UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO

Mayra Camila Romero

Aplicando técnicas de *Machine Learning* para avaliar resultados do ENEM

São Carlos

2021

Mayra Camila Romero

Aplicando técnicas de *Machine Learning* para avaliar resultados do ENEM

Trabalho de conclusão de curso apresentado ao Centro de Ciências Matemáticas Aplicadas á Indústria do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, como parte dos requisitos para conclusão do MBA em Ciências de Dados.

Área de concentração: Ciência de Dados

Orientador: Prof. Dr. Adriano Kamimura Suzuki

Versão original

São Carlos

2021

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

R763a

Romero, Mayra Camila
Aplicando técnicas de Machine Learning para avaliar resultados do ENEM / Mayra Camila Romero; orientador Adriano Kamimura Suzuki. -- São Carlos, 2021.
72 p.

Trabalho de conclusão de curso (MBA em Ciência de Dados) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2021.

1. ENEM. 2. Educação. 3. Machine Learning. 4. Classificação. I. Kamimura Suzuki, Adriano, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

AGRADECIMENTOS

Agradeço à Deus pelas oportunidades oferecidas em minha vida, minha família e amigos pelo apoio durante todo o processo.

Obrigada aos professores e tutores que estiveram disponíveis para nos ensinar.

Não posso deixar de agradecer ao meu Jimmy, que hoje não esta mais aqui, mas me fez companhia em muitos momentos durante o processo.

RESUMO

ROMERO, M. **Aplicando técnicas de *Machine Learning* para avaliar resultados do ENEM**. 2021. 78p. Dissertação - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2021.

O objetivo deste trabalho era desenvolver e comparar modelos de classificação para encontrar características socioeconômicas relevantes que indicassem candidatos que tem maior chance de atingir uma pontuação média maior que 500 pontos no ENEM. Utilizando a base disponibilizada pelo INEP, foram feitas análises de distribuição para escolher as informações que seriam utilizadas e os tratamentos que seriam feitos. Três técnicas de *Machine Learning* foram utilizadas e através da matriz de confusão calculou-se métricas desempenho, concluindo que o *Random Forest* foi o modelo que teve melhor desempenho. Atráves dos resultados dos modelos foi possível analisar as informações que impactavam a previsibilidade do modelo, entre elas: renda familiar e número de computadores. Este resultado comprovou o quanto a desigualdade no Brasil atinge a educação.

Palavras-chave: ENEM. Educação. *Machine Learning*. Classificação.

ABSTRACT

ROMERO, M. **Applying Machine Learning techniques to evaluate ENEM results**. 2021. 78p. Dissertação - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2021.

The purpose of this work was to develop and compare classification models in order to find relevant socioeconomic characteristics that indicate candidates with most chances to get an average score greater than 500 on ENEM. Distribution analyzes were made in the database provided by INEP to select the data along with the treatments in the raw data. Three Machine Learning techniques were used and performance metrics were computed through confusion matrix, concluding that the Random Forest has the best performance. Regarding the models output it was possible to analyze the information that impacted the most in predictability, among them: family income and numbers of computers. This result proved how much the Brazil inequality impacts the education.

Keywords: ENEM. Education. Machine Learning. Classification.

LISTA DE FIGURAS

LISTA DE TABELAS

ÍNDICE DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

ENEM	Exame Nacional do Ensino Médio
SiSu	Sistema de Seleção Unificada
ProUni	Programa Universidade para Todos
FIES	Fundo de Financiamento Estudantil
INEP	Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira
ML	Machine Learning
AP	Aprendizado de Máquina
RF	*Random Forest*
EM	Ensino Médio

SUMÁRIO

1 INTRODUÇÃO

A desigualdade social no Brasil infelizmente é muito alta, o que prejudica a população em diferentes aspectos, inclusive todas as fases da educação. Com o passar do tempo a educação pública foi deixando de se preocupar com a qualidade de ensino. Muitas crianças atualmente não tem acesso a escola, ás vezes por conta que da falta de vagas na região ou até mesmo por não existirem escolas em determinadas regiões do Brasil, dado o seu tamanho. Mas o fato de uma criança frequentar a escola não significa que terá disponível todas as ferramentas necessárias, algumas escolas nõ possuem elementos básicos como uma cadeira. Segundo o IBGE[1], em 2018, 99,3% das crianças de 6 a 14 anos frequentavam a escola. Mas o ensino que é dado, é padronizado? Todas essas escolas tem todas as ferramentas e condições básicas necessárias para dar um ensino de qualidade? Infelizmente não.

Quando olhamos a faixa etária de 15 a 17 anos, esse percentual cai para 88,2%. Ou seja, claramente os cidadãos não seguem estudando. Se pensarmos somente no cenário em que toda a população estudasse em escola pública, já fica claro que nem todos tem as mesmas oportunidades e isso pode influênciar uma vida inteira. Mas quando olhamos o cenário atual do país, onde um parte da população estuda em escolas particulares, essa desigualdade se torna ainda maior.

Dado isto, o projeto vai utilizar os dados extraídos do INEP[2] sobre o ENEM para aplicar técnicas de *Machine Learning*. O objetivo é avaliar a diferença de desempenho de alunos de rede pública e privada utilizando a pontuação final do exame. [1] [2]

1.1 ENEM

O Exame Nacional do Ensino Médio (ENEM) é uma prova do Governo Federal criada em 1998 com o objetivo de avaliar o desempenho dos alunos que finalizam o ensino médio. A partir de 2009, o exame começou a ser aceito para os estudantes ingressarem no Ensino Superior. Atualmente, os alunos são avaliados em 180 questões objetivas a partir dos conhecimentos: linguagens, códigos e suas tecnologias, ciências humanas e suas tecnologias, e, mátematica e suas tecnologias. Além das questões, os alunos devem redigir uma redação (dissertativo-argumentativo), onde a situação é dada pelo exame. O exame é aplicado em dois dias (INEP, 2021b).

Hoje em dia, existem três principais programas que utilizam a nota do ENEM:

- Sisu (Sistema de Seleção Unificada): Reúne vagas de universidades federais e

[1] https://biblioteca.ibge.gov.br/visualizacao/livros/liv101657_informativo.pdf
[2] https://www.gov.br/inep/pt-br/acesso-a-informacao/dados-abertos/microdados/enem

estaduais. Algumas universidades portuguesas já estão aceitando como meio de ingresso. Para conseguir usar a nota existem dois requisitos: ter feito o ENEM no ano anterior e não zerar a redação. Não existe nota mínima para se inscrever, mas as Universidades podem colocar notas em cada curso e além disso, as notas variam de acordo com a concorrência e número de vagas disponíveis. Com isso, cada curso em cada Universidade tem sua nota de corte (nota mínima do ENEM para tentar a vaga). As vagas são divididas em dois grupos: vagas para qualquer aluno concorrer e vagas para alunos que se encaixam nos perfis que dão direito as cotas.

- ProUni (Programa Universidade para Todos): Reúne vagas de bolsas (100% ou 50%) em universidades privadas. Os requisitos são os mesmos do SiSu, porém no ProUni é obrigatória uma pontuaçõ mínima no ENEM de 450 pontos. Os estudantes são classificados por curso e instituição atráves de sua nota e critérios para participação (BRASIL, 2018). O programa é focado em alunos de baixa renda (renda mensal bruta de até 3 salários mínimos) que não possuem ensino superior completo;

- FIES (Fundo de Financiamento Estudantil): Programa onde o Governo Federal faz o financiamento para os alunos que não possuem condições de pagar uma universidade privada. Os requisitos para se candidatar ao financiamento são: ter realizado o ENEM a partir de 2010, pontuação acima de 450 pontos, não zerar a redação e possuir renda mensal bruta de até 3 salários mínimos. Os alunos inscritos são classificados de acordo com o grupo de preferência esolhido e a nota do ENEM (descrescente) (BRASIL, 2021). Os alunos só começam a pagar após concluírem o curso, possuem prazo extenso para pagamento das parcelas e taxa de juros baixas.

O exame tem como foco alunos que estão no último ano do ensino médio. Porém, com o passar dos anos, muitos alunos que já finalizaram ou até mesmo os que não concluíram ainda o ensino médio, começaram a realizar o exame como forma de praticar. No ano de 2020 foram 5.783.357 de inscritos, um aumento de 13,5% em relação a 2019 (INEP, 2021a).

Todos os candidatos que fazem o ENEM são cobrados de forma idêntica, mesmos assuntos, mesmas questões. A dúvida é: os alunos de escolas públicas têm a chance de ter o mesmo desempenho que alunos de escolas privadas? Este ponto é importante pois a desigualdade social e educacional no Brasil é muito grande. A desigualdade começa pela própria Lei nº 13.415/2017, onde não exige que a rede particular siga a mesma regra que a pública (SILVA; MELO, 2018).

Barros (2014) citou a questão dessa desigualdade como relevante:

> "Em se tratando de estudantes oriundos da rede pública, a baixa autoestima é vista como um dos maiores causadores da autoexclusão nos vestibulares e no ENEM. Os principais argumentos de muitos adolescentes para justificar a falta

de interesse pelos exames é a crença de que não são capazes de conseguir bons resultados ou de competir com alunos de outras escolas."

Corroborando, Sampaio e aes (2009), levantaram outros pontos que estão diretamente correlacionados com a classe social do aluno: nível de educação dos pais e acesso á internet. Este último pode fazer uma grande diferença nos estudos, dado que é possível acessar muitas informações e aprofundar os estudos. Teoricamente, a rede pública deveria fornecer acesso á internet e livros aos alunos, mas a maioria das vezes não possuem recursos para investir.

O problema da desigualdade educacional é muito mais profundo do que somente a classe social. Até mesmo para os alunos que ocupam a mesma classe e estudam em rede pública pode existir uma diferença. Com o passar dos anos, para atingir as metas estipuladas, as ações do Governo podem ter causado uma desigualdade educacional no próprio ensino público. Isso se deve ao fato que muitas vezes os Governos Estaduais tomavam suas próprias decisões, sem haver um padrão. O resultado disso foi que o ensino público em estados mais ricos são melhores que outros estados (SAMPAIO; OLIVEIRA, 2015).

É importante salientar que não é porque um aluno estuda em rede pública que não é possível ter um bom desempenho no ENEM. Até porque mesmo que raras, existem escolas públicas que conseguem dar a base suficiente para os alunos. Isso somado ao esforço dos alunos chegam a resultados positivos. A ideia é avaliar de forma geral os alunos que ocupam essas duas posições diferentes.

1.2 *Machine Learning*

Machine Learning (ML) ou aprendizado de máquina (AM), são técnicas utilizadas para tentar encontrar padrões em determinadas situações. Segundo Géron (2019): "*Machine Learning* é a ciência (e arte) da programação de computadores para que eles aprendam com os dados" (Tradução própria).

Essas técnicas utilizam uma combinação entre computação, estatística e matemática. Outra definição que podemos ter de ML é:

"*Machine Learning* é sobre computadores e outros sistemas artificiais aprendam sem programá-los explicitamente. É aqui que queremos que esses sistemas vejam alguns dados, aprenda com eles e use esse conhecimento para inferir coisas de outros dados."(Tradução própria) (SHAH, 2020).

Podemos dividir em dois grupos as situações principais que usamos ML: onde temos um alvo que gostaríamos de acertar e outro onde não sabemos o que queremos acertar, mesmo assim queremos encontrar um padrão ou um grupo.Essas situações são chamadas

de supervisionadas e nção supervisionadas respectivamente. Os algoritmos de ML usam diversos atributos que mostram características sobre aquele grupo que se quer encontrar um padrão e através delas os algoritmos classificam dado o alvo ou calculam a probabilidade de ele fazer parte daquele alvo. Nos casos não supervisionados, os algoritmos usam os atributos para tentar encontrar grupos parecidos. Os algoritmos conseguem encontrar padrões aprendendo com a experiência passada daquela determinada situação (FACELI et al., 2011). Os algoritmos que podemos aplicar em ML podem ser simples ou complexos. Como exemplos de técnicas simples, temos:

Árvore de Decisão (*Decision Tree*): é uma técnica supervisionada usada para soluções de classificação. Ela usa os atributos (variéveis explicativas) dividindo em grupos e em subgrupos, com isso uma árvore de decisão é desenvolvida. A árvore é formada por nós e folhas (SHAH, 2020).

Regressão Logística: também é uma técnica supervisionada. Encontra a relação dos atributos com a variável resposta. O resultado é uma probabilidade (de 0 a 1) da observação fazer parte do alvo ou não.

Já para modelos complexos:

***Random Forest* (Florestas Aleátorias):** este modelo utiliza diversas árvores de decisão. A diferença é que ele cria diferentes árvores de decisão para amostras diferentes dos seus dados e não usa todas as variéveis explicativas em todas as árvores. Isso é feito para evitar a correlação entre elas e possuir um resultado mais confiável. O resultado é a média de todas as árvores criadas (JAMES; WITTEN; TIBSHIRANI, 2013).

Com o passar dos anos e os avanços da técnologia, aplicar ML se tornou mais fácil dado os computadores eficientes que foram surgindo. Temos um amplo universo onde podemos aplicar as técnicas de ML hoje em dia. São utilizados em áreas totalmente diferentes, possuindo dados é possível aplicar qualquer técnica. Existem diversas aplicações que são muito utilizadas atualmente e fazem sucesso dado sua assertividade. Alguns exemplos de soluções são:

- Área da Saúde: modelos de ML são usados para diagnosticar câncer através de características de exames;

- Risco de Crédito: são aplicados para prever se um cliente pode ficar inadimplente no futuro;

- Fraudes: as técnicas de ML ajudam a detectar a probabilidade de uma transação no cartão de crédito ser fraudulenta;

- Compras Online: com o aumento do comércio online, uma das estratégias das lojas é fazer recomendações de produtos baseado em um produto que o cliente comprou. Essas recomendações são feitas através de modelos que encontram padrões de compras

para perfis similares;

- Música: treinar modelos para inspirar novas melodias com base em outras melodias de um mesmo artista (JÚNIOR et al., 2019).

Na área da educação não é diferente, conseguimos criar soluções usando ML. Porém, atualmente ainda não são muito exploradas. Tavares, Meira e Amaral (2020) acredita que usar a inteligência artificial (técnicas de ML estão dentro do escopo de IA) na educação só tende a agregar:

> "Estudar o uso da IA na educação é uma forma de buscar soluções que possam agregar valor para o processo de ensino-aprendizagem, para apoiar professores e alunos, porém, sem negligenciar o aspecto humano, sem esquecer habilidades como ética e responsabilidade, trabalho em equipe e flexibilidade (. . .) ".

Como citado acima, a aplicação de técnicas de ML na área da educação ainda não são muito comuns. Muitos trabalhos são focados em criar alguns algortimos específicos e não na aplicação dos já existentes. Mas será apresentado dois exemplos práticos a seguir.

Primeiro temos o estudo de Leão et al. (2021), onde aplicou técnicas de AP (aprendizado de máquina) para apoiar a aprendizagem adaptiva. A primeira técnica utilizada foi de agrupamento (*K-means*, não supervisionada). O objetivo era encontrar grupos que tivessem preferências de aprendizado parecidos. As preferências podiam ser: áudio, imagem, gráficos, páginas da web, fóruns entre outros. O segundo passo foi utilizar o resultado do *K-means* em uma técnica de classificação (árvore de decisão). O resultado do agrupamento serviu de variável resposta (alvo) para aplicar a classificação. Com isso, foi possível encontrar quais preferências de aprendizagem faziam parte de cada grupo de alunos e poderia ser utilizado para encontrar qual grupo um novo aluno pertenceria por exemplo.

Já o trabalho de Ezaki (2020) tinha um próposito bem diferente. O objetivo era analisar a equidade de oportunidade educacional das crianças no Nepal. Utilizou-se uma regressão logística com atributo alvo binário: aluno que estuda em escola privada e aluno que estuda em escola pública. A área escolhida para realizar o estudo foi o subúrbio de Bhaktapur District, onde o número de crianças que estudam em escolas privadas tem aumentado e a transferência entre escolas públicas para privadas também. Os atributos explicativos utilizados na regressão logística focaram nas características individuais das crianças e informações do histórico familiar. Como resultado da aplicação do algoritmo, os atributos que tiveram uma maior importância para conseguir explicar a probabilidade de crianças frequentarem escolas privadas foram: ser do genêro masculino, ser o filho mais velho em uma família, a ocupação do pai ser relacionada a trabalho intelectual, a mãe ser

alfabetizada e a família ter posses (patrimônio). O trabalho citado tem o objetivo muito parecido com o que queremos aplicar neste.

1.3 Objetivos Gerais

O objetivo geral deste trabalho é análisar as características pessoais e socioeconômicas do aluno e identificar através de técnicas de ML se existe uma correlação entre elas e a pontuação do ENEM. Para este trabalho foi assumido como alvo ter pontuação maior ou menor que 500 pontos. Esse corte específico foi escolhido dado que a pontuação mínima para conseguir uma vaga no SiSu no ano de 2020 girou em torno de 500 (MAURO, 2021).

Além disso, a pontuação mínima que um candidato deve ter para tentar vagas no ProUni e FIES são 450 pontos.

1.4 Objetivos Específicos

- Usar modelos de predição/classificação para encontrar as características socioeconômicas relevantes;

- Comparar quais modelos de ML possuem uma melhor performance para acertar o objetivo;

- Analisar se diferentes regiões do Brasil possuem as mesmas informações para explicar a chance de o aluno conseguir tirar mais de 500 pontos no ENEM.

1.5 Organização

O trabalho esta organizado em quatro partes principais. A primeira apresenta a introdução ao tema e problema, referencial bibliográfico, objetivos e como esta organizado. A segunda parte esta focada para apresentar os principais tratamentos e técicas que serão utilizados na base de dados. Nos dois últimos são apresentados o resultado da aplicação dos tratamentos e técnicas e a conclusão dos resultados, respectivamente.

2 MATERIAL E METODOLOGIA

2.1 Introdução

A metodologia deste trabalho irá abordar técnicas de *Machine Learning* supervisionadas que serão utilizadas, tratamento de dados e medidas de desempenho dos modelos (algoritmos que foram ensinados). Segundo Géron (2019):

> "O aprendizado de máquina é a ciência (e arte) de programar computadores para que eles possam aprender com os dados. "

Os dados que usamos para ensinar as técnicas são as informações/características que temos de determinada situação ou problema. No caso das técnicas supervisionadas além das características precisamos do rótulo do problema. O rótulo nada mais é do que a resposta do seu problema. Para exemplificar podemos pensar em um problema de saúde. O objetivo é ensinar o algoritmo através de algumas informações extraídas de um exame se a pessoa está doente ou não. O fato de estar ou não doente é o rótulo do problema. Neste trabalho, as características do problema são as informações dos alunos, como região de moradia, indíces socioeconómicos entre outros, e o rótulo é se o aluno teve ou não uma pontuação maior que 500 pontos no ENEM.

Para conseguir ensinar bem um algoritmo de AM é necessário ter um cuidado com os dados. Existem dados de diversos formatos, podem ser números, textos, imagens etc. Normalmente são divididos em dados estruturados e não estruturados.

- Dados estruturados: são informações organizadas em uma determinada estrutura/padrão. A mais comum chamamos de "base de dados", onde os dados ficam representados de forma tabular. As colunas são as características sobre o problema e as linhas são as observações daquela situação. Neste caso, cada observação é um aluno.

- Dados não estrurados: acontece quando os dados não estão representados de forma organizada, as informações estão em diferentes padões ou está em um formato não "interpretável" para os algoritmos, por exemplo: áudios, fotos etc.

Além de uma base de dados estruturada, nécessário trazer tratamentos nos dados para possíveis problemas: balanceamento de base, dados categóricos, ruídos, *outliers*. Esses assuntos e o conjunto de dados que será utilizado neste trabalho serão detalhados nos próximos tópicos.

2.2 Conjunto de Dados

Os dados utilizados neste trabalho são referentes ao ENEM de 2019 e foram extraídos no site do INEP. Os dados disponibilizados não possuem informações que podem ser utilizadas para identificar os alunos. Abaixo seguem as informações (que chamaremos de atributos) que que serão usadas em algum momento no trabalho (no arquivo original existem informações que não serão utilizados e não serão indicadas neste trabalho):

1. NU_INSCRICAO: Número da incrição;
2. NU_ANO: Ano em que o ENEM foi realizado;
3. CO_MUNICIPIO_RESIDENCIA: Código do município da residência do candidato.
4. NO_MUNICIPIO_RESIDENCIA: Nome do município da residência do candidato.
5. CO_UF_RESIDENCIA: Código da unidade de federação (estado) de residência do candidato;
6. SG_UF_RESIDENCIA: Sigla da unidade de federação (estado) de residência do candidato;
7. NU_IDADE: Idade do candidato;
8. TP_SEXO: Sexo do candidato;
9. TP_ESTADO_CIVIL: Indica o estado civil do candidato;
10. TP_COR_RACA: Indica a raça que o o candidato se identifica;
11. TP_NACIONALIDADE: Indica a nação (país) em que o candidato nasceu;
12. SG_UF_NASCIMENTO: Sigla da unidade de federação (estado) em que o candidato nasceu;
13. TP_ST_CONCLUSAO: Situação de conclusão do ensino médio do candidato. Indica se já finalizou ou não;
14. TP_ESCOLA: Indica o tipo de escola que o candidato estudou no ensino médio;
15. TP_ENSINO: Indica o tipo de instituição que estudou no ensino médio;
16. TP_LOCALIZACAO_ESC: Tipo de localização da escola;
17. IN_BAIXA_VISAO: Indica se o aluno apontou que tem baixa visão;
18. IN_CEGUEIRA: Indica se o aluno apontou que sofre de cegueira;

19. IN_SURDEZ: Indica se o aluno apontou que sofre de surdez;
20. IN_DEFICIENCIA_AUDITIVA: Indica se o aluno apontou que sofre de alguma deficiência auditiva;
21. IN_SURDO_CEGUEIRA: Indica se o aluno apontou que sofre de surdez e cegueira;
22. IN_DEFICIENCIA_FISICA: Indica se o aluno apontou que tem alguma deficiência fisíca;
23. IN_DEFICIENCIA_MENTAL: Indica se o aluno apontou que tem alguma deficiência mental;
24. IN_DEFICIT_ATENCAO: Indica se o aluno apontou que tem déficit de atenção;
25. IN_DISLEXIA: Indica se o aluno apontou que tem dislexia;
26. IN_DISCALCULIA: Indica se o aluno apontou que tem discalculia;
27. IN_AUTISMO: Indica se o aluno apontou que tem autismo;
28. IN_VISAO_MONOCULAR: Indica se o aluno apontou que tem visão monocular;
29. IN_OUTRA_DEF: Indica se o aluno apontou que possui alguma outra deficiência ou condição especial que não foi citada em outra pergunta;
30. TP_PRESENCA_CN: Indica se o aluno estava presente na prova de Ciências da Natureza;
31. TP_PRESENCA_CH: Indica se o aluno estava presente na prova de Ciências Humanas;
32. TP_PRESENCA_LC: Indica se o aluno estava presente na prova de Linguagens e Códigos;
33. TP_PRESENCA_MT: Indica se o aluno estava presente na prova de Matemática;
34. NU_NOTA_CN: Nota do aluno na prova de Ciências da Natureza;
35. NU_NOTA_CH: Nota do aluno na prova de Ciências Humanas;
36. NU_NOTA_LC: Nota do aluno na prova de Linguagens e Códigos;
37. NU_NOTA_MT: Nota do aluno na prova de Mátematica;
38. TP_STATUS_REDACAO: Indica o status da redação do aluno;
39. NU_NOTA_REDACAO: Nota do aluno na redação;

40. Q001: Pergunta até que série o pai ou homem responsável pelo aluno estudou;
41. Q002: Pergunta até que série a mãe ou mulher responsável pelo aluno estudou;
42. Q003: Indica em qual grupo de ocupação o pai ou homem responsável se identifica;
43. Q004: Indica em qual grupo de ocupação a mãe ou mulher responsável se identifica;
44. Q005: Pergunta quantas pessoas moram na residência do aluno;
45. Q006: Pergunta qual a renda da famiília (somando de todos que trabalham e moram na residência);
46. Q007: Pergunta se na residência do aluno trabalha empregado(a) doméstico(a);
47. Q008: Pergunta se na residência do aluno tem banheiro, se sim quantos;
48. Q009: Pergunta se na residência do aluno tem quartos para dormir, se sim quantos;
49. Q010: Pergunta se na residência do aluno tem carro, se sim quantos;
50. Q019: Pergunta se na residência do aluno tem TV em cores, se sim quantos;
51. Q021: Pergunta se na residência do aluno tem TV por assinatura;
52. Q022: Pergunta se na residência do aluno tem celular, se sim quantos;
53. Q024: Pergunta se na residência do aluno tem computador, se sim quantos;
54. Q025: Pergunta se na residência tem acesso á internet.

A partir do Capítulo 3 será mostrado como os atributos serão utilizados e a distribuição dos principais atributos.

2.3 Tratamento dos Dados

Para conseguir fazer um bom tratamento de dados, é necessário conhecer todos os seus atributos (características sobre o problema), chamamos de "Exploração dos Dados". A exploração tem como objetivo mostrar como os dados estão disponíveis. Pode ser feito através de fórmulas estatísticas ou análise visual (FACELI et al., 2011). A exploração será divididas em algumas partes: Identificação do tipo do dado para cada atributo, distribuição dos dados, tratamento de outliers, tratamento de dados faltantes, tratamento de dados categóricos e por fim tratamento de dados redundantes.

2.3.1 Identificação do Tipo de Atributo

O objetivo é identificar o que e como o atributo representa a informação. Podemos ter dados em 3 formatos: número, data e texto. Para cada tipo é necessário ver como esta 'preenchido' na base de dados. Para dados números é importante ver a unidade e escala que estão representados, números de casa decimais e o número máximo/mínimo possível. Em atributos que representam datas, verificar o período que esta sendo mostrado e o formato da data. Já para atributos que estão em texto, precisamos conhecer a informação representada naquele atributo e quais são as respostas possíveis. Por exemplo: tem informações sobre o tempo (sol, chuva, nublado), tipo de sangue (A+, B-, O-).

2.3.2 Distribuição dos Dados

Nesta parte da exploração podem ser utilizados fórmulas estatísticas e formas visuais. As análises baseadas em estatística mais comum usados em dados númericos são: média, mediana, desvio padrão e quartis.

- Média: A média de um conjunto é a soma dos valores de todas as observações dividido pela quantidade de observações. Sua forma mais comum de representação é $x_{\bar{O}bs}$.

$$x_{\bar{O}bs} = \frac{1}{n}\sum_{i=1}^{n} x_i,$$

  em que n representa a quantidade total de observações no conjunto e x_i é cada uma das observações.

- Mediana: é o valor que representa a posição de 50% dos dados. Pode ser representado por M_d. Se o número de observações for ímpar a mediana é o valor central de todos os números daquele atributo. Se for par:

$$M_d = \frac{a+b}{2},$$

  em que a e b são os números centrais do conjunto.

- Desvio Padrão: É uma medida de dispersão dos dados, o quanto eles são semelhantes. Pode ser representado por σ.

$$\sigma = \sqrt{\frac{1}{n}\sum(x_i - x_{\bar{O}bs})^2},$$

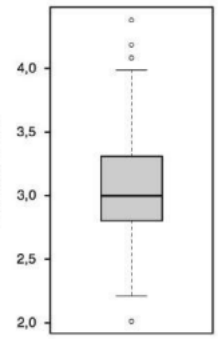
  em que n representa a quantidade total de observações no conjunto, x_i é cada uma das observações e $x_{\bar{O}bs}$ é a média observada do conjunto.

- Quartis: temos 3 quartis, o primeiro representa 25% das primeiras observações do conjunto, o segundo representa 50% (significa a mesma coisa que a mediana) e o terceiro quartil representa 75%. Para conseguir calcular esses quartis os dados precisam estar ordenados (MAGALHãES; LIMA, 2015).

2.3.3 *Outliers*

Outliers são observações que quando comparado a outras observações do mesmo atributo são diferentes, os valores distoam muito. Uma das técnicas utilizadas para identificar esses casos é o boxplot, que é uma medida de visualização. Além de visual, o boxplot trás informações do valor minímo, valor máximo, primeiro quartil, mediana e terceiro quartil. Tudo que está antes do valor minímo ou depois do valor máximo, são valores extremos que podemos considerar como *outliers*. Na Figura 1 temos um exemplo de boxplot:

Figura 1 – Exemplo de Boxplot



Fonte: Faceli et al. (2011)

Os pontos que estão "fora" dos intervalos do boxplot podemos considerar como *outliers*. São 3 pontos acima do 4 e 1 ponto abaixo do 2.

2.3.4 **Dados Faltantes**

É normal em uma base de dados termos atributos onde algumas observações não estão preenchidas. Por exemplo: se eu tenho um atributo que mostra a idade de cada exemplo, pode ter casos onde parte dos exemplos venha vazio. São os dados faltantes. Esse problema pode acontecer por diversos motivos. Alguns deles são: erro de preenchimento, falta da informação no momento, a informação não não era obrigatória entre outros. O ideal é tratar esses dados faltantes pois alguns algoritmos não conseguem trabalhar com essa situação. Existem algumas técnicas para resolver esse problema, alguns exemplos são: excluir os exemplos que tem dados faltantes da base de dados (não é uma boa saída quando se tem uma amostra pequena), substituição dos dados faltantes por média ou

moda ou mediana do atributo. Além disso, é possível treinar um algoritmo para substituir os dados faltantes (FACELI et al., 2011).

2.3.5 Dados Categóricos

A maioria dos algoritmos de AM só trabalham com dados numéricos. Isso pode ser um problema caso a base de dados conter dados categóricos. Para resolver este tipo de problema é preciso fazer uma transformação nos dados. Se o atributo categórico só for preenchido por duas possibilidades (ter ou não ter determinada característica), podem ser transformado em uma "flag", que é composta por 1 e 0. Caso o atributo tenha mais de 2 possibilidades uma solução é usar o *One-hot-enconding*, que transforma cada possível resposta do atributo em um novo atributo sendo preenchido por 1 ou 0.

2.3.6 Dados Redundantes

Dados redundantes são dados que tem o mesmo significado. Isso é um dos passos essenciais na hora de treinar um algoritmo. O que acontece muitas vezes é que alguns atributos, para o algoritmo, explicam a mesma informação e isso pode afetar no resultado final do ajuste do algoritmo. Nesses casos, na maior parte das vezes escolhemos a variável com maior importância. Uma das formas de identificarmos se existem atributos redundantes é calcular a correlação:

$$\rho_{x,y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x}) - (y_i - \bar{y})}{\sigma x.\sigma y},$$

em que n representa a quantidade total de observações no conjunto, x_i é cada uma das observações do atributo X, $\bar{x}$ é a média do atributo X, y_i é cada uma das observações do atributo Y, $\bar{y}$ é a média do atributo Y,σx é o desvio padrão de X (explicado no item 2.2.2) e σy é o desvio padrão de Y.

O numerador

$$\sum_{i=1}^{n}(x_i - \bar{x}) - (y_i - \bar{y})$$

também é conhecido como a covariância de x,y ($\text{Cov}_{x,y}$). Segundo Magalhães e Lima (2015):

> "... o coeficiente de correlação é o quociente entre covariância e o produto dos desvios padrão de X e Y. A divisão pelo produto dos desvios padrão tem a função de padronizar a medida e torná-la possível de ser utilizada para comparações com outras variáveis."

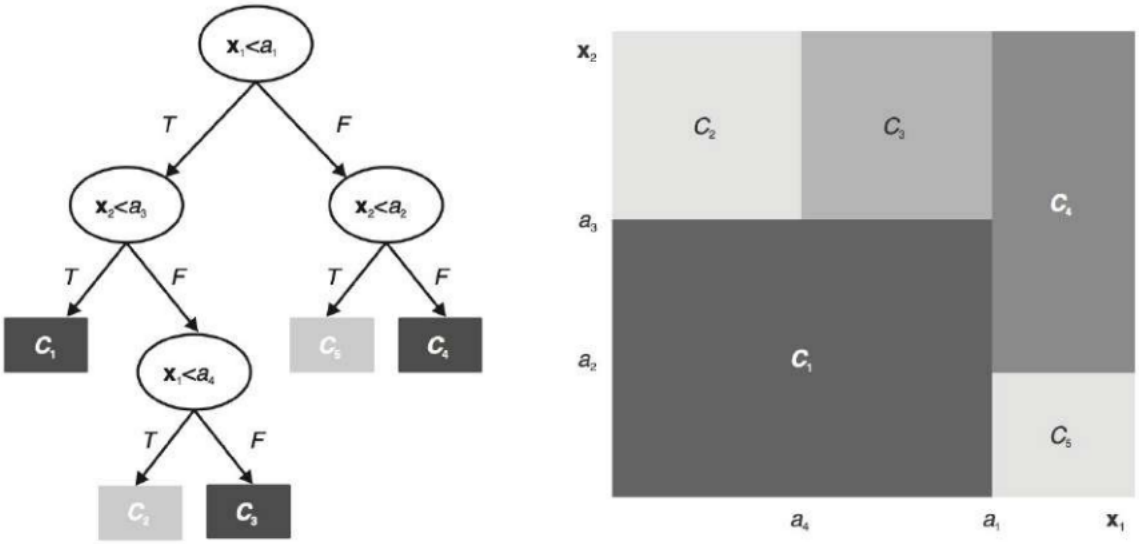
2.4 Técnicas de *Machine Learning*

2.4.1 Árvores de Decisão

Árvore de Decisão é um algoritmo baseado em procura e geralmente utilizado em problemas de classificação (supervisionado). O algoritmo quebra o conjunto total de observações em subconjuntos para conseguir separar a classe que se deseja predizer. Os subconjuntos são criados atráves de um corte no valor de um dos atributos, que é uma condição (ex: idade >= 20 anos). A primeira quebra é chamado de nó raiz, dela vai sair como alguns subconjuntos. Desses, se forem subconjuntos que conseguem discriminar bem uma classe, é finalizado e chamado de folha, caso contrário é necessário usar outro atributo e fazer outra quebra. Neste caso, chama-se nó de decisão.

Segundo Faceli et al. (2011), a árvore de decisão basicamente usa um problema complexo e divide em problemas mais simples. Onde em cada problema simples é usado a mesma estratégia de conseguir separar as classes. A divisão em subconjuntos é feita até o critério de parada ser atingido. No final, a solução dos problemas mais simples são combinados. Na Figura 2 podemos ver um exemplo simples de uma árvore de decisão:

Figura 2 – Exemplo Árvore de Decisão



Fonte: Faceli et al. (2011)

No exemplo, são usados como atributos preditores x_1 e x_2. As condições criadas pela árvore estão representadas no quadro ao lado. Quanto maior a entropia, maior a incerteza.

Os atributos que são escolhidos para quebrar os nós são selecionados de acordo com um critério. O mais comum é chamado de Entropia. Shah (2020) descreve como:

"Entropia (E) é uma medida de desordem, incerteza ou aleatoriedade."

Sua fórmula é representada da seguinte forma:

$$E = -\sum_{i=1}^{k} p_i . \log 2(p_i),$$

em que k é a quantidade de classes, p_i é o número de vezes que aparece na classe i.

Na árvore de decião é necessário darmos a situação em que a árvore deve parar de "crescer", ou seja, quando ela deve parar de selecionar novos atributos. Caso isso não seja feito, pode resultar em alguns problemas,como o *overffiting* ou uma árvore muito complexa (difícil de interpretar). Este procedimento pode ser chamado de critério de parada ou "poda" da árvore. Existem 2 formas de ser feito, antes da construção da árvore (pré poda) e após o algporitmo já ter criado a árvore (pós poda). Os critérios mais utilizados entre essas duas possibilidades são:

- **Pré-poda:** Podem ser utilizadas diversas abordagens, como por exemplo: determinar um número mínimo de observações em cada folha, determinar o máximo de nós que podem ser criados (profundidade da árvore) ou quando a folha tiver 100% das observações de uma mesma classe.

- **Pós-poda:** Neste caso, permitimos que o algoritmo cria a árvore sem nenhuma restrição. Provavelmente terá um resultado complexo e superajustado aos dados (*overffiting*). Uma das maneiras mais simples é usar a diferença entre as medidas "Erro estático" e "Erro de *backed-up*". Já a maneira mais utilizada é "custo de complexidade". Esse tipo de poda utiliza a taxa de erro e o quão grande a árvore é (FACELI et al., 2011).

O algoritmo abaixo, mostra o passo a passo para a construção de uma árvore de decisão:

Algoritmo 1: Construção de uma árvore de decisão

Entrada: Um conjunto de treinamento D = (X_i, Y_i), i = 1, ..., n
Resultado: Árvore de Decisão
1 **/* Função GeraÁrvore(D) */**;
2 **se** *critério de parada*(**D**) = *Verdadeiro* **então**
3 **Retorna:** um nó folha rotulado com a constante que minimiza a função perda;
4 **fim**
5 Escolha o atributo que maximiza o critério de divisão em **D**;
6 **para cada** *partição dos exemplos* D_i *baseado nos valores do atributo escolhido* **faça**
7 Induz uma subárvore $Árvore_i = GeraÁrvore(D_i)$;
8 **fim**
9 **Retorna:** *Árvore* contendo um nó de decisão baseado no atributo escolhido, e descendentes $Árvore_i$;

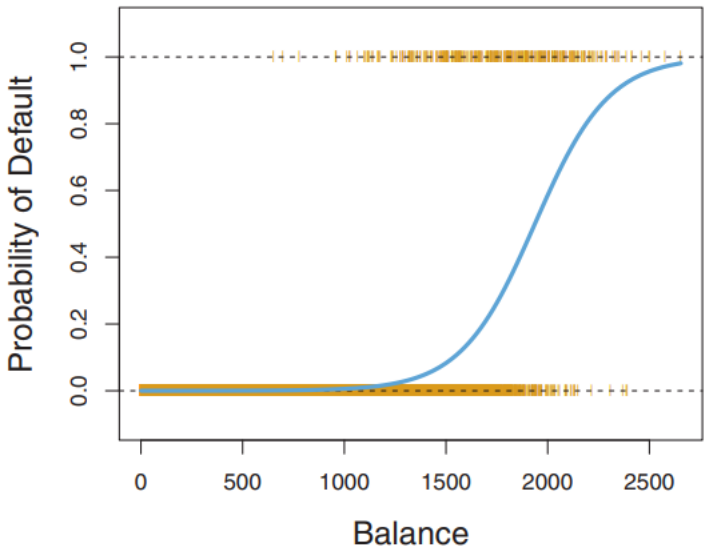
Fonte: Adaptado de Faceli et al. (2011)

2.4.2 Regressão Logística

Regressão Logística é um dos algoritmos mais conhecidos e aplicados em AM. Também é supervisionado e uma de suas aplicações é em problemas binários (0 e 1), que será o foco deste trabalho. De acordo com Diniz e Louzada (2012) este algoritmo é ideal para definir a probabilidade de uma observação ser da classe 1 ou 0 de acordo com os atributos utilizados. Normalmente usamos como valor 1 a situação que queremos predizer, por exemplo, se eu quero saber se uma pessoa está doente utilizo 1, caso contrário 0.

A regressão logística é parecida com a regressão linear multipla, onde quero encontrar a relação dos atributos com o atributo alvo. Mas dado que se trata de um problema de classificação a saída do modelo deve ter uma resposta entre 0 e 1. Na Figura 3 temos um exemplo da distribuição de probabilidade da regressão logística:

Figura 3 – Exemplo da distribuição de probabilidade de Regressão logística



Fonte: James, Witten e Tibshirani (2013)

A Figura 3 nos mostra como funciona a distribuição de probabilidade da regressão logística, onde confirmamos que temos valores entre 0 e 1 somente.

Como citado acima, queremos ter uma probabilidade entre 0 e 1, por isso é necessário aplicar a função logística na regressão, senão podemos ter probabilidades negativas, o que não faz sentido na interpretabilidade de um modelo. A função logística é representada por:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\ldots+\beta_k X_k}}{1+e^{\beta_0+\beta_1 X_1+\ldots+\beta_k X_k}},$$

em que $p(X)$ representa a probabilidade de X acontecer, β_0 é o intercepto do modelo, β_1 é um dos coeficientes e X_1 representa um dos atributos preditivos. Na fórmula temos β_k e X_k, que signfica mais atributos que podemos adicionar no modelo, para cada atributo temos um coeficiente (β).

Após manipular a fórmula temos:

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X_1+\ldots+\beta_k X_k},$$

em que $\frac{p(X)}{1-p(X)}$ é chamado de *odds*, que possui valor de 0 até ∞. Quanto mais próximo de 0 menor a probabilidade, e quanto mais próximo de ∞ maior é a probabilidade (JAMES; WITTEN; TIBSHIRANI, 2013).

Após aplicar a função logarítmica:

$$\log\left\{\frac{p(X)}{1-p(X)}\right\} = \beta_0 + \beta_1 X_1 + ... + \beta_k X_k,$$

em que, $p(X)$ é a probabilidade da classe, os β_k são os coeficientes de cada atributo k e X_k são os valores para cada atributo k. A $\log\left\{\frac{p(X)}{1-p(X)}\right\}$ pode ser chamada de logito. Segundo Bruce e Bruce (2019), logito é: "A função que mapeia a probabilidade de pertencimento a uma classe com amplitude de $\pm\infty$".

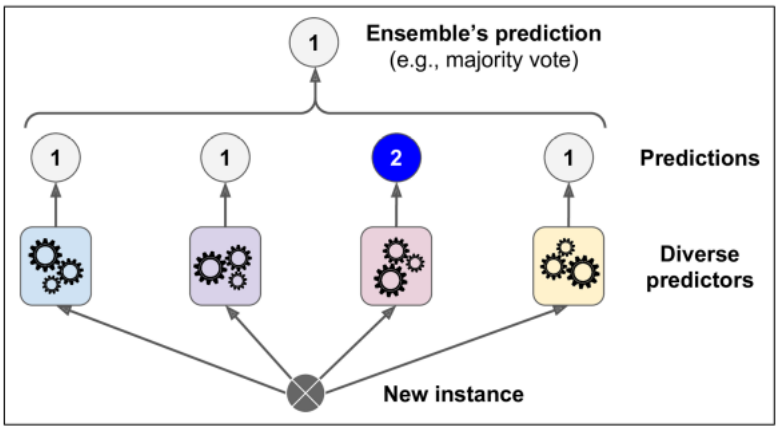
Diferente da regressão linear, em que existe uma relaçõ linear entre $p(X)$ e X, a estimação dos coeficientes em regressão logística é feito de forma diferente. Para fazer a estimação dos coeficientes na regressão logística normalmente é utilizado o método geral da máxima verossimilhança. De acordo com James, Witten e Tibshirani (2013), esse método tenta encontrar um valor para β_0 e β_k de modo que $p(X)$ seja próximo de 1 para todas as observações quando o evento desejado acontece ou próximo de 0 para todas as observações quando o evento não acontece. A equação que representa a função de verossimilhança é:

$$l(\beta_0, \beta_k) = \prod_{i:y_i=1} p(x_i) \prod_{i':y'_i=0} (1 - p(x'_i)).$$

2.4.3 *Randon Forest* (Floresta Aleatória)

O RF (*Random Forest*) é um algoritmo que faz parte do método *"ensemble"*. Este método basicamente utiliza como algoritmo vários algoritmos de AM. Por exemplo: se para fazer uma predição utilizar mais de um algortimo ou o mesmo algortimo mas várias vezes, combinando o resultado final, é considerado um *"ensemble"*. Na Figura 4 temos uma ilustração de como seria um *ensemble*:

Figura 4 – Método *ensemble*



Fonte: Géron (2019)

A Figura 4 representa diversos algoritmos (que podem ser diferentes ou iguais), que se encontram no final para combinar as predições e ter um resultado final.

Mesmo se o algoritmo utilizado for fraco, quando são combinados diversos algortimos para dar o resultado final, tende a ter uma precisão melhor do que somente 1 algoritmo forte (JAMES; WITTEN; TIBSHIRANI, 2013). Dado a natureza do nome "Florestas", o RF utiliza várias Árvores de Decisões no seu desenvolvimento. Isso é feito para conseguir a melhor predição possível.

A técnica utilizada no RF é o *bagging*, que é um algoritmo que foca em reduzir a variância dos dados, ou seja, evita o *overfitting*. Isso é feito pois normalmente Árvores de Decisões tendem a ter alta variância. Para que isso seja possível cada árvore que o algoritmo treina usa somente uma amostra do conjunto de dados. A amostragem é feita por uma técnica chamada *"Bootstrap"*, onde a amostragem é por reposição. Ou seja, é possível ter duas observações exatamente iguais no mesmo conjunto de treinamento de determinada árvore. A amostragem é feita em cada árvore treinada no RF. As árvores são treinadas de forma paralelas e são diferentes uma das outras (observações e atributos), justamente para conseguir previsões diferentes. A resposta final do modelo treinado é a moda dos resultados das árvores. Moda é o elemento mais frquente em um conjunto de observações.

A descrição do RF é descrita pelo Algoritmo 2:

Algoritmo 2: *Random Forest*

Entrada: Dado um conjunto de dados X = $x_1, x_2, ..., x_j$ e Y = $y_1, y_2, ..., y_k$.

Resultado: Gera o modelo final: $\hat{f}(x) = \sum_{b=1}^{B} f^b(x)$, que calcula os votos obtidos por cada modelo f^b, resultando uma classificação final de acordo com a votação majoritária.

1 Para b = 1,2,3,... B,repita:
2 Crie uma amostra *bootstrap* (X_b, Y_b) com n exemplos de (X,Y).
3 Ajusta uma árvore de decisão f^b para o conjunto de treinamento (X_b, Y_b), utilizando m atributos para a escolha de cada nó.
4 Fim da repetição.

Fonte: BREINMAN *apud* Vieira, Oliveira e Paiva (2015)

Uma das vantagens de utilizar o RF é sua robustez em relaçõ ao *overfitting*. Isso só é possível pois cada árvore é feita com uma amostra diferente de observações e atributos (para cada nó, os atributos são selecionados de forma aleatoria de um subconjunto de todos os atributos). No que resulta em não ter correlação entre as árvores (PANG-NING et al., 2019).

2.5 Avaliação do Modelo

Após treinar um algoritmo de AM temos como resultado um modelo que consegue predizer dados futuros. Mas como sabemos o quanto o modelo acerta na predição? o Quanto podemos confiar em seu resultado? Para isso calculamos algumas métricas de avaliação, que serão apresentadas em breve. Mas antes de calcular métricas é necessário falar sobre como separar o conjunto de dados para fazer esses calculos.

2.5.1 Base de Treino e Teste

Só é possível saber o desempenho de um modelo, identificar se houve *overffiting* como novos dados, que não foram utilizados para treinar. Dado isso, dividimos o conjunto de dados que temos disponíveis em 2 partes:

1) Treino: dados utilizados para o algoritmo aprender, para o ajuste do modelo final.

2) Teste: dados nunca vistos pelo algoritmo, para poder identificar o quanto ele aprendeu a predizer novos valores. Se na parte do treinamento houve *overffiting*, quando aplicado aos dados de teste o resultado será insatisfatório, pois o algoritmo se ajustou demais aos dados de teste.

Geralmente, a divisão utilizada para no conjunto é de 70% de observações para teste e 30% para treinamento. Essas proporções podem mudar dependendo do problema.

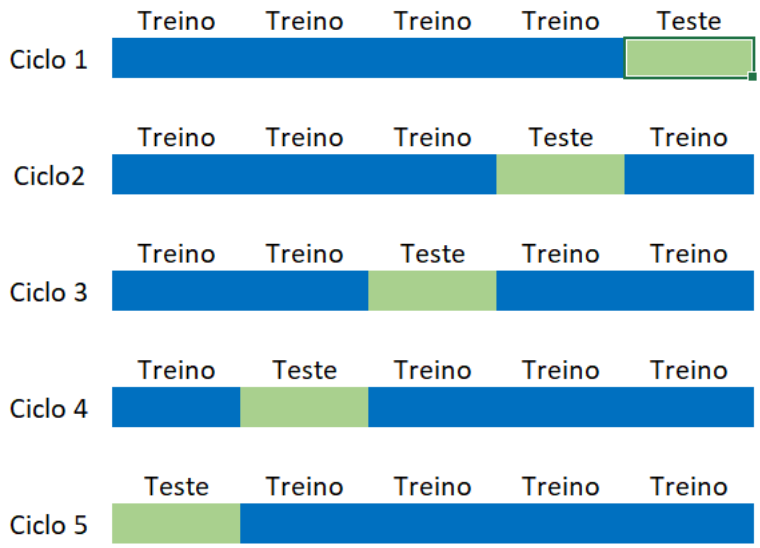
2.5.1.1 *Hould-out Validation:*

É a forma mais simples de dividir os dados. Os dados são selecionados de forma aleatória, respeitando as proporções solicitadas para treino e teste. Um problema desta técnica é o fato de a amostra selecionada para fazer parte do treino não represente da mesma forma o de teste. O ideal é usar esta técnica quando se tem muitos dados.

2.5.1.2 *K-fold Cross Validation:*

Também conhecido como "Validação Cruzada", visa treinar o algoritmo com partições diferentes do conjunto de dados. O "k" é o número de particições que será feito na base de treino. É muito comum encontrar trabalhos com k= 5 ou k=10, mas pode-se escolher qualquer número. Usando como exemplo k=5: a base de treino será dividida em 5 partes, sendo 80% para treinar e 20% para testar o modelo. O algoritmo será treinado 5 vezes, em cada vez a amostra de treino e teste será diferente. A avaliação final do modelo, é a média dos 5 modelos gerados (FACELI et al., 2011). A Figura 5 representa o processo:

Figura 5 – Partições do *k-fold*



Fonte: Elaborada pela autora

A Figura 5 mostra como em cada ciclo (onde o algoritmo é treinado) utiliza-se diferentes partições de treino e teste, até que todas elas tenham sido testadas como treino e teste.

2.5.1.3 *Leave-One-Out Cross-Validation:*

Segundo Shah (2020), este método pode ser considerado uma variação do *k-fold*. Isso porque ao invés de escolher um número de "k" este número já é estabelecido pelo número de observações que se tem no conjunto de dados. A cada ciclo do treino do algoritmo, somente uma observação será utilizada para fazer o teste do modelo. Isso se repetirá até todas as observações terem feito parte do teste. O problema que este método pode causar é o custo computacional.

2.5.2 **Métricas de Avaliação**

Existem diversas métricas para avaliar o desempenho dos modelos. Elas variam quando o modelo é de regressão ou classificação. Neste trabalho iremos focar nas métricas para modelos de classificação.

2.5.2.1 Matriz de Confusão

A matriz de confusão é uma tabela que nas colunas tem a previsão do modelo e nas linhas o resultado verdadeiro (atributo alvo). Essas informações mostrar os resultados classificados corretamente e os incorretos. Cada "caixa" da matriz de confusão recebe um nome, conforme mostra a Tabela 1:

Tabela 1 – Matriz de Confusão

	Classe Predita	
Classe Real	$\hat{Y} = 1$	$\hat{Y} = 0$
Y = 1	VP	FN
Y = 0	FP	VN

Fonte: Adaptado de Faceli et al. (2011)

A matriz da Tabela 1 mostra quatro indicadores:

- Verdadeiro Positivo(VP): modelo preditor classificou corretamente o 1.

- Falso Positivo(FP): modelo preditor classificou como 1 sendo na realidade 0.

- Falso Negativo(FN): modelo preditor classificou como 0 sendo na realidade 1.

- Verdadeiro Negativo(VN): modelo preditor classificou corretamente o 0.

Através desses 4 indicadores são geradas algumas métricas:

- Acurácia (AC): Mostra o acerto total do modelo (independente da classe). Pode não ser uma boa opção para dados desbalanceados.

$$AC = \frac{VP + VN}{n},$$

em que n é a soma de todos os indicadores da matriz: VP + VN + FN + FP. Igual o total de observações do conjunto de dados.

- Precisão (P): mede o quanto foi preciso na predição da classe positiva (neste exempo o 1). Ou seja, de todos classificados como positivos, quantos eram positivos reais (BRUCE; BRUCE, 2019).

$$P = \frac{VP}{VP + FP}$$

Recall (Revocação) (R): Também é conhecido como "sensibilidade". Mede o quanto o modelo consegue prever uma classe positiva real (neste exemplo 1). Ou seja, de todos os positivos reais, quantos o modelo acertou a predição (BRUCE; BRUCE, 2019).

$$R = \frac{VP}{VP + FN}$$

Especificidade (Es): Mede o quanto de negativos (neste exemplo 0) reais o modelo conseguiu classificar corretamente.

$$Es = \frac{VN}{VN + FP}$$

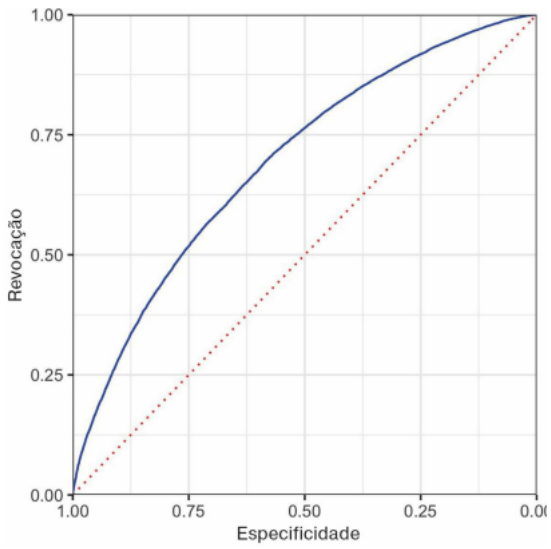
Medida F-1: Combina as medidas de Revocação e Precisão. É dado o mesmo grau de importância (FACELI et al., 2011).

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}$$

2.5.2.2 Curva ROC (*Receiving Operating Characteristics*)

A curca ROC (Característica Operátoria do Receptor) é representada em um espaço de duas dimensões, onde o eixo x é a medida de Especificidade e eixo y medida de Revocação. Para poder visualizar, a Figura 6 mostra o que é uma curva ROC:

Figura 6 – Exemplo de Curva ROC



Fonte: Bruce e Bruce (2019)

Na Figura 6 temos um exemplo de curva ROC. A linha diagonal representa qualquer classificador que tem probabilidades aleatórias. Quanto mais próximo do ponto (0,1), melhor será o classificador, pois significa que está acertando um grande número de observações positivas e negativas. Ao contrário disso, o ponto (1,0) mostra predições muito ruins. Quando os pontos são iguais, seja (0,0) ou (1,1), significa classificações sempre negativas e positivas, respectivamente (FACELI et al., 2011).

3 RESULTADOS

Neste capiítulo será mostrado quais atributos serão utilizados, as distribuições dos principais atributos e a aplicação das metodologias e métodos apresentados no capítulo 2.

3.1 Atributo Alvo

O atributo alvo é o evento que queremos ensinar ao algoritmo. Neste trabalho nosso atributo alvo é binário, a pontuação média do ENEM ser maior de 500 pontos ou não. O alvo será construído calculando a média entre as variáveis NU_NOTA_CN, NU_NOTA_CH, NU_NOTA_LC, NU_NOTA_MT e NU_NOTA_REDACAO. Mas antes de calcular a média, é importante fazer alguns filtros: deixar somente as observações em que os alunos compareceram em todas as provas e não deixaram a redação em branco ou teve ela anulada. Esse filtro é importante pois queremos saber a média geral da prova, se houve alguma falta ou anulação a média não será real.

Na Tabela 2 vemos a distribuição do status da redação dos alunos:

Tabela 2 – Distribuição status da redação

Status	**Quantidade**	**Percentual**
Sem problemas	3.779.455	74,18%
Nulo	1.172.126	23,00%
Em branco	56.901	1,12%
Fuga ao tema	40.623	0,80%
Cópia texto motivador	23.265	0,46%
Texto insuficiente	8.578	0,17%
Anulada	5.659	0,11%
Parte desconectada	4.863	0,10%
Não atendimento ao tipo textual	3.800	0,07%
Total	5.095.270	100%

Fonte: Elaborada pela autora

Na Tabela 3 temos o percentual de alunos que compareceram nas provas:

Tabela 3 – Percentual de alunos que compareceram

Situação	**CN**	**CH**	**LC**	**MT**
Presente	72,82%	77,00%	77,00%	72,82%
Faltou	27,14%	22,92%	22,92%	27,14%
Eliminado	00,04%	00,08%	00,08%	00,04%

Fonte: Elaborada pela autora

As provas de Ciências da Natureza (CN) e Matemática (MT) tem o mesmo percentual dado que são realizadas no mesmo dia. O mesmo raciocínio serve para as provas de Ciências Humanas (CH) e Linguagens e Códigos (LC).

Após realizar os filtros de alunos presentes em todas as provas e redações válidas (exceção nulos, em branco ou anuladas), foi possível calcular a pontuação média da prova. Sendo assim, as observações que tiveram mais ou 500 pontos, será o nosso atributo alvo (representado como 1), quem tiver menos de 500 pontos será o 0. A distribuição do atributo alvo é mostrado na Tabela 4:

Tabela 4 – Distribuição do atributo alvo

Alvo	**Quantidade**	**Percentual**
>= 500 pontos	2.125.397	58,04%
<500 pontos	1.536.581	41,96%
Total	3.661.978	100%

Fonte: Elaborada pela autora

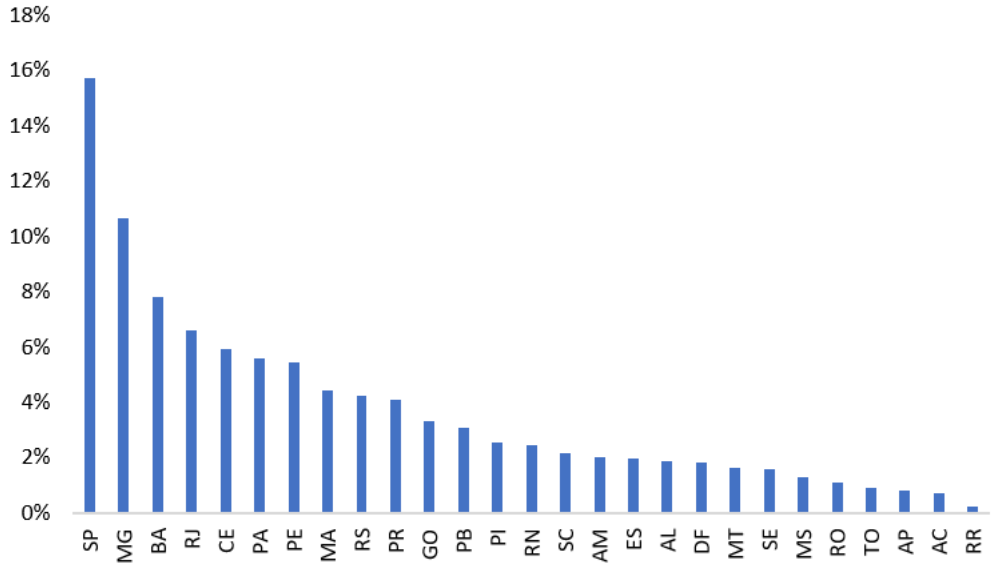
3.2 Distribuição dos Atributos Explicativos

Verificar a distribuição dos valores de cada atributo explicativo é uma parte muito importante antes de desenvolver o modelo. A exploração é feita para verificar se determinado atributo tem uma boa distribuição (que consiga discriminar) ou se está concentrado somente em certos valores. Isso é feito para evitar de colocar no modelo atribbutos que não ajudam no poder de predição.

A seguir será apresentado a distribuição dos atributos que estão disponíveis e foram escolhidos para serem utilizados. Os gráficos serão divididos em duas partes: a primeira é referente as informações dos alunos e a segunda parte é referente as informações socioeconômica do aluno e sua família.

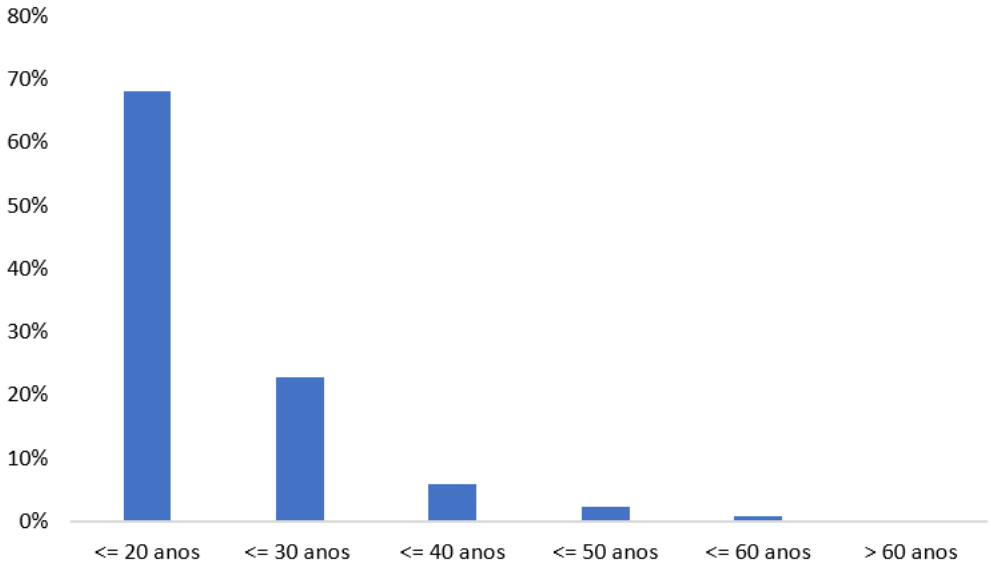
3.2.1 Informações dos Alunos

Figura 7 – Unidade da Federação de residência do aluno



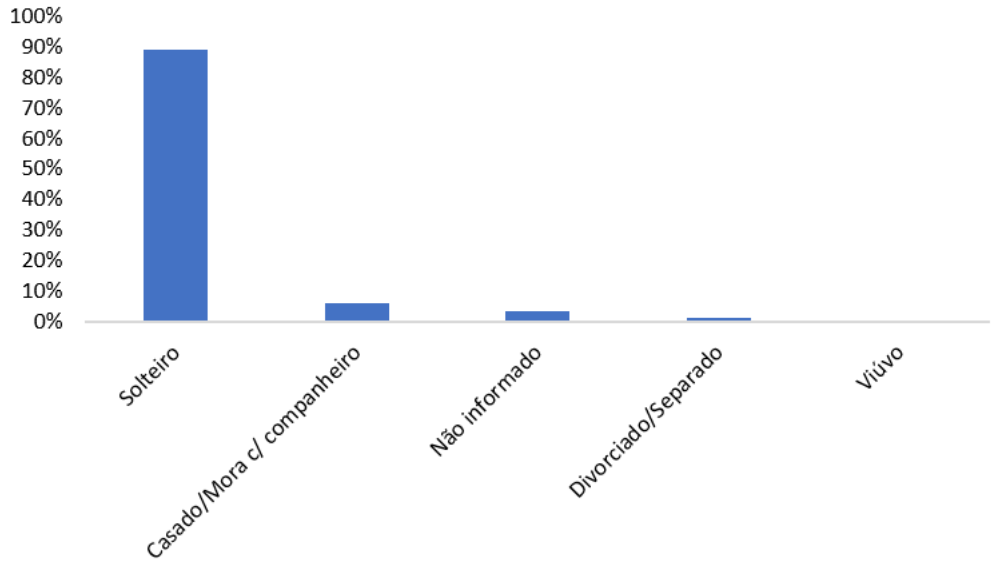
Fonte: Elaborada pela autora

Figura 8 – Idade dos Alunos



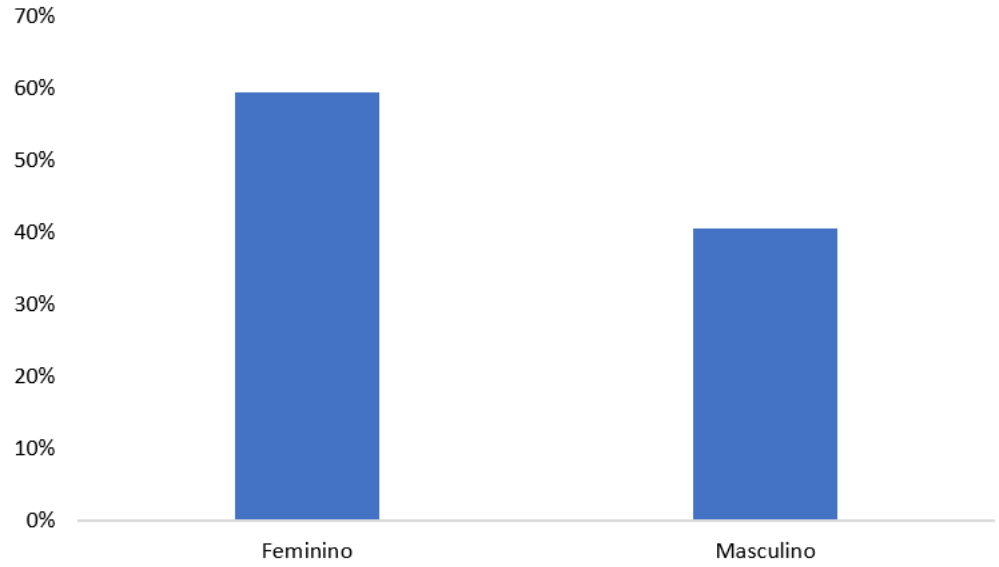
Fonte: Elaborada pela autora

Figura 9 – Estado Civil dos Alunos



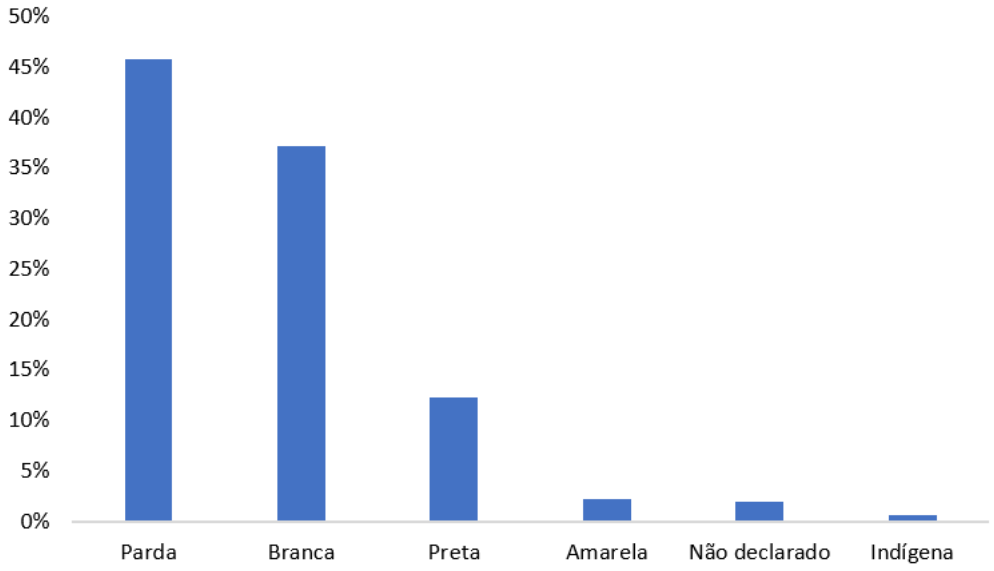
Fonte: Elaborada pela autora

Figura 10 – Sexo em que os alunos se identificam



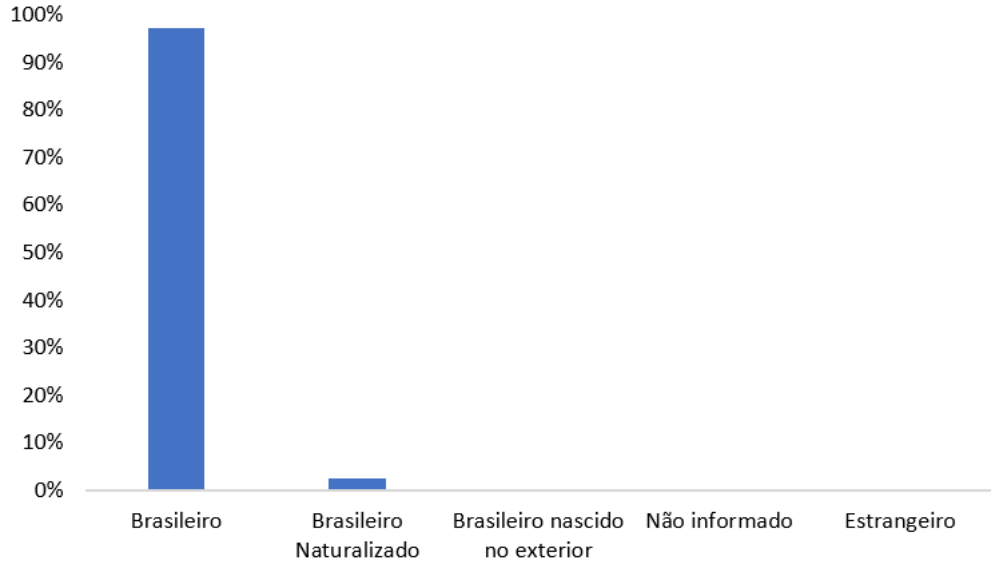
Fonte: Elaborada pela autora

Figura 11 – Cor/Raça em que os alunos se identificam



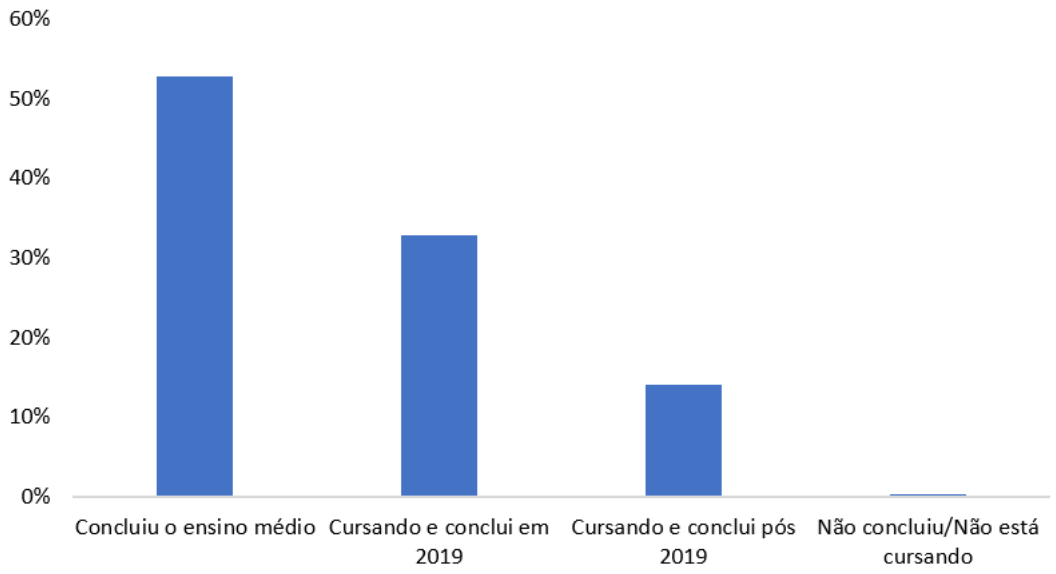
Fonte: Elaborada pela autora

Figura 12 – Nacionalidade dos Alunos



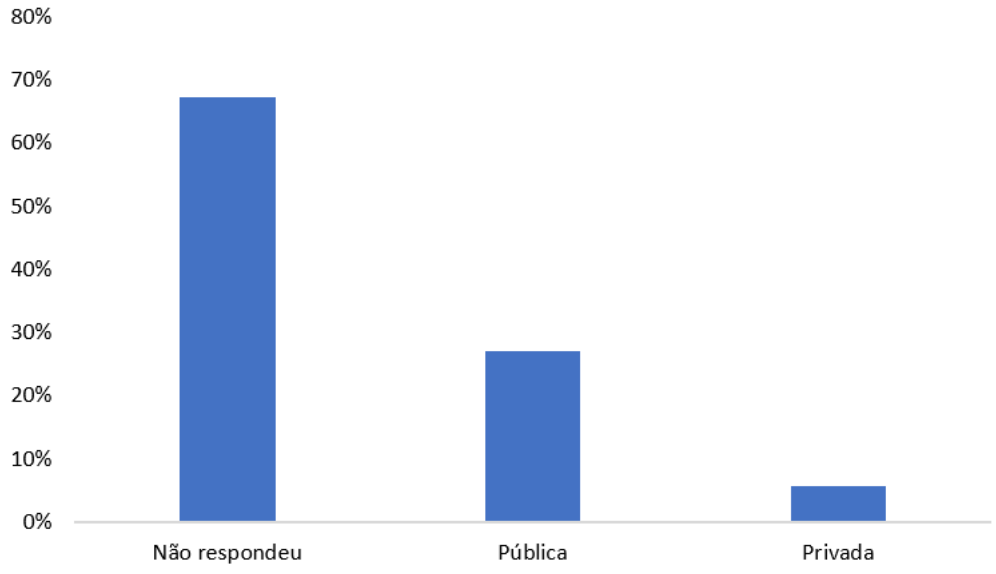
Fonte: Elaborada pela autora

Figura 13 – Tipo de conclusão do Ensino Médio



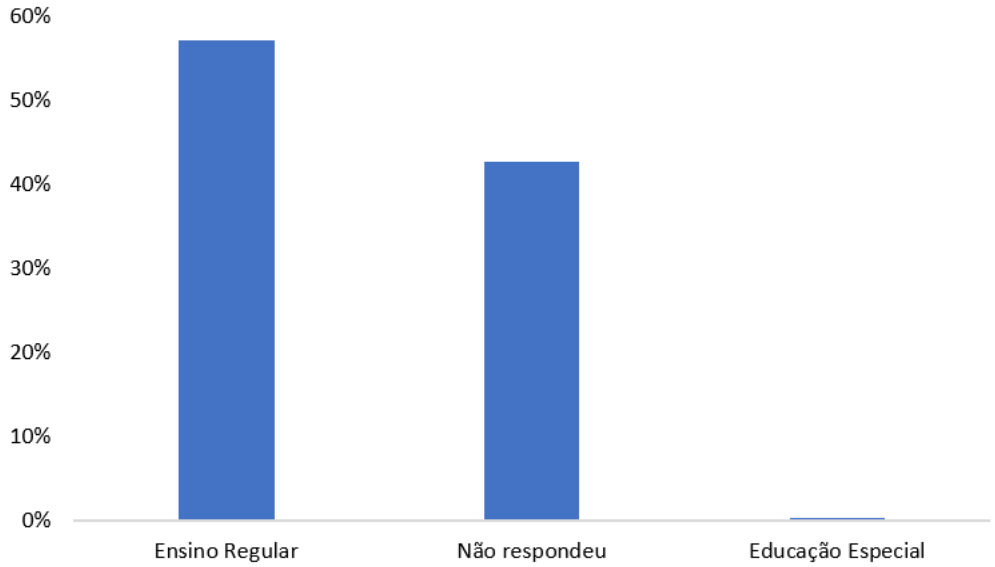
Fonte: Elaborada pela autora

Figura 14 – Tipo de escola em que o candidato estudou



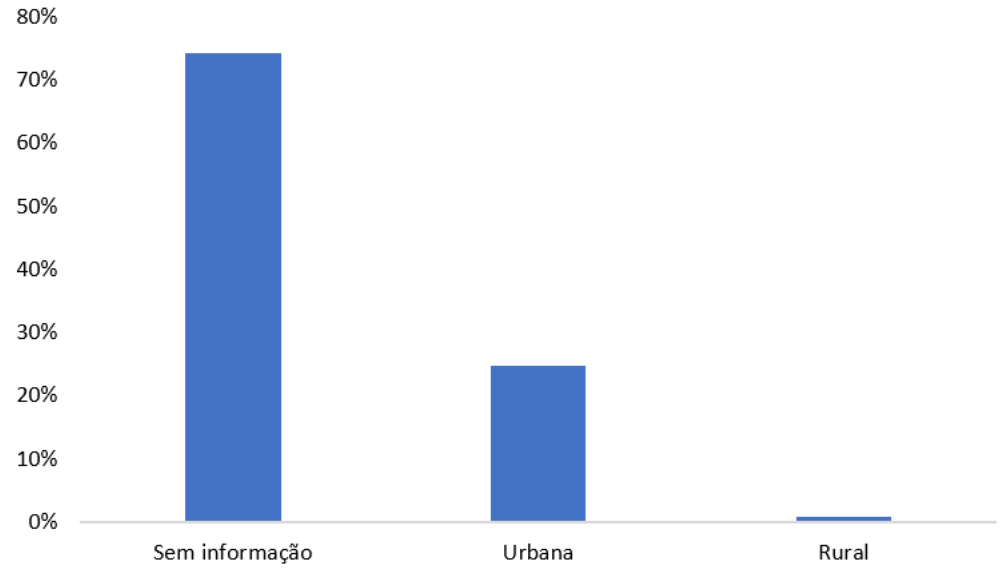
Fonte: Elaborada pela autora

Figura 15 – Tipo de ensino da escola em que o candidato estudou



Fonte: Elaborada pela autora

Figura 16 – Localização da escola em que o candidato estudou

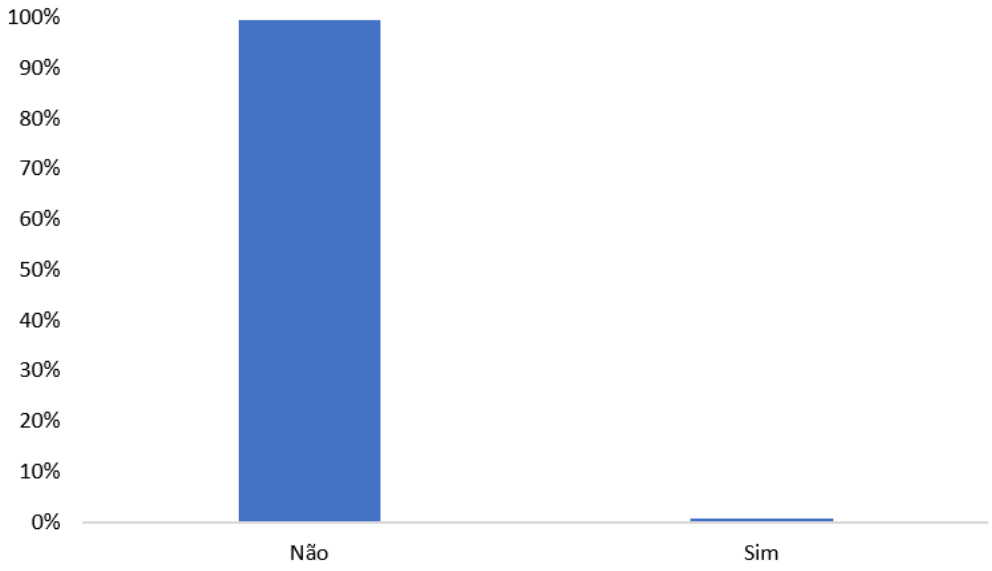


Fonte: Elaborada pela autora

A última variável explicativa sobre o aluno que será apresentada foi construída através de outras. As variáveis utilizadas foram: 'IN_BAIXA_VISAO', 'IN_SURDEZ', 'IN_DEFICIENCIA_AUDITIVA', 'IN_SURDO_CEGUEIRA', 'IN_DEFICI_FISICA', 'IN_DEFICI_MENTAL', 'IN_DEFICIT_ATENCAO', 'IN_DISLEXIA', 'IN_DISCALCULIA',

'IN_AUTISMO', 'IN_VISAO_MONOCULAR', 'IN_CEGUEIRA', 'IN_OUTRA_DEF'. Cada uma das variáveis indicava se o aluno tinha aquela deficiência especifica, então a variável criada considerou qualquer uma das deficiências.

Figura 17 – Indicador se o candidato possui alguma deficiência

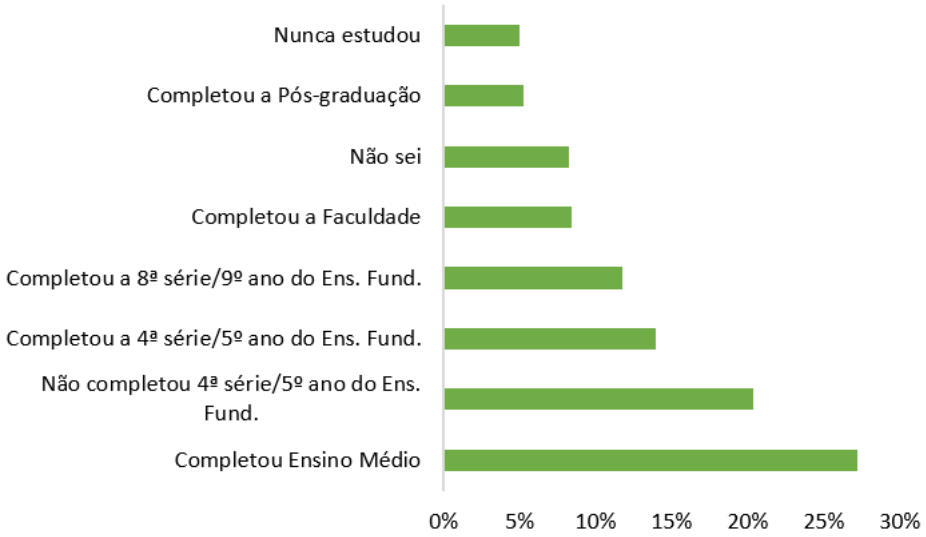


Fonte: Elaborada pela autora

Das onze variáveis apresentadas, algumas foram escolhidas para serem retiradas e não entrarem no modelo: Figura 9, Figura 12, Figura 15, Figura 16 e Figura 17, O motivo para serem retiradas é pelo fato que não mostram discriminar as informações nelas contidas.

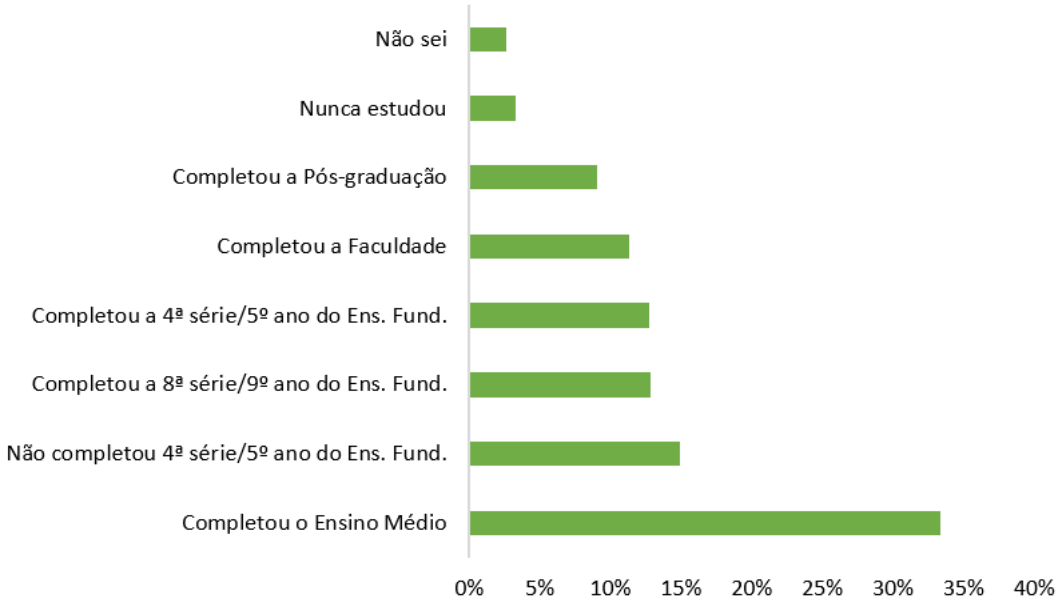
3.2.2 Informações Socioeconômicas dos Candidatos

Figura 18 – Série em que o pai/homem responsável pelo candidato estudou



Fonte: Elaborada pela autora

Figura 19 – Série em que a mãe/mulher responsável pelo candidato estudou



Fonte: Elaborada pela autora

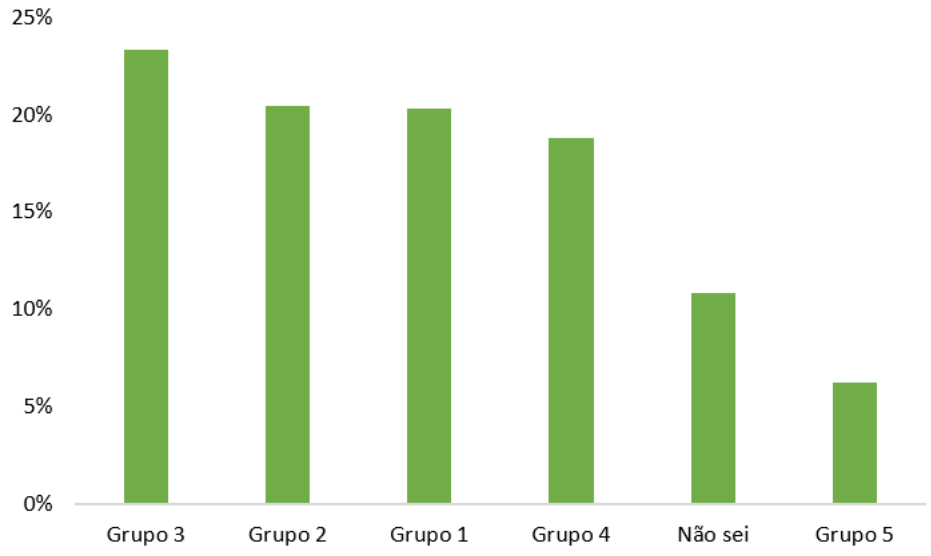
As duas próximas figuras representam "Grupos de Ocupações". Abaixo é detalhado quais ocupações pertencem a cada grupo:

- **Grupo 1:** "Lavrador, agricultor sem empregados, bóia fria, criador de animais (gado,

porcos, galinhas, ovelhas, cavalos etc.), apicultor, pescador, lenhador, seringueiro, extrativista.";

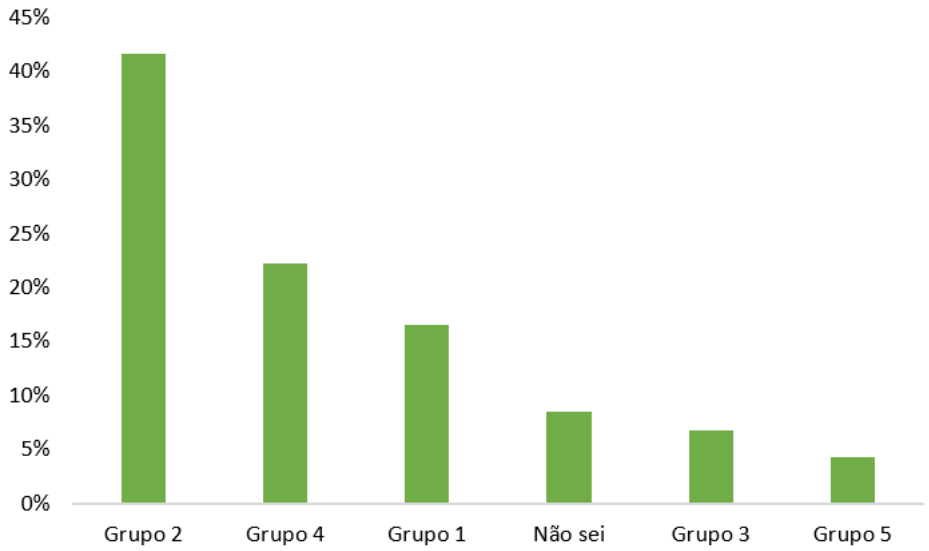
- **Grupo 2:** "Diarista, empregado doméstico, cuidador de idosos, babá, cozinheiro (em casas particulares), motorista particular, jardineiro, faxineiro de empresas e prédios, vigilante, porteiro, carteiro, office-boy, vendedor, caixa, atendente de loja, auxiliar administrativo, recepcionista, servente de pedreiro, repositor de mercadoria.";
- **Grupo 3:** "Padeiro, cozinheiro industrial ou em restaurantes, sapateiro, costureiro, joalheiro, torneiro mecânico, operador de máquinas, soldador, operário de fábrica, trabalhador da mineração, pedreiro, pintor, eletricista, encanador, motorista, caminhoneiro, taxista.";
- **Grupo 4:** "Professor (de ensino fundamental ou médio, idioma, música, artes etc.), técnico (de enfermagem, contabilidade, eletrônica etc.), policial, militar de baixa patente (soldado, cabo, sargento), corretor de imóveis, supervisor, gerente, mestre de obras, pastor, microempresário (proprietário de empresa com menos de 10 empregados), pequeno comerciante, pequeno proprietário de terras, trabalhador autônomo ou por conta própria."
- **Grupo 5:** "Médico, engenheiro, dentista, psicólogo, economista, advogado, juiz, promotor, defensor, delegado, tenente, capitão, coronel, professor universitário, diretor em empresas públicas ou privadas, político, proprietário de empresas com mais de 10 empregados."

Figura 20 – Grupo de ocupação em que o pai/homem responável pelo candidato trabalha/trabalhou



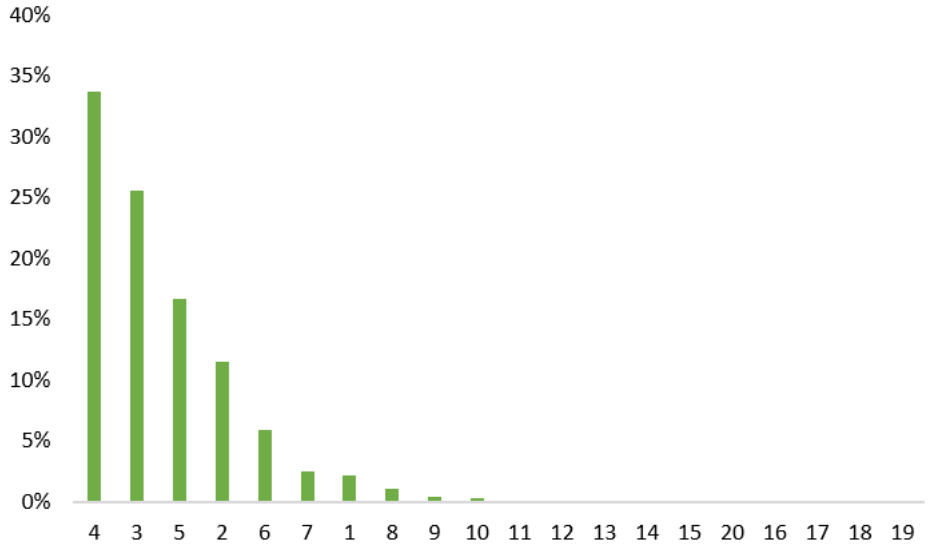
Fonte: Elaborada pela autora

Figura 21 – Grupo de ocupação em que o mãe/mulher responável pelo candidato trabalha/trabalhou



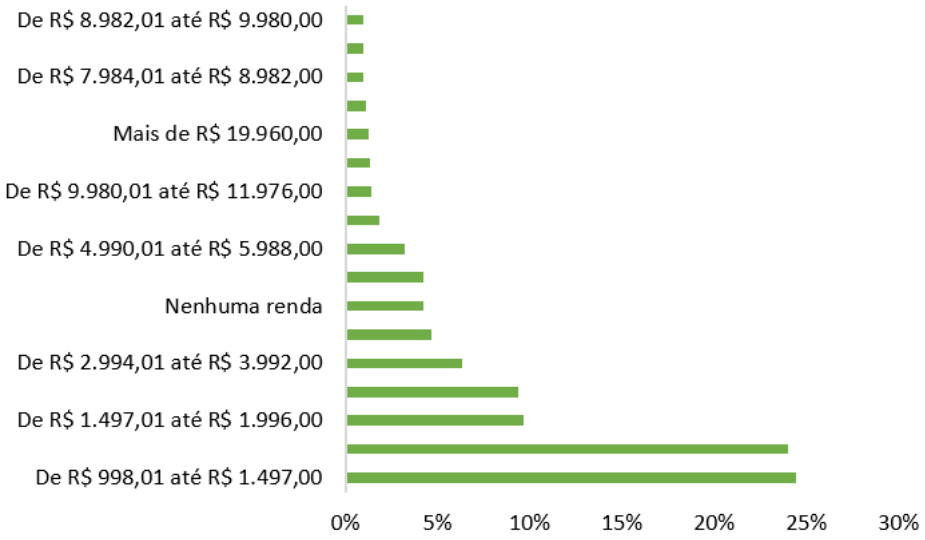
Fonte: Elaborada pela autora

Figura 22 – Quantidade de pessoas que moram na residência (incluindo candidato)



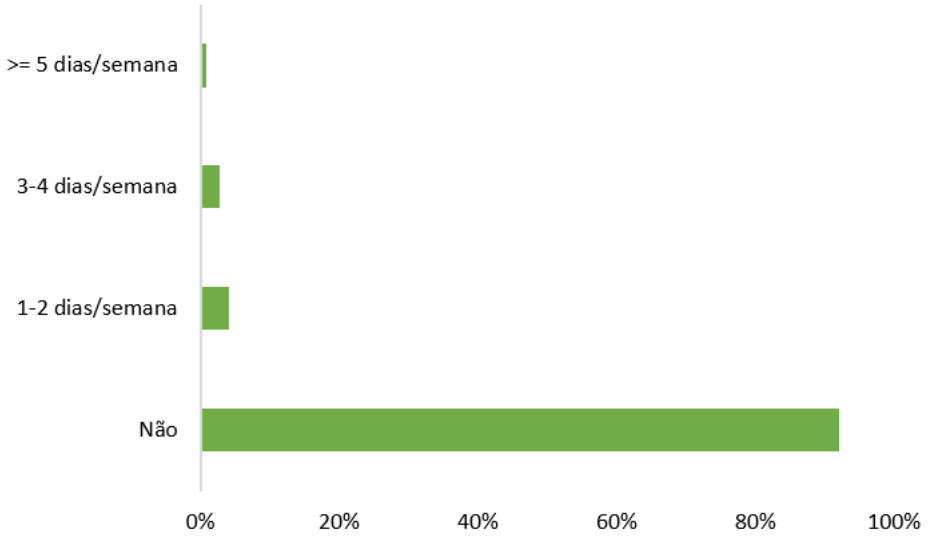
Fonte: Elaborada pela autora

Figura 23 – Renda mensal da família do candidato



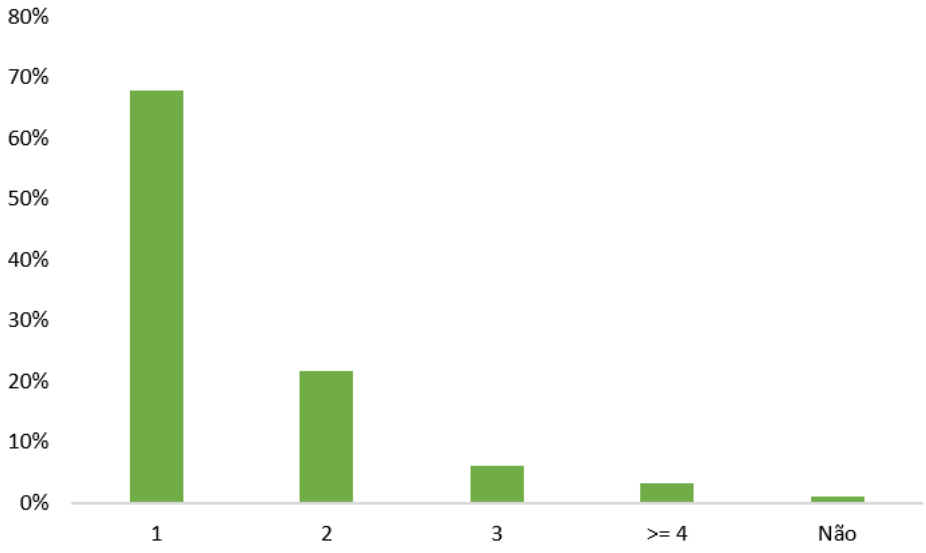
Fonte: Elaborada pela autora

Figura 24 – Empregado(a) doméstico(a) que trabalham na residência do candidato



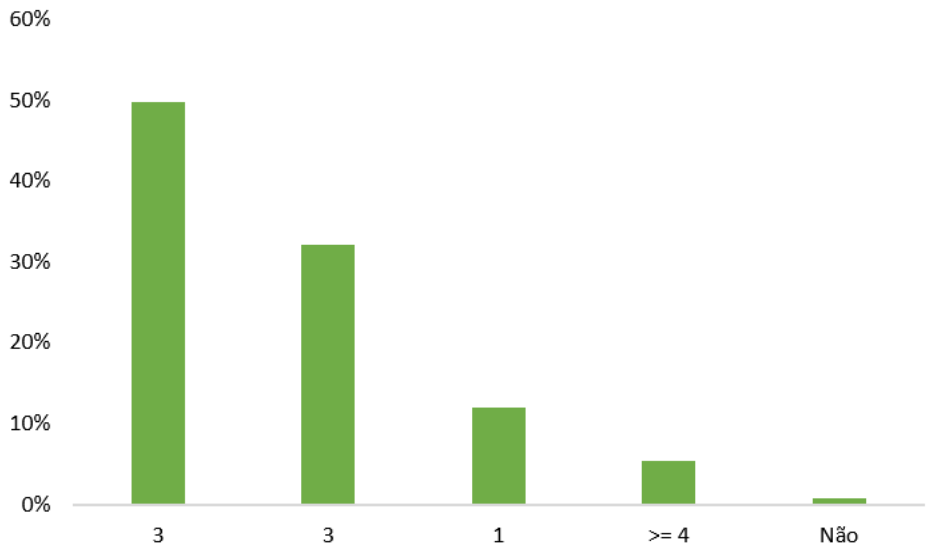
Fonte: Elaborada pela autora

Figura 25 – Quantidade de banheiros na residência do candidato



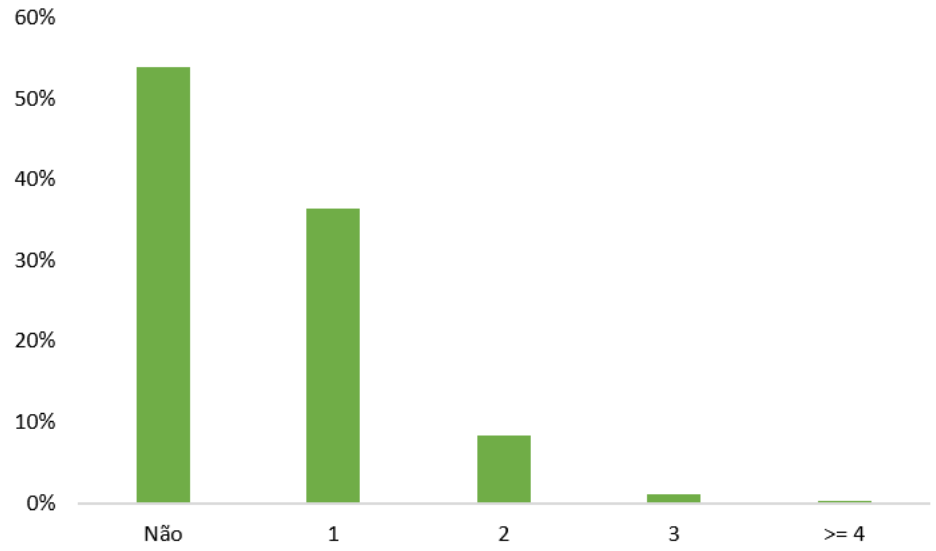
Fonte: Elaborada pela autora

Figura 26 – Quantidade de quartos na residência do candidato



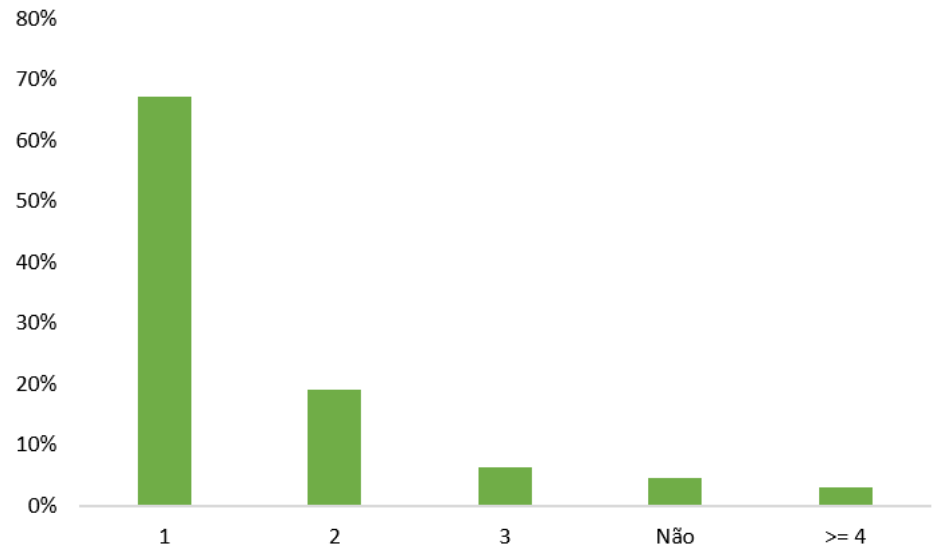
Fonte: Elaborada pela autora

Figura 27 – Quantidade de carros na residência do candidato



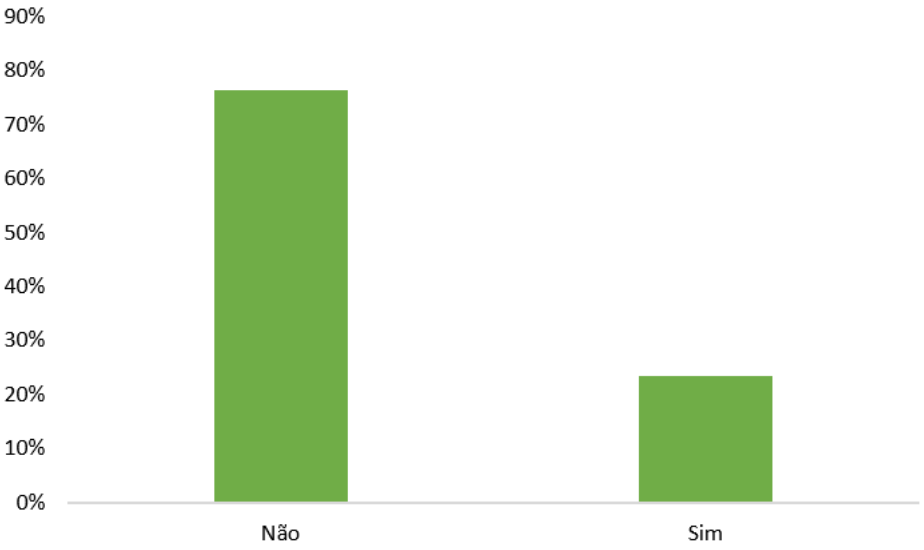
Fonte: Elaborada pela autora

Figura 28 – Quantidade de TVs (em cores) na residência do candidato



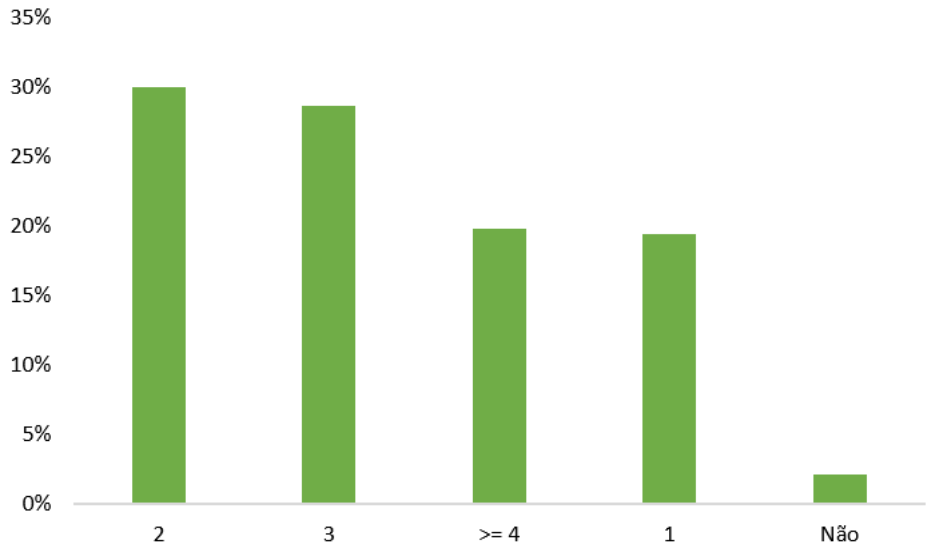
Fonte: Elaborada pela autora

Figura 29 – Indicador de TV por assinatura na residência do candidato



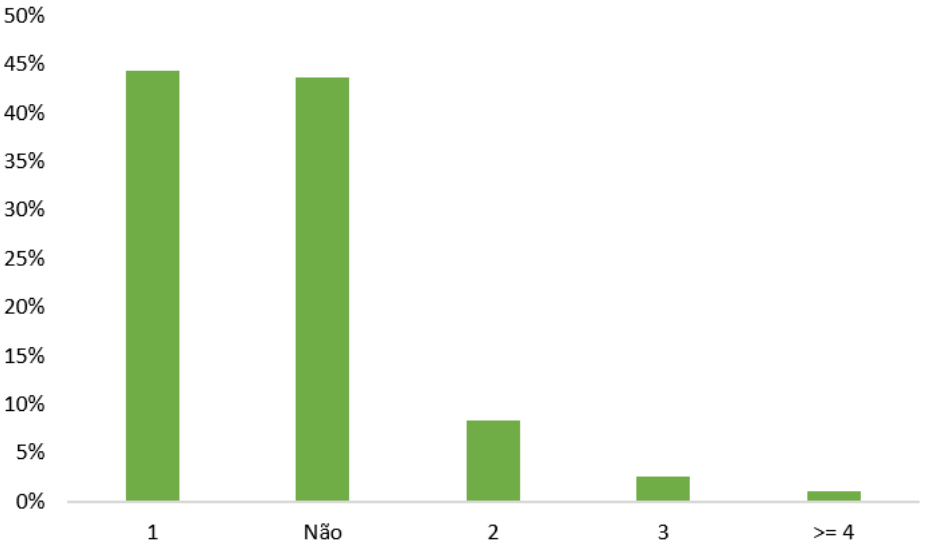
Fonte: Elaborada pela autora

Figura 30 – Quantidade celulares na residência do candidato



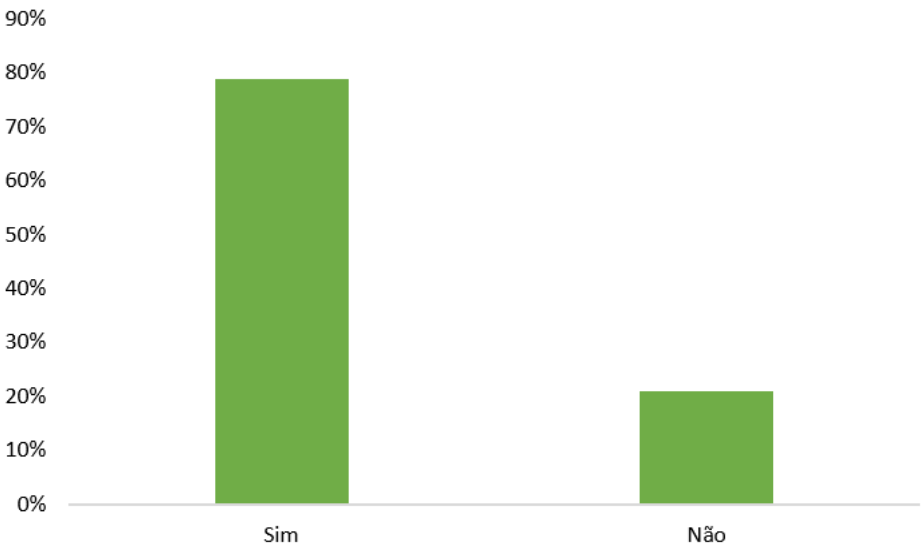
Fonte: Elaborada pela autora

Figura 31 – Quantidade computadores na residência do candidato



Fonte: Elaborada pela autora

Figura 32 – Indicador de acesso a internet na residência do candidato



Fonte: Elaborada pela autora

Das quinze variáveis apresentadas, todas serão testadas para usar no modelo, dado que apresentam boas distribuições nos valores.

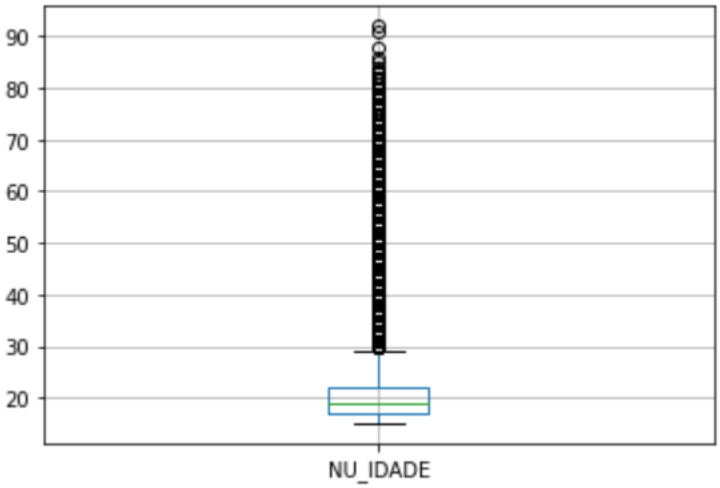
3.3 Limpeza e Tratamento

Após analisar as distribuições das variáveis explicativas foram encontradas oportunidades de limpeza da base. O primeiro filtro feito foi retirar os valores *missings* (nulos) em 'NU_IDADE', onde possuia 60 valores *missings*. Neste caso, como a base de dados tem muitos casos, retirar os *missings* não é um problema, mas se a base de dados fosse pequena o ideal seria preencher esses dados faltantes usando alguma técnica como média ou mediana por exemplo. As outras variáveis estavam 100% preenchidas. Além disso, é importante avaliar os *outliers* que podem constar nas variáveis. Entre todas, duas chamaram a atenção em relação a *outliers*:

- Idade do candidato: analisando os valores das idades foram encontrados dois tipos de *outliers*: contextuais e globais. Os contextuais são casos onde o valor preenchido não faz sentido com o significado da variável ou não faz sentido com a situação como um todo. No caso da idade, foram encontrados valores abaixo de 15 anos, o que neste caso em específico não encaixa muito bem dado que é uma idade muito nova para estar no ensino médio e poder prestar o ENEM, mesmo que seja somente como treino. Se encaixavam nessa situação estavam 70 casos que foram removidos da base. Já os *outliers* globais são casos em que o valor distoa muito da distribuição dos valores e ocorre com uma frequência baixa. Para auxiliar a identificar estes casos

usamos o boxplot. A Figura 33 representa o boxplot da variável 'NU_IDADE' (já com o filtro de idade abaixo de 15 anos):

Figura 33 – Box-plot da idade do candidato

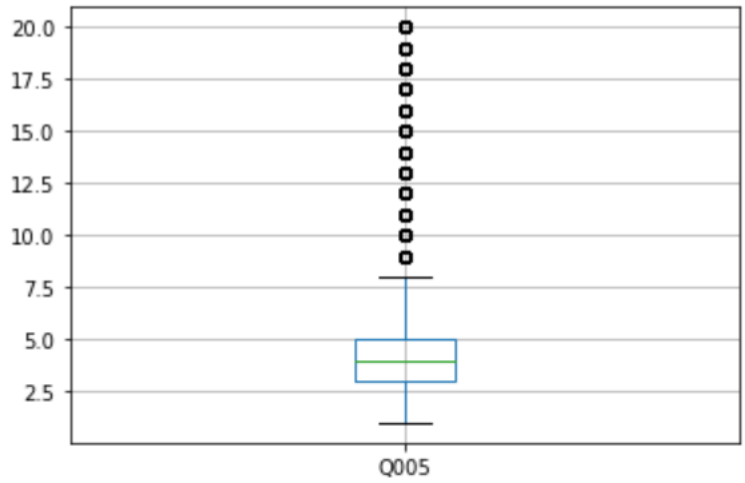


Fonte: Elaborada pela autora

Geralmente, todos os pontos pretos podem ser interpretados como *outliers*. Neste caso, somente os pontos que representam acima de 70 anos serão retirados, dado que começam a ter um espaçamento maior e se retirarmos todos os pontos pretos (acima de 30 anos) perdemos a variabilidade da variável. Somente 542 casos estão acima de 70 anos.

- Quantidade de pessoas que moram na residência: Para esta variável não foram encontrados preenchimentos que não estivessem inseridos no contexto, mas foram encontrados *outliers* globais. Na Figura 34 temos o boxplot da variável:

Figura 34 – Box-plot da quantidade de moradores na residência do canditato



Fonte: Elaborada pela autora

Neste caso, não temos muito pontos pretos, mas não serão retirados todos pois senão muitos casos serão perdidos. Para retirar os *outliers*, mas tentar preservar a quantidade de dados, os casos a partir de 15 moradores na residência serão retirados. Nesta caso, 1342 observações se encaixam no filtro.

Após os filtros retirando casos *missings* e *outliers* de duas variáveis, a distribuição do atributo alvo é apresentada na Tabela 5:

Tabela 5 – Distribuição do atributo alvo - pós filtros

Alvo	**Quantidade**	**Percentual**
>= 500 pontos	2.122.042	58,05%
<500 pontos	1.533.666	41,95%
Total	3.655.708	100%

Fonte: Elaborada pela autora

Outra informação importante que as distribuições nos mostram é o tipo de variável (numérica ou texto). A grande parte estão em texto e são preenchidas por categorias diferentes. É preciso tratar essas variáveis pois os algoritmos só conseguem gerar um modelo com variáveis numéricas. Abauxo segue os tratamentos que cada variável irá receber:

- Unidade de Federação: As unidades de federação serão agrupadas em 5 regiões: Norte, Nordeste, Sul, Sudeste e Centro-Oeste. Para as 5 regiões será aplicado a técnica *One-hot-enconding* (seção 2.3.5);
- Idade: já está no formato numérico mas a escala será alterada, a idade será dividida por 10. Isso é feito para a escala ficar mais próxima das outras variáveis;

- Sexo: transformação em flag, 1 sendo sexo feminino e 0 masculino;
- Cor/Raça: aplicação da técnica *One-hot-enconding*;
- Tipo de conclusão do EM: aplicação da técnica *One-hot-enconding*;
- Tipo de escola que estudou: aplicação da técnica *One-hot-enconding*;
- Série em que pai/homem responsável estudou: aplicação da técnica *One-hot-enconding*;
- Série em que mãe/mulher responsável estudou: aplicação da técnica *One-hot-enconding*;
- Grupo de ocupação do pai/homem responsável: aplicação da técnica *One-hot-enconding*;
- Grupo de ocupação do mãe/mulher responsável: aplicação da técnica *One-hot-enconding*;
- Quantidade de moradores na residência: sem tratamento, já está no formato numérico;
- Renda mensal: para renda, será utilizado o valor da faixa inferior. Po exemplo: para a faixa R$ 998,01 até R$ 1.497,00, o valor será R$ 998. Para o caso onde não tem renda será utilizado 0. O valor já estará no formato numérico mas a escala será alterada, a renda será dividida por 1000. Isso é feito para a escala ficar mais próxima das outras variáveis;;
- Empregado(a) doméstico(a): será utilizado o valor da faixa inferior. Para o caso onde a resposta for 'não' será utilizado 0;
- Quantidade de banheiros: será utilizado o número de banheiros. Para o caso de mais de 4 banheiros, será utilizado somente o 4. Para o caso onde a resposta for 'não' será utilizado 0;
- Quantidade de quartos: será utilizado o número de quartos. Para o caso de mais de 4 quartos, será utilizado somente o 4. Para o caso onde a resposta for 'não' será utilizado 0;
- Quantidade de carros: será utilizado o número de carros. Para o caso de mais de 4 carros, será utilizado somente o 4. Para o caso onde a resposta for 'não' será utilizado 0;
- Quantidade de TVs: será utilizado o número de TVs. Para o caso de mais de 4 TVs, será utilizado somente o 4. Para o caso onde a resposta for 'não' será utilizado 0;
- Indicador de TV por assinatura: transformação em flag, 1 sendo que possui TV por assinatura e 0 caso constrário;

- Quantidade de celulares: será utilizado o número de celulares. Para o caso de mais de 4 celulares, será utilizado somente o 4. Para o caso onde a resposta for 'não' será utilizado 0;
- Quantidade de computadores: será utilizado o número de computadores. Para o caso de mais de 4 computadores, será utilizado somente o 4. Para o caso onde a resposta for 'não' será utilizado 0;
- Indicador de acesso a internet: transformação em flag, 1 sendo que possui acesso a internet e 0 caso constrário;

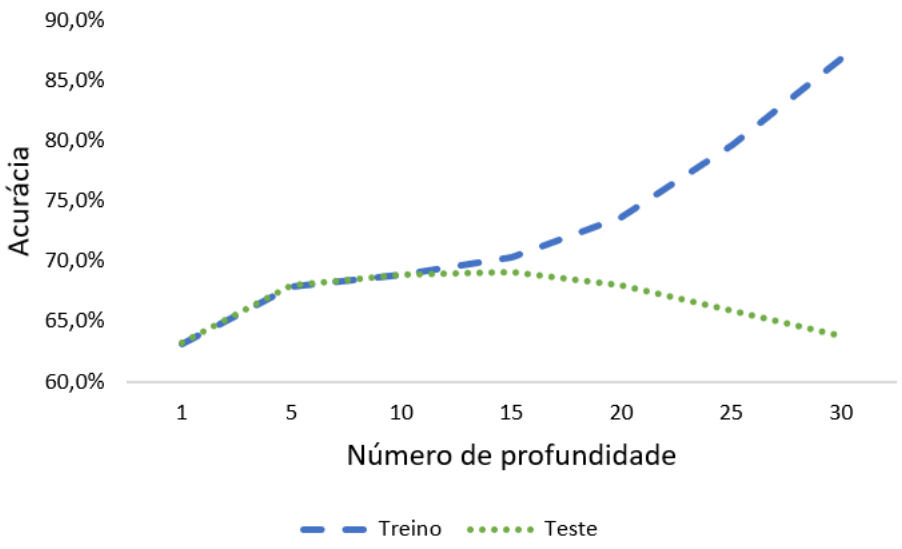
3.4 Desenvolvimento do Modelo

Dado a grande quantidade de dados que temos na base, para dividir em treino e teste será utilizado *Hould-out Validation*, onde 70% vai ser utilizado para treinar o modelo e 30% para testar o modelo e verificar seu desempenho.

3.4.1 Árvore de Decisão

Para desenvolver o modelo de classificação de árvore de decisão foi utilizado a biblioteca *sklearn-tree.DecisionTreeClassifier* no python. É possível alterar diversos parâmetros que podem influenciar o resultado final do modelo. Neste trabalho foram testados diferentes valores para dois parâmetros: *max_depth* (profundidade máxima que a árvore pode ter) e *max_leaf_nodes* (número máximo de nós que a árvore pode ter). A Figura 35 e 36 mostra a acurácia para diferentes valores de profundidade e nós respectivamente:

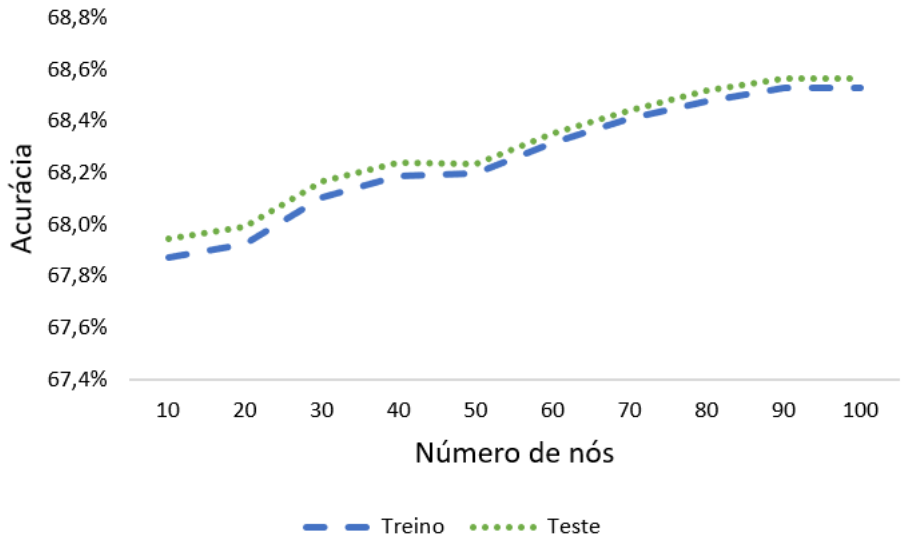
Figura 35 – Número máximo de profundidade x Acurácia



Fonte: Elaborada pela autora

Na Figura 35 podemos ver que a medida que aumentamos a quantidade de níveis de profundidade da árvore, a acurácia do treino aumenta, mas após o nível 15 a acurácia do teste não acompanha a do treinamento. Isso ocorre porque o modelo esta sofrendo com *overffiting*.

Figura 36 – Número máximo de nós x Acurácia



Fonte: Elaborada pela autora

Na Figura 36 vemos que ao aumentar o número de nós a acurácia aumenta, do treinamento e teste. Mas a partir da quantidade 50 o aumento da acurácia é bem baixo, pouco relevante.

Como o objetivo deste trabalho é encontrar a maior acurácia possível, foi escolhido limitar a árvore em no máximo 15 níveis de profundidade. Isso significa que o nosso critério de parada é a profunfidade da árvore. Mas se o objetivo fosse desenvolver um modelo com a menor complexidade seria possível limitar a quantidade de nós. Pois que quanto menor a quantidade de nós, menor a profundidade e complexidade da árvore e ao mesmo tempo estaria se perdendo somente 1% de acurácia. Além disso, o critério utilizado foi entropia. A acurácia do treino foi de 70,3% e no teste 69,1%, o que é um bom sinal pois significa que o modelo não sofreu com *overffiting*. Se o modelo tivesse sofrido *overffiting* haveria uma grande queda de desempenho na avaliação do teste (como mostrado na figura 35). Abaixo serão exploradas outras métricas de avaliação e a importância de cada atributo explicativo.

A Tabela 6 mostra como fica a matriz de confusão para os dados de teste (30% da base - 1.096.713 casos):

Tabela 6 – Matriz de Confusão - Árvore de Decisão

	Classe Predita	
Classe Real	Alvo = 0	Alvo = 1
Alvo = 0	25,2%	16,7%
Alvo = 1	14,2%	43,8%

Fonte: Elaborada pela autora

A Tabela 7 mostra as métricas: precisão, revocação, especificidade e medida F-1:

Tabela 7 – Métricas - Árvore de Decisão

Métrica	**%**
Precisão	72,4%
Revocação	75,5%
Especificidade	60,2%
Medida F-1	73,9%

Fonte: Elaborada pela autora

A Tabela 8 mostra a importância de cada variável explicativa na árvore de decisão:

Tabela 8 – Importância das variáveis - Árvore de Decisão

Variável Explicativa	**% de Importância**
Renda Familiar	51,3%
Quantidade de computadores na residência	13,0%
Idade do candidato	6,5%
Grupo de ocupação do pai/homem responsável	5,5%
Situação de conclusão do EM	2,9%
Quantidade de pessoas que moram na residência	2,7%
Série que a mãe/mulher responsável estudou	2,7%
Região do país que o candidato mora	2,0%
Cor/Raça do candidato	2,0%
Tipo de escola do EM	1,9%
Série que a pai/homem responsável estudou	1,8%
Quantidade de celulares na residência	1,7%
Grupo de ocupação do mãe/mulher responsável	1,3%
Sexo do candidato	1,3%
Serviço de empregado(a) doméstico(a)	0,7%
Quantidade de banheiros na residência	0,6%
Quantidade de TVs na residência	0,6%
Quantidade de quartos na residência	0,6%
Quantidade de carros na residência	0,5%
Indicador de TV por assinatura na residência	0,3%
Indicador de acesso a internet na residência	0,3%

Fonte: Elaborada pela autora

3.4.2 Regressão Logística

Para criar o modelo de regressão logística foi usada a biblioteca *statsmodels* - *GLM* do python. O parâmetro *family* utilizado foi o "Binomial". Ao contrário dos outros algoritmos de AM que será mostrado neste trabalho, adicionamos uma variável 'const' (preenchida com número 1). Isso foi feito para o modelo ser gerado com um intercepto. A Figura 37 mostra o resultado da regressão logística e logo após as métricas de desempenho serão apresentadas.

Figura 37 – Resultado da regressão logística

	coef	std err	z	P>|z|	[0.025	0.975]
const	18.7869	1.77e+04	0.001	0.999	-3.47e+04	3.48e+04
IDADE	-0.2848	0.002	-114.645	0.000	-0.290	-0.280
SEXO	-0.1649	0.003	-56.559	0.000	-0.171	-0.159
Q005	-0.1058	0.001	-95.895	0.000	-0.108	-0.104
renda_a	0.1531	0.001	133.389	0.000	0.151	0.155
emp_dom	-0.0917	0.002	-39.830	0.000	-0.096	-0.087
banheiro	0.1205	0.003	39.059	0.000	0.114	0.127
quartos	-0.0258	0.002	-10.936	0.000	-0.030	-0.021
carros	0.0155	0.003	5.321	0.000	0.010	0.021
tvs	-0.0182	0.003	-6.961	0.000	-0.023	-0.013
Q021_2	-0.0703	0.004	-17.182	0.000	-0.078	-0.062
celular	0.0921	0.002	54.447	0.000	0.089	0.095
computador	0.3856	0.003	144.579	0.000	0.380	0.391
Q025_2	0.1576	0.004	40.882	0.000	0.150	0.165
reg_uf_CO	-0.1893	0.007	-27.705	0.000	-0.203	-0.176
reg_uf_NO	-0.2213	0.007	-33.515	0.000	-0.234	-0.208
reg_uf_NOD	0.0320	0.006	5.786	0.000	0.021	0.043
reg_uf_SUD	0.1547	0.005	29.498	0.000	0.144	0.165
cor_raca_A	0.6582	0.021	30.848	0.000	0.616	0.700
cor_raca_B	0.7202	0.019	37.941	0.000	0.683	0.757
cor_raca_C	0.4269	0.019	22.276	0.000	0.389	0.464
cor_raca_D	0.4743	0.019	25.133	0.000	0.437	0.511
cor_raca_E	0.5104	0.021	24.400	0.000	0.469	0.551
st_conclu_A	1.2723	0.034	37.762	0.000	1.206	1.338
st_conclu_B	-19.0095	1.77e+04	-0.001	0.999	-3.48e+04	3.47e+04
st_conclu_C	0.7158	0.034	21.134	0.000	0.649	0.782
tipo_em_A	-20.7205	1.77e+04	-0.001	0.999	-3.48e+04	3.47e+04
tipo_em_B	-0.8748	0.010	-90.448	0.000	-0.894	-0.856
Q001_A	-0.1843	0.009	-20.654	0.000	-0.202	-0.167
Q001_B	-0.0668	0.006	-10.392	0.000	-0.079	-0.054
Q001_C	0.0079	0.007	1.189	0.235	-0.005	0.021
Q001_D	0.0351	0.007	5.144	0.000	0.022	0.049
Q001_E	0.1327	0.006	21.279	0.000	0.120	0.145
Q001_F	0.3514	0.009	39.578	0.000	0.334	0.369
Q001_G	0.1925	0.012	16.589	0.000	0.170	0.215
Q002_A	0.1052	0.013	8.228	0.000	0.080	0.130
Q002_B	0.2978	0.010	29.560	0.000	0.278	0.318
Q002_C	0.3779	0.010	37.598	0.000	0.358	0.398
Q002_D	0.3973	0.010	39.637	0.000	0.378	0.417
Q002_E	0.5286	0.010	54.919	0.000	0.510	0.547
Q002_F	0.7063	0.011	64.882	0.000	0.685	0.728
Q002_G	0.6049	0.012	51.493	0.000	0.582	0.628
Q003_A	-0.0240	0.006	-3.804	0.000	-0.036	-0.012
Q003_B	0.1034	0.006	18.525	0.000	0.092	0.114
Q003_C	0.1729	0.005	31.471	0.000	0.162	0.184
Q003_D	0.4133	0.006	65.469	0.000	0.401	0.426
Q003_E	0.2343	0.011	20.865	0.000	0.212	0.256
Q004_A	0.0157	0.007	2.269	0.023	0.002	0.029
Q004_B	0.0942	0.005	17.206	0.000	0.083	0.105
Q004_C	0.1604	0.007	22.047	0.000	0.146	0.175
Q004_D	0.2402	0.007	36.785	0.000	0.227	0.253
Q004_E	0.1778	0.013	13.768	0.000	0.152	0.203

Fonte: Elaborada pela autora

Como o resultado da regressão logística é uma pontuação (entre 0 e 1) que indica a probabilidade de um caso ser ou não alvo, para ser considerado alvo (igual a 1) bastava a pontuação ser maior que 50%. A Tabela 9 mostra como fica a matriz de confusão para os dados de teste (30% da base - 1.096.713 casos):

Tabela 9 – Matriz de Confusão - Regressão Logística

	Classe Predita	
Classe Real	Alvo = 0	Alvo = 1
Alvo = 0	25,2%	16,8%
Alvo = 1	13,8%	44,3%

Fonte: Elaborada pela autora

A Tabela 10 mostra as métricas: precisão, revocação, especificidade e medida F-1:

Tabela 10 – Métricas - Regressão Logística

Métrica	**%**
Precisão	72,6%
Revocação	76,3%
Especificidade	60,1%
Medida F-1	74,4%

Fonte: Elaborada pela autora

Além destas métricas, a acurácia do treino foi de 69,4% e do teste 69,5%, o que comprova que o modelo não teve problema de *overffiting*. Mas pela figura 37 vemos na coluna P>|z| os valores de significância para cada variável explicativa no modelo. As variáveis "st_conclu_B", "tipo_em_A" e "Q001_C" são categorias das variáveis "st_conclu", "tipo_em" e "Q001" que aparecem com um alto valor de P>|z|, ou seja, não são significantes para o modelo e não ajudam a fazer a previsão. Dado isto, será testado um segundo modelo, retirando estas três variáveis. A Figura 38 mostra o resultado da regressão logística sem as três variáveis:

Figura 38 – Resultado da regressão logística (modelo 2)

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.1685	0.023	-50.972	0.000	-1.213	-1.124
IDADE	-0.1539	0.002	-71.016	0.000	-0.158	-0.150
SEXO	-0.1713	0.003	-59.228	0.000	-0.177	-0.166
Q005	-0.1128	0.001	-102.870	0.000	-0.115	-0.111
renda_a	0.1612	0.001	141.474	0.000	0.159	0.163
emp_dom	-0.0961	0.002	-42.157	0.000	-0.101	-0.092
banheiro	0.1267	0.003	41.489	0.000	0.121	0.133
quartos	-0.0328	0.002	-14.009	0.000	-0.037	-0.028
carros	0.0019	0.003	0.656	0.512	-0.004	0.008
tvs	-0.0174	0.003	-6.721	0.000	-0.023	-0.012
Q021_2	-0.0718	0.004	-17.713	0.000	-0.080	-0.064
celular	0.0985	0.002	58.713	0.000	0.095	0.102
computador	0.4044	0.003	152.803	0.000	0.399	0.410
Q025_2	0.1661	0.004	43.365	0.000	0.159	0.174
reg_uf_CO	-0.1902	0.007	-28.025	0.000	-0.204	-0.177
reg_uf_NO	-0.1764	0.007	-26.920	0.000	-0.189	-0.164
reg_uf_NOD	0.0619	0.005	11.313	0.000	0.051	0.073
reg_uf_SUD	0.1766	0.005	33.867	0.000	0.166	0.187
cor_raca_A	0.6392	0.021	30.082	0.000	0.598	0.681
cor_raca_B	0.7219	0.019	38.132	0.000	0.685	0.759
cor_raca_C	0.4359	0.019	22.809	0.000	0.398	0.473
cor_raca_D	0.4724	0.019	25.096	0.000	0.436	0.509
cor_raca_E	0.5150	0.021	24.708	0.000	0.474	0.556
Q002_A	0.0292	0.012	2.458	0.014	0.006	0.053
Q002_B	0.3073	0.009	32.783	0.000	0.289	0.326
Q002_C	0.4072	0.009	43.229	0.000	0.389	0.426
Q002_D	0.4399	0.009	46.741	0.000	0.421	0.458
Q002_E	0.6091	0.009	67.741	0.000	0.591	0.627
Q002_F	0.8247	0.010	80.276	0.000	0.805	0.845
Q002_G	0.7017	0.011	62.926	0.000	0.680	0.724
Q003_A	-0.0717	0.006	-12.169	0.000	-0.083	-0.060
Q003_B	0.1149	0.005	22.179	0.000	0.105	0.125
Q003_C	0.1741	0.005	34.218	0.000	0.164	0.184
Q003_D	0.4988	0.006	85.173	0.000	0.487	0.510
Q003_E	0.3836	0.010	37.030	0.000	0.363	0.404
Q004_A	0.0096	0.007	1.399	0.162	-0.004	0.023
Q004_B	0.0903	0.005	16.704	0.000	0.080	0.101
Q004_C	0.1416	0.007	19.656	0.000	0.127	0.156
Q004_D	0.2265	0.006	35.126	0.000	0.214	0.239
Q004_E	0.1670	0.013	13.069	0.000	0.142	0.192

Fonte: Elaborada pela autora

A Tabela 11 mostra como fica a matriz de confusão para os dados de teste do segundo modelo da regressão:

Tabela 11 – Matriz de Confusão - Regressão Logística (modelo 2)

	Classe Predita	
Classe Real	Alvo = 0	Alvo = 1
Alvo = 0	24,9%	17,0%
Alvo = 1	13,8%	44,2%

Fonte: Elaborada pela autora

A Tabela 12 mostra as métricas para o segundo modelo de regressão: precisão, revocação, especificidade e medida F-1:

Tabela 12 – Métricas - Regressão Logística

Métrica	%
Precisão	72,2%
Revocação	76,2%
Especificidade	59,4%
Medida F-1	74,1%

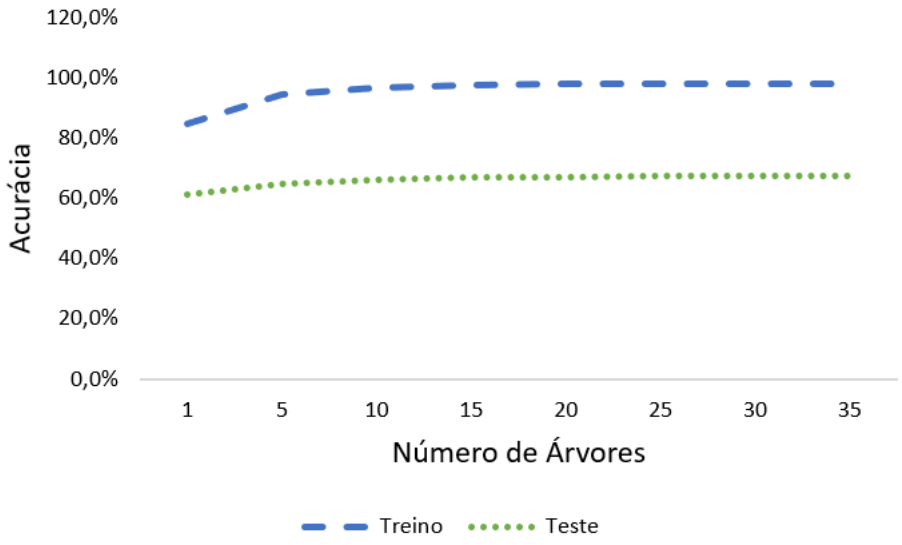
Fonte: Elaborada pela autora

Olhando as métricas da Tabela 12 constatamos que o modelo teve uma pequena queda de desempenho. Seguindo na mesma linha, a acurácia do treino e teste ficaram em 69,1% e caíram um pouco em relação ao primeiro modelo. Mas igual ao modelo anterior, na figura 38 observamos que algumas variáveis continuam com P>|z| alto: "carros", "Q002_A" e "Q004_A".

3.4.3 *Random Forest*

O modelo *Random Forest* foi desenvolvido na biblioteca *sklearn.ensemble - RandomForestClassifier* no python. Igual na árvore de decião, existem diversos parâmetros que podem ser alterados, mas existe um parâmetro essencial para esta técnica: *n_estimators* (número de árvores que serão criadas). Não existe um número ideal de quantas árvores o modelo deva ter, cada problema é específico. Para este trabalho, foram testados alguns números de árvores (sem nenhum outro parâmetro alterado) conforme a Figura 39 mostra:

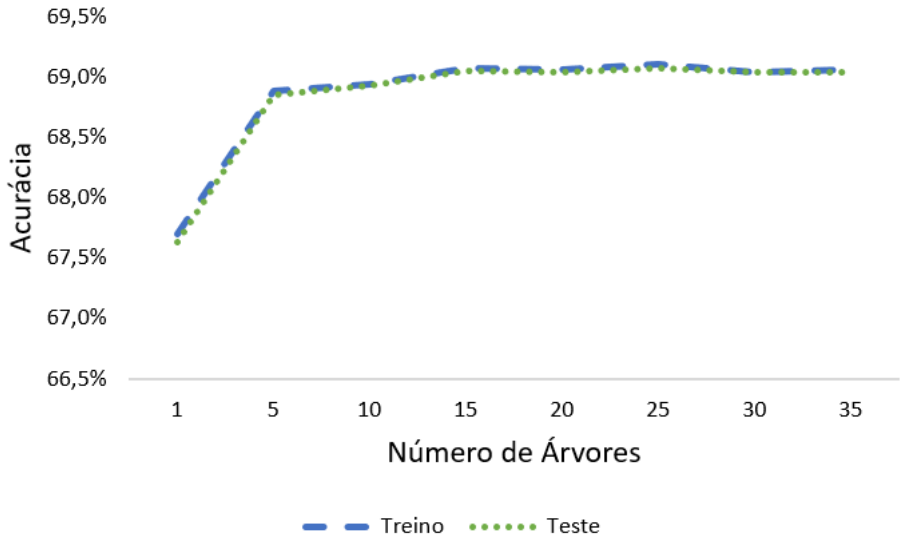
Figura 39 – Número de árvores x Acurácia



Fonte: Elaborada pela autora

Pela Figura 39 podemos concluir que somente imputando o número de árvores e sem especificar nenhuma outra restrição, o modelo sofre de *overffiting*. Quando aplicado a base de teste, o desempenho cai em torno de 20%. Para tentar resolver este problema, além do número de árvores vamos imputar uma restrição de *max_depth* (profundidade de cada árvore). A Figura 40 mostra o número de árvores versus acurácia (com todas as arvores com no máximo 10 de profundidade):

Figura 40 – Número de árvores x Acurácia (profundidade máxima 10



Fonte: Elaborada pela autora

Dada a Figura 40, podemos confirmar que limitar o número de profundidade das árvores ajudou a manter um bom desempenho tanto no treino quanto no teste, sem sofrer *overffiting*. Com base nesses resultados, o *Random Forest* que será treinado terá 25 árvores e máximo 10 de profundidade (combinação que teve a maior acurácia, mostrada na Figura 40).

A Tabela 13 mostra a matriz de confusão para os dados de teste do modelo de *Random Forest*:

Tabela 13 – Matriz de Confusão - *Random Forest*

	Classe Predita	
Classe Real	Alvo = 0	Alvo = 1
Alvo = 0	23,8%	18,1%
Alvo = 1	12,1%	45,2%

Fonte: Elaborada pela autora

A Tabela 14 mostra as métricas para o modelo de *Random Forest*: precisão, revocação, especificidade e medida F-1:

Tabela 14 – Métricas - *Random Forest*

Métrica	**%**
Precisão	71,4%
Revocação	77,8%
Especificidade	56,8%
Medida F-1	74,4%

Fonte: Elaborada pela autora

A Tabela 15 mostra a importância de cada variável explicativa:

Tabela 15 – Importância das variáveis - *Random Forest*

Variável Explicativa	**% de Importância**
Renda Familiar	22,5%
Quantidade de computadores na residência	21,7%
Grupo de ocupação do pai/homem responsável	11,1%
Quantidade de banheiros na residência	6,3%
Grupo de ocupação do mãe/mulher responsável	5,8%
Indicador de TV por assinatura na residência	5,7%
Série que a mãe/mulher responsável estudou	3,6%
Quantidade de carros na residência	3,6%
Série que a pai/homem responsável estudou	3,4%
Cor/Raça do candidato	3,2%
Quantidade de TVs na residência	3,1%
Quantidade de celulares na residência	3,0%
Idade do candidato	1,6%
Tipo de escola do EM	1,1%
Indicador de acesso a internet na residência	1,1%
Região do país que o candidato mora	1,0%
Quantidade de pessoas que moram na residência	0,7%
Situação de conclusão do EM	0,5%
Serviço de empregado(a) doméstico(a)	0,5%
Sexo do candidato	0,4%
Quantidade de quartos na residência	0,1%

Fonte: Elaborada pela autora

3.4.4 *Comparação entre modelos*

Com os três modelos de classificação treinados podemos analisar os resultados e compara-lós. A Tabela 16 mostra a comparação das métricas dos três modelos:

Tabela 16 – Comparação das métricas entre modelos

Métrica	Árvore	Reg. Logística	*Random Forest*
Acurácia	69,1%	69,1%	69,0%
Precisão	72,4%	72,2%	71,4%
Revocação	75,5%	76,2%	77,8%
Especificidade	60,2%	59,4%	56,8%
Medida F-1	73,9%	74,1%	74,4%

Fonte: Elaborada pela autora

Quando olhamos somente a acurácia os três modelos tem praticamente o mesmo desempenho e conseguem prever de forma correta 69% de todos os casos que foram passados no teste (aproximadamente 1.000.000 de casos). Mas o ideal é avaliar por outros ângulos, por isso utilizamos outras métricas de desempenho.

A precisão avalia de todos os casos que o modelo classificou como "alvo", quantos realmente eram reais. Neste caso, o modelo que teve a melhor precisão foi a árvore de decisão, com uma pequena diferença da regressão logística, e o pior *Random Forest*. Já a revocação verifica de todos os casos "alvo" reais, quando o modelo previu de forma correta. Olhando por esse ponto de vista as conclusões se invertem, dado que o RF teve um melhor resultado, com 2,3% a mais que a árvore (modelo que tem a melhor precisão). A regressão logística também tem um melhor desempenho nessa métrica do que a árvore, mas 0,6% abaixo do RF.

Ao contrário da revocação, temos a especificidade. Essa métrica avalia de todos os casos que não são alvos reais, quantos o modelo previu de forma correta. Por este ponto de vista, a árvore teve um melhor resultado, seguido da regressão e RF. Por fim, temos a medida F-1 que combina as medidas precisão e revocação.

Cada problema de classificação se difere em diferentes aspectos, por isso não existe a melhor métrica para ser usada, vai depender de cada problema específico. O objetivo deste trabalho é classificar de forma correta o máximo de casos reais alvos. Dado isto, as melhores métricas para serem consideradas são: precisão e revocação. Como a medida F-1 engloba essas duas métricas, o modelo que desempenha melhor é o RF, por isso seria escolhido. Mas se o objetivo fosse diferente, outro modelo poderia ter um resultado melhor.

Além das métricas, outro ponto que difere entre os algoritmos é a "importância" de cada variável explicativa. Cada algoritmo considera diferentes variáveis como principais. No caso dos algoritmos de árvore e RF a função *feature_importances_*, do *sklearn*, foi utilizada para obter o valor da importância de cada variável. Segundo a documentação da biblioteca, scikitlearn (2021): "A importância de um recurso é calculada como a redução total (normalizada) do critério trazido por esse recurso" (Tradução própria). Dado isso, na árvore as duas principais variáveis foram: renda familiar e quantidade de computadores. No RF as mesmas variáveis entraram como as duas mais importantes, mas com pesos

bem diferentes. A árvore concentrou um peso grande em uma variável, enquanto o RF distribuiu entre outras informações.

No caso da regressão logística, analisaremos a "importância" de forma diferente, dado que não temos a mesma informação igual aos dois algoritmos. Vamos olhar o valor do coeficiente de cada variável, pois quando aquela informação existe, é o quanto vai impactar no valor final do score. Com o coeficiente positivo, as duas principais são: série que mãe/mulher responsável estudou e cor/raça do candidato. Com o coeficiente negativo são: sexo e idade do candidato.

4 CONCLUSÃO

Com os resultados apresentados neste trabalho conseguimos atingir o objetivo inicial desejado. Dos três algoritmos treinados para criar os modelos de classificação, com o objetivo de acertar os alunos que conseguem nota acima de 500 pontos, o *Random Forest* se mostrou melhor quando analisamos as métricas de precisão e revocação. Além disso, outro objetivo era conseguirmos encontrar quais características dos alunos e socioeconômicas estavam relacionadas a chance de sucesso (modelo acertar o alvo). De todas as informações utilizadas, as que se mostraram com maior relação ao alvo foram: renda familiar, qauntidade de computadores, grupo de ocupação do pai/homem responsável, série que mãe/mulher responsável estudou, quantidade de banheiros entre outros. Grande parte dessas características estão relacionadas a desigualdade social. Infelizmente, comprovamos que nem todos no país tem acesso as mesmas oportunidades. Ou seja, mesmo que um aluno de baixa renda tente conseguir uma vaga no SiSu, sua sitação financeira poderá impactar. Isso não quer dizer que alunos de baixa renda não possam conseguir as vagas, pois muitos conseguem, apenas mostra que pode ser um caminho mais dificíl. Além disso, mostramos como diferentes algoritmos interpretam e dão pesos para as informações de forma diferentes, sendo que a base de dados utilizada foi exatamente igual nos três modelos. Outro ponto importante é como tomar uma decião de qual modelo usar olhando somente acurácia pode nos enganar. Os três modelos tiveram a mesma acurácia, mas quando analisamos outras métricas isso mudou.

Como oportunidade futura há diversas possibilidades de exploração no assunto, uma delas seria testes com a alteração nos parâmetros dos modelos, que foi trabalhado pouco neste estudo. A mudança desses parâmetros podem trazer melhores resultados. Ainda considerando os algoritmos, seria interessante treinar outros modelos de AM. Existem diversos outros algoritmos supervisionados que podem ser testados para verificar se tem um desempenho melhor, como por exemplo: *Gradien Boosting*, *Support Vector Machine*, K-Vizinhos mais próximos entre outros.

Outra alternativa é trabalhar com as variáveis explicativas. Testar cada algoritmo com um conjunto de variáveis diferentes, tirando uma de cada vez para encontrar o conjunto de informações que desempenha melhor. Além disso, fazer a verificação de correlação entre as variáveis explicativas pode ajudar a reduzir o número de informações no momento de treinar o algoritmo. Esse estudo geralmente é realizado para reduzir variáveis que explicam o mesmo evento. A Figura 41 mostra a correlação existente para algumas variáveis numéricas utilizadas nos modelos:

Figura 41 – Correlação entre variáveis

	Sexo	Banheiro	Quartos	Carros	TVs	Celular	Computador	Acesso internet	Renda	Idade
Sexo	1,00	-0,06	-0,04	-0,06	-0,08	-0,04	-0,10	-0,06	-0,07	-0,01
Banheiro	-0,06	1,00	0,53	0,55	0,55	0,39	0,49	0,26	0,61	-0,13
Quartos	-0,04	0,53	1,00	0,42	0,40	0,45	0,35	0,25	0,37	-0,16
Carros	-0,06	0,55	0,42	1,00	0,48	0,43	0,49	0,31	0,54	-0,14
TVs	-0,08	0,55	0,40	0,48	1,00	0,42	0,47	0,25	0,51	-0,13
Celular	-0,04	0,39	0,45	0,43	0,42	1,00	0,42	0,37	0,36	-0,18
Computador	-0,10	0,49	0,35	0,49	0,47	0,42	1,00	0,38	0,53	-0,06
Acesso internet	-0,06	0,26	0,25	0,31	0,25	0,37	0,38	1,00	0,23	-0,07
Renda	-0,07	0,61	0,37	0,54	0,51	0,36	0,53	0,23	1,00	-0,11
Idade	-0,01	-0,13	-0,16	-0,14	-0,13	-0,18	-0,06	-0,07	-0,11	1,00

Fonte: Elaborada pela autora

Na Figura 41 em verde temos a correlação da variável com ela mesmo, por isso seu valor é 1. Nos campos vermelhos temos correlações altas entre variáveis. Isso não é interessante para o modelo. O ideal seria escolher somente uma delas (que tem correlação alta) para manter no modelo.

Por fim, explorar a busca de novas informações para tentar melhorar o desempenho dos modelos seria um bom desafio. Para o desenvolvimento destes modelos foram exploradas somente as informações que estavam na base de dados do ENEM, mas é possível trazer outras informações, como: indicadores de idade, indicadores socioeconômicos da cidade ou estado de residência do candidato entre outros.

REFERÊNCIAS

BARROS, A. S. X. Vestibular e enem: um debate contemporâneo. **Ensaio: aval. pol. públ. Educ**, Rio de Janeiro, v. 22, n. 85, p. 1057–1090, 2014.

BRASIL, M. d. E. **ProUni - Como Funciona**. Brasília, DF: Ministro da Educação: [s.n.], 2018. Disponível em: <:http://portal.mec.gov.br/prouni-sp-1364717183/como-funciona>. Acesso em: 26 jun. 2021.

_____. **Novo FIES: Sobre inscrição, seleção e cursos**. Brasília,DF: Ministro da educação: [s.n.], 2021. Disponível em: <:http://portalfies.mec.gov.br/?pagina=faq#sobreinscricao>. Acesso em: 26 jun. 2021.

BREINMAN, L. *Random forests*. ***Machine Learning***, Boston, v. 45, n. 1, p. 5–32, 2001.

BRUCE, P.; BRUCE, A. **Estatística Prática para Cientista de Dados**: 50 conceitos essenciais. 1. ed. Brasil: Alta Books, 2019.

DINIZ, C.; LOUZADA, F. **Modelagem Estatística para Risco de Crédito**. João Pessoa, Paraíba: ABE - Associação Brasileira de Estatística, 2012.

EZAKI, N. *A study of equality of educational opportunity in Nepal using logistic regression analysis*. ***International Journal of Comparative Education and Development***, Japan, v. 22, n. 4, p. 249–262, 2020.

FACELI, K. et al. **Inteligência Artificial**: Uma abordagem de aprendizado de máquina. Rio de Janeiro: LTC, 2011.

GÉRON, A. **Hands-on Machine Learning with Scikit-Learn, Keras TensorFlow**: Concepts, tools, and techniques to build intelligent systems. 2. ed. São Paulo: Novatec, 2019.

INEP. **5,8 milhões estão inscritos para fazer o Enem 2020**. Brasília, DF: [s.n.], 2021. Disponível em: <:http://portal.inep.gov.br/artigo/-/asset_publisher/B4AQV9zFY7Bv/content/5-8-milhoes-estao-inscritos-para-fazer-o-enem-2020/21206>. Acesso em: 26 jun. 2021.

_____. **Exame Nacional do Ensino Médio (Enem)**. Brasília, DF: [s.n.], 2021. Disponível em: <:https://www.gov.br/inep/pt-br/areas-de-atuacao/avaliacao-e-exames-educacionais/enem>. Acesso em: 26 jun. 2021.

JAMES, G.; WITTEN, D.; TIBSHIRANI, T. H. R. **An Introduction to Statistical Learning**: with applications in r. Nova York: Springer, 2013.

JÚNIOR, J. A. S. et al. Geração de música com aprendizado de máquina. **Colloquium Exactarum**, v. 11, n. 2, p. 56–65, 2019.

LEãO, J. J. C. C. et al. Inteligência artificial na educação: aplicações do aprendizado de máquina para apoiar a aprendizagem adaptativa. **ReviVale**, Araçuaí, v. 1, n. 1, p. 1–19, 2021.

MAGALHãES, M. N.; LIMA, A. C. P. **Noções de Probabilidade e Estatística**. São Paulo: Editora da Universidade de São Paulo, 2015.

MAURO, G. **Maiores e Menores notas de corte do SISU 2020: Saiba quais são e se prepare!** 2021. Disponível em: <https://noticiasconcursos.com.br/maiores-e-menores-notas-de-corte-do-sisu-2020/>. Acesso em: 07 jul. 2021.

PANG-NING, T. et al. **Introduction to Data Mining**. 2. ed. New York, NY, publisher = Pearson,: [s.n.], 2019.

SAMPAIO, B.; AES, J. G. Diferenças de eficiência entre ensino público e privado no brasil. **Economia Aplicada**, São Paulo, v. 13, n. 1, p. 45–68, 2009.

SAMPAIO, G. T. C.; OLIVEIRA, R. P. Dimensões da desigualdade educacional no brasil. **RBPAE**, v. 31, n. 3, p. 511 – 530, 2015.

SCIKITLEARN. **sklearn.tree.DecisionTreeClassifier**. 2021. Disponível em:<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier.feature_importances_>. Acesso em: 28 dez. 2021.

SHAH, C. **A hands-on introduction to data science**. Cambridge: Cambridge University Press, 2020.

SILVA, R. C. D.; MELO, S. D. G. Enem: Propulsão ao mercado educacional brasileiro no século xxi. **Educação e Realidade**, Porte Alegre, v. 43, n. 4, p. 1385–1404, 2018.

TAVARES, L. A.; MEIRA, M.; AMARAL, S. F. Inteligência artificial na educação: Survey. **Brazilian Journal of Development**, Curitiba, v. 6, n. 7, p. 48699–48714, 2020.

VIEIRA, F. D.; OLIVEIRA, S. R. M.; PAIVA, S. R. Metodologia baseada em técnicas de mineração de dados para suporte à certificação de raças de ovinos. **Journal of the Brazilian Association of Agricultural Engineering**, Brasília, v. 35, n. 6, p. 1172–1186, 2015.