Análise de dados do Exame Nacional do Ensino Médio para previsão da nota de matemática dos candidatos

Marcella Andrade da Rocha*

23 de agosto de 2022

Introdução

A análise de dados educacionais é um campo emergente que consiste em atribuir valor estatístico a um grande volume de dados educacionais, onde, técnicas e algoritmos de aprendizagem de máquina podem ser aplicados para manipular e obter padrões gerando novos conhecimentos relacionados ao contexto ensino-aprendizagem, não somente utilizando desempenho dos alunos mas também o cenário econômico e social nos quais as escolas estão introduzidas com a finalidade de otimizar o uso dos recursos disponíveis (JINDAL; BORAH, 2013).

O Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) é o órgão responsável pela realização do Exame Nacional do Ensino Médio (ENEM), avaliação elaborada e aplicada pelo Ministério da Educação (MEC) e utilizada para aferir o desempenho dos estudantes no final do ensino médio como também dar acesso à educação superior (ENEM/INEP, 2022). O ENEM é distribuído em quatro áreas: Matemática e suas tecnologias; Ciências humanas e suas tecnologias; Linguagens, códigos e suas tecnologias; Ciências da natureza e suas tecnologias. Como também é realizada a redação para testar a aptidão de escrita.

O Programa Internacional de Avaliação de Estudantes (Pisa) com o propósito de avaliar os sistemas educacionais medindo o desempenho escolar de alunos de 15 anos de idade em matemática, ciências e leitura, expôs que o Brasil possui baixa competência em matemática, ciências e leitura se equiparado com outros 78 países participantes. A última edição publicada em 2018 indica que 68,1% dos estudantes brasileiros com a idade de 15 anos, não dispõe de nível básico mínimo em matemática para exercício pleno da cidadania e esse índice está estacionado desde 2009 (PISA, 2018).

Ao comparar o Brasil aos países vizinhos na América do Sul também analisados pelo PISA, ele é o pior pais na disciplina de matemática com empate estatístico com a Argentina, Brasil com 384 e Argentina com 379 pontos. Uruguai, Chile, Peru e Colômbia

*Programa de Pós graduação em Engenharia Elétrica e Computação UFRN

estão na frente com 418, 417, 400 e 391 pontos, respectivamente. Mais de 40% dos jovens que possuem o nível básico de conhecimento não tem capacidade de resolver problemas simples e corriqueiros. Dos 10.961 estudantes que participaram do PISA apenas 0,1% obtiveram nível máximo de proficiência em matemática (PISA, 2018).

Observando os índices do PISA e levando em consideração que o ENEM apresenta importância no cenário educacional brasileiro, sendo uma das avaliações mais importantes para os estudantes e fornecendo um grande número de dados e informações não explorados, este projeto tem como finalidade extrair os dados educacionais do ENEM através do banco de dados do ano 2021 (INEP, 2022) e analisar quais aspectos regionais/sociais influenciam na previsão da nota de matemática, como também o desenvolvimento de um programa computacional utilizando sistemas inteligentes capaz de prever a nota de matemática utilizando as notas das outras disciplinas. Para esse fim, algoritmos supervisionados e não supervisionados serão aplicados e algumas análises a serem realizadas subdividirão o banco de dados em tipos de escola (privadas, públicas e estrangeiras) ou em regiões (norte, nordeste, sul, centro-oeste e sudeste).

1 Revisão Bibliográfica

Nesta seção será apresentado alguns trabalhos na literatura com escopo e objetivos semelhantes ao aplicado nessa pesquisa.

O trabalho relacionado com finalidade mais próxima foi o de Alves, Cechinel e Queiroga (2018), onde os autores tiveram como propósito descobrir padrões e fazer um modelo de previsão para indicar o desempenho (baixo, médio ou alto) das notas nas provas de matemática do ENEM 2015 utilizando técnicas de mineração de dados. Os dados utilizados apresentam valores médios de aproveitamento dos alunos agrupados por escola e compreendem 15.598 instâncias.

Os autores Alves, Cechinel e Queiroga utilizaram o software Waikato Environment for Knowledge Analysis (WEKA) com as técnicas de aprendizagem de máquina Naive Bayes e J48. Para mais, Alves, Cechinel e Queiroga implementaram a categorização na variável a ser predita e alcançaram uma acurácia de 71,9384% com o algoritmo J48, demonstrando que a condição mais simples de ser predita é a categoria "Alta"(notas acima de 502) e a mais complexa é a "Média"(notas entre 451 e 502).

A abordagem de Rodrigues, Pinto e Souza (2016) possui uma análise do conjunto de dados fornecido pelo site qedu.org.br que reúne resultados de avaliações empregadas para indicar a qualidade do ensino básico no Brasil. Os autores utilizaram dados de provas do ENEM de 2009 a 2014 para analisar a situação das escolas públicas estaduais do município de Viçosa no estado de Minas Gerais (MG). Com esses resultados foi possível realizar comparações entre as escolas privadas, públicas e o Colégio de Apicação da UFV (COLUNI) em relação à média geral no ENEM, à taxa de participação dos estudantes e as médias em Ciências da Natureza e suas tecnologias.

Os resultados alcançados pelos autores Rodrigues, Pinto e Souza após a análise dos dados foi que a taxa de participação dos estudantes de escola pública no ENEM ainda é baixa e o desempenho adquirido foi inferior ao esperado. Esses valores podem estar relacionados a várias adversidades, com ênfase ao pouco acesso aos patrimônios culturais levando em consideração que as escolas públicas atendem à classe mais pobre da população.

Um estudo mais recente de Bravin, Lee e Rissino (2019) utilizando o software RStudio para aplicação de um algoritmo de classificação e um de regressão na linguagem de programação R, árvore de Decisão e Regressão Linear, respectivamente. Os autores utilizaram técnicas de mineração de dados para adquirir conhecimento útil a partir dos dados do ENEM 2015 com 15.497 instâncias e 42 atributos após o pré-processamento.

Os resultados do trabalho de Bravin, Lee e Rissino foi adquirido a partir da mineração dos dados do ENEM. Utilizando a árvore de decisão foi possível observar os grupos socioeconômicos das escolas e a relação dos grupos com as notas de Português mas obteve um baixo desempenho, com a regressão linear foi possível prever as notas de português utilizando indicadores de nível socioeconômico.

LOBO, CASSUCE e CIRINO (2017) fez um modelo hierárquico de dois níveis utilizando os dados do ENEM 2013 para os estudantes do 3º ano do ensino médio da região nordeste que tinham interesse em ingressar em uma instituição de ensino superior. Com a análise estatística feita em LOBO, CASSUCE e CIRINO identificou-se a importância da condição socioeconômica do estudante no desempenho escolar que neutraliza as consequências negativas que o tipo de escola gera em relação a proficiência em matemática.

O estudo feito por Simon e Cazella (2017) implementou um modelo preditivo para

indicar a média de desempenho dos estudantes do ensino médio em Ciências da Natureza e suas tecnologias no ENEM 2015. O software utilizado foi o WEKA para aplicação das técnicas de mineração de dados com o algoritmo J48. Os resultados adquiridos pelos autores com a árvore de decisão foi que o algoritmo identificou corretamente 77,02% instâncias das 15998 utilizadas.

A pesquisa exposta neste artigo se distingue das tratadas nesta seção pois averígua-se prever as notas de matemática através dos aspectos socioeconômicos e as notas das outras disciplinas do ENEM 2022. Para isso, será implementado um modelo preditivo utilizando algoritmos de aprendizagem de máquina com a biblioteca *Sklearn* da linguagem de programação *Python*.

2 Descrição da Base de Dados

O Banco de dados utilizado para esse projeto é o dos microdados do ENEM 2021 que possui aproximadamente 1,5 GB com 76 atributos abrangendo dados do participante, dados da escola, dados dos pedidos de atendimento especializado, dados dos pedidos de atendimento específico, dados dos pedidos de recursos especializados e específicos para realização das provas, dados do local de aplicação da prova, dados da prova objetiva, dados da redação e dados do questionário socioeconômico como também 3.389.832 de instâncias. A base de dados possui 37 atributos categóricos/alfanuméricos, 39 numéricos e 44.648.935 valores faltosos ou nulos e pode ser adquirida em (INEP, 2022).

Para o desenvolvimento do projeto inicialmente está sendo utilizado 22 atributos como mostrado na tabela 1 e 3.389.832 instâncias.

Abaixo uma descrição detalhada dos atributos:

- Nota: número real de 0 a 1000;
- Siglas das UF possuem 2 dígitos que representam os estados do Brasil;
- Sexo: feminino e masculino;
- Estado civil: Solteiro, Casado/Mora com companheiro, Divorciado/Desquitado/Separado, Viúvo;
- Cor/Raça: Branca, Preta, Parda, Amarela, Indígena, Não dispõe da Informação;
- Situação de Conclusão do Ensino Médio: Concluiu, cursando e conclui em 2018, cursando e conclui após 2018, Não concluiu e não está cursando, Não informado;
- Ano de conclusão representa o ano com 4 dígitos;
- Tipo de instituição que concluiu/concluirá o EM: Ensino Regular, Educação Especial e Educação Jovens e Adultos;
- Tipo de escola: Não respondeu, Pública, Privada, Exterior;
- Localização da Escola: Urbana e Rural;
- Nacionalidade: Não informado, Brasileiro(a), Brasileiro(a) Naturalizado(a), Estrangeiro(a) e Brasileiro(a) Nato(a), nascido(a) no exterior;

Tabela 1 – Atributos utilizados inicialmente

Atributos	Tipo	Tamanho
Nota de Matemática	Numérico	4
Nota de Ciências Naturais	Numérico	4
Nota de Ciências Humanas	Numérico	4
Nota de Linguagens e Códigos	Numérico	4
Nota de Redação	Numérico	4
Sexo	Categórico	2
Estado Civil	Categórico	4
Cor/Raça	Categórico	6
Situação de Conclusão do Ensino Médio (EM)	Categórico	4
Ano de conclusão do EM	Numérico	4
Tipo de instituição que concluiu/concluirá o EM	Categórico	3
Tipo de escola do EM	Categórico	4
Localização da Escola	Categórico	2
Nacionalidade	Categórico	5
Renda Mensal da Família	Alfanumérico	17
Presença em CN	Categórico	3
Presença em CH	Categórico	3
Presença em LC	Categórico	3
Grau de instrução do pai	Categórico	8
Grau de instrução da mãe	Categórico	8
Candidato possui computador	Categórico	5
Candidato tem acesso a internet	Categórico	2

- Renda Mensal da Família possui os valores entre nenhuma renda até maior que R$ 19.080,00;
- Presença na prova objetiva de Ciências da Natureza (CN);
- Presença na prova objetiva de Ciências Humanas (CH);
- Presença na prova objetiva de Linguagens e Códigos (LC);
- Grau de instrução do(a) pai/mãe é uma pergunta do questionário socioeconômico do ENEM: Até que série seu pai/mãe, ou o homem/mulher responsável por você, estudou? e possui 8 respostas desde nunca estudou até pós-graduação;
- Candidato possui computador, tem acesso à internet ou exerceu atividade remunerada também representam perguntas do questionário socioeconômico.

3 Análise dos Dados

O atributo 'Nota de Matemática' que é o *target* será dividido em classes, essa divisão foi definida observando a distribuição dos dados no histograma da figura 1.

As classes ficaram definidas como:

1. Menor que 440 - 727.129;

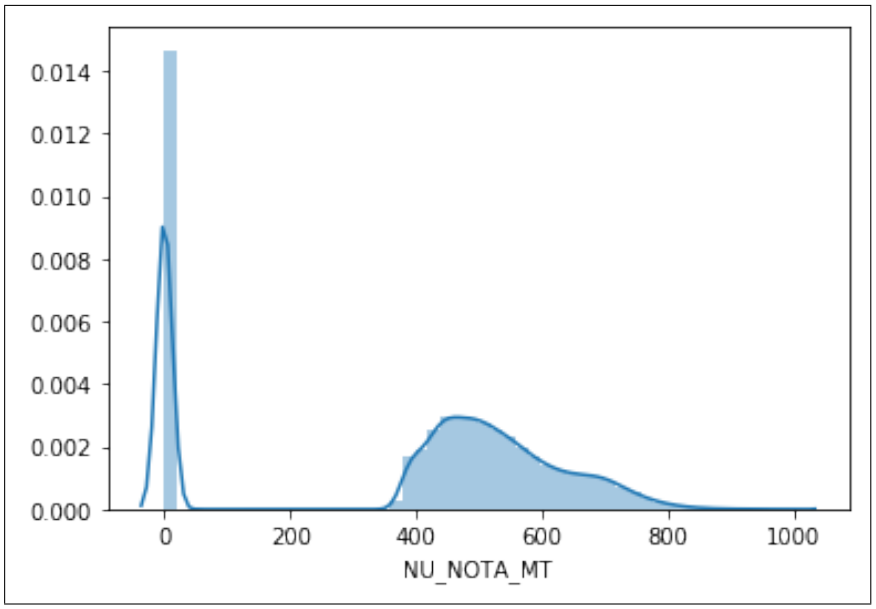

Figura 1 – Histograma de distribuição das notas de matemática.

2. Entre 440 e 490 - 810.255;
3. Entre 490 e 540 - 743.169;
4. Entre 540 e 620 - 796.532;
5. Maior que 620 - 828.014.

Como podemos observar na figura 1 existem muitas instâncias nulas que podem representar tanto notas zero como alunos que faltaram no dia de realização do exame, retirando-se esses dados o número de instâncias fica em 3.389.409. Balanceando os dados pela quantidade de candidatos por estado do Brasil é possível diminuir mais instâncias da base de dados, deixando 2469 instâncias para cada estado e um total de 66.663.

Calculando a média, desvio padrão, mediana e variância do atributo nota de matemática, temos os seguintes valores 536.62, 103.48, 517.3 e 10708.7, respectivamente. A distribuição dos dados das notas de matemática entre as escolas: 1- Não Informadas pelo candidato; 2- Públicas; 3- Privadas; 4- Estrangeiras e pode ser vista na figura 2 e percebe-se que as notas dos alunos da escola pública é menor que as dos demais, a mediana das notas dos candidatos de escola pública deu 497.3 enquanto que os não informados 517.4, escolas privadas 516.6 e escolas estrangeiras 626.4.

A matriz de correlação dos atributos nota antes do pré processamento é possível ser vista na figura 3 e pode-se ver que as notas que mais se correlacionam são as notas de linguagens e códigos (NU_NOTA_LC) e ciências humanas (NU_NOTA_CH) com 0.7 e os atributos menos correlacionados são as notas de Matemática (NU_NOTA_MT) e Redação (NU_NOTA_REDACAO) com 0.49.

Na figura 4 tem-se a distribuição das notas de matemática por sexo e na figura 5 a distribuição por raça.

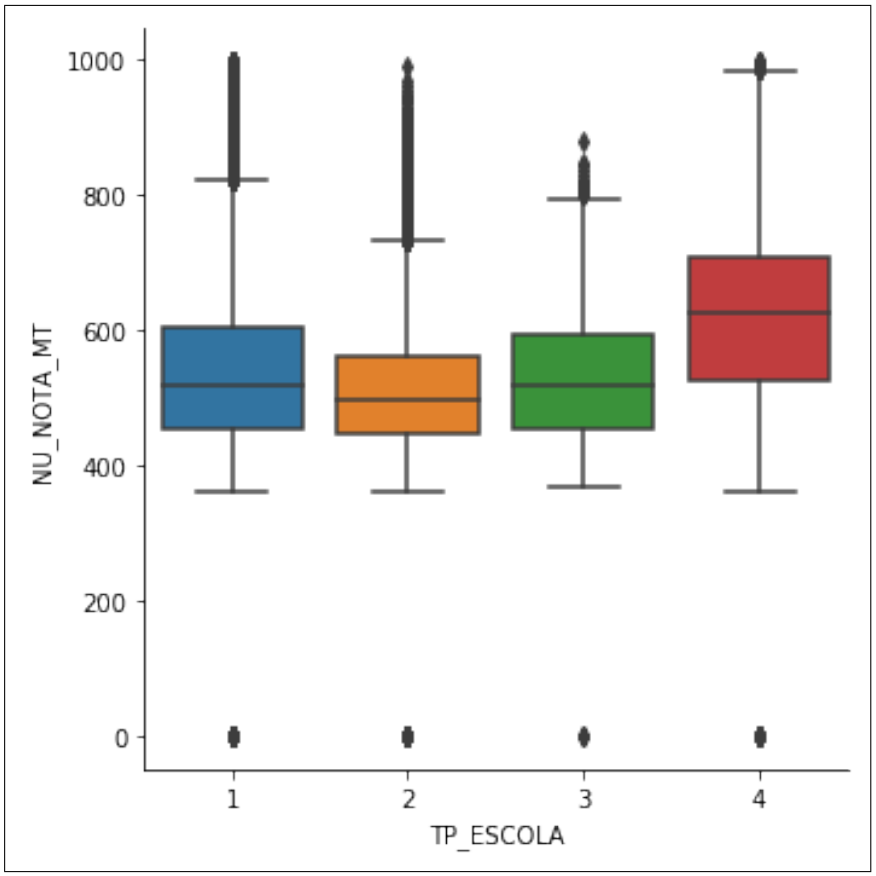

Figura 2 – Boxplot com a distribuição das notas de matemática por tipo de escola.

Na distribuição pelo sexo podemos observar que possui mais mulheres que homens fazendo a prova do ENEM e suas notas são maiores.Na distribuição pela raça podemos perceber que as maiores notas são adquiridas pela raça branca seguida de não declarado e amarela.

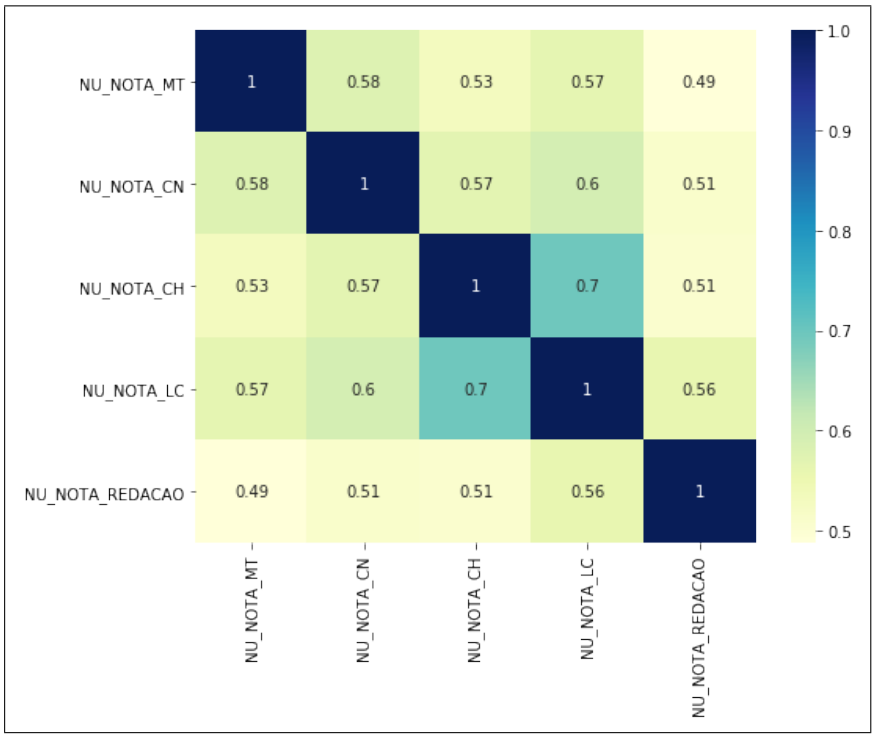

Figura 3 – Matriz de correlação entre todas as notas.

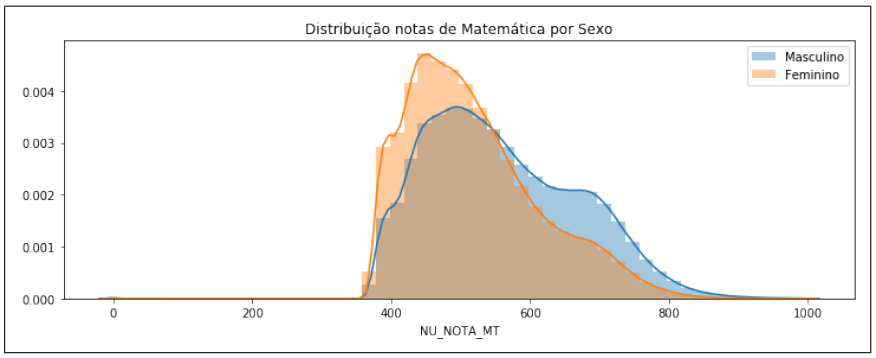

Figura 4 – Distribuição das notas de matemática pelo sexo.

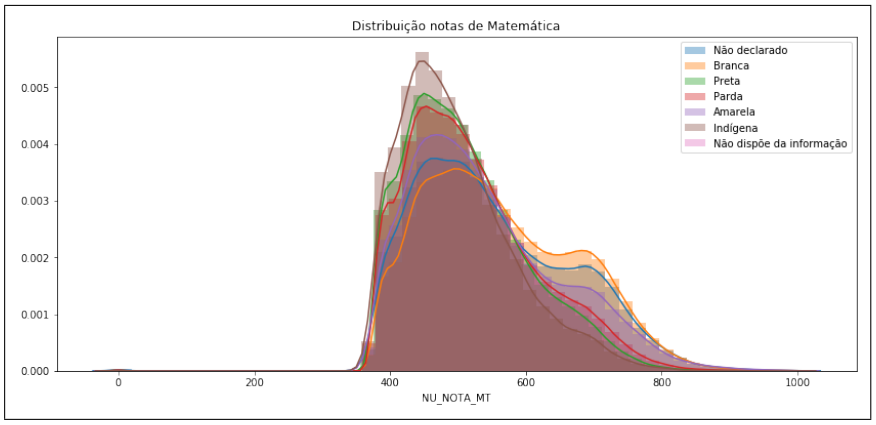

Figura 5 – Distribuição das notas de matemática pela raça.

3.1 Transformação de dados

Os dados do tipo categórico que estavam com valores nominais como: UF de Residencia, Sexo, Cor/Raça, Nacionalidade, Situação de Conclusão do EM, Tipo de escola do EM, Presença em CN, CH e LC, Renda Mensal da Família, Grau de instrução do pai/mãe e candidato possui computador, tem acesso à internet ou exerceu atividade remunerada foram convertidos em numéricos, para aplicação de algumas técnicas de Aprendizagem de Máquina que manipulam melhor valores do tipo numérico.

4 Pré-processamento

O banco de dados possui um número elevado de instâncias e atributos e necessita de pré processamento, nas próximas seções estará detalhado as técnicas de pré processamento utilizadas para redução desses dados tentando preservar dados relevantes para a etapa de treinamento e regressão.

4.1 Ausência de Valores

A base de dados do ENEM possui 44.648.935 valores incompletos ou ausentes somando todos os 24 atributos selecionados inicialmente. Analisando alguns atributos com dados ausentes temos que:

- 1.354.127 candidatos foram desclassificados ou faltaram a prova de matemática;
- 1.608.648 candidatos faltaram ou foram desclassificados na prova de Ciências da Natureza;
- 1.365.483 candidatos faltaram ou foram desclassificados na prova de Ciências Humanas;
- 1.365.483 candidatos faltaram ou foram desclassificados na prova de Linguagens e Códigos;
- 99% dos candidatos presentes no primeiro dia foram ao segundo dia de prova.

Tabela 2 – Porcentagem de valores faltantes por atributo

Atributos	**Porcentagem (%)**
Nota de Matemática	29.17
Nota de Ciências Naturais	29.17
Nota de Ciências Humanas	24.76
Nota de Linguagens e Códigos	24.76
Nota de Redação	24.76
Sigla da UF de residência	0.0
Sexo	0.0
Estado Civil	3.95
Cor/Raça	0.0
Situação de Conclusão do Ensino Médio (EM)	0.0
Ano de conclusão do EM	0.0
Tipo de instituição que concluiu/concluirá o EM	36.82
Tipo de escola do EM	0.0
Localização da Escola	73.72
Nacionalidade	0.0
Renda Mensal da Família	0.0
Presença em CN	0.002
Presença em CH	0.002
Presença em LC	0.002
Grau de instrução do pai	0.0
Grau de instrução da mãe	0.0
Candidato possui computador	0.0
Candidato tem acesso a internet	0.0
Candidato exerceu atividade remunerada	0.00007

Na tabela 2 observa-se a percentagem de valores faltantes por atributos.

O primeiro filtro para eliminação de dados faltantes foi eliminar os atributos com porcentagem acima de 60% de valores faltantes, assim, Localização da Escola foi removido por conter um número significativo de dados faltantes. Em seguida foram removidos os dados dos candidatos faltaram nas provas de CN, CH ou teve a nota de matemática igual a zero.

A distribuição das notas levando em consideração os valores faltantes pode ser vista na figura 6.

Foi inferido nota zero aos dados faltantes, levando em consideração que falta implica em nota zero na prova. Na figura 7 aplicando-se a correlação das notas após o pré-processamento dos dados faltantes pode-se perceber um aumento significativo das correlações das notas.

O banco de dados inicial foi reduzido para 3.341.297 com todas as técnicas acima aplicadas.

4.2 Filtro de correlação

Por meio das correlações positivas e mais altas será selecionado os atributos mais relevantes para o modelo. Verificando a correlação de Pearson de todas as variáveis do

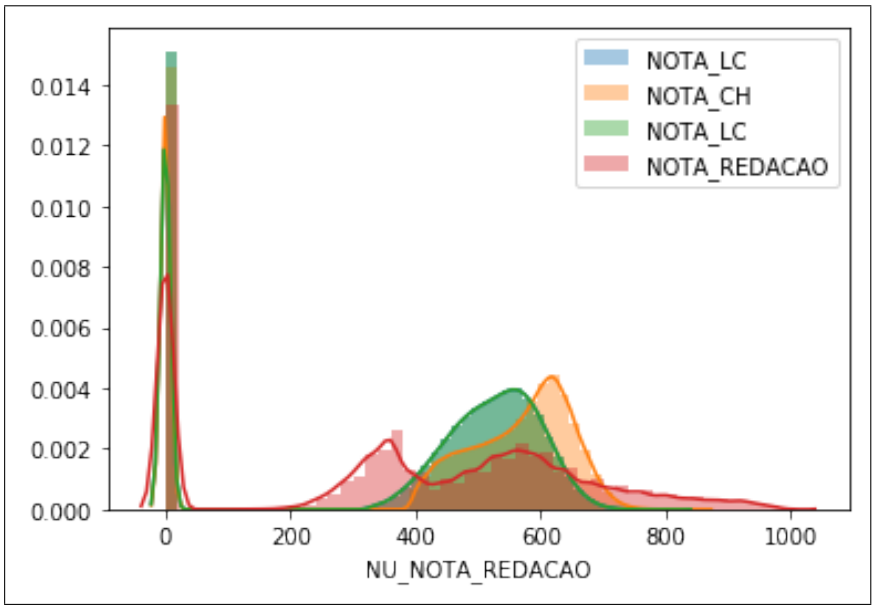

Figura 6 – Distribuição das notas

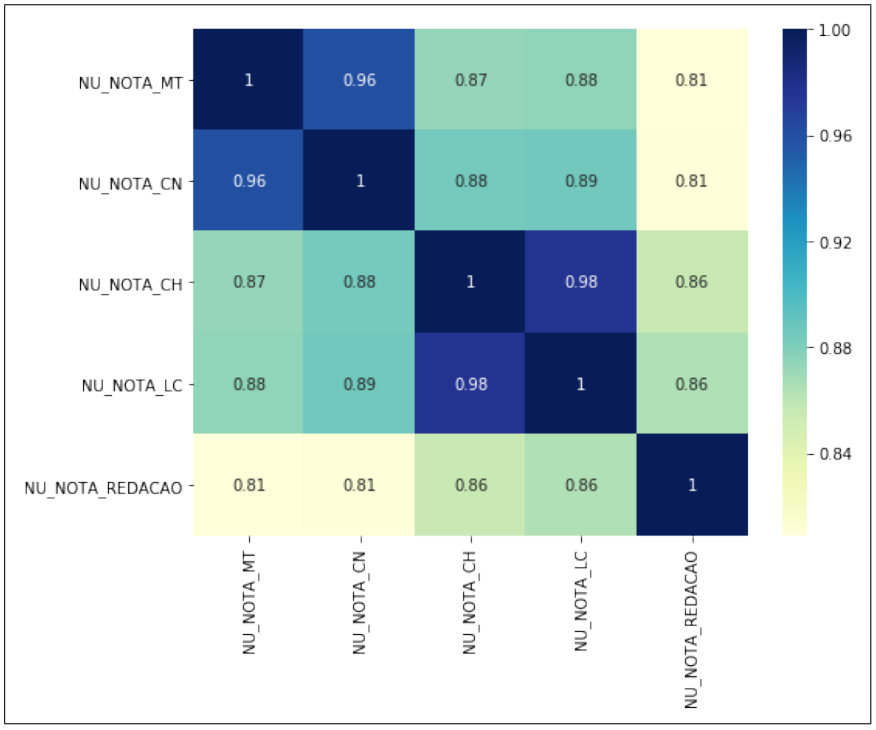

Figura 7 – Correlação das notas após pré processamento de dados faltantes.

conjunto, Figura 8, é possível ter uma visão macro das variáveis mais relevantes.

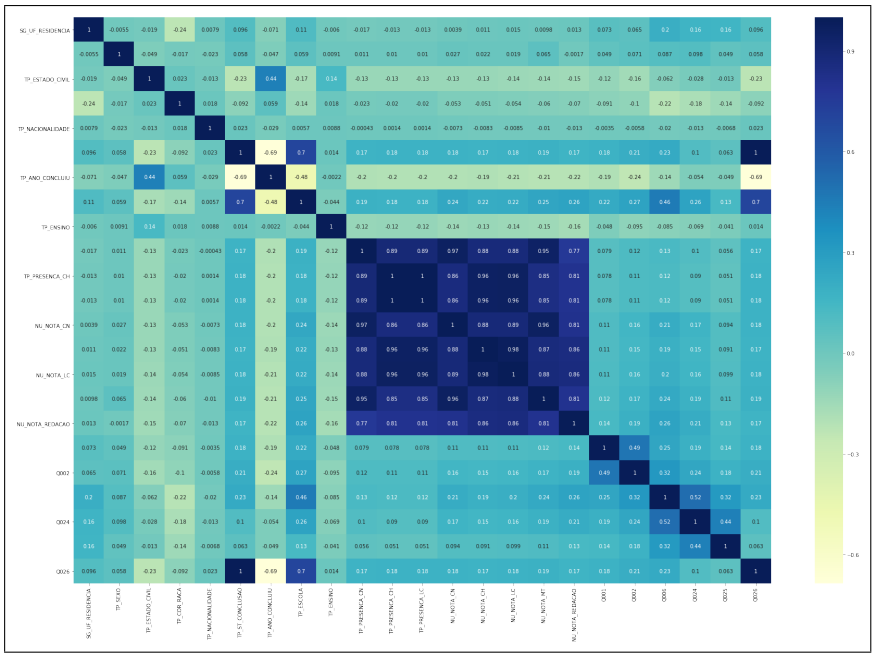

Figura 8 – Correlação de todos os atributos.

Como pode-se observar, apenas os recursos TP_PRESENCA_CN, TP_PRESENCA_CH, TP_PRESENCA_LC, NU_NOTA_CN, NU_NOTA_CH, NU_NOTA_LC e NU_NOTA_REDACAO estão altamente correlacionados com a variável de saída NU_NOTA_MT.

4.3 Wrapper

O método Wrapper precisa de um algoritmo de aprendizado de máquina e usa seu desempenho como critério de avaliação. Isso significa que se alimentam os atributos com um algoritmo de aprendizado de máquina selecionado e, com base no desempenho do modelo, adicionam-se/removem-se os atributos. É um processo iterativo e computacionalmente custoso, mas é mais eficiente que o método do filtro.

Existem diferentes métodos Wrapper, como Backward Elimination, Forward Selection, Bidirectional Elimination e RFE. Nas próximas seções será abordado o Backward Elimination e o RFE.

4.3.1 Backward Elimination

Nesse método, primeiro alimentam-se todos os atributos possíveis que serão utilizados no modelo. Verifica-se o desempenho do modelo e, em seguida, remove-se iterativamente os atributos com pior desempenho, um por um, até que o desempenho geral do modelo chegue ao intervalo aceitável.

A métrica de desempenho usada para avaliar o desempenho do atributo é *pvalue*. Se o *pvalue* estiver acima de 0,05, o recurso será removido, caso contrário, será mantido. Foi utilizado o método dos Mínimos Quadrados Ordinários (OLS). Este modelo é usado para realizar regressão linear. Essa abordagem foi implementada e o conjunto final adquirido só removeu um dos atributos, "Candidato tem acesso a internet".

4.3.2 Recursive Feature Elimination

O método Recursive Feature Elimination (RFE) funciona removendo recursivamente atributos e construindo um modelo nos atributos que permanecem. Ele usa a métrica de precisão para classificar o recurso de acordo com sua importância. O método RFE leva o modelo a ser usado e o número de recursos necessários como entrada. Em seguida, fornece a classificação de todas as variáveis, sendo 1 a mais importante. Ele também oferece suporte, sendo True um recurso relevante e False um recurso irrelevante.

Utilizando o modelo Regressão Linear com o RFE classificou o número ideal de recursos, para os quais a precisão é mais alta. Fazendo isso usando um loop que começa com 1 atributo e vai até o 23. Em seguida, seleciona-se aquele para o qual a precisão é mais alta. Como resultado desse método nenhum atributo foi removido e a acurácia deu 94.1%.

4.4 PCA

Primeiro os dados de teste e treino são normalizados, em seguida, cria-se a instância do modelo PCA, pode-se transmitir a variação desejada que o PCA vai capturar, na base de dados foi utilizada as variações: 0.85, 0.90, 0.95 e 0.99. Utilizar 0.9 como parâmetro para o modelo significa que o PCA manterá 90% dos atributos. Na tabela 3 tem o resultado do PCA utilizando o algoritmo de Regressão Linear.

Tabela 3 – Número de componentes x Precisão

Variação	**Número de componentes**	**Precisão (%)**
0.85	10	89
0.9	12	89
0.95	14	89
0.99	17	97

Utilizar 17 atributos adquire-se um excelente resultado.

5 Metodologia

5.1 Modelos Supervisionados

Para previsão das notas de matemática através dos outros atributos selecionados no pré-processamento foram utilizados quatro modelos supervisionados para problemas de regressão: Regressão Linear, Multilayer Perceptron Regressor (MLP Regressor) e Nearest Neighbors Regressor (KNN regressor) explicados com detalhes nas próximas subseções.

5.1.1 Regressão Linear

Utiliza-se o modelo de regressão linear múltipla no banco de dados, ajustando um modelo linear com as instâncias, como mostrado no vetor 1, para minimizar a soma dos quadrados dos resíduos entre as instâncias observadas no conjunto de dados e a instância alvo prevista pela aproximação linear.

$$NOTA\_MT = (NOTA\_CN, ..., SG\_UF\_RESIDENCIA) \tag{1}$$

De modo geral, a instância alvo Y é relacionada a um número p de instâncias de entrada. O modelo de regressão linear múltipla com p instâncias é dado por

Tabela 4 – Matriz de distribuição dos dados

y	x_1	x_2	...	x_p
y_1	x_11	x_12	...	x_1p
y_2	x_21	x_22	...	x_2p
...	...	...	...	...
y_n	x_n1	x_n2	...	x_np

$$Y_i = \beta_0 + \beta_1 x_i 1 + \beta_2 x_i 2 + ... + \beta_p x_i p + \epsilon_i i = 1, \ldots, n. \quad (2)$$

onde,

- $x_i 1, x_i 2, \ldots, x_i p$ são valores das instâncias conhecidas, por exemplo: NOTA_LC;
- $\beta_1, \beta_2, \ldots, \beta_p$ são os parâmetros da regressão;
- ϵ_i são os erros.

O modelo descreve um hiperplano p-dimensional referente as instâncias conhecidas. Supondo que tem n atributos ($n > p$) da instância NOTA_MT (nota de matemática) e das p instâncias conhecidas (Outras instâncias exceto a nota de matemática). Assim, y_i é o valor da NOTA_MT no i-ésimo atributo enquanto que x_{ij} é o valor da instância conhecida x_j no i-ésimo atributo, $j = 1, \ldots, p$. Os dados de uma regressão linear podem ser representados da seguinte forma:

Na figura 9 exibe o modelo de Regressão Linear para os dados (Notas reais x Notas preditas).

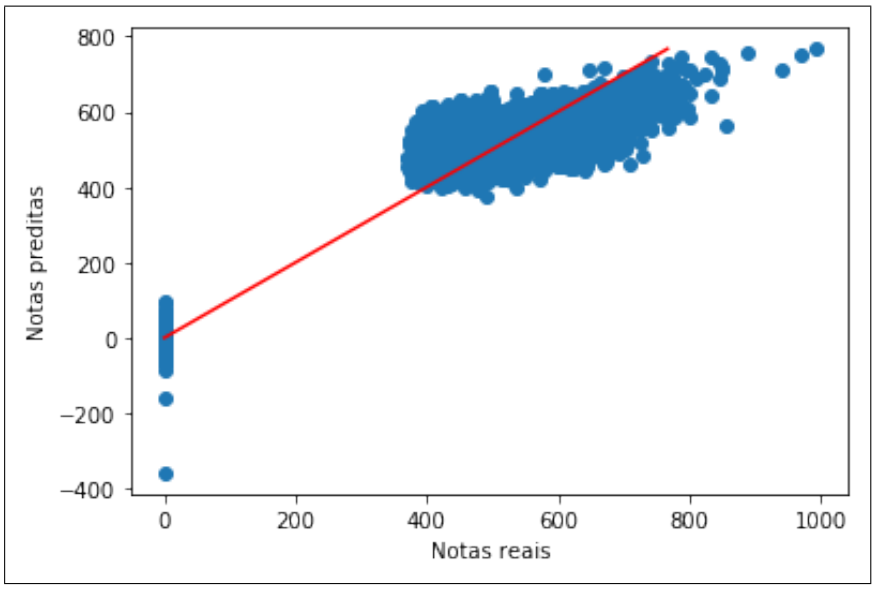

Figura 9 – Gráfico de Regressão Linear

E que cada atributo atende a equação 2. Os resultados obtidos por esse modelo podem ser vistos na seção 7 onde foi comparado com os outros três métodos.

5.1.2 Multilayer Perceptron Regressor

A arquitetura da rede consiste em 23 neurônios na camada de entrada (número de instâncias conhecidas do problema), 12 neurônios da camada oculta e 1 na camada de saída (nota de matemática prevista). A função ReLU $max(0, x)$ foi usada como a função de transferência nas camadas ocultas e a função identidade foi usada na camada de saída. O uso da função identidade na saída da rede geralmente é feito para que saídas com valores reais sejam obtidas que é o caso das notas de matemática. Se a função logística fosse empregada as saídas seriam alocadas no intervalo [0, 1].

A função de ativação é a transformação não linear feita através de todas as instâncias de entrada. Uma rede neural sem a função de ativação é simplesmente um modelo de regressão linear, logo, a função de ativação torna a rede neural capaz de aprender e executar tarefas mais complexas. A escolha da função ReLU como função de ativação das camadas internas foi devida à vantagem dela em relação as outras pois ela não ativa todos os neurônios ao mesmo tempo, o que significa que para a função ReLU se a entrada for negativa ela será convertida em zero e o neurônio não será ativado, tornando a rede esparsa, eficiente e fácil para o computador processar os dados.

A partir das outras instâncias para a qual o *Multilayer Perceptron* funcionará como um mecanismo de previsão. Uma das instâncias de entrada geralmente é fixada em um valor unitário para produzir o mesmo efeito de ter um limite nos neurônios da próxima camada. A otimização da rede é discutida abaixo.

O objetivo é que a rede aprenda a produzir as notas de matemática na saída Y quando apresentado com os valores conhecidos X. Seja U os pesos nas linhas que conectam a entrada às camadas ocultas; U é uma matriz 2312. W o peso nas linhas que conectam as camadas ocultas à camada de saída; W é uma matriz 121. Os pesos que ligam as camadas ocultas são dados pelo produto entre $x_i'U$ (onde $'$ representa a transposta) e na saída das camadas ocultas, tem a função de ativação ReLU. Para determinados valores de Y e atributos conhecidos X_ij, buscam-se os valores de W e de U iterativamente, uma vez que a cada etapa as derivadas parciais da função de perda com relação aos parâmetros do modelo são calculadas para atualizar os parâmetros. Ele também pode ter um termo de regularização adicionado à função de perda que reduz os parâmetros do modelo para evitar sobreajuste. A previsão das notas de matemática em relação a nota de redação pode ser vista na imagem 10, essa distribuição é similar nos outros atributos mas por questão computacional não foi possível plotar.

5.1.3 Nearest Neighbors Regressor

A instância de saída é excluída das instâncias de referência/conhecidas e para as quais as estimativas são calculadas. O método *Nearest Neighbors* é baseado na estimativa do vizinho mais próximo ponderada à distância, onde k posições mais semelhantes são usadas para prever a distribuição no bairro da instância alvo. Antes que o método do vizinho mais próximo possa ser aplicado, as seguintes variantes devem ser decididas:

1. A distância a ser usada para encontrar os pontos de referência mais semelhantes;
2. O número de vizinhos mais próximos;
3. A função peso para ponderar as instâncias conhecidas.

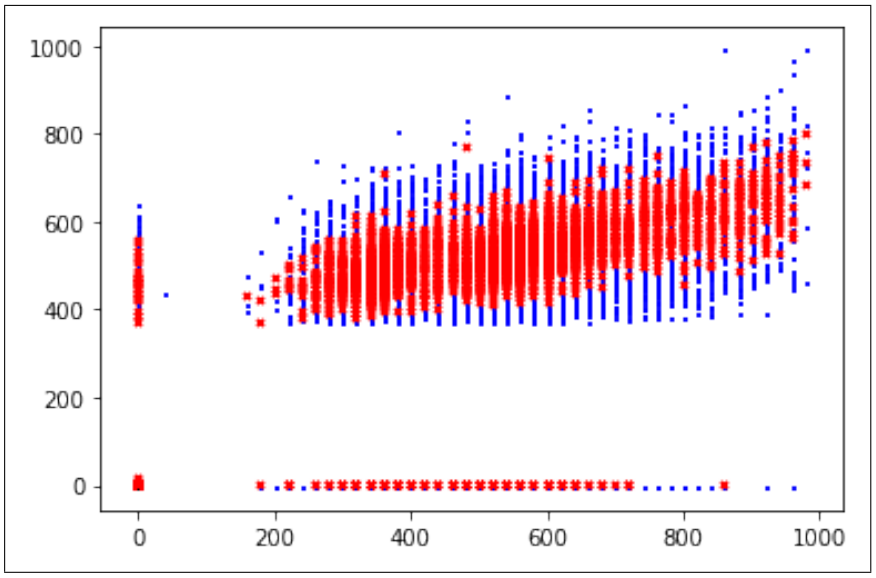

Figura 10 – Distribuição da previsão da nota de matemática(azul) x nota da redação (vermelho)

A distância escolhida para o *Nearest Neighbors Regressor* padrão e única é a **minkowski** que é equivalente à métrica euclidiana mais utilizada no algoritmo *k-nearest neighbors*. Nessa métrica, X é assumido como uma matriz de distância, a distância representa o quão similares são os atributo entre si.

O número de vizinhos mais próximos para o problema foi feito através do cálculo do erro/Curva do Cotovelo para os conjuntos de treinamento e teste e pode ser vista na figura 11.

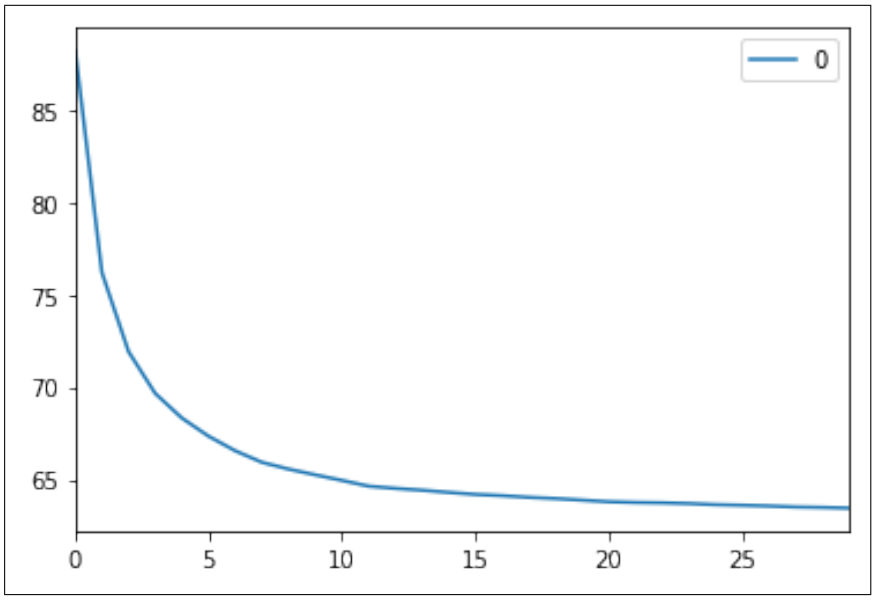

Figura 11 – Curva do cotovelo para escolha do melhor valor de K

Para um valor muito baixo de k (por exemplo, k = 1), o modelo superajusta os dados de treinamento, o que leva a uma alta taxa de erro no conjunto de validação. Em contrapartida, um valor alto de k, o modelo tem um desempenho ruim tanto no conjunto de treinamento quanto no de teste. O cotovelo de erro de validação atinge o final da curva no valor de k = 9 este valor de k é o valor quase ótimo, pois não foi encontrado o valor mínimo de erro para o conjunto de treinamento dentro do intervalo $k = [1, 30]$. Para decidir o melhor valor de k sem traçar a curva do cotovelo, foi feito também o *GridSearch* e o valor de k encontrado foi igual a 9, esse foi o valor final utilizado no modelo.

A função peso utilizada na previsão utilizou os seguintes parâmetros:

- Uniformidade: Pesos uniformes onde todos os pontos em cada bairro são ponderados igualmente;
- Distância: Vizinhos mais próximos de um determinado ponto terão uma influência maior do que vizinhos mais distantes.

Essa função peso com esses parâmetros é única para o algoritmo utilizado, logo foi a selecionada. Na seção 7 possui a comparação entre o *Nearest Neighbors Regressor* e todos os métodos descritos.

5.1.4 Árvore de Regressão

Nas árvores de regressão, cada nó terminal ou folha contém uma constante (geralmente, uma média) ou uma equação para o valor previsto de um determinado conjunto de dados. As árvores de decisão podem ser aplicadas a problemas de regressão, usando a *DecisionTreeRegressor* (DTR) do *Sklearn*, esse foi o algoritmo utilizado para o banco de dados.

A árvores de regressão utilizada no problema usa o erro quadrático médio (MSE) para decidir dividir um nó em dois ou mais sub nós. O algoritmo primeiro escolhe um valor e divide os dados em dois subconjuntos, para cada subconjunto, ele calcula o MSE separadamente. A árvore escolhe como resultado o menor valor do MSE.

Adiante está com mais detalhes a divisão decidida para a árvore de regressão. A primeira etapa para criar a árvore é gerar a primeira decisão binária:

- Escolhe-se uma variável e o valor para dividir de forma que os dois grupos sejam tão diferentes um do outro quanto possível;
- Vê qual é o melhor valor possível para cada variável;
- Para determinar o melhor, pega-se a média ponderada de dois novos nós ($mse * num_amostras$).

Para recapitular, agora tem-se:

- Um único número que representa a qualidade da divisão, que é a média ponderada dos erros quadráticos médios dos dois grupos que se criam.

- Uma maneira de encontrar a melhor divisão é tentar todas as variáveis e todos os valores possíveis dessa variável e ver qual variável e qual valor se dá uma divisão com a melhor pontuação.

Como na configuração de classificação, o método de ajuste assumirá como matrizes de argumento X e y, apenas que, neste caso, y terá valores de ponto flutuante em vez de valores inteiros.

Esta é a totalidade da criação de uma árvore de regressão e irá parar quando tiver alguma condição de parada (definida por hiperparâmetros, no problema foi utilizada a profundidade máxima da árvore - $max\_depth = 5$) for atendida, como por exemplo, atingir o limite solicitado ou quando os nós folhas têm apenas um valor neles (nenhuma divisão adicional é possível, MSE para os dados de treinamento será zero).

Foi necessário definir restrições na profundidade da árvore (profundidade vertical), pois sem esse ajuste o modelo deu um MSE zero no conjunto de treinamento, que no pior caso, acabou fazendo uma folha para cada observação, desse modo, foi necessário prevenir esse sobreajuste ao treinar a árvore. O valor encontrado foi definido levando em consideração que uma profundidade mais alta permite que o modelo aprenda relações muito específicas para uma amostra particular e quanto mais alto pode levar ao sobreajuste, portanto, foi ajustado e testado valores menores, iniciando em 2 e indo até 30, o valor definido ficou em 5, como pode ser visto nas figuras 12 e 13.

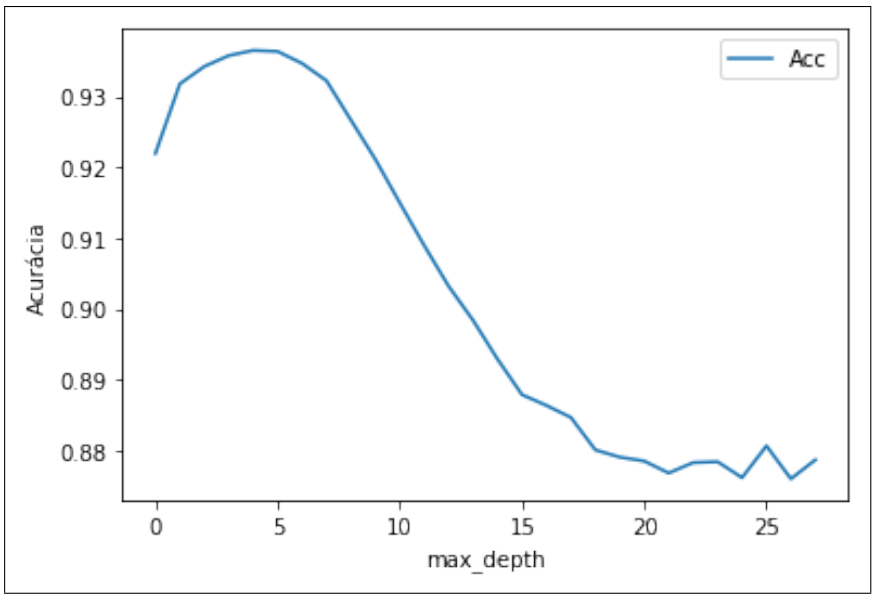

Figura 12 – Acurácia com max_depth = [2,30]

Parece que neste cenário o erro aumenta quando se aumenta a profundidade da árvore, o melhor valor com maior acurácia e menor erro foi o escolhido.

O número de instâncias a serem consideradas para a divisão são selecionadas aleatoriamente pelo algoritmo. Como regra geral, utiliza-se a raiz quadrada do número total de instâncias, mas deve-se verificar até 30-40% do número total de instâncias, pois valores mais altos também levam ao sobreajuste. Na figura 14 está a árvore de regressão após os ajustes necessários.

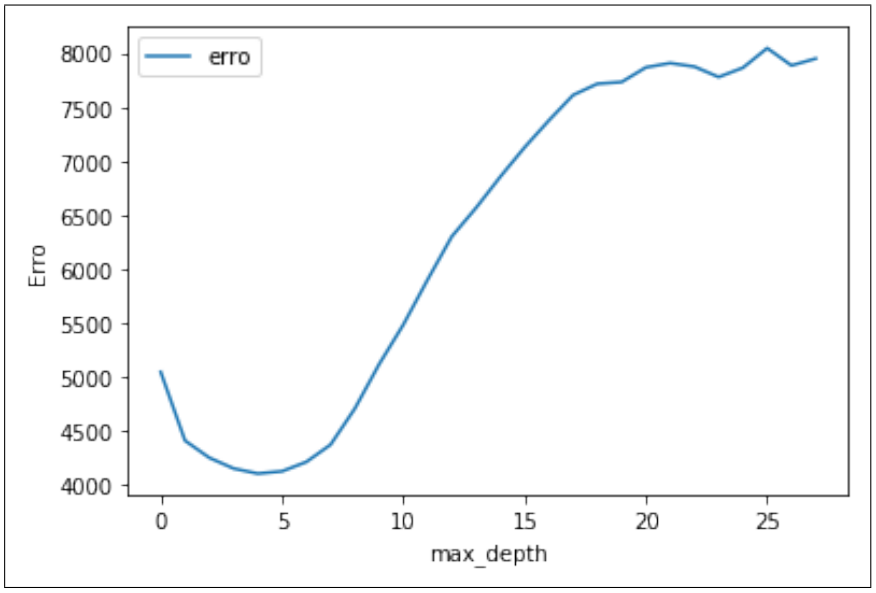

Figura 13 – Erro com max_depth = [2,30]

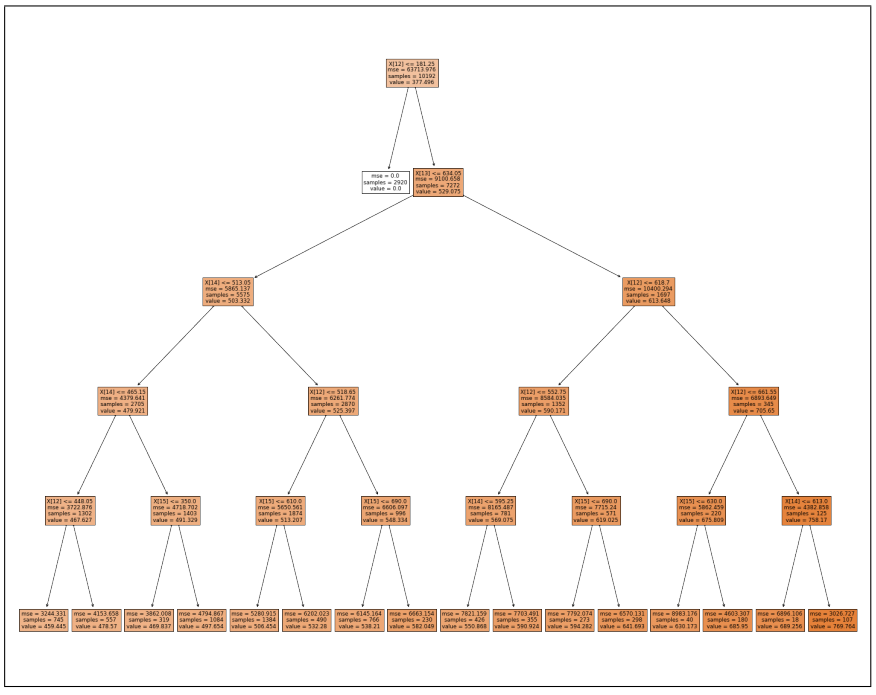

Figura 14 – Árvore de Regressão com max_depth = 5

Apenas ajustando a profundidade da árvore já se evitou o sobreajuste, então não foi necessário utilizar a poda, pois a árvore já está bem simplificada. Qualquer ajuste adicional não agregaria melhoras à árvore de regressão. Os resultados adquiridos e comparados aos outros modelos estão na seção 7.

5.2 Testes estatísticos dos modelos supervisionados

O teste de Wilcoxon não foi projetado para comparar múltiplas variáveis aleatórias. Portanto, ao comparar vários métodos de regressão ou classificação, uma abordagem "intuitiva"seria aplicar o teste de Wilcoxon a todos os pares possíveis. Porém, quando múltiplos testes são conduzidos, alguns deles rejeitarão a hipótese nula (DEMŠAR, 2006). Para a comparação de múltiplos métodos, Demšar recomenda o teste de Friedman. O teste de Friedman ordena os algoritmos em ordem decrescente para cada conjunto de dados com relação a seus desempenhos. Sua hipótese nula (H_0) afirma que todos os algoritmos são equivalentes e seus resultados médios são iguais.

Para o teste, utilizou-se a acurácia dos algoritmos considerando dois dos quatro conjuntos de teste, pois o algoritmo MLPR e KNNR por questões de hardware não foi possível obter os resultados para a base original e base reduzida 1. O teste de Friedman gerou um valor $p = 0,112$, considerando um nível de significância $\alpha = 0,05$, pode se concluir que os desempenhos dos métodos são equivalentes, pois o p-valor é maior que α, logo não deve-se rejeitar a hipótese nula.

O teste de Friedman produz um valor p muito pequeno. Para muitos níveis de significância α pode-se concluir que o desempenho de todos os algoritmos são não equivalentes.

Aplicando o teste Wilcoxon aos pares, MLPR com KNN (2 bases de dados) e RL com Árvore de Regressão (4 bases de dados), os valores para o teste foram: $p-valor = 0.179$ e $p - valor = 0.065$, respectivamente.

5.3 Modelos Não Supervisionados

5.3.1 K-Means

Para execução do k-means convencional são feitos apenas alguns passos. O primeiro passo é eleger aleatoriamente o valor de k centroides, onde k é igual ao número de clusters que será escolhido. Os centroides representam o centro de um cluster.

O corpo principal do algoritmo funciona por um processo de dois passos denominados expectativa e maximização. O passo da expectativa atribui cada ponto ao seu centroide mais próximo. Em seguida, o passo de maximização calcula a média de todos os pontos para cada cluster e define um novo centroide.

A etapa de inicialização aleatória faz com que o algoritmo k-means seja não-determinístico, o que significa que as atribuições do cluster irão variar se você executar o mesmo algoritmo duas vezes no mesmo conjunto de dados. Nesse trabalho foram executadas cinco inicializações variando o valor do *seed* entre 0 e 100.

Para escolher o número apropriado de clusters foram utilizados quatro algoritmos, o método do joelho (Elbow Method) e os índices Silhouette, Davies Bouldin (DB) e Correctly Rand (CR). Na Figura 15 mostra o Elbow Method após a execução do k-means variando o k entre 2 e 20, incrementando k a cada iteração e registrando o SSE.

Ao plotar o SSE como uma função pelo número de clusters, observa-se que o

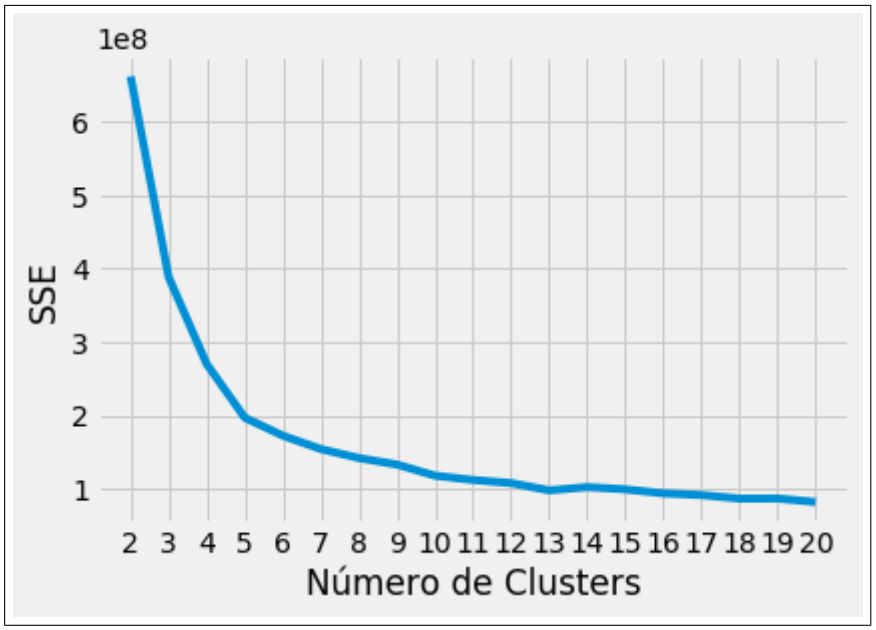

Figura 15 – Curva do cotovelo para escolha do melhor valor de clusters

SSE continua diminuindo conforme aumenta-se o k. À medida que mais centroides são adicionados, a distância de cada ponto até seu centroide mais próximo diminui. Há um ponto ideal onde a curva SSE começa a dobrar, conhecido como ponto do cotovelo. O valor k deste ponto é considerado uma compensação razoável entre o erro e o número de clusters. No ponto da figura 15, o cotovelo está localizado em k = 5.

O índice Silhouette é uma medida de coesão e separação do cluster que quantifica o quão bem um ponto se encaixa em seu cluster atribuído com base em dois fatores:

1. Quão próximo o ponto está de outros pontos no cluster;
2. A que distância o ponto está de pontos em outros clusters.

Os valores do índice variam entre -1 e 1. Números maiores indicam que as amostras estão mais próximas de seus clusters do que de outros clusters.

Na implementação do *scikit-learn* para o índice de validação Silhouette, o índice médio de todas as amostras é resumido em uma pontuação. A função de pontuação precisa de no mínimo dois clusters ou gerará uma exceção. Na figura 16 mostra o resultado da implementação.

O maior índice no Silhouette foi com 2 clusters. Na figura 17 está o índice DB gerado para os valores de k entre 2 e 20.

Nesse índice o menor valor define o número de clusters exatamente o inverso do Silhouette, resultando também em 2 clusters. O último índice é mostrado na Figura 18.

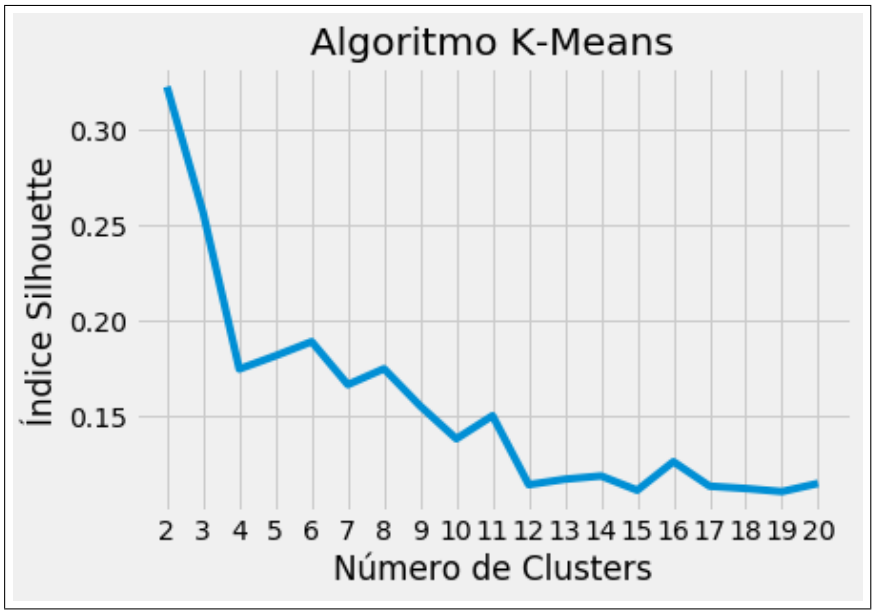

Figura 16 – Índice Silhouette aplicado no K-means

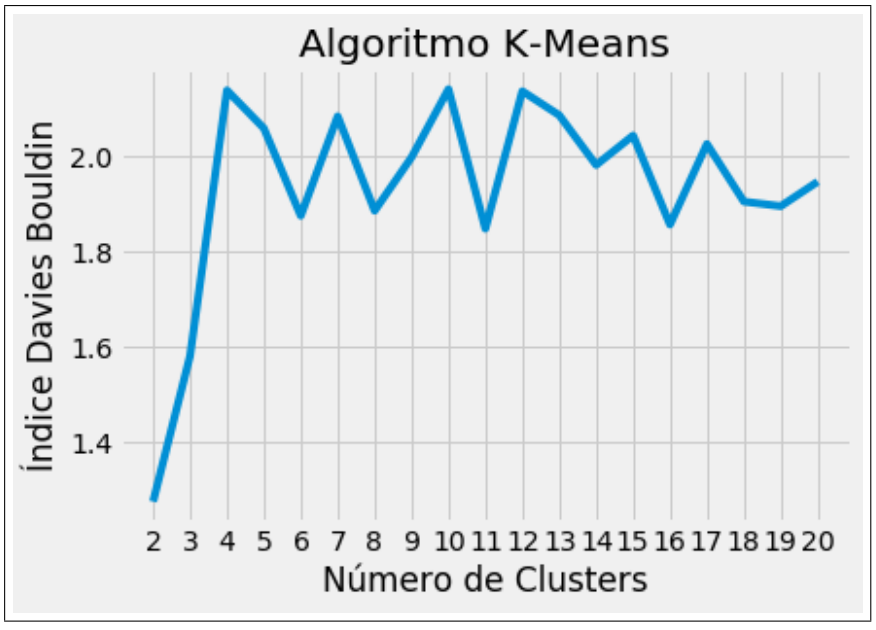

Figura 17 – Índice DB aplicado no K-means

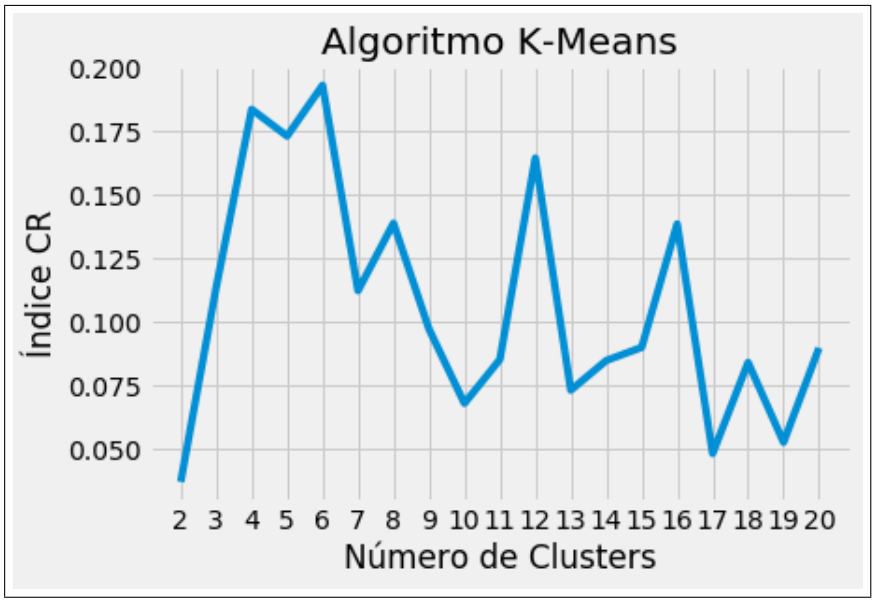

Figura 18 – Índice CR aplicado no K-means

No índice CR como os rótulos de verdade são conhecidos, é possível usar uma métrica de agrupamento que considera os rótulos da avaliação. Utilizando a implementação do *scikit-learn* do índice CR. Ao contrário dos índices Silhouette e DB, o CR usa atribuições das classes definidas (5 classes descritas na seção 3) para medir a similaridade entre rótulos verdadeiros e previstos.

Os valores de saída do CR variam entre -infinito e 1. Uma pontuação próxima a 0 indica atribuições aleatórias e uma pontuação próxima a 1 indica clusters perfeitamente rotulados. Na figura 18 o maior valor adquirido foi 12 clusters.

5.3.2 Hierárquico Aglomerativo

Um tipo hierárquico de agrupamento aplica o método "de cima para baixo"ou "de baixo para cima"para agrupar os dados. Aglomerativo é um método de agrupamento hierárquico que aplica a abordagem "ascendente"para agrupar os elementos em um conjunto de dados. Neste método, cada elemento inicia seu próprio cluster e se funde progressivamente com outros clusters de acordo com determinados critérios.

O dendrograma da figura 19 permitiu reconstruir o histórico de fusões, que resultou no agrupamento hierárquico de baixo para cima dos dados.

Por meio do dendrograma é possível notar a formação de dois grupos. O método utilizado para o agrupamento hierárquico é o *Ward* pois é um dos métodos que minimiza o quadrado da distância euclidiana às médias dos grupos utilizando a variância para gerar grupos e a distância euclidiana é uma das mais utilizadas em algoritmos de agrupamento. Calculando para cada objeto o quadrado da distância euclidiana no grupo, depois soma-se todos os objetos. Em cada passo combinam-se os dois grupos que possuem a menor variância entre si.

A Figura 20 mostra a aplicação do método *ward* utilizando o índice Silhouette.

Com 2 grupos o valor do índice silhouette foi o mais próximo de 1, logo é a melhor

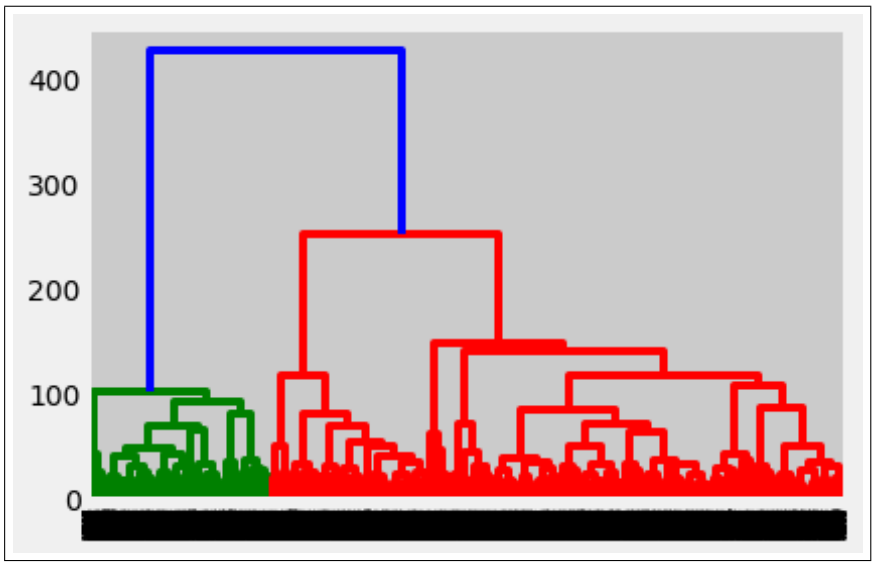

Figura 19 – Dendrograma dos dados

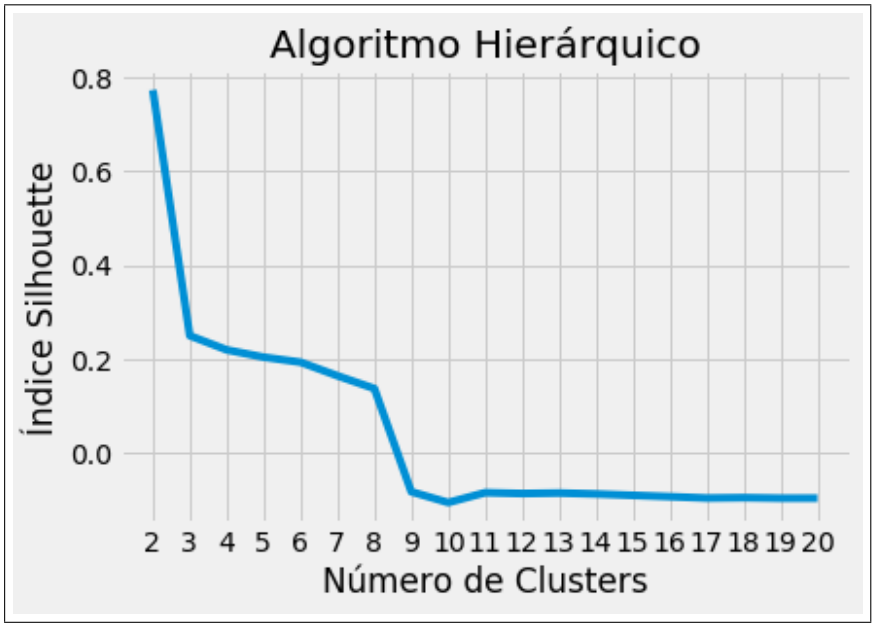

Figura 20 – Índice silhouette aplicado no Hierárquico

partição.

Na Figura 21 tem o índice DB aplicado ao método *ward* do hierárquico aglomerativo.

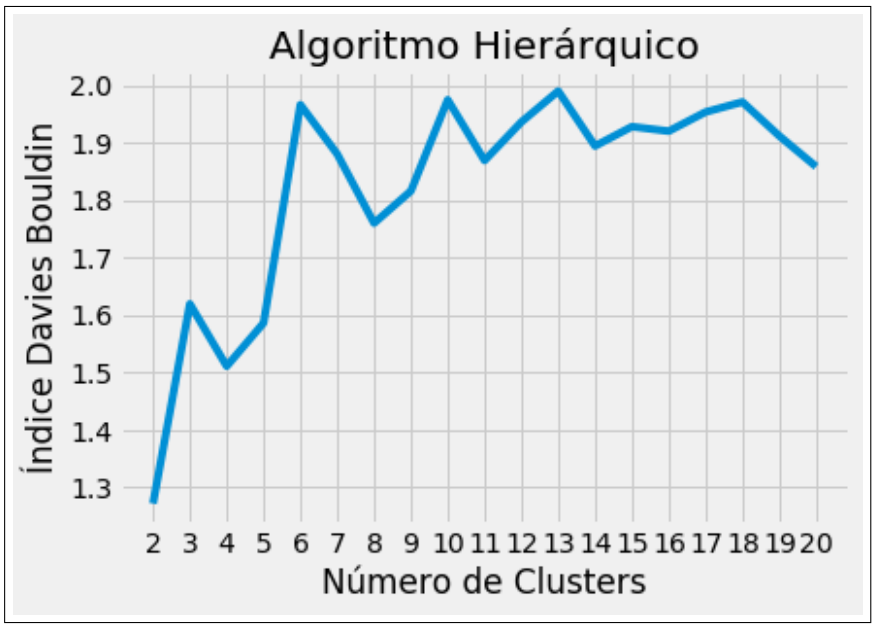

Figura 21 – Índice DB aplicado no Hierárquico

Nesse índice os valores são adquiridos por meio da similaridade relativa entre dois grupos. O comportamento desse índice é o inverso do Silhouette mas adquiriu o mesmo resultado em termo de quantidade de clusters = 2.

E por último o índice CR na figura 22.

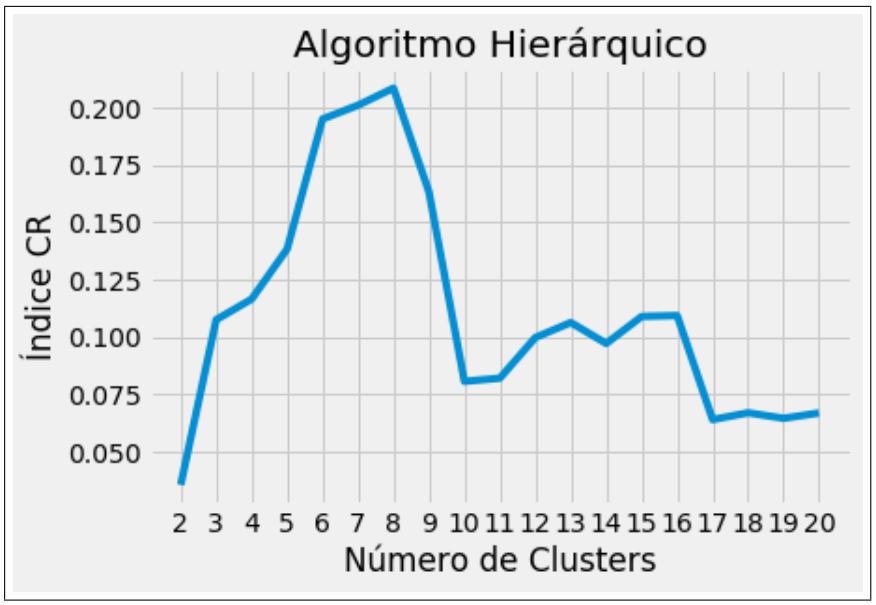

Figura 22 – Índice CR aplicado no Hierárquico

O número de grupos mais adequado no CR ficou em 8 clusters. O valor do índice CR nos dois algoritmos Kmeans e Hierárquico divergiu dos outros dois índices pois os rótulos de verdade são conhecidos e o método não supervisionado obteve uma baixa acurácia em comparação com o método supervisionado e quando utiliza o método CR e faz a avaliação com os rótulos preditos e os verdadeiros dá diferença em relação aos outros dois índices que utilizam apenas os rótulos preditos.

5.4 Testes estatísticos modelos não supervisionados

Normalmente, ao comparar dois métodos, a hipótese nula afirma que seus desempenhos são equivalentes. Para esta situação, Demšar 2006 recomenda o teste de Wilcoxon. Na tabela 5 estão os p-valores do teste estatístico para os modelos Kmeans e Hierárquico para cada índice.

Tabela 5 – Teste de Wilcoxon para dois métodos

Índice	**p-valor**
Silhouette	0.01758
BD	0.00549
CR	0.546

O teste de Wilcoxon produziu um valor $p = 0.01758$ para o índice Silhouette, $p = 0.00549$ para o índice DB e $p = 0.546$ para o índice CR. Considerando um nível de significância α de $0,05$, pode-se concluir que os desempenhos de kmeans e hierárquico não são equivalentes (rejeitar a hipótese nula) para os índices Silhouette e DB, já os desempenhos dos métodos com o índice CR são equivalentes.

Considerando que a hipótese nula foi rejeitada, geralmente tem-se dois cenários para um teste post-hoc (DEMŠAR, 2006):

- Todos os métodos são comparados entre si. Nesse caso, aplica-se o teste post-hoc de Nemenyi;
- Todos os métodos são comparados a um método de controle. Neste cenário, aplica-se o teste post-hoc de Bonferroni-Dunn.

O teste de Nemenyi tem a vantagem de ter um diagrama associado para representar os resultados da comparação. O diagrama pode ser visto na figura 23.

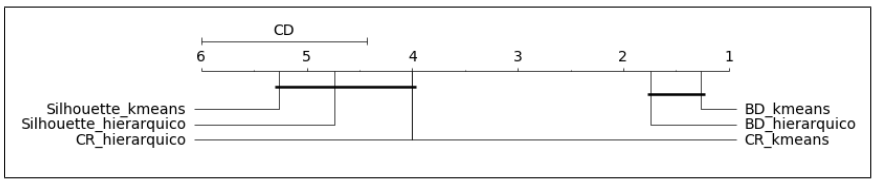

Figura 23 – Diagrama de Nemenyi

Neste gráfico, cada algoritmo é colocado em um eixo de acordo com seu resultado médio. Em seguida, os algoritmos que não apresentam diferenças significativas são agrupados usando uma linha horizontal. O gráfico também mostra o tamanho da diferença crítica

necessária para considerar dois algoritmos como significativamente diferentes. Quaisquer dois algoritmos cuja diferença de desempenho seja maior que a diferença crítica são considerados significativamente diferentes. Logo, os algoritmos kmeans e hierárquico não possuem diferenças significativas, pois estão agrupados pela linha horizontal.

6 Comitês de máquina

6.1 Bagging

Bagging combina vários algoritmos de aprendizagem de forma a reduzir a variação das estimativas. Para o Bagging foi feita a tabela 6 contendo 10, 15 e 20 regressores base. Foram utilizados os algoritmos de regressão: Regressão linear (RL), Multi-layer Perceptron regressor (MLPR), DecisionTreeRegressor (DTR) e KNeighborsRegressor (KNNR).

Tabela 6 – Score do Bagging utilizando 4 algoritmos de regressão

Estratégia	**10**	**15**	**20**	**Média**
MLPR	69,4	56,1	60,8	62,1
DTR	93,9	93,9	93,9	93,9
KNNR	93,5	93,5	93,5	93,5
RL	93,7	93,7	93,7	93,7
Média	87,6	84,3	85,4	85,8

6.2 Boosting

Boosting é um algoritmo que converte algoritmos de apredizagem fracos em fortes e são menos afetados pelo problema de overfitting. A tabela 7 a seguir possui os resultados adquiridos pelo algoritmo de boosting com os quatro algoritmos de regressão RL, MLPR, DTR e KNNR.

Tabela 7 – Score do Boosting utilizando 4 algoritmos de regressão

Estratégia	**10**	**15**	**20**	**Média**
MLPR	9,4	93,9	93,9	65,7
DTR	93,9	93,9	93,8	93,86
KNNR	92,5	92,5	91,9	92,3
RL	93,5	93,2	93,2	93,3
Média	72,3	93,3	93,2	86,3

6.3 Bagging e Boosting

Os *scores* adquiridos pelos algoritmos individualmente foram:

- RL - 93,7
- MLPR - 93,9
- DTR - 93,7
- KNNR - 93,4

Esses valores foram para a redução da base com 20.000 instâncias que também foi utilizada para os testes com bagging e boosting. O resultado do Bagging RL comparada com o RL individualmente foi a mesmo, o MLPR no Bagging teve um desempenho bem inferior, o DTR teve um desempenho um pouco melhor no Bagging e o KNNR também melhorou de 93,4 para 93,5 no Bagging.

O algoritmo Boosting teve como resultado para o RL um valor inferior ao individual, o MLPR no Boosting teve um valor muito baixo em comparação ao individual, o DTR teve um leve aumento no desempenho e o KNNR individualmente teve um score melhor que no boosting.

Para o algoritmo boosting o único algoritmo que teve uma melhora foi o DTR, no algoritmo de Bagging teve uma melhora de desempenho apenas no DTR e KNNR. Ambos tiveram baixo desempenho no algoritmo MLPR em comparação com o individual.

Pode-se observar na última linha das tabelas 6 e 7 os resultados adquiridos pelos tamanhos e para o Bagging e Boosting foi o 15. A última coluna das tabelas está a média dos modelos e o modelo com melhor desempenho para ambos os comitês foi o DTR.

6.4 Stacking

Stacking é uma técnica de aprendizagem que combina múltiplas previsões de modelos de regressão base em um novo conjunto de dados. Esses novos dados são tratados como dados de entrada para outro regressor.

6.4.1 Homogêneo

Para o stacking homogêneo foi feita a tabela 8 do mesmo modo que as tabelas do Boosting e Bagging.

Tabela 8 – Score do Stacking Homogêneo utilizando 4 algoritmos de regressão

Estratégia	**10**	**15**	**20**	**Média**
MLPR	91,2	91,3	91,3	91,2
DTR	93,4	93,3	93,4	93,36
KNNR	90.4	90,5	90.6	90,5
RL	91,2	91,3	91,3	91,2
Média	91,55	91,6	91,6	91,56

O Stacking homogêneo teve um melhor resultado em comparação aos outros dois comitês no modelo MLPR, nos outros modelos o resultado foi inferior.

6.4.2 Heterogêneo

A tabela 9 do Stacking heterogêneo foram escolhidos três regressores base, os que possuíram melhor desempenho e menor custo computacional, RL, DTR e KNNR. Foram feitas quatro configurações:

1. 50% do método RL com 50% do método DTR;
2. 50% do método RL com 50% do método KNNR;
3. 50% do método DTR com 50% do método KNNR;

4. 33% do método RL, 33% do método DTR e 33% do método KNNR.

Tabela 9 – Score do Stacking Heterogêneo utilizando 3 algoritmos de regressão

Configurações	**Score**
1	93,9
2	93,9
3	93,7
4	94,5
Média	94

O melhor desempenho foi com a combinação dos 3 métodos e o algoritmo Stacking heterogêneo teve uma média de desempenho melhor do que o Bagging e o Boosting.

6.5 Seleção de atributos em comitês

Aplicando um filtro randômico com 50% dos dados da base original foi criado 9 bases diferentes e testado o Bagging com dois algoritmos de regressão, RL e DTR. Os resultados dos testes pode ser observado na tabela 10.

Tabela 10 – Score utilizando 2 algoritmos de regressão para diferentes bases

Bases	**RL**	**DTR**
1	93,4	93,8
2	93,7	93,9
3	93,6	93,9
4	93,7	94
5	93,8	94,2
6	93,7	94,1
7	94,4	94,7
8	94,1	94,4
9	94	94,2
Média	93,82	94,13

Avaliando a variabilidade do modelo RL foi feito o intervalo de confiança para a média para as 9 bases criadas, com um intervalo de confiança em 95%, valor de z = 1,96, e média 93,82 o acerto médio do modelo deve variar entre 93,5 e 94,05. A variabilidade do modelo DTR o acerto médio do modelo deve variar entre 93,9 e 94,3. Comparando com o algoritmo Bagging da tabela 6 esse método teve melhor desempenho.

6.5.1 Teste Estatísticos para os comitês

Utilizando o teste de Wilcoxon para as 9 bases criadas foi obtido o $p - valor = 0.00752$ demonstrando que o AD e o RL são de distribuições diferentes, indicando uma evidência muito forte contra a hipótese nula.

O teste de Friedman aplicado às médias de acurácia dos quatro algoritmos de regressão para o Bagging, Boosting e Stacking, resultaram em um $p - valor = 0.368$ indicando que os resultados dos comitês são equivalentes e não deve-se rejeitar a hipótese nula.

Para calcular o testes de hipóteses para essas bases, foi necessário o uso da média (XeY) e desvio padrão (s_xes_y) dos dois métodos. Aplicando os valores adquiridos na equação 3.

$$Estatistica\_de\_teste = \frac{(X - Y)}{\sqrt{\frac{s_x^2}{n} + \frac{s_y^2}{n}}} \tag{3}$$

Sendo, $X = 94,13$, $Y = 93,82$, $s_x = 0,282$, $s_y = 0,299$ e o número de bases $n = 9$. Buscando o resultado da estatística de teste na tabela Z o valor encontrado é 0,9981. Logo o valor de p é igual a 0.9981, para o teste bicaudal o valor deve ser multiplicado por dois, $2 * p = 0.9962$. Escolhendo um valor de alfa = 0,05, assim o p-valor é maior que o nível de significância alfa. Valores altos de p-valor evidenciam que a hipótese nula é verdadeira e não existe uma diferença estatística significante entre DTR e RL.

6.6 Discussão

1. Qual o impacto do número de regressores base em comitês homogêneos?
2. Qual o impacto do número de regressores base em comitês heterôgeneos?
3. Qual o impacto do tipo de regressores base em comitês homogêneos?
4. Qual o impacto do tipo de regressores base em comitês heterogêneos?
5. Qual a melhor escolha de tipo de comitê, homogêneo ou heterogêneo?
6. Qual o impacto da seleção de atributos nos comitês de regressores?
7. Existe realmente uma relação entre diversidade e precisão nos comitês de regressores?

Em resposta aos questionamentos acima o impacto do número de regressores nos comitês homogêneos e heterogêneos não foi muito significante, a diferença de desempenho com o aumento dos regressores não impactou quase nada no desempenho, apenas no Boosting MLPR de 10 regressores para 15 houve uma diferença significativa. Comparando os tipos de regressores e seu impacto no desempenho dos comitês, o único que teve um desempenho mais baixo que os outros foi o MLPR além de ter um maior custo computacional. O comitê heterogêneo obteve melhor desempenho, logo seria a melhor escolha. A seleção de atributos teve um impacto muito melhor no desempenho, o que demonstra que alguns atributos tem maior influência na previsão das notas de matemática. Para o Boosting e Stacking existiu a relação pois quando a diversidade de regressores era 10 a precisão era menor, mas quando aumentou para 15 ou 20 as precisão aumentou. Mas para o Bagging foi o inverso, menor diversidade maior precisão.

7 Resultados

Os resultados no banco de dados para os métodos de pré-processamento conhecidos utilizados pode ser visualizado na tabela 11.

Os modelos supervisionados foram aplicados em quatro configurações da base de dados, a base de dados original após a etapa de pré-processamento com 3.341.297 instâncias, a base de dados reduzida para 1.000.000 de instâncias, 100.000 instâncias e

Tabela 11 – Resultado dos métodos de pré processamento

Métodos de pré processamento	Número de Instâncias	Número de Atributos
Base original	3.389.832	76
Manual	66.663	24
Ausência de valores	3.341.297	23
Filtro de correlação	3.341.297	7
Wrapper	3.341.297	22
PCA	3.341.297	17

20.000 instâncias. A tabela 12 possui a acurácia de previsão das notas de matemática em cada um dos métodos de regressão escolhidos.

Tabela 12 – Resultados da Regressão

Base de Dados	Regressão Linear	MLP Regressor	KNN Regressor	Árvore Regressão
Original	94,08%	-	-	94,1%
Reduzida 1	93,05%	-	-	93,07%
Reduzida 2	93,3%	93,7%	93,2%	93,3%
Reduzida 3	93,7%	93,9%	93,4%	93,7%

A base reduzida 3 foi a melhor opção devido ao tempo de treinamento da máquina e eficiência. Alguns resultados utilizando a base original não foram possíveis de serem obtidos pois o hardware utilizado não foi suficiente para treinar a máquina.

Analisando as quatro métricas mais comuns para avaliar previsões sobre problemas de aprendizado de máquina de regressão, a tabela 13 foi elaborada com os valores resultantes.

Tabela 13 – Métricas para avaliação dos modelos de Regressão

Modelos	MAE	MSE	RMSE	R^2
MLPR	42,799	3.922,56	62,63	0,937
KNNR	43,970	4.255,01	65,23	0,934
DTR	43,767	4.153,97	64,45	0,935
RL	46,764	4.062,88	63,74	0,936

O Mean Absolute Error (MAE) é a média das diferenças absolutas entre as previsões e os valores reais. Dá uma ideia de como as previsões estavam erradas, magnitude do erro. Um valor zero indica nenhum erro ou previsões perfeitas, o MLPR teve o menor erro em comparação aos outros métodos.

O Mean Squared Error (MSE) é muito parecido com o MAE, pois fornece uma ideia geral da magnitude do erro. Obter a raiz quadrada do MSE converte as unidades de volta às unidades originais da variável de saída e pode ser significativo para descrição e apresentação. Isso é chamado de Root Mean Squared Error (RMSE). Resumidamente, RMSE é o desvio padrão dos erros que ocorrem quando uma previsão é feita em um conjunto de dados. O MLPR também teve o melhor resultado dentre os quatro algoritmos.

O R Square (R^2) é também conhecido como coeficiente de determinação. Essa métrica fornece uma indicação de quão bem um modelo se ajusta a um determinado conjunto de dados. Ele indica quão próxima a linha de regressão (ou seja, os valores previstos plotados) está dos valores de dados reais. O valor de R^2 está entre 0 e 1, onde 0

indica que este modelo não se ajusta aos dados fornecidos e 1 indica que o modelo se ajusta perfeitamente ao conjunto de dados fornecido. Todos os modelos tiveram bons resultados neste coeficiente.

Os métodos supervisionados geraram gráficos de dispersão dos dados demonstrando que há uma certa linearidade nos resultados. As figuras 24, 25, 26 e 27 mostram a distribuição dos dados previstos contra os reais.

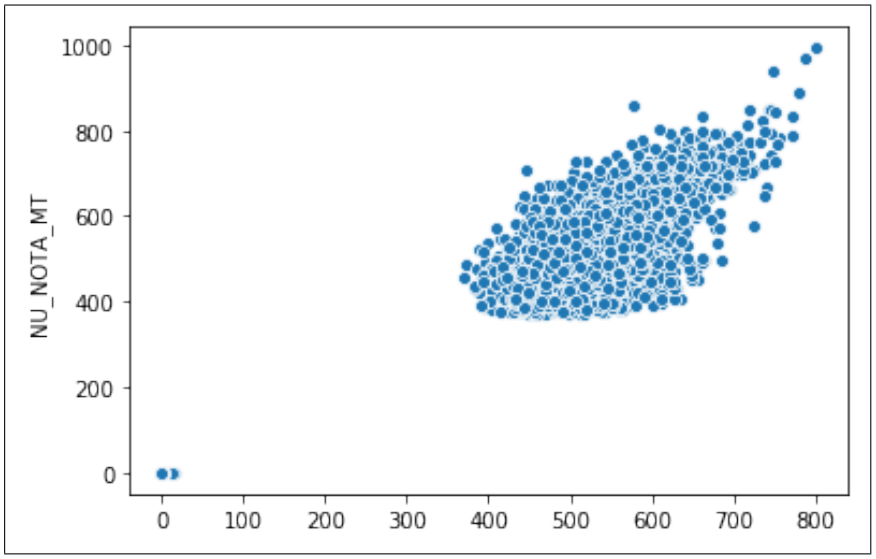

Figura 24 – Gráfico de dispersão do MLPR

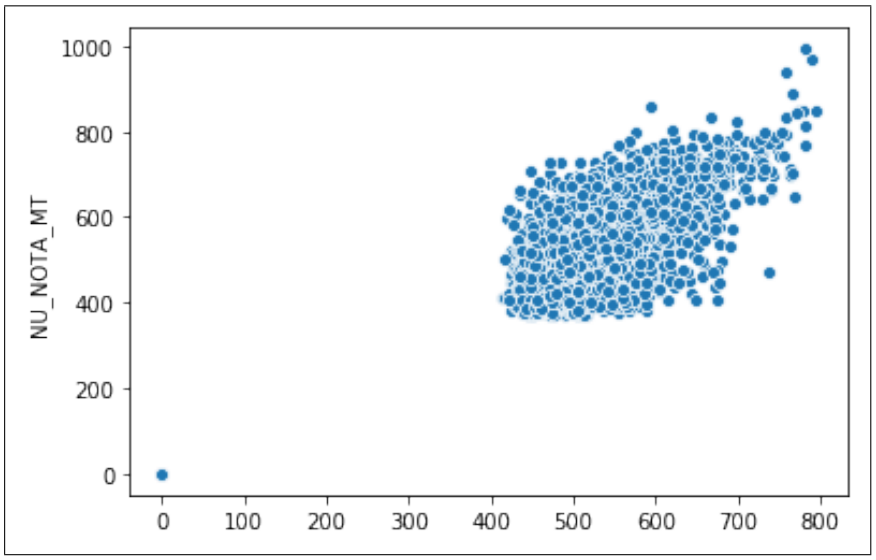

Figura 25 – Gráfico de dispersão do KNNR

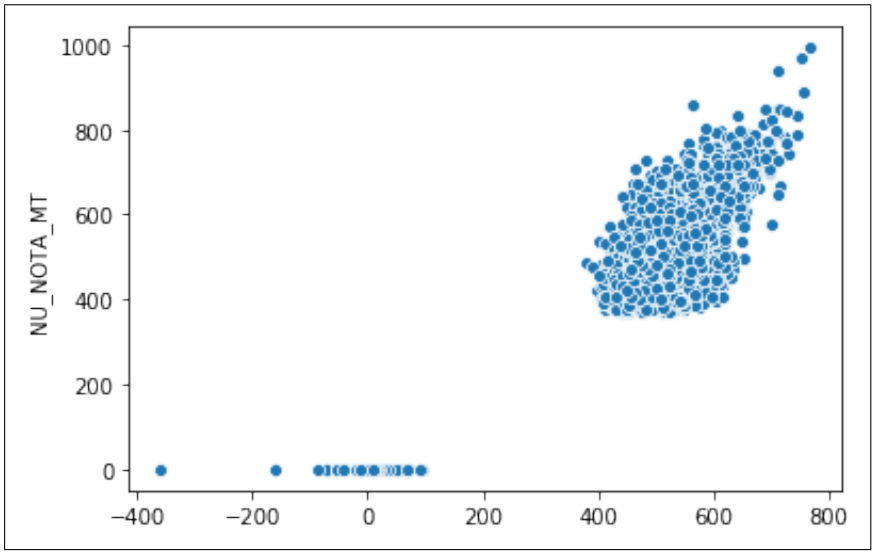

Figura 26 – Gráfico de dispersão da RL

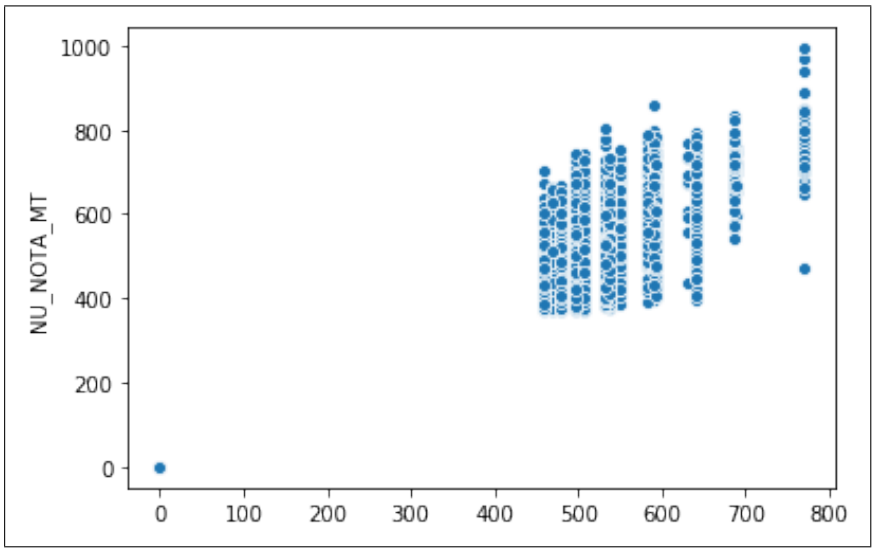

Figura 27 – Gráfico de dispersão da DTR

Nos gráficos pode-se observar a existências de alguns pontos distantes da nuvem de maioria (*outliers*), mas o algoritmo deve prever algumas notas zero devido à adição dessas notas na etapa de pré-processamento, o algoritmo que gerou *outliers* foi o RL, dando algumas notas negativas.

Para os resultados em relação aos algoritmos não supervisionados aplicados à base de dados teve como a melhor partição k-means com dois clusters já o hierárquico para a base de dados demorou mais tempo para ser treinado. Os dois algoritmos utilizando os índices não categorizaram os clusters como na categorização original (5 clusters), apenas o método do joelho utilizando k-means encontrou esse valor.

Os resultados adquiridos com os métodos Boosting, Bagging e Stacking foram semelhantes. Embora não tenha sido significante o uso de comitês para o problema houve uma pequena melhora em alguns métodos de aprendizado de máquina, ver seção 6.

8 Conclusões

Este trabalho teve como objetivo investigar se seria factível a previsão das notas de matemática a partir de alguns dados socioeconômicos e notas das outras disciplinas dos microdados do ENEM 2021. Os resultados apresentados demonstraram que técnicas de aprendizagem supervisionadas podem ser empregadas para prever essas notas, demonstrando que os valores geram um padrão. Entre os métodos de aprendizagem supervisionada, o método MLPR teve o melhor desempenho em todos os aspectos e indica grande eficiência na previsão da nota de matemática.

As técnicas de aprendizagem não supervisionadas tiveram resultados mais baixos, mas foi possível detectar que os padrões podem se agrupar de modo diferente dependendo do algoritmo utilizado. Os comitês de máquina tiveram boas acurácias, mas exigiram maior processamento em comparação com os algoritmos individuais, logo não seria tão viável utilizá-los neste problema.

A redução da base de dados foi satisfatória, os resultados de acurácia se mantiveram elevados e o processamento para treinar os dados diminuiu significativamente. Os testes estatísticos detectaram que os algoritmos supervisionados geraram desempenhos equivalentes e os algoritmos não supervisionados com o teste de Wilcoxon, dependendo do índice os algoritmos Kmeans e Hierárquico possuiam desempenhos correspondentes. Para os comitês, utilizando as médias de acurácia resultantes dos quatro algoritmos para Bagging, Boosting e Stacking demonstraram equivalência em seus resultados.

Esses são resultados encorajadores em relação aos desafios para o problema em questão e indicam a importância dos atributos na determinação das notas de matemática.

8.1 Limitações

O aprendizado de máquina requer computadores com *hardware* robusto para uma quantidade grande de dados, o banco de dados do ENEM é muito grande, 1,5 Gigabytes, logo é necessário grande processamento para utilizar todos esses dados.

8.2 Trabalhos Futuros

Como sugestão de trabalho futuro a continuidade dessa pesquisa conjectura a utilização de mais dados com fatores socioeconômicos que podem possuir relevância na

nota de matemática dos estudantes e também qual a influência desses fatores em outras disciplinas do ENEM. Outra abordagem que seria interessante é buscar várias bases de dados dos ENEM anteriores, cerca de cinco anos de diferença e fazer uma análise exploratória dos dados para relacionar os padrões encontrados e se historicamente o desempenho do estudante melhorou ou piorou por influência socioeconômica.

Referências

ALVES, R. D.; CECHINEL, C.; QUEIROGA, E. Predição do desempenho de matemática e suas tecnologias do enem utilizando técnicas de mineração de dados. In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. [S.l.: s.n.], 2018. v. 7, n. 1, p. 469. Citado na página 3.

BRAVIN, G. F.; LEE, L.; RISSINO, S. das D. Mineração de dados educacionais na base de dados do enem 2015. *Brazilian Journal of Production Engineering-BJPE*, p. 186–201, 2019. Citado na página 3.

DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, v. 7, n. Jan, p. 1–30, 2006. Citado 2 vezes nas páginas 20 e 26.

ENEM/INEP. *ENEM, Exame Nacional do Ensino Médio*. BR, 2022. Disponível em: <https://enem.inep.gov.br/>. Citado na página 1.

INEP. *Microdados - INEP*. BR, 2022. Disponível em: <http://portal.inep.gov.br/microdados>. Citado 2 vezes nas páginas 2 e 4.

JINDAL, R.; BORAH, M. D. A survey on educational data mining and research trends. *International Journal of Database Management Systems*, Academy & Industry Research Collaboration Center (AIRCC), v. 5, n. 3, p. 53, 2013. Citado na página 1.

LOBO, G. D.; CASSUCE, F. C. da C.; CIRINO, J. F. Avaliação do desempenho escolar dos estudantes da região nordeste que realizaram o enem: uma análise com modelos hierárquicos. *Revista Espacios*, v. 6, 2017. Citado na página 3.

PISA. *Relatório do Programa Internacional de Avaliação de Alunos*. BR, 2018. Disponível em: <http://download.inep.gov.br/acoes_internacionais/pisa/documentos/2019/relatorio_PISA_2018_preliminar.pdf>. Citado 2 vezes nas páginas 1 e 2.

RODRIGUES, A. A.; PINTO, B. N. S.; SOUZA, V. C. de A. Análise dos resultados do enem 2009-2014 como um dos indicadores da aprendizagem de ciências da natureza nas escolas públicas de viçosa (mg). *The Journal of Engineering and Exact Sciences*, v. 2, n. 2, p. 082–094, 2016. Citado na página 3.

SIMON, A.; CAZELLA, S. Mineração de dados educacionais nos resultados do enem de 2015. In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. [S.l.: s.n.], 2017. v. 6, n. 1, p. 754. Citado na página 3.