado o mesmo exemplo do caso de regressão, porém será estimada a Saúde dos ratos (Tabela 1), e o ponto de corte de probabilidade será definido como 0.5, isto é, se a estimativa for maior que o ponto de corte considera-se o rato obeso. Assim como no método anterior é necessário definir uma função perda diferenciável para avaliar o quão boa são as estimativas. A função perda utilizada é dada pela Equação (2.24),

$$L(y_i, F(x)) = -y_i \times ln(\frac{p}{1-p}) + ln\left(1 + \exp^{ln(\frac{p}{1-p})}\right). \tag{2.24}$$

Onde y_i são os valores observados e, como é uma variável qualitativa, para a realização dos cálculos define-se y_i como:

$$y_i = \begin{cases} 1 & \text{, se o i-ésimo elemento apresenta a característica de interesse.} \\ 0 & \text{, caso contrário.} \end{cases}$$

A razão entre p e $1-p$, presente na Equação (2.24) é definida como chance de ocorrência do evento na Equação (2.25)

$$Chance = \frac{p}{1-p} \tag{2.25}$$

Respectivamente, p pode ser escrito como:

$$\begin{aligned} chance &= \frac{p}{1-p} \\ \Leftrightarrow p &= \frac{chance}{1+chance} \\ \Leftrightarrow p &= \frac{exp^{ln(chance)}}{1+\exp^{ln(chance)}} \end{aligned} \tag{2.26}$$

Para o exemplo, define-se os ratos com a Saúde normal como 0 e os ratos obesos como 1. Em seguida é necessário calcular a primeira estimativa. Esta estimativa é calculada pela mesma equação do método de regressão (Equação 2.14), onde, y_i é referente ao i-ésimo valor observado e γ é o ln(chance).

Para determinar a primeira estimativa precisa-se encontrar o valor que minimiza a função perda. O valor encontrado é o ln(chance) (demonstrado na Equação 2.33), desta maneira, seguindo o exemplo obtêm-se que a proporção de ratos obesos é 1/3, calculando o valor ln(chances) = ln(p/1-p) = ln(1/2). Com isso, é obtida a estimativa inicial que representa a primeira árvore toco apresentada na Figura 10.

$$log(1/2) = -0,69$$

Figura 10: Primeira árvore de *Grandient Boosting* de Classificação(Fonte: Autor)

Como o ponto de corte foi definido como 0.5 e a probabilidade encontrada foi 0.33, nesta primeira estimativa todos os ratos seriam classificados como normais. Torna-se necessário então construir uma nova árvore de decisão baseada nos erros de classificação cometidos pela folha. Os erros cometidos são obtidos pela diferença entre o valor observado

da variável de interesse (y_i) e a probabilidade estimada pela primeira árvore, onde a probabilidade é encontrada através da Equação (2.26). Os valores dos erros podem ser observados na coluna Resíduos da Tabela 3.

Tabela 3: Resíduo dos ratos. (Fonte:Autor)

	Comprimento	Sexo	Ração	Idade	Peso	IMC	Resíduos
1	7	M	A	4	39	Obeso	0.667
2	8	M	B	6	43	Obeso	0.667
3	8	F	B	3	40	Normal	-0.333
4	10	M	A	4	40	Normal	-0.333
5	8	F	B	5	38	Normal	-0.333
6	9	M	A	5	42	Normal	-0.333

Calculados os erros, a árvore será construída utilizando as variáveis que não serão estimadas. Esta árvore não visa estimar valores para a variável de interesse, mas sim estimar os resíduos. Para o exemplo, seriam utilizadas as variáveis Comprimento, Idade, Peso para estimar os resíduos e construir a árvore. A melhor árvore é apresentada na Figura 11.

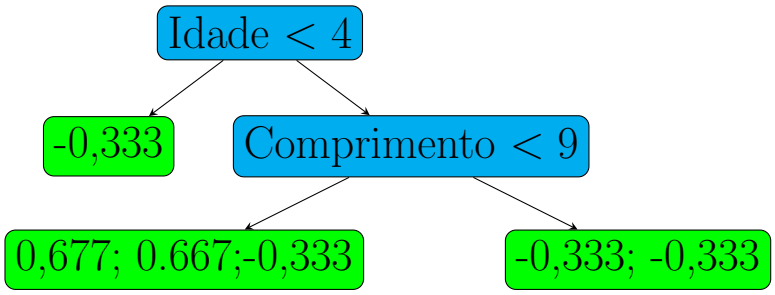


Figura 11: Árvore dos resíduos. (Fonte:Autor)

Criada a árvore, para realizar a nova estimação seria aplicado o mesmo processo do método de regressão onde há a soma da primeira árvore toco com a segunda árvore. Entretanto esta soma não pode ocorrer neste método pois o valor presente na primeira árvore toco está em função do ln(chance) enquanto os valores na segunda árvore estão em função da probabilidade. Nota-se que nas folhas que apresentam mais de um elemento não foi cálculada a média assim como na regressão, pois cada valor será utilizado para realizar a transformação de probabilidade em ln(chance).

Torna-se necessário então realizar uma transformação nos valores presentes nas folhas da árvore que viabilizem a soma. A transformação é dada pela Equação (2.27), onde o numerador é a soma dos resíduos da folha e denominador é a soma do produto entre a probabilidade do i-ésimo indivíduo e sua probabilidade complementar. Vale ressaltar

que na primeira iteração os valores de (Probabilidade Anterior)$_i$ × (1 − (Probabilidade Anterior)$_i$) são iguais, porém eles podem não ser mais iguais na iteração seguinte.

$$\frac{\sum Residuos}{\sum (ProbabilidadeAnterior)_i \times (1 - (ProbabilidadeAnterior)_i)} \tag{2.27}$$

Criada a nova árvore, com as devidas transformações, é realizado então o mesmo procedimento do método de regressão. Soma-se a primeira folha com a árvore escalonada por uma taxa de aprendizado obtendo-se assim a nova estimativa. Este passo apresenta-se na imagem 12, onde os valores abaixo das folhas são as transformações realizadas pela Equação 2.27.

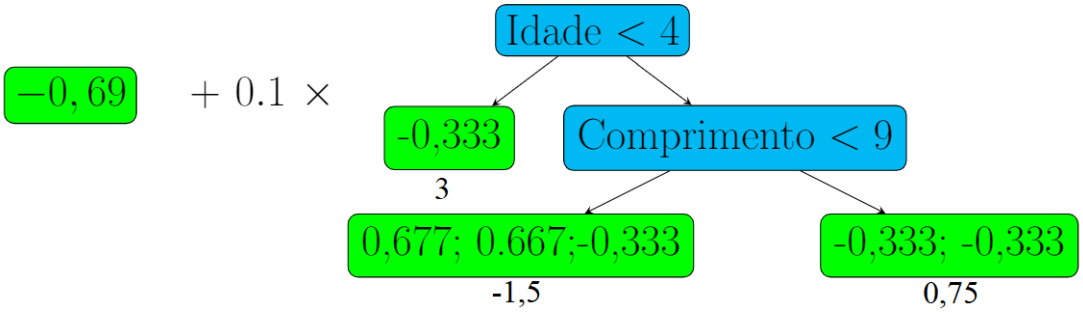


Figura 12: Segunda estimação pelo método de *Gradient Boosting* (Fonte:Autor)

A estimação gerada esta em ln(chance) então não apresenta fácil interpretação, para facilitar o entendimento do valor obtido deve-se transformá-lo em probabilidade através da Equação (2.26). Seguindo o exemplo dos ratos, se o rato da terceira linha fosse passado pela árvore obteria-se que o ln(chance) relacionada a ele é -0.69 + 0.1x3 que é igual a -0.39. Transformando a chance em probabilidade têm-se entao que a probabilidade referente a saúde deste rato é 0.4, sendo 0.5 o ponto de corte observa-se então que este rato é classificado como normal.

A partir de agora os passos a cima serão repetidos, ou seja, serão calculados os novos pseudo resíduos com base nas probabilidades estimadas e os valores observados. Com os novos resíduos, constrói-se uma nova árvore que fornece o ln(chance), que será utilizada para calcular novas probabilidades. É possível que as vezes as estimativas apresentam pioras, por esse motivo são realizadas várias iterações. Esta cadeia de árvores é gerada até atingir o máximo de árvores pré-definido ou até que adição de uma nova árvore não reduza os residuos de forma significante.

2.5.2.2 Algoritmo

Dada a explicação anterior, agora será apresentado o algoritmo e seus cálculos serão abertos. O input do algoritmo de Gradiente Boosting de classificação é a função perda apresentada na Equação (2.24) e os dados de treino. Estes dados apresentam a seguinte forma:

Dados de treino $(x_i, y_i)_{i=1}^{n}$, onde:

- x_i é referente as variáveis utilizadas para predição da i-ésima linha;
- y_i é referente ao i-ésimo valor observado da variável a ser predita.

A função perda (Equação 2.24) deriva-se da log-verossimilhança da Regressão Logística (Equação 2.49), onde p é a probabilidade estimada e y_i os valores observados da variável de interesse.

$$\ln(\text{verossimilhança}) = \sum_{i=1}^{N} y_i \times \ln(\text{p}) + (1 - y_i) \times \ln(\text{1-p}) \tag{2.28}$$

Quando se está fazendo Regressão Logística melhor é a predição quanto maior o valor da log-verossimilhança, isto é, o objetivo é maximizar a log-verossimilhança. Entretanto, em Gradient Boosting, o objetivo é utilizar a log-verossimilhança como função perda onde menores valores representam modelos melhores treinados, para resolver este problema multiplica-se a Equação (2.49) por -1. Têm-se então que:

$$\begin{aligned}
-\sum_{i=1}^{N} y_i \times \ln(\text{p}) + (1 - y_i) \times \ln(\text{1-p}) &= -\sum_{i=1}^{N} y_i \times \ln(\text{p}) + \ln(\text{1-p}) - y_i \times \ln(\text{1-p}) \\
&= \sum_{i=1}^{N} -y_i \times [\ln(\text{p}) - \ln(\text{1-p})] - \ln(\text{1-p}) \\
&= \sum_{i=1}^{N} -y_i \times [\ln(\text{chance})] - \ln\left(1 - \frac{exp^{ln(chance)}}{1 + \exp^{ln(chance)}}\right) \\
&= \sum_{i=1}^{N} -y_i \times \ln(\text{chance}) + \ln\left(1 + \exp^{ln(chance)}\right) \quad (2.29)
\end{aligned}$$

Pode-se observar, pela Equação (2.29), que partindo da log-verossimilhança (Equação 2.49), através de manipulações matemáticas, chega-se na função perda (Equação 2.24). É necessário agora verificar se a Equação (2.29) é diferenciável em relação a ln(chance).

$$\begin{aligned}\frac{\partial}{\partial \ln(\text{chance})}\sum_{i=1}^{N} -y_i \times \ln(\text{chance}) + \ln\left(1+\exp^{ln(chance)}\right) &= \sum_{i=1}^{N} -y_i + \frac{\exp^{ln(chance)}}{1+\exp^{ln(chance)}} \\ &= \sum_{i=1}^{N} -y_i + p \end{aligned} \quad (2.30)$$

Através da Equação (2.30) demonstra-se que a função perda (Equação 2.24) é diferenciável. Além disto a derivada da função perda pode ficar em função do ln(chance) ou da probabilidade p. Como será visto mais a frente algumas vezes é mais fácil utilizar a derivada da função perda em função do ln(chance) e em outras vezes em função da probabilidade p.

Inserido os inputs da função o primeiro passo, assim como realizado no *Gradient Boosting* para regressão, é calcular uma predição inicial para a iteração. Esta predição é calculada através da função na Equação (2.14).

$$\begin{aligned} F_0(x) &= argmin_\gamma \sum_{i=1}^{N} L(y_i, \gamma) \\ &= argmin_\gamma \sum_{i=1}^{N} -y_i \times \ln(\text{chance}) + \ln\left(1+\exp^{ln(chance)}\right) \end{aligned} \quad (2.31)$$

Em seguida deriva-se $F_0(x)$ em relação ao ln(chance) para encontrar o valor que minimiza $L(y_i, \gamma)$.

$$\begin{aligned} \frac{\delta}{\delta \ln(\text{chance})} \sum_{i=1}^{N} & -y_i \times \ln(\text{chance}) + \ln\left(1+\exp^{ln(chance)}\right) = 0 \\ \Leftrightarrow & \sum_{i=1}^{N} -y_i + \frac{\exp^{ln(chance)}}{1+\exp^{ln(chance)}} = 0 \\ \Leftrightarrow & \sum_{i=1}^{N} -y_i + p = 0 \\ \Leftrightarrow & \; p = \frac{\sum_{i=1}^{N} y_i}{N} \end{aligned} \quad (2.32)$$

Após substituir a função perda do método de classificação na Equação (2.14) e realizar a derivação para encontrar o valor que minimiza $L(y_i, \gamma)$ observa-se que a predição inicial no método de *Gradient Boosting* de classificação é a proporção de elementos de interesse, isto é, a proporção de y_i igual a 1. Convertendo p em função do ln(chance) (Equação

2.33) obtém-se a predição inicial para o ln(chance), que será a folha inicial.

$$F_0(x) = ln(chance) = ln\left(\frac{p}{1-p}\right) = ln\left(\frac{\sum_{i=1}^{N} y_i}{N - \sum_{i=1}^{N} y_i}\right) \tag{2.33}$$

O passo seguinte é criar o loop que construirá todas as árvores de decisão. Assim, como no método de regressão, serão construídas M árvores. A primeria ação dentro do loop é calcular os resíduos gerados pela primeira estimativa para cada indivíduo. Calcula-se os resíduos através da Equação (2.18), onde r_{im} é o i-ésimo resíduo da m-ésima árvore.

$$\begin{aligned} r_{im} &= -\left[\frac{\delta L(y_i, F(x_i))}{\delta F(x_i)}\right]_{F(x)=F_{m-1}(x)} \\ &= \frac{\delta}{\delta F(x_i)} - y_i \times ln(chance) + ln\left(1 + \exp^{ln(chance)}\right) \\ &= y_i - \frac{\exp^{ln(chance)}}{1 + \exp^{ln(chance)}} \\ &= (y_i - p) \end{aligned} \tag{2.34}$$

Observa-se na Equação (2.34) que os resíduos são a diferença entre o valor observado e o valor predito. O valor predito a ser utilizado para o cálculo do resíduo é indicado pelo índice $F(x) = F_{m-1}(x)$, logo, quando m=1 têm-se que os resíduos são calculados através da diferença entre o valor observado (y_i) e o valor predito ($F_0(x)$) que é o valor da predição inicial.

$$\begin{aligned} \gamma_{j,m} &= argmin_\gamma \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma) \\ &= argmin_\gamma \sum_{x_i \in R_{jm}} -y_i \times [F_{m-1}(x_i) + \gamma] + ln\left(1 + \exp^{F_{m-1}(x_i)+\gamma}\right) \end{aligned} \tag{2.35}$$

Calculados os resíduos o algoritmo agora constrói a árvore de decisão para os valores de r_{im} e cria as regiões finais R_{jm} para j = 1,...,J_m. As regiões finais são as folhas das árvores e o índice j refere-se a j-ésima folha da m-ésima árvore. Como as árvores podem ser diferentes J_m refere-se ao total de folhas na m-ésima árvore. Após criar as regiões finais calculam-se os valores de saída das folhas da árvore, para cada folha será calculado um γ_{jm}. O valor de saída para cada folha é o valor que minimiza a soma de $L(y_i, F_{m-1}(x_i)+\gamma)$.

Para obter o valor que minimiza γ na Equação (2.35) é necessário realizar a derivada com relação a γ para obter a solução, entretanto, derivar esta função é complicado e

trabalhoso então será utilizada uma abordagem diferente. Como resolver a derivada da função perda em relação a γ é complicado, pode-se aproximar a função perda com um polinômio de Taylor de segunda ordem (2.36) e derivá-lo em relação a γ.

$$L(y_i, F_{m-1}(x_i) + \gamma) \approx L(y_i, F_{m-1}(x_i)) + \frac{\delta}{\delta F()} L(y_i, F_{m-1}(x_i))\gamma + \frac{1}{2}\frac{\delta^2}{\delta F()^2} L(y_i, F_{m-1}(x_i))\gamma^2 \tag{2.36}$$

Reorganizando a Equação (2.36) temos

$$\frac{\delta}{\delta\gamma} L(y_i, F_{m-1}(x_i) + \gamma) \approx \frac{\delta}{\delta F()} L(y_i, F_{m-1}(x_i)) + \frac{\delta^2}{\delta F()^2} L(y_i, F_{m-1}(x_i))\gamma \tag{2.37}$$

Manipulando algébricamente a Equação (2.37) têm-se que γ pode ser escrito como a razão entre a primeira derivada da função perda e a segunda derivada (Equação 2.38)

$$\begin{aligned} \frac{\partial}{\partial\gamma} L(y_i, F_{m-1}(x_i) + \gamma) &\approx \frac{\partial}{\partial F()} L(y_i, F_{m-1}(x_i)) + \frac{\partial^2}{\partial F()^2} L(y_i, F_{m-1}(x_i))\gamma = 0 \\ \Leftrightarrow \frac{\partial^2}{\partial F()^2} L(y_i, F_{m-1}(x_i))\gamma &= -\frac{\partial}{\partial F()} L(y_i, F_{m-1}(x_i)) \\ \gamma &= \frac{\frac{\partial}{()} L(y_i, F_{m-1}(x_i))}{\frac{\delta^2}{\delta F()^2} L(y_i, F_{m-1}(x_i))} \end{aligned} \tag{2.38}$$

A primeira derivada da função perda apresenta-se na Equação 2.34, agora será encontrada a segunda derivada da função perda.

$$\begin{aligned} \left[\frac{\partial^2 L(y_i, F(x_i))}{\partial F(x_i)^2}\right] &= \frac{\partial^2}{\partial F(x_i)^2}\left(\sum_{x_i \in R_{jm}} -y_i \times ln(chance_i) + ln\left(1 + \exp^{ln(chance_i)}\right)\right) \\ &= \sum_{x_i \in R_{jm}} -\frac{\exp^{ln(chance_i)}}{\left(1 + \exp^{ln(chance_i)}\right)^2} \times \exp^{ln(chance)} + \frac{\exp^{ln(chance_i)}}{1 + \exp^{ln(chance_i)}} \end{aligned} \tag{2.39}$$

Multiplicando e dividindo a segunda parcela da Equação (2.40) por $(1 + exp^{ln(chance_i)})$ obtêm-se

$$
\begin{aligned}
&= \sum_{x_i \in R_{jm}} -\frac{\exp^{2\times ln(chance_i)}}{\left(1+\exp^{ln(chance_i)}\right)^2} + \frac{\exp^{ln(chance_i)} + \exp^{2\times ln(chance_i)}}{\left(1+\exp^{ln(chance_i)}\right)^2} \\
&= \sum_{x_i \in R_{jm}} \frac{\exp^{ln(chance)_i}}{\left(1+\exp^{ln(chance_i)}\right)} \frac{1}{\left(1+\exp^{ln(chance_i)}\right)} \\
&= \sum_{x_i \in R_{jm}} p_i(1-p)_i
\end{aligned} \tag{2.40}
$$

Na Equação (2.40), através de manipulações algébricas, pode-se observar que a segunda derivada da função perda é igual ao produto entre a probabilidade do i-ésimo indivíduo e sua probabilidade complementar $(p_i(1-p)_i)$, substituindo a primeira e a segunda derivada da Equação (2.38) têm-se que γ é:

$$
\gamma_{j,m} = \frac{\sum_{x_i \in R_{jm}} y_i - \frac{\exp^{ln(chance_i)}}{1+\exp^{ln(chance)}}}{\sum_{x_i \in R_{jm}} p_i(1-p)_i} \tag{2.41}
$$

Chega-se que γ é a soma dos resíduos presentes na j-ésima folha da m-ésima árvore, dividido pela soma do produto entre a probabilidade do i-ésimo indivíduo e sua probabilidade complementar. Com isso, foram calculados os valores de saída para cada folha da m-ésima árvore. Em seguida, é necessário atualizar os valores da predição. A atualização é feia através da Equação (2.23) igual no método de regressão.

A nova previsão $F_m(x)$ (Equação 2.23) é baseada na previsão anterior $F_{m-1}(x)$ e na árvore recém criada. O somatório indica que os valores de saída ($\gamma_{j,m}$'s) para todas as folhas $R_{j,m}$ sejam somados caso um indivíduo caia em mais de uma folha. E a letra grega ν (nu) é a taxa de aprendizado que vai escalonar a árvore. Ao fim do loop o algoritmo retorna $F_M(x)$ que é a função preditora após M iterações. Esta função é utilizada para a classificação de novos dados.

2.6 Regressão Logística

A Regressão Logística é um método de predição para variáveis qualitativas, o que a torna comparável ao método de Gradient Boosting para classificação. Além disso, fornece resultados em termo de probabilidades assim como o método de Gradient Boosting para classificação. Esta seção explicará o método de Regressão Logística para possibilitar a comparação desses métodos.

A Regressão Logística é um caso particular dos Modelos Lineares Generalizados(MLG).

Os MLG's tem como finalidade permitir a modelagem, não apenas utilizando os modelos lineares clássicos, os quais assumem que a variável dependente (Y_i) siga uma distribuiçao Normal. Assim estes modelos admitem que Y_i possa seguir outras distribuições, as quais pertençam à família exponencial.

O objetivo dos modelos lineares generalizados é descrever a relação entre a média da variável resposta Y_i e as variáveis explicativas X_i através de uma combinação linear com os parâmetros representados por $\beta_0, \beta_1, \ldots, \beta_{k-1}$ mais um erro aleatório ε_i (Equação 2.42).

$$g(E[y_i]) = \sum_{i=0}^{N} x_{i,j}\beta_i + \varepsilon_i \; ; \;\; j = 1, 2, \ldots, n \tag{2.42}$$

O modelo pode ser escrito em forma matricial tornando a expressão mais simples (Equação 2.43).

$$g(E(y_i)) = X^t\beta + \varepsilon \tag{2.43}$$

Onde y e ε são vetores $n \times 1$, X^t uma matriz $n \times k$ e β um vetor $k \times 1$. Como a variável resposta (y_i) é dicotômica, sabe-se que

$$Y_i \sim Bernoulli\,(p_i) \tag{2.44}$$

Desta forma, a componente linear é definida como

$$E[y_i] = p_i \tag{2.45}$$

E a função de ligação utilizada é a função de ligação canônica de uma Bernoulli.

$$g(p_i) = log(\frac{p_i}{1 - p_i}) \tag{2.46}$$

Desta forma, têm-se que:

$$ln\left(\frac{p_i}{1 - p_i}\right) = X_i^t\beta, \qquad i = 1, 2, \ldots, n \tag{2.47}$$

Em que $X_i^t = (1, X_1, X_2, \ldots, X_{k-1})$ representa o vetor de variáveis explicativas referentes ao i-ésimo indivíduo , p_i é a probabilidade do evento de interesse ocorrer para o i-ésimo indivíduo e $\beta = (\beta_0, \beta_1, \ldots, \beta_{k-1})$ é o vetor dos parâmetros desconhecidos do

modelo. O termo $p_i/(1-p)$ é chamado de logit, como foi explicado na Equação (2.25) ele representa a chance de ocorrência do evento.

Vale ressaltar que a função logística se apresenta como uma curva em forma de "S", e como ela estima através de probabilidade seus resultados ficam contidos no intervalo de zero a um, como pode ser observado na Figura 13.

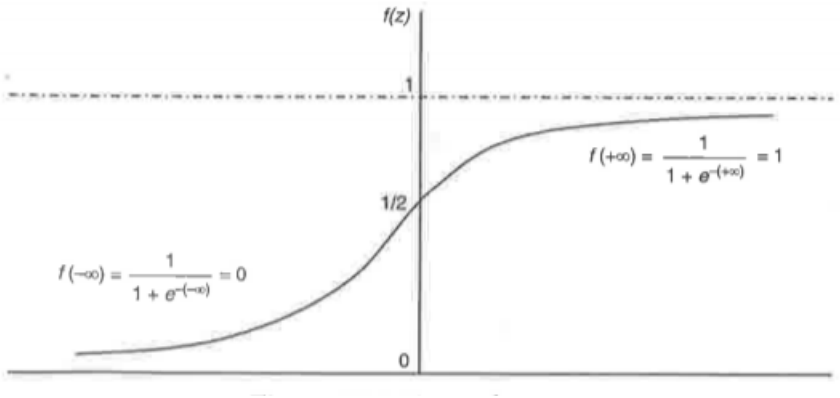


Figura 13: Função Logística(Fonte:(CHAN et al., 2009))

2.6.1 Método da Máxima Verossimilhança

O método de Máxima Verossimilhança (MV) é utilizado para estimar os parâmetros do modelo de regressão logística. O método consiste em estimar os parâmetros de um modelo utilizando as estimativas que maximizam o valor da função de verossimilhança.

Como a variável resposta apresenta distribuição Bernoulli ($Y_i \sim \mathit{Bernoulli}(p_i)$), sua função de probabilidade é dada então por:

$$P(Y_i = y_i|p_i) \;=\; p_i^{y_i}\;(1-p_i)^{1-y_i},\;\; i = 1, 2, \ldots, n$$

Como $y_1, y_2, \ldots, y_n$ são independentes, a função de verossimilhança é:

$$L(\beta; y) = \prod_{i=1}^{N} p_i^{y_i}(1-p_i)^{1-y_i} \tag{2.48}$$

Para facilitar os cálculos aplica-se a função logaritmo na Equação 2.48, têm-se então que:

$$\begin{aligned} lnL(\beta; y) &= ln \prod_{i=1}^{N} p_i^{y_i} (1-p_i)^{1-y_i} \\ &= \sum_{i=1}^{N} [y_i ln(p_i) + (1-y_i) ln(p_i)] \\ &= \sum_{i=1}^{N} \left[y_i ln\left(\frac{p_i}{1-p_i}\right) + ln(1-p_i) \right] \end{aligned} \tag{2.49}$$

Substituindo a Equação 2.47 na Equação 2.49 obtém-se que:

$$lnL(\beta; y) = \sum_{i=1}^{N} \left[y_i X_i^t \beta - ln(1 - \exp^{X_i^t \beta}) \right] \tag{2.50}$$

O estimador de máxima verossimilhança para β é encontrado ao se derivar a Equação (2.50) em relação a cada β_k e igualar a zero. Entretanto, as equações geradas são não-lineares nos parâmetros e não apresentam solução analítica. Neste caso, para cada amostra será utilizado um método iterativo para encontrar uma aproximação para a estimativa de máxima verossimilhança de β.

A estimativa pontual para a média da variável resposta é dada por:

$$\hat{p_i} = \frac{\exp^{X_i^t \hat{\beta}}}{1 + \exp^{X_i^t \hat{\beta}}}; \;\; i = 1, 2, \ldots, n \tag{2.51}$$

Ajustado o modelo de regressão logística, pode-se estimar a probabilidade da ocorrência do evento para cada observação. Os indivíduos são classificados como 1 se a probabilidade do evento de interesse ocorrer for maior que o ponto de corte p, e classificados como 0 caso o contrário. Assim como em *Gradient Boosting de Classi cacao*, o ponto de corte escolhido é aquele que maximiza as medidas de sensibilidade (Equação 2.6) e especificidade (Equação 2.7). Desta forma, a classificação é representada por uma matriz de confusão (Figura 4).

O ponto de corte p, que maximiza as medidas de sensibilidade e especificidade, é obtido através da curva ROC. A Curva ROC é um gráfico entre a sensibilidade (Verdadeiros positivos) e o complementar da especificidade (Falsos Positivos) calculado em diferentes pontos de corte. Esta curva permite visualizar melhor qual é o ponto de corte que maximiza as medidas. O ponto de corte escolhido é aquele que está mais próximo do canto superior esquerdo do gráfico, veja o exemplo de Curva ROC na Figura 14. Já o AUC (*Area Under Curve*) é a área calculada abaixo da curva ROC, ela mede o quão bem o

modelo separa as classes da variável resposta, seu valor está entre zero e um, e quanto maior melhor.

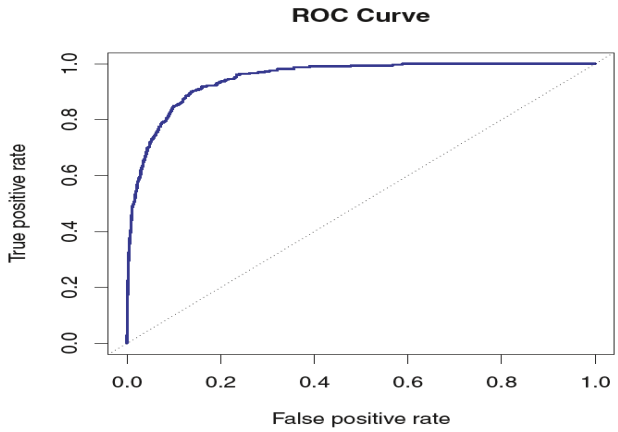


Figura 14: Curva ROC. Fonte:(JAMES et al., 2014)

2.7 Base de dados

Os métodos de *gradiente boosting* e regressão logistica serão utilizados para realizar a predição, logo é necessário uma base de dados para ajustar os modelos. Para a realização deste projeto foram obtidas, no site Kaggle ((INC, 2019)), duas bases de dados relacionadas por uma variável identificadora. A primeira base de dados, que será chamada de **Base 1**, apresenta 438.557 observações e 18 variáveis com informações sócio-econômicas dos indivíduos, como gênero, se tem carro, se têm propriedades, todas as variáveis podem ser observadas na Tabela 17 presente no Apêndice 1. A segunda base de dados, que será chamada de **Base 2**, representava a situação do indivíduo quanto ao pagamento do crédito mensalmente. Esta base apresenta 1.048.575 observações e 3 variáveis, uma identificadora, outra referente ao mês que o dado foi extraído, e a terceira sobre a situação do crédito, a descrição destas variáveis está presente na Tabela 18, presente no Apêndice 1.

A fim de utilizar o método de regressão logística é necessário que a variável resposta (Y) seja binária. Como o objetivo dos modelos é prever se um indivíduo é um bom pagador ou não, através da variável STATUS, da Base 2, foi criada uma nova variável chamada Bom_pagador. A variável STATUS representa a situação do crédito, e é composta de 7 categorias, as categorias de **1** a **5** representam as categorias referentes ao tempo de atraso no pagamento de crédito, a categoria **C** representa que o crédito foi pago devidamente.

Para criar a variável Bom pagador foi calculada a frequência do indivíduo em cada

uma das categorias da variável STATUS, um indivíduo é classificado como **bom pagador** quando sua frequência na categoria C da variável STATUS é maior que as demais categorias. O indivíduo é dito **mau pagador** quando a categoria C da variável STATUS não é aquela com maior freqûencia. Desta forma, definiu-se a variável da seguinte forma:

$$\text{Bom pagador} = \begin{cases} 1 & \text{, se o indivíduo é bom pagador} \\ 0 & \text{, caso contrário.} \end{cases} \tag{2.52}$$

Para exemplificar a classificação, pode-se observar na Tabela 4 dois exemplos de indivíduos classificados. O indivíduos com ID 5001718 apresentou maior frequência no STATUS 0, indicando que a maior parte do tempo está com pagamento atrasado, desta forma ele foi classificado como mau pagador. Já o indivíduo com ID 5001739 foi classificado como bom pagador, pois ele apresenta maior frequência na categoria C.

Tabela 4: Exemplo de classificação de indivíduos

ID	STATUS	Frequência	Bom pagador
5001718	0	24	Não
	1	2	
	C	3	
5001739	0	12	5001739
	C	46	

Em seguida, ligou-se a Base 1 a Base 2 através da variável identificadora como referência, gerando uma base de dados que será chamada de **Base Crédito**. A Base Crédito apresenta 33.110 observações, onde cada indivíduo foi classificado como bom ou mau pagador.

Em uma análise preliminar da Base Crédito, percebeu-se que haviam linhas exatamente iguais, isto é, a base apresentava linhas com as mesmas informações. Algumas dessas linhas apresentavam diferença apenas na classificação quanto a variável resposta, gerando linhas com os mesmos valores nas variáveis explicativas porém gerando classificações diferentes. Considerou-se que estas repetições poderiam gerar algum problema na classificação dos modelos, desta forma, decidiu-se remover as linhas que apresentaram exatamente os mesmos valores nas variáveis explicativas. Com isso, a base de dados passou a ter 5.602 observações.

Em seguida, foram alteradas as variáveis DAYS_BIRTH e DAYS_EMPLOYED, que representam respectivamente o número de dias desde o nascimento e o número de dias empregado. A primeira foi transformada na variável Idade que representa a idade dos

indivíduos em anos e a segunda foi trasformada na variável Tempo_trabalho que representa o tempo empregado em anos dos indivíduos. A variável OCCUPATION_TYPE é a única que apresentou valores faltantes, para resolver este problema optou-se por remover a variável, uma vez que a remoção das linhas com dados faltantes reduziria ainda mais a base de dados. Antes de removê-la, foi calculado o coeficiente de contingência ajustado entre ela e a variável resposta, a fim de verificar se havia relação entre estas variáveis, e foi obtido um valor de 0.08, indicando baixa relação.

3 Análise dos Resultados

O objetivo deste trabalho é utilizar o método de *gradient boosting* de classificação, apresentado na Seção 2.5, para predizer se um indivíduo é um bom ou mau pagador, e comparar este modelo com o modelo de regressão logística apresentado na Seção 2.6. Para o ajuste destes modelos será utilizada a base de dados discutida na Seção 2.7.

Tabela 5: Tabela de contingência entre as variáveis qualitativas e a variável bom pagador

		Bom pagador			
		Sim	Não	Sim	Não
		Valores absolutos		Valores Relativos	
Total		1905	3697	0,340	0,660
Possui carro	Sim	700	1359	0,367	0,368
	Não	1205	2338	0,633	0,632
Tem propriedades	Sim	1269	2510	0,666	0,679
	Não	636	1187	0,334	0,321
Categoria da renda	Comerciante/ empresário	468	912	0,246	0,247
	Pensionista	304	569	0,160	0,154
	Servidor público	150	274	0,079	0,074
	Assalariado	983	1942	0,516	0,525
Educação	Superior Completo	496	905	0,260	0,245
	Superior Incompleto	72	138	0,038	0,037
	Médio Completo	1306	2613	0,686	0,707
	Médio Incompleto	31	41	0,016	0,011
Gênero	Masculino	700	1312	0,367	0,355
	Feminino	1205	2385	0,633	0,645
Telefone Celular	Sim	1905	3697	1,000	1,000
	Não	0	0	0,000	0,000

A fim de entender melhor os dados e como eles estão distribuídos foi realizada uma análise descritiva da base. A base apresenta 5.602 observações e 17 variáveis, sendo 12 variáveis qualitativas, em que uma delas a variável de interesse Bom pagador, e 5 variáveis quantitativas. As Tabelas 5 e 6 apresentam os valores absolutos e os valores relativos das variáveis qualitativas, divididos pela variável bom pagador. Pode-se observar que a base apresenta 1.905 indivíduos considerados bom pagadores e 3.697 classificados como mau pagadores, indicando que 66% da base é classificada como mau pagadores. Além disso, observa-se que há mais mulheres do que homens, mais pessoas com propriedades e carros do que sem, e maior frequência de indivíduos com ensino médio completo.

Tabela 6: Tabela de contingência entre algumas variáveis qualitativas e a variável bom pagador

		Bom pagador			
		Sim	Não	Sim	Não
		Valores absolutos		Valores Relativos	
Estado Civil	Solteiro	285	556	0,150	0,150
	Casamento Civil	194	327	0,102	0,088
	Casamento Religioso	1246	2435	0,654	0,659
	Divorciado	114	227	0,060	0,061
	Viúvo	66	152	0,035	0,041
Moradia	Casa/ apartamento próprio	1700	3293	0,892	0,891
	Apartamento escritório	19	27	0,010	0,007
	Apartamento alugado	24	68	0,013	0,018
	Apartamento Municipal	73	124	0,038	0,034
	Com os pais	84	171	0,044	0,046
	Compartilhado	5	14	0,003	0,004
Telefone de trabalho	Sim	414	817	0,217	0,221
	Não	1491	2880	0,783	0,779
Telefone Residencial	Sim	560	1061	0,294	0,287
	Não	1345	2636	0,706	0,713
Email	Sim	148	334	0,078	0,090
	Não	1757	3363	0,922	0,910

Quando observa-se os valores relativos, percebe-se que as proporções de cada variável comparadas com a variável bom pagador são muito próximas, um exemplo seria a pro-

porção de bons pagadores que moram em casa/ apartamento que é 0.892, enquanto que a proporção de maus pagadores moram em casa/ apartamento é 0.891. Este padrão foi observado em todas as variáveis qualitativas.

A fim de verificar a relação entre a variável resposta e as variáveis qualitativas foi calculado o coeficiente de contingência modificado (C), que verifica a relação entre variáveis qualitativas. A Tabela 7 apresenta os valores do coeficiente de contingência modificado, a variável Telefone Celular não aparece pois ela não apresenta variação. Pode-se observar que nenhuma variável apresentou forte relação com a variável resposta.

Tabela 7: Relações entre as variáveis qualitativas e a variável resposta.

Variáveis Qualitativas	C
Gênero	0.0170
Possui carro	0.0000
Possui propriedades	0.0177
Tipo de renda	0.0172
Escolaridade	0.0408
Estado civil	0.0372
Tipo de moradia	0.0427
Telefone de trabalho	0.0053
Telefone residencial	0.0097
Email	0.0293

A Tabela 8 apresenta algumas medidas para as variáveis quantitativas da base de dados. Pode-se observar que a menor renda anual é 27.000, e a maior é 1.350.000. Indivíduos com tempo de trabalho igual a zero são considerados indivíduos desempregados. Vale ressaltar que um indivíduo apresenta 19 filhos, entretanto este valor não refletiu na variável tamanho da família. Optou-se por manter esse indivíduo, uma vez que é apenas uma observação.

Tabela 8: Medidas descritivas para as variáveis quantitativas

	Minimo	Mediana	Média	Máximo	Desvio Padrão
Idade	21	42	42,91	68	11,46
Tempo de trabalho	0	3,7	5,63	43	6,23
Renda anual	27000	157500	180886	1350000	99839,01
Número de filhos	0	0	0,4365	19	0,8
Tamanho da família	1	2	2.165	4	0,86

A fim de verificar a relação entre a variável resposta e as variáveis quantitativas foi calculada a correlação bisserial (R), que é utilizada nos casos em que uma das variáveis

é dicotômica. A Tabela 9 apresenta os valores da correlação, pode-se observar que todas as variáveis apresentaram fraca correlação.

Tabela 9: Relações entre as variáveis quantitativas e a variável resposta.

Variável Quantitativa	R
Idade	0.0140
Tempo de trabalho	0,0030
Renda anual	-0.0050
Número de filhos	-0,0070
Tamanho da família	-0,0013

Através da Figura 15 que apresenta o gráfico de barras para o tamanho da família, pode-se observar como esta distribuída a variável referente ao tamanho da família. Percebe-se que mais da metade dos indivíduos compõem uma família de duas pessoas, e menos de 30% dos indivíduos compõem uma famíla com 3 ou mais indivíduos. É possível observar que todas os tamanhos de família apresentaram a mesma proporção entre bons e maus pagadores, isto é, cada categoria apresentava 1/3 de indivíduos bons pagadores e 2/3 de maus pagadores. Essa característica também pode ser observada nas Figuras 17 e 16

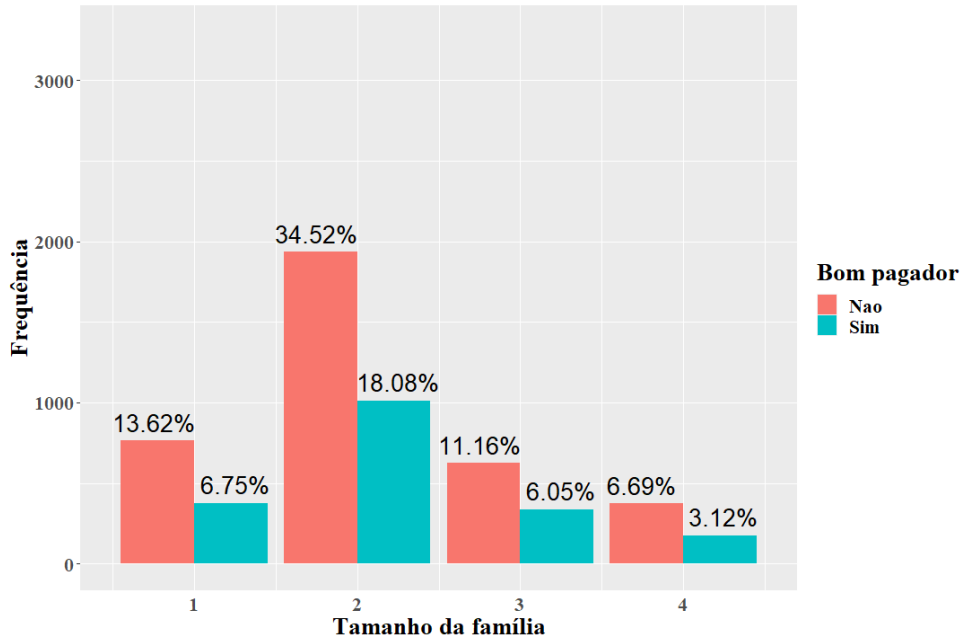


Figura 15: Gráfico de barras para o tamanho da família dos indivíduos

Observando a Figura 16 que apresenta o de barras para a variável Idade, que separa os indivíduos em intervalos de 10 anos, percebe-se que a maior parte dos indivíduos se encontra na faixa entre 31 e 40 anos, representando 28,74% da base, enquanto que a menor parte dos indivíduos se encontra na faixa entre 61 e 70 anos, representando 7.5% da base.

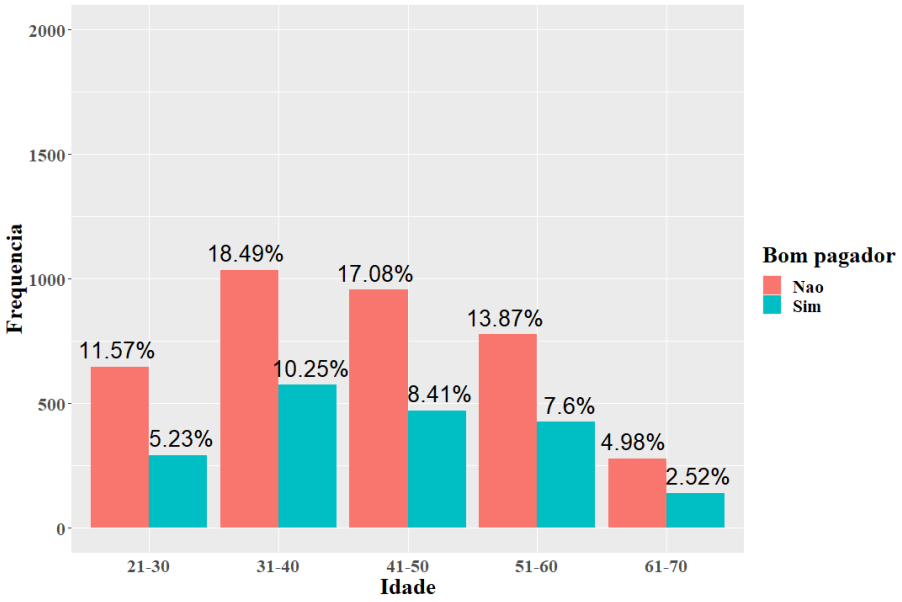


Figura 16: Gráfico de barras da Idade dos indivíduos

Através da Figura 17 que apresenta o gráfico de barras para a variável tempo de trabalho dividida em intervalos de 10 anos, observa-se que 66,9% dos indivíduos apresentam tempo de trabalho entre 1 e 10 anos. Vale ressaltar que a segunda categoria com maior frequência é a de indivíduos desempregados.

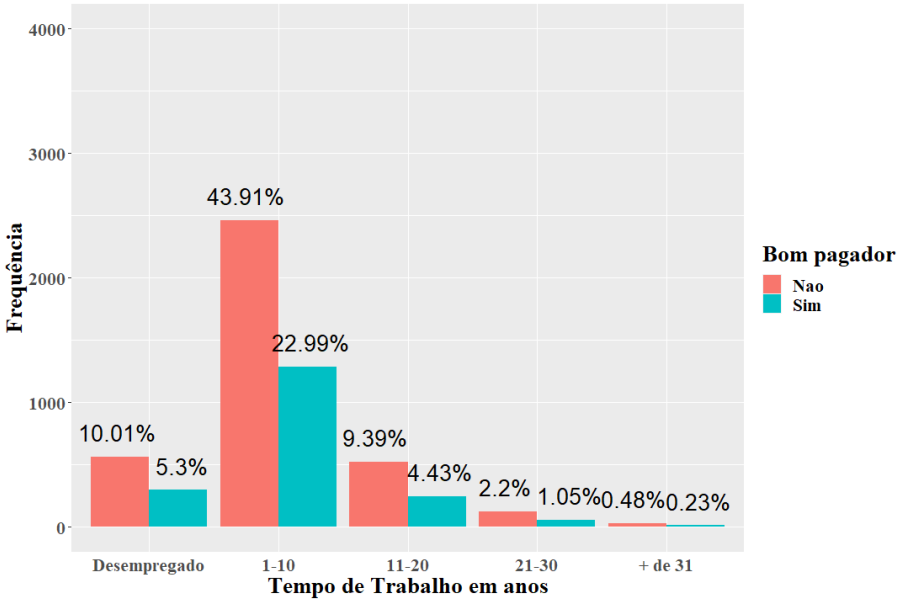


Figura 17: Gráfico de barras do Tempo de trabalho dos indivíduos

Para entender a distribuição da variável renda anual, foi construído um gráfico histograma, apresentado na Figura 18. Pode-se observar que a maior parte dos indivíduos estão

concentrados com renda anual abaixo de 250.000. Analisando os quantis desta variável obtém-se que apenas 3% dos indivíduos apresentam renda anual superior a 400.000. Apesar de apresentarem diferentes frequências, é possível notar uma semelhança na distribuição das rendas para ambas as categorias.

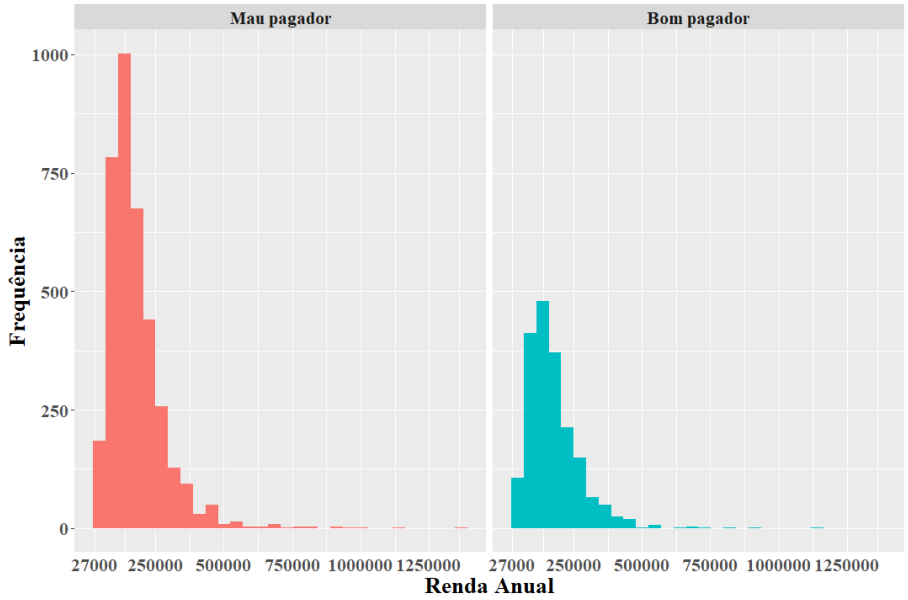


Figura 18: Histograma da renda anual dos indivíduos.

A Figura 19 apresenta o gráfico boxplot da renda anual, para melhorar a visualização do gráfico limitou-se o eixo y, que representa a renda anual, até o valor de 500.000. Pela Figura 19 observa-se que os boxplots são mutio semelhantes.

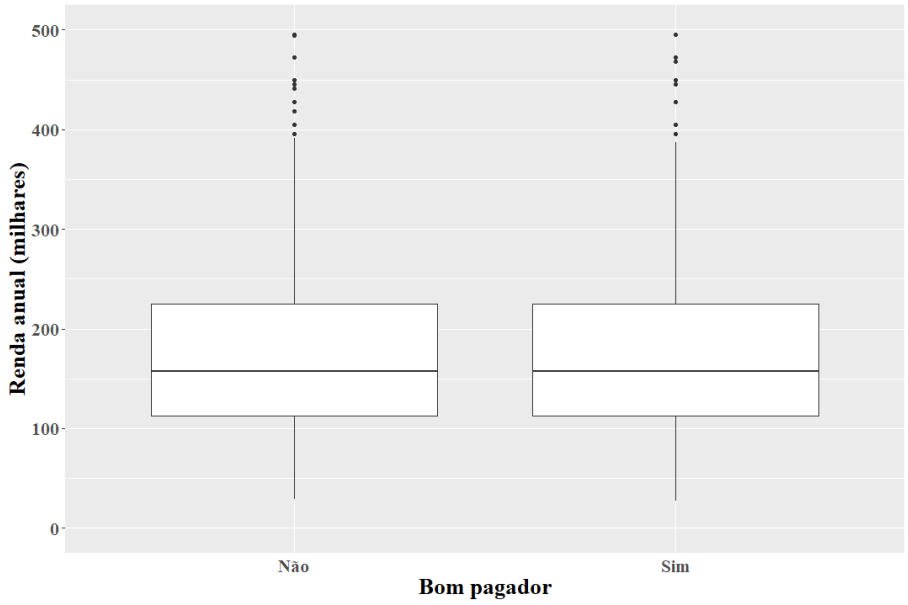


Figura 19: Boxplot da renda anual dos indivíduos separados pela variável Bom pagador.

Através da Tabela 10 pode-se observar a semelhança mostrada na Figura 19. A diferença entre a renda média dos grupos é muito baixa, ambos os grupos apresentam a mesma mediana. Observa-se também que a maior renda da base é de 1.350.000 de um indivíduo classificado como mau pagado, e a menor renda da base de 27.000 é de um indivíduo classificado como bom pagador.

Tabela 10: Medidas descritivas da renda divididas pela variável bom pagador

	Média	Mediana	Desvio Padrão	Máximo	Mínimo
Não	181777	157500	102498	1350000	29250
Sim	179156	157500	94468	1125000	27000

3.1 Modelagem e análise dos modelos

Nesta subseção serão explicadas as decisões tomadas para a modelagem dos dados, e também a análise dos modelos ajustados. A base inicialmente apresentava 18 variáveis, sendo uma a variável de interesse, para a modelagem pelo modelo de *Gradient Boosting* foram removidas duas variáveis. A variável FLAG_MOBIL que indica se o indivíduo possui celular não apresentava variação, e a variável OCCUPATION_TYPE apresentava um elevado valor de dados faltantes, não foram utilizadas no ajuste. Desta forma, a base utilizada para os ajustes dos modelos apresenta 16 variáveis e 5602 observações.

A base de dados foi dividida em treino e teste, nas seguintes proporções, 70% e 30% respectivamente. Para o ajuste dos modelos de *Gradient Boosting* foi utilizada a função `train` do pacote Caret do software R ((KUHN, 2020)). Ela permite a maipulação de vários hiperparâmetros do método, como o número de árvores, o tamanho da árvore (Ntree), coeficiente de aprendizado (C.A.), número mínimo de observação por nó (Min. Obs).

A fim de verificar como a função lidaria com a base de dados, primeiramente foi ajustado um modelo sem a alteração de nenhum hiperparâmetro. O método de reamostragem utilizado foi o *bootstrap* (A coluna Boots, presente nas tabelas a seguir, representa o número de *bootstraps* realizados). A função retorna alguns modelos, e a Tabela 11 apresenta as informações do modelo com maior AUC. Pode-se observar que o modelo apresentou um baixo valor de AUC, indicando que ele acerta em torno de 50%. Como os valores dos hiperparâmetros utilizados para a modelagem são pré-definidos pela função `train`, acredita-se que o modelo possa melhorar ao modificar estes valores.

Tabela 11: Medidas do Modelo de teste

Ntree	Tamanho	C.A.	Min Obs	Boots	AUC
50	3	0,1	10	50	0,5985

A fim de melhorar as medidas do modelo, foi feita uma análise de sensibilidade com valores para os hiperparâmetros encontrados na literatura. Empiricamente (FRIEDMAN, 2001), verificou que pequenos valores para o coeficiente de aprendizado (< 0.1) levam a melhores generalizações. Friedman afirma também que, para pequenos conjuntos de dados, um valor ótimo para o número de árvores é 500, enquanto que para conjuntos de dados maiores é necessário entorno de 5000 árvores. O tamanho da árvore indica quantos nós finais a árvore apresentará, os valores mais usuais para *Gradient Boosting* são entre 8 a 32 nós dependendo do tamanho da base de dados (STARMER, 2018). Desta forma, foram escolhidos os seguintes valores para o ajuste dos modelos:

- Número de árvores: 500, 1000, 2000, 10000;
- Tamanho da árvore: 8, 16, 32;
- Coeficiente de aprendizado: 0.1, 0.05.

Foram gerados 24 modelos de *Gradient Boosting*, e as medidas dos 4 melhores modelos é apresentada na Tabela 12. Utilizando o AUC como métrica para comparar os modelos, percebe-se que houve um aumento no AUC se comparado ao primeiro modelo construído.

Tabela 12: Análise de sensibilidade dos Modelo ajustado com a base treino

Ntree	Tamanho	C.A.	Min Obs	Boots	AUC
500	32	0,05	100	30	0,8094
1000	32	0,05	100	30	0,8790
2000	8	0,05	100	30	0,8149
10000	8	0,05	100	30	0,9553

Observou-se uma melhora nos modelos de treino ao alterar os hiperparâmetros. Em seguida, a fim de reduzir a complexidade do modelo, decidiu-se por reduzir o número de variáveis para a construção do modelo. A Figura 20 apresenta a influência relativa das variáveis para o ajuste do modelo. Baseando-se nos resultados obtidos pela função `train`, foram selecionadas as 8 variáveis mais relevantes para a construção dos modelos acima. Desta forma, foram selecionadas as variáveis tempo de trabalho, idade, renda anual,

escolaridade, possui carro, gênero, tipo de moradia e estado civil. Utilizando essas oito variáveis foram ajustados novamente 24 modelos seguindo os mesmos hiperparâmetros utilizados nos modelos completos.

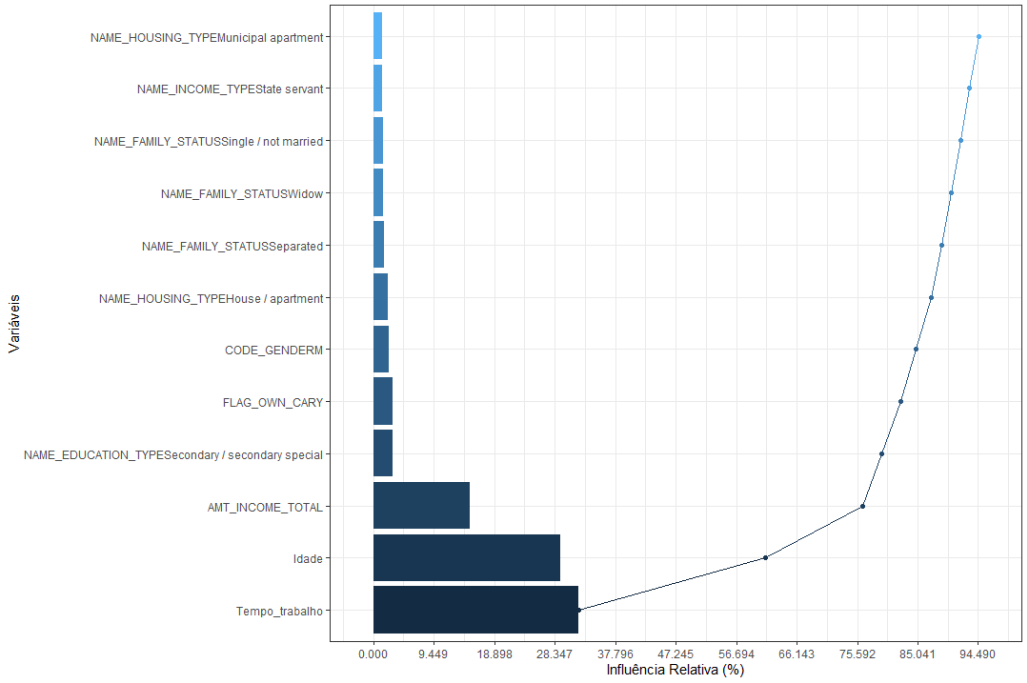


Figura 20: Influência relativa das variáveis no ajuste do modelo GBM.

A Tabela 13 apresenta os melhores modelos após a redução de variáveis explicativas. Pode-se observar que os valores de AUC foram bem próximos dos modelos completos. Vale ressaltar que os modelos construídos com 10000 árvores foram modelos que apresentaram *over tting*, isto é, eles apresentaram bom desempenho nos dados de treino, mas nos dados de teste apresentaram resultados semelhantes dos modelos com menor número de árvores, desta forma, eles não serão utilizados para as análises.

Tabela 13: Medidas dos Modelos GBM ajustados com a base treino

	Ntree	Tamanho	C.A.	Min Obs	Boots	AUC (treino)
GBM1	2000	8	0,05	100	30	0,7831
GBM2	2000	16	0,05	100	30	0,8648
GBM3	1000	32	0,05	100	30	0,8790
GBM4	10000	8	0,05	100	30	0,9191

Obtidos os modelos de *gradient boosting*, serão ajustados agora, os modelos de regressão logística, Para o ajuste dos modelos de regressão logística, foi utilizada a função `glm` (R Core Team, 2014). A função de ligação utilizada foi a função *logit*. Para escolher o número ótimo de variávei para a construção do modelo, foi utilizada a técnica de lasso

através da função `cv.glmnet` do pacote `glmnet` (FRIEDMAN; HASTIE; TIBSHIRANI, 2010). Utilizando a função, foi obtido que o valor ótimo de variáveis para o ajuste do modelo é zero, isto é, o melhor modelo segundo o método de lasso é o modelo nulo. A fim de ratificar o resultado do lasso, foi utilizado também o teste de comparação de Wald.

Através do teste de comparação de Wald foram ajustados 16 modelos, no qual cada modelo contém uma variável explicativa a menos que o modelo anterior, isto é, estes modelos foram ajustados removendo variável menos significativa. Utilizando deste método, obteve-se o mesmo resultado do lasso, indicando que o melhor modelo é o modelo nulo. Desta forma, acredita-se que as variáveis que compõem a base de dados não são boas para explicar a variável resposta.

A Tabela 14 indica quais foram os 4 melhores modelos, com base nas métricas dos modelos de treino, pelo método de *gradient boosting*. Como foi observado anteriormente que o AUC do modelo utilizando os parâmetros pré-definidos pela função `train` gerou melhor resultado melhor do que os demais modelos, foi ajsutado tambem um modelo com as 8 variáveis mais relevantes usando os hiperparâmetros pré-definidos. Observou-se que dos 4 melhores modelos, 3 deles foram ajustados com 8 variáveis explicativas e que dois deles são modelos que utilizam os hiperparâmetros pré-definidos. Cada modelo foi nomeado, indo de GBM1 até GBM4.

Tabela 14: Hiperparâmetros e AUC dos 4 melhores modelos GBM

	Variáveis explicativas	Ntree	Tamanho	C.A.	Min Obs	Boots	AUC (treino)	Ponto de corte
GBM1	8	2000	16	0,05	100	30	0,8648	0,3662
GBM2	15	2000	8	0,05	100	30	0,8149	0,3783
GBM3	15	500	32	0,05	100	30	0,8094	0,3410
GBM4	15	1000	32	0,05	100	30	0,879	0,3598

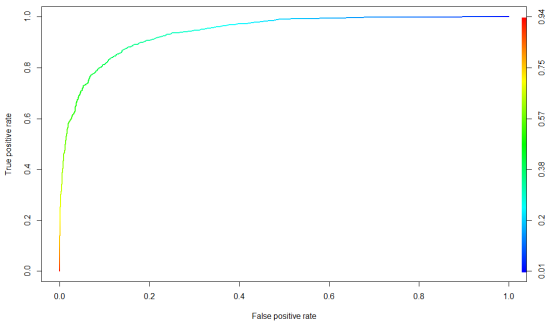
A Tabela 15 indica quais foram os 4 melhores modelos, com base nas métricas dos modelos de treino, pelo método de regressão logística. Foi possível observar que para o método de regressão logística, os modelos que apresentaram melhores resultados no AUC foram os 4 modelos com menores números de variáveis explicativas. Cada modelo foi nomeado, indo de GLM1 até GLM4.

Para definir o ponto de corte para classificar os indivíduos, foram ajustadas as curvas ROC para cada modelo, a Figura 21 mostra as curvas ROC dos modelos GBM, as curvas ROC dos modelos GLM são mostradas na Figura 22 no Apêndice 2. Os valores de corte para cada ajuste esta definido nas Tabelas 14 e 15. Definido o ponto de corte, é possível

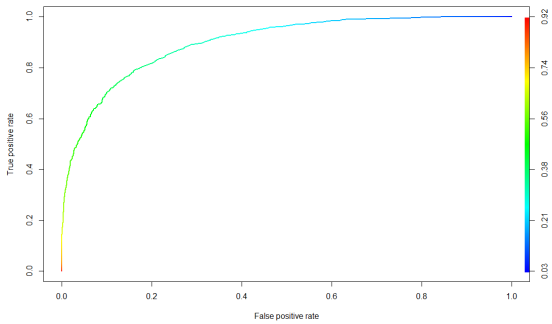
Tabela 15: AUC dos 4 melhores modelos GLM

	Variáveis explicativas	Boots	AUC (treino)	Ponto de corte
GLM1	15	50	0,5242	0,3612
GLM2	10	50	0,524	0,3352
GLM3	7	50	0,5237	0,3325
GLM4	0	50	0,5242	0,3612

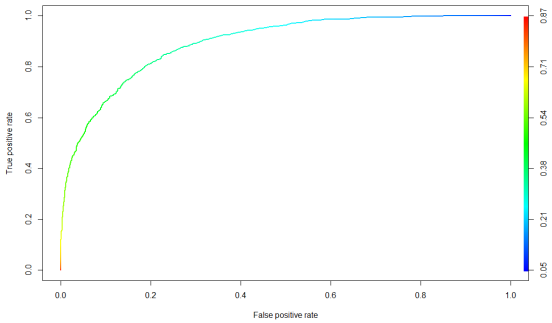
classifcar os indivíduos como bons e maus pagadores, se o indivíduo apresentou valor estimado menor que o ponto de corte ele é classificado como mau pagador, caso contrário é definido como bom pagador.



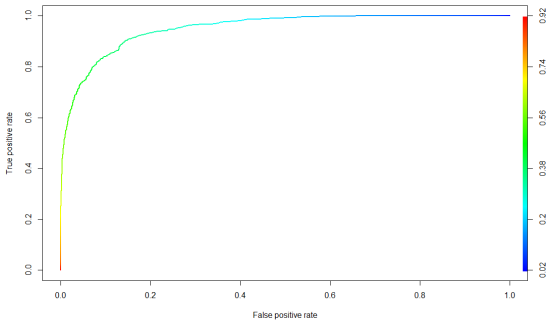
(a) GBM1



(b) GBM2



(c) GBM3



(d) GBM4

Figura 21: Curvas ROC dos 4 melhores modelos GBM ajustados.

Aplicando a base de teste nos 8 modelos e utilizando os pontos de corte, foram obtidas as seguintes métricas, Acurácia, Sensibilidade e Especificidade. Estas métricas podem ser observadas na Tabela 16.

Pode-se observar que os modelos apresentaram valores baixos para acurácia. Os modelos GBM apresentaram valores mais próximos entre a sensibilidade e a especificidade, enquanto que os modelos GLM apresentaram maiores discrepâncias nessas métricas, indicando que os modelos GBM acertaram de forma semelhante as categorias, enquanto

Tabela 16: Métrica dos melhores modelos ajustados na base de teste

	Acurácia	Sensibilidade	Especificidade
GBM1	0,5405	0,592	0,4396
GBM2	0,5685	0,652	0,4063
GBM3	0,5417	0,572	0,4834
GBM4	0,5542	0,614	0,4378
GLM1	0,6054	0,81	0,2084
GLM2	0,4685	0,394	0,613
GLM3	0,4506	0,328	0,6883
GLM4	0,6601	1	0

que os modelos GLM acertaram mais uma categoria do que a outra. O modelo que apresentou maior Acurácia foi o modelo GLM4, entretanto, ele classificou todos os indivíduos na mesma categoria, isto é, classificou todos como maus pagadores, visto que sua sensibilidade é 1 e sua especificidade é zero.

O modelo GLM1 foi o segundo modelo com a maior acurácia, este modelo é ajustado com todas as variáveis explicativas. Este modelo apresentou uma acurácia de 60.54%. O modelo que melhor classificou os indivíduos como bons pagadores foi o modelo GBLM3, que classificou corretamente os indivíduos como bons pagadores 68,83%, entretanto apresentou uma acurácia menor que os demais modelos. O melhor modelo GBM foi o GBM2 que apresentou acurácia de 56.85%.

Observado as métricas dos modelos na base de teste, pode-se concluir que o melhor modelo depende do que se busca prever. Se o objetivo é acertar mais na categorização de bons pagadores, pode ser bom utilizar o modelo GLM3. Se o objetivo é acertar mais na classificação de indivíduos mau pagadores, os modelos GLM1 e GBM2 são modelos que apresentaram boas métricas na classificação de indivíduos mau pagadores.

4 Conclusões

O objetivo deste trabalho era comparar os modelos de *Gradient Boosting* e Regressão Logística para a classificação de indivíduos quanto ao pagamento de crédito. Primeiramente, foi possível observar que os métodos de *gradient boosting* e regressão logística não apresentaram resultados satisfatórios na modelagem, uma vez que a acurácia dos modelos foi baixa. Acredita-se que este problema é decorrente da base de dados utilizada e não dos modelos utilizados.

Analisando o desempenho dos modelos, foi possível notar que o modelos GBM geraram melhores métricas quando utilizou-se de todas as variáveis explicativas, e com isso, geraram estimativas mais balanceadas entre a sensibilidade e especificidade. Já o modelo de regressão logística, apresentou melhor resultado de acurácia e de sensibilidade nos modelos completo e nulo, e apresentou melhores resultados na especificidade quando foi ajustada com um número de variáveis explicativas intermediário.

Desta forma, apesar dos modelos ajustados não serem modelos com uma alta acurácia, ainda é possível comparar os métodos. A partir dos resultados, observa-se que o modelo de regressão logística apresenta melhores resultados quando se busca acertar mais na classificação de uma das categorias da variável de interesse. Já o GBM se encaixa melhor nas situações no qual não é necessário acertar mais em uma categoria do que na outra.

Referências

AFENDRAS, G.; MARKATOU, M. Optimality of training/test size and resampling effectiveness in cross-validation. *Journal of Statistical Planning and Inference*, 2018.

CHAN, B. et al. *Analise de dados: modelagem multivariada para tomada de decisoes*. [S.l.]: Elsevier, 2009. ISBN 9788535230468.

EFRON, B.; TIBSHIRANI, R. *An Introduction to the Bootstrap*. [S.l.]: CRC Press, 1994. (Chapman & Hall/CRC Monographs on Statistics & Applied Probability). ISBN 9781000064988.

FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, v. 33, n. 1, p. 1–22, 2010.

FRIEDMAN, J. H. Greedy function approximation: A gradient boostingmachine. *The Annals of Statistics*, Institute of Mathematical Statistics, v. 29, n. 5, p. 1189 – 1232, 2001.

GOODFELLOW, I. et al. *Deep learning*. [S.l.]: MIT press Cambridge, 2016. v. 1.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. [S.l.]: Springer, 2009. (Springer series in statistics). ISBN 9780387848846.

INC, K. *Credit Card Approval Prediction*. 2019. (Acesso em 2020). Disponível em: ⟨https://www.kaggle.com/rikdifos/credit-card-approval-prediction⟩.

JAMES, G. et al. *An Introduction to Statistical Learning: With Applications in R*. [S.l.]: Springer Publishing Company, Incorporated, 2014. ISBN 1461471370.

KUHN, M. *caret: Classi cation and Regression Training*. [S.l.], 2020. R package version 6.0-86. Disponível em: ⟨https://CRAN.R-project.org/package=caret⟩.

MACDONALD, K. *k-Folds Cross Validation in Python*. 2017. (Acesso em 2020). Disponível em: ⟨http://www.kmdatascience.com/2017/07/k-folds-cross-validation-in-python.html⟩.

NARKHEDE, S. *Understanding Confusion Matrix*. 2018. (Acesso em 2020). Disponível em: ⟨https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62⟩.

R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2014. Disponível em: ⟨http://www.R-project.org/⟩.

RASCHKA, S. *Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning*. 2020. Disponível em: ⟨https://arxiv.org/pdf/1811.12808.pdf⟩.

SIMON, P. *Too Big to Ignore: The Business Case for Big Data*. [S.l.]: Wiley, 2013. (Wiley and SAS Business Series). ISBN 9781118642108.

STARMER, J. *Machine Learning*. [s.n.], 2018. (Acesso em 2020). Disponível em: ⟨https://www.youtube.com/watch?v=Gv9_4yMHFhI&list=PLblh5JKOoLUICTaGLRoHQDuF_7q2GfuJF⟩.

APÊNDICE 1 – Informações da base de dados

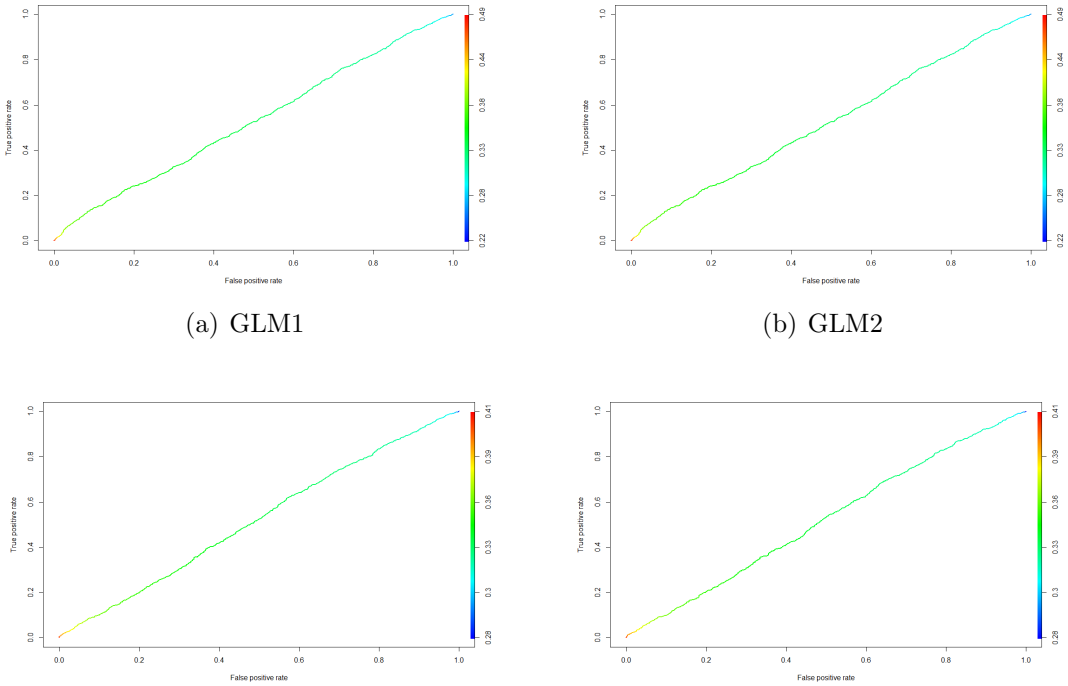
Tabela 17: Variáveis da base de dados sócio-econômica

Variável	Descrição
ID	Número do cliente
CODE_GENDER	Gênero
FLAG_OWN_CAR	Se tem carro
FLAG_OWN_REALTY	Se tem propriedade
CNT_CHILDREN	Número de filhos
AMT_INCOME_TOTAL	Renda anual
NAME_INCOME_TYPE	Categoria da renda
NAME_EDUCATION_TYPE	Nível de educação
NAME_FAMILY_STATUS	Estado civil
NAME_HOUSING_TYPE	Tipo de residencia
DAYS_BIRTH	Número de dias do dia corrente até o dia do nascimento, -1 significa ontem
DAYS_EMPLOYED	Número de dias do dia corrente até o dia que inicou o emprego,-1 ontem
FLAG_MOBIL	Se tem telefone celular
FLAG_WORK_PHONE	Se tem telefone de trabalho
FLAG_PHONE	Se tem telefone
FLAG_EMAIL	Se tem email
OCCUPATION_TYPE	Ocupação
CNT_FAM_MEMBERS	Tamanho da família

Tabela 18: Variáveis da base de dados quanto ao pagamento do crédito

Variáveis	Descrição
ID	Número do cliente
MONTHS_BALANCE	O mês dos dados extraídos é o ponto de partida, 0 é o mês atual, -1 é o mês anterior e assim por diante
STATUS	Status do crédito, sendo: 0 indica 1-29 dias em atraso; 1indica 30-59 dias em atraso; 2 indica 60-89 dias em atraso; 3 indica 90-119 dias em atraso; 4 indica 120-149 dias em atraso; 5 indica dívidas vencidas ou incobráveis por mais de 150 dias; C indica quitado naquele mês; X indica nenhum empréstimo para o mês.

APÊNDICE 2 – Gráfico



(a) GLM1

(b) GLM2

(c) GLM3

(d) GLM4

Figura 22: Curvas ROC dos 4 melhores modelos GLM ajustados.