PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

Elias da Silva Cruz

Análise Preditiva - Enem 2019
Impactos Socioeconômicos

Belo Horizonte
2021

Elias da Silva Cruz

Análise Preditiva - Enem 2019
Impactos Socioeconômicos

Trabalho de Conclusão de Curso apresentado ao Curso de Especialização em Ciência de Dados e Big Data como requisito parcial à obtenção do título de especialista.

Belo Horizonte
2021

SUMÁRIO

AGRADECIMENTO

Agradeço a Deus pela saúde e oportunidade de concluir esse trabalho.

A minha esposa Cintia Lima Cruz e meu filho Pedro Lima Cruz pela compreensão, apoio, carinho e amor.

Aos meus pais e irmãos por me acompanharem em muitos momentos importantes da minha vida.

Aos professores e a PUC Minas por todo o conhecimento partilhado durante todo o curso.

1. Introdução

1.1. Contextualização

O ENEM Exame Nacional do Ensino Médio, considerado dentre as maiores provas do país, criado em 1998 com o objetivo de avaliar o desempenho dos candidatos que tenham ou não concluído o Ensino Médio também serve como ferramenta de seleção usada pelas maiores universidades públicas e privadas. São cerca de 180 questões objetivas mais uma de redação. Essa avaliação é realizada durante 2 dias com questões divididas entre (Ciências Humanas, Ciências da Natureza, Linguagens e Códigos, Matemática).

O INEP "Instituto Nacional de Estudos e Pesquisas Anísio Teixeira" responsável por toda a logística de prova no Brasil Inteiro, fazendo com que essas provas cheguem até o seu destino com um imenso corpo de funcionários envolvidos, cuidando de toda segurança não só físicas, mas também através de monitoramento por câmeras. Após o candidato concluir a prova a mesma é transportada com segurança evitando vazamentos ou violações. As provas serão escaneadas e digitalizadas passando por um processo rigoroso de correção sendo a prova de redação corrigida de forma manual e o cartão resposta sendo scaneado por um processo automático de correção.

A partir de 2019 passou por uma reformulação com relação a correção das notas onde foi incluído uma nova metodologia bastante complexa e específica de correção. Essa metodologia considera que as questões que tiveram o menor percentual de acertos elas vão ter um impacto menor um pacto maior nota do aluno porque elas são consideradas mais difíceis e as questões que tiveram maior percentual de acertos elas são consideradas questões fáceis então ela vai ter um pacto menor nota do aluno e isso implica que com essa complexidade da metodologia o aluno ele simplesmente não consegue calcular a nota dele baseado simplesmente nos acertos.

O Enem é bastante procurado por muitos candidatos dado sua importância pois proporciona um número relevante de oportunidades de ingresso em diversas Faculdades e Universidades (Públicas ou Privadas). Há também entre os inscritos aqueles que tem a pretensão de medir seus conhecimentos "IN_TREINEIRO" através do exame. As instituições utilizam principalmente o Enem como forma de seleção além de fundo de auxílio do Governo como FIES "Fundo de Investimento Estudantil". Os resultados individuais podem ser utilizados por instituições que possuem convênios com o INEP.

Este trabalho pretende buscar identificar similaridades e diferenças entre as questões socioeconômicas com base nas respostas / notas obtidas no exame. Para tanto, obteve de

bases públicas (Ministério da Educação) e dados referente IDH através do Programa das Nações Unidas para o Desenvolvimento (PNUD). O Índice de Desenvolvimento Humano (IDH) é uma medida resumida do progresso a longo prazo em três dimensões básicas do desenvolvimento humano: renda, educação e saúde. Disponível em: **https://www.br.undp.org/content/brazil/pt/home/idh0.html.** Acesso em: 24 de out. 2021.

O IDH é peça chave nessa comparação pois trata-se da unidade de medida utilizada para aferir o grau de desenvolvimento de uma determinada sociedade tendo como base 3 pilares principais educação, saúde e renda.

1.2. O problema proposto

Objetiva se com a presente análise o impacto socioeconômico e suas influências nas notas dos candidatos assim também como a identificação de padrões e a relação das notas obtidas nos exames. Dessa forma podemos visualizar através dos dados se o desenvolvimento da região (consultados através do IDH) teve influência direta com o desempenho da nota obtida pelo candidato.

É importante mapearmos as regiões com seus índices pois entendemos que isto está diretamente ligado ao desenvolvimento de cidadãos conscientes, trazendo resultados diretamente na economia do País. Nossas habilidades e competências são fruto de uma educação que tem seu complemento na medida em que avançamos durante ciclos do nosso cotidiano.

Temos como target principal mapear a relação entre os Fatores socioeconômicos X Nota do candidato. Através dessa análise podemos observar diversos comportamentos por exemplo se a nota alcançada por candidato possui relação sobre a condição financeira ou se o fator escolaridade dos Pais tem influência direta com as notas dos filhos e até mesmo observar os comportamentos de evasão por candidato.

O dado analisado diz respeito as notas de candidatos distribuídos pelo Brasil bem como seu estado socioeconômico.

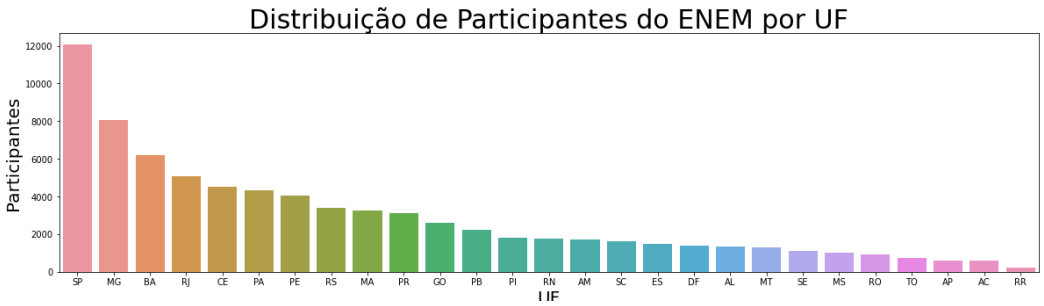

Figura 1: Distribuição de participantes por UF

Baseado nas Notas dos candidatos e utilizando como guia o dicionário de dados micro dados temos um mapeamento claro dos resultados separados por localidade sobre a qualidade do ensino médio brasileiro.

Utilizando a técnica dos 5-Ws (principais perguntas que devem ser feitas e respondidas ao investigar e relatar um fato ou situação, sendo aplicável a várias atividades profissionais), podemos organizar assim o problema para obter melhores insights do projeto:

Why? (Por quê?): A análise dos dados do INEP é importante pois o assunto abordado cria diversas oportunidades de análise principalmente no que tange a qualidade do ensino no Brasil. Os resultados mostram com que prioridades as atenções são voltadas para a Educação e o quanto isso significa para uma parcela da população e para o Governo que tem a missão de administrar o investimento para cada Estado / Município. A análise e a busca comportamentos em relação as notas obtidas.

Who? (Quem?) Os dados objeto da análise são abertos, disponibilizados pelo próprio site do INEP.

What? (O quê?): O problema proposto é a análise dos dados de condidatos do Enem 2019, assim como a previsão das notas obtidas na prova com o objetivo de detectar comportamentos socioeconômicos em relação as notas.

Where? (Onde?): Os dados geográficos são de abrangência nacional.

When? (Quando?): O período analisado é da totalidade do ano de 2019. Todo o desenvolvimento do projeto foi realizado utilizando a linguagem de programação de alto nível

Python. Escolhemos a linguagem por ter diversas bibliotecas disponíveis para análise de dados e aprendizado de máquina.

Utilizamos o ambiente Google Colab Notebook baseado na versão Python 3.7.10 da linguagem Python presente na plataforma Google Colaboratory, executado na nuvem. Jupyter Notebook é uma aplicação web (criada e mantida pela Google) que permite criar documentos utilizando códigos e textos em um mesmo Notebook.

1.3. Objetivos

Temos como objetivo destacar a relação das notas alcançadas pelos candidatos do Enem com impactos socioeconômico (IDH-REGIÃO, renda mensal por família, etc..) e se há uma influência direta com os mesmos obtendo insights dos principais comportamentos obtidos com o resultado final da análise.

2. Coleta de Dados

A fonte dos dados foi adquirida através do site do INEP porém deixou de disponibilizar os dados a partir de fevereiro 2022, com isso tomei a decisão em disponibilizar via Drive. Os dados brutos para esta análise estão disponíveis em: https://drive.google.com/file/d/1hg1PKOht8TLnaiUHNFBAgQQqUBU7To-i/view?usp=sharing.

Os dados de IDHM separados por Estados no período de 2010 foram extraídos diretamente do site https://www.kaggle.com/BrasilComCenso/atlas-idhm-brasil-1991-2000-e-2010-lat-e-long/version/1 . Realizamos a extração somente das principais colunas referente o Índice de Desenvolvimento Humano. Acesso em: 24 de out. 2021. Os formatos disponíveis estão em csv e xls.

Segue abaixo a descrição dos campos obtidos no dataset extraído do INEP.

Nome da Coluna / Campo	Descrição
CO_MUNICIPIO_RESIDENCIA	Código município
NO_MUNICIPIO_RESIDENCIA	Nome do município de residência
CO_UF_RESIDENCIA	Código da Unidade da Federação de residência
SG_UF_RESIDENCIA	Sigla da Unidade da Federação de residência
NU_IDADE	Idade do candidato

TP_SEXO	Sexo
TP_ESTADO_CIVIL	Estado Civil
TP_COR_RACA	Cor Raça
TP_NACIONALIDADE	Código do município de nascimento
TP_ST_CONCLUSAO	Situação de conclusão do Ensino Médio
TP_ESCOLA	Tipo de escola do Ensino Médio
IN_TREINEIRO	Indica se o inscrito fez a prova com intuito de apenas treinar seus conhecimentos
TP_DEPENDENCIA_ADM_ESC	Dependência administrativa (Escola)
NU_NOTA_CN	Nota da prova de Ciências da Natureza
NU_NOTA_CH	Nota da prova de Ciências Humanas
NU_NOTA_LC	Nota da prova de Linguagens e Códigos
NU_NOTA_MT	Nota da prova de Matemática
NU_NOTA_REDACAO	Nota da prova de redação
Q001	Até que série seu pai, ou o homem responsável por você, estudou?
Q002	Até que série sua mãe, ou a mulher responsável por você, estudou?
Q005	Incluindo você, quantas pessoas moram atualmente em sua residência?
Q006	Qual é a renda mensal de sua família? (Some a sua renda com a dos seus familiares.
Q024	Na sua residência tem computador?
Q025	Na sua residência tem acesso à Internet?

Tabela 1: Descrição das colunas do dataset.

Descrição dos dados de IDH extraídos da fonte do Kaggle.

Ano	Ano do Censo.
UF	Estado Unidade da Federação (Estado).

Codmun6	Código utilizado pelo IBGE para identificação do município
Codmun7	Código utilizado pelo IBGE para identificação do município(com digito verificador).
Município	Nome do município.
IDHM	Índice Desenvolvimento Humano Município
IDHM_E	Índice Desenvolvimento Humano Educação
IDHM_L	Índice Desenvolvimento Humano Longevidade
IDHM_R	Índice Desenvolvimento Humano Renda

Tabela 2: Descrição das colunas IDHM.

3. Processamento/Tratamento de Dados

O Dataset original possui 5095271 linhas e 136 colunas com seus dados variando entre Dados do Participante, Dados da Escola, Dados do local de aplicação da prova, Dados das provas (Ciências Humanas, Ciências da Natureza, Linguagens e Códigos, Matemática) além do questionário preenchimento do dado socioeconômico.

Estamos executando o notebook na nuvem do Google Colab, armazenamos os dados no Google Drive (Link), precisamos montar o drive para importar os datasets. Isso é feito com o seguinte comando:

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

Figura 2: drive.mount

Para que otimizar o desempenho durante as análises realizamos a importação inicial de 76428 linhas, optamos pela redução da quantidade de linhas sendo assim possível tornar o processamento viável.

```
1 base.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8660 entries, 0 to 8659
Data columns (total 23 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   NO_MUNICIPIO_RESIDENCIA  8660 non-null   object
 1   CO_UF_RESIDENCIA         8660 non-null   int64
 2   SG_UF_RESIDENCIA         8660 non-null   object
 3   NU_IDADE                 8660 non-null   int64
 4   TP_SEXO                  8660 non-null   object
 5   TP_ESTADO_CIVIL          8660 non-null   int64
 6   TP_COR_RACA              8660 non-null   int64
 7   TP_NACIONALIDADE         8660 non-null   int64
 8   TP_ST_CONCLUSAO          8660 non-null   int64
 9   TP_ESCOLA                8660 non-null   int64
 10  IN_TREINEIRO             8660 non-null   int64
 11  TP_DEPENDENCIA_ADM_ESC   1978 non-null   float64
 12  NU_NOTA_CN               6245 non-null   float64
 13  NU_NOTA_CH               6590 non-null   float64
 14  NU_NOTA_LC               6590 non-null   float64
 15  NU_NOTA_MT               6245 non-null   float64
 16  NU_NOTA_REDACAO          6590 non-null   float64
 17  Q001                     8660 non-null   object
 18  Q002                     8660 non-null   object
 19  Q005                     8660 non-null   int64
 20  Q006                     8660 non-null   object
 21  Q024                     8660 non-null   object
 22  Q025                     8660 non-null   object
dtypes: float64(6), int64(9), object(8)
memory usage: 1.5+ MB
```

Figura 3: Colunas dataframe.

Para a base de dados IDHM importamos trazendo os resultados de IDH por Município. A intensão é cruzar estes dados com a base principal e realizar insights e plotar gráficos que possam auxiliar na comparação e amostragem dos dados.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5565 entries, 0 to 5564
Data columns (total 9 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   ANO        5565 non-null   int64
 1   UF         5565 non-null   object
 2   Codmun6    5565 non-null   int64
 3   Codmun7    5565 non-null   int64
 4   Município  5565 non-null   object
 5   IDHM       5565 non-null   float64
 6   IDHM_E     5565 non-null   float64
 7   IDHM_L     5565 non-null   float64
 8   IDHM_R     5565 non-null   float64
dtypes: float64(4), int64(3), object(2)
memory usage: 391.4+ KB
```

Figura 4: Importação Dataframe IDHM.

Realizamos o tratamento de dados faltantes “NaN” com a média das notas dos candidatos, dessa forma podemos evitar a perda desnecessária de dados importantes. Decidimos criar um campo “SOMA_NOTA” contendo a soma das notas para compor a análise, assim podemos enriquecer os dados.

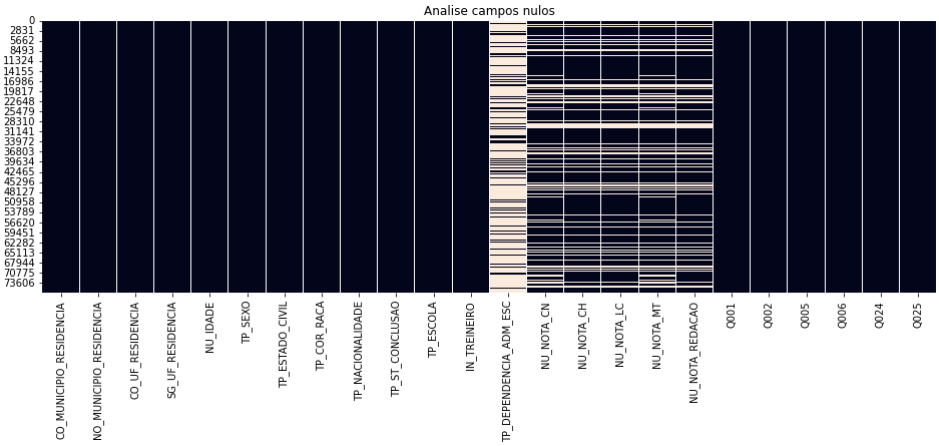

Figura 5: Campos Nulos.

Removendo valores = 0, indicando que o candidado zerou a prova ou não compareceu.

Esse tipo de tratamento é importante e evita alguns outliers o que pode nos ajudar quando executarmos os modelos de Machine Learning.

Utilizamos o módulo pickle para serializar objetos e salvá-los em um arquivo. Você pode desserializar o arquivo serializado para carregá-los de volta quando necessário. Pickle tem uma grande vantagem sobre outros formatos - você pode usá-lo para armazenar qualquer objeto Python.

Isso mesmo, você não está limitado aos dados. Uma das funcionalidades mais utilizadas é salvar modelos de aprendizado de máquina após a conclusão do treinamento. Dessa forma não precisamos treinar novamente o modelo toda vez que executar o script.

Salvamos o conteudo treinado no dataframe base_enem_corr.csv.

4. Análise e Exploração dos Dados

A primeira análise foi realizada com o intuito de obter mais insights incluindo mais um dataframe IDHM com apoio as informações de dados Socioeconômicos. As Bibliotecas utilizadas foram Matplotlib, seaborn, Numpy, Pandas, Scipy como observados no notebook.

Inicialmente para a exploração dos dados, foi sugerido responder algumas perguntas acerca da natureza do dataframe, separamos algumas associadas ao questionário socioeconômico. Até que série seu pai, ou o homem responsável por você, estudou? Até que série sua mãe, ou a mulher responsável por você, estudou?

A	'Nunca estudou'
B	'Não completou 5o ano Fundamental'
C	'Completou 5o ano Fundamental mas não completou 9o ano'
D	'Completou 5o ano Fundamental mas não completou Médio'
E	'Completou Médio mas não completou Faculdade'
F	'Completou Faculdade mas não completou Pós Graduação'
G	'Completou Pós Graduação'
H	'Não sei'

Tabela 3. Legenda Gráfico até que série seu responsável estudou?

Apesar de ser uma pergunta importante para o mapeamento, percebemos que em relação a nota de Matemática algumas pessoas responderam a opção H - 'Não sei'. Neste gráfico utilizamos ao boxplot através da biblioteca seaborn, incluímos a opção "showmeans=True" para incluir uma seta verde indicando a média.

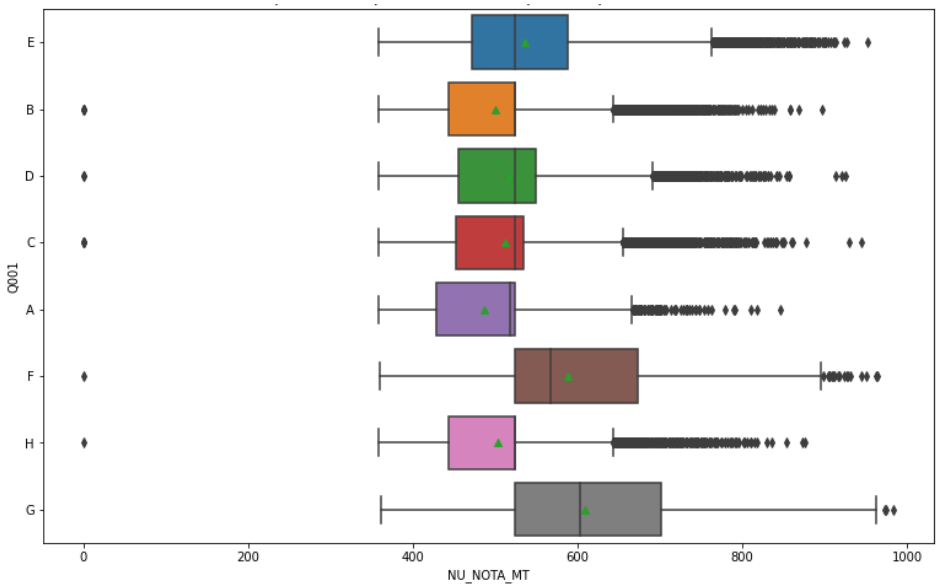

Figura 6: Escolaridade do Pai / Homem responsável.

Podemos perceber no gráfico acima que há uma relação forte sobre a escolaridade dos pais / responsáveis sobre a nota do candidato em relação a Nota de Matemática. Isso denota um incentivo e conscientização com relação a prioridade e importância da educação como item primordial da família. Somente com uma Educação melhor teremos acesso à cultura e isso com certeza nos levará a lugares que nunca imaginamos chegar. Nas opções G e F ('Completou Faculdade, mas não completou Pós Graduação' e 'Completou Pós Graduação') há um pico entre os outliers, reforçando a relação entre os comportamentos observados.

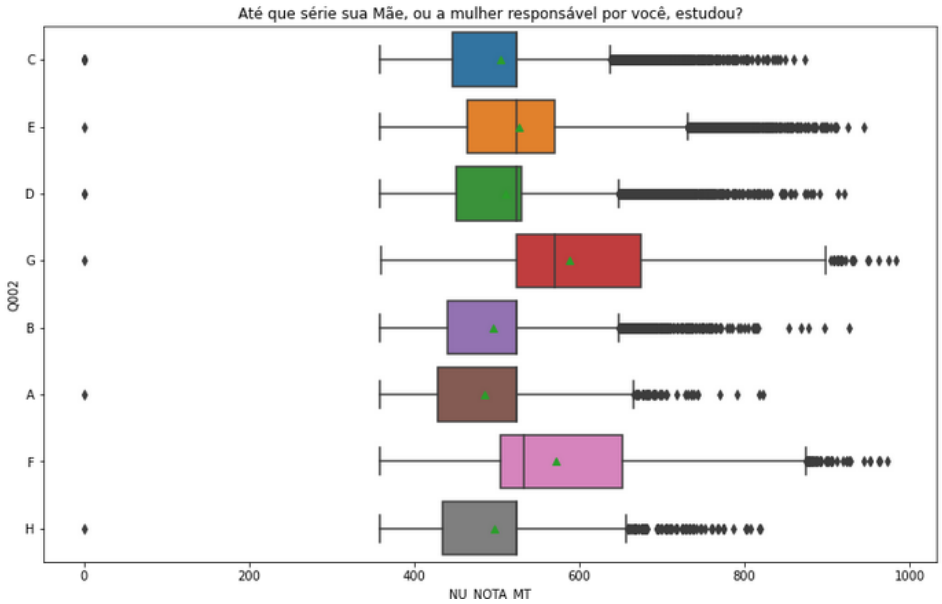

Figura 7: Escolaridade do Mãe / Mulher responsável.

Sobre a distribuição de notas por sexo, observamos que entre os homens principalmente entre a mediana e o terceiro quartil tivemos a participação maior de notas contidas, apesar do resultado existe outliers que mostram notas ainda maiores após o limite superior. Entre as mulheres notamos uma proporção maior para os outliers. No notebook mostramos essa diferença para cada prova aplicada.

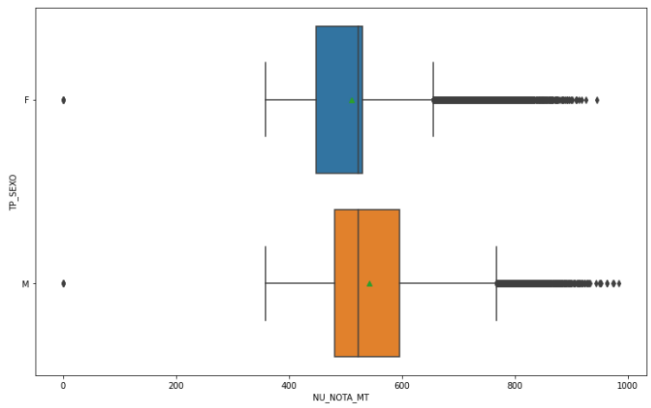

Figura 8: Escolaridade do Mãe / Mulher responsável.

O questionário referente a renda mensal de sua família é observado mediante a nota alcançada pelo candidato. A renda influenciou diretamente no resultado.

Podemos observar que pelo comportamento pessoas com maior poder aquisitivo podem contribuir no preparo atuando de forma preditiva com o auxílio através de bons livros, métodos, escolas ou professores particulares auxiliando no sucesso do candidato.

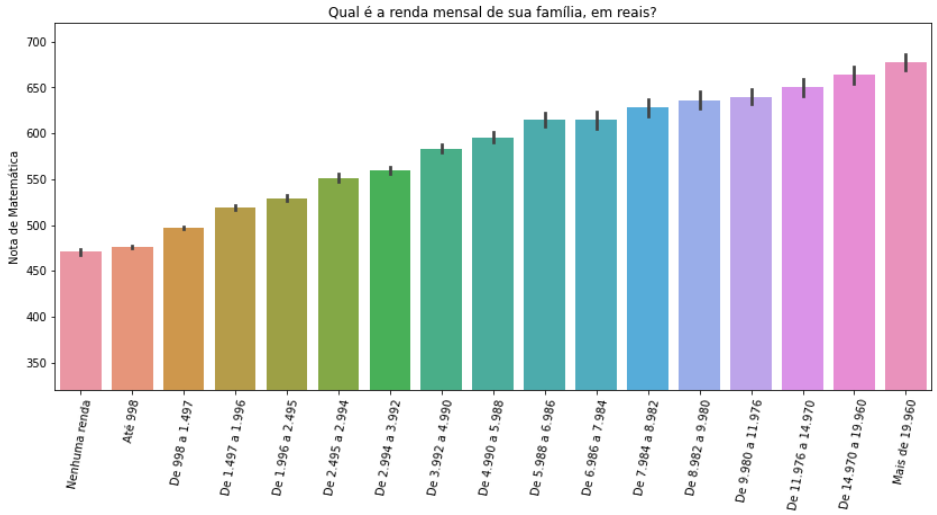

Figura 9: Incluindo você, quantas pessoas moram atualmente em sua residência?

Durante minha pesquisa percebi que muitos estudos focados na qualidade do ensino, principalmente entre as instituições. Essas informações são relevantes e fundamental para o mapeamento da qualidade do ensino no Brasil. Elevar o nível da Educação trará uma taxa de desemprego menor visto que mais pessoas estarão preparadas para o Mercado além da diminuição da taxa de criminalidade.

As informações sobre dependência administrativa x nota matemática mostra que a média das escolas Privadas está acima do terceiro quartil das outras instituições mesmo apresentando outliers superior.

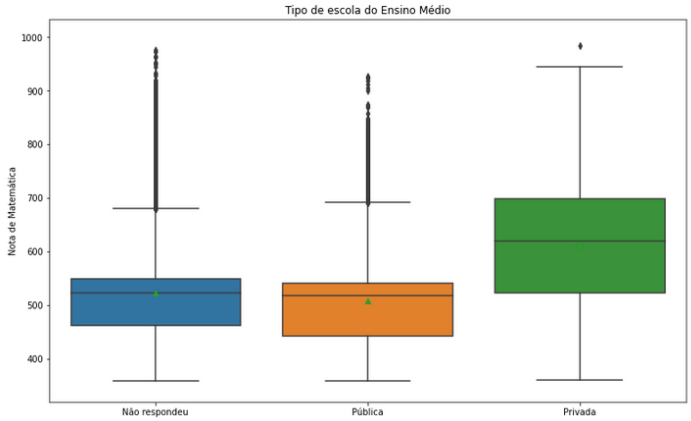

Figura 10: Tipo de Escola Ensino Médio

No gráfico abaixo há uma concentração de participações entre próximo à idade de 20 anos, posterior há uma queda na medida em que avançamos na idade a nota diminui. Esse comportamento não ocorreu entre as notas (Ciências Humanas, Ciências da Natureza, Linguagens e Códigos) pois houve uma concentração de notas entre as idades 15 a 35 anos conforme mostrado no notebook.

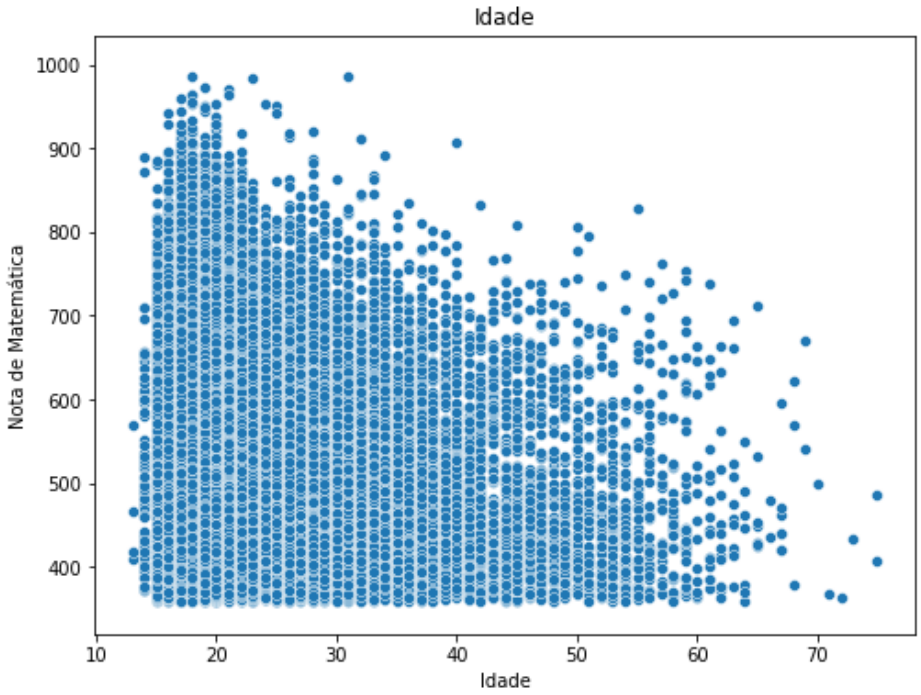

Figura 11: Distribuição de notas por idades

Realizamos uma análise exibindo a idade mínima contida no dataframe 12 anos, idade média 19 anos e idade máxima 82.

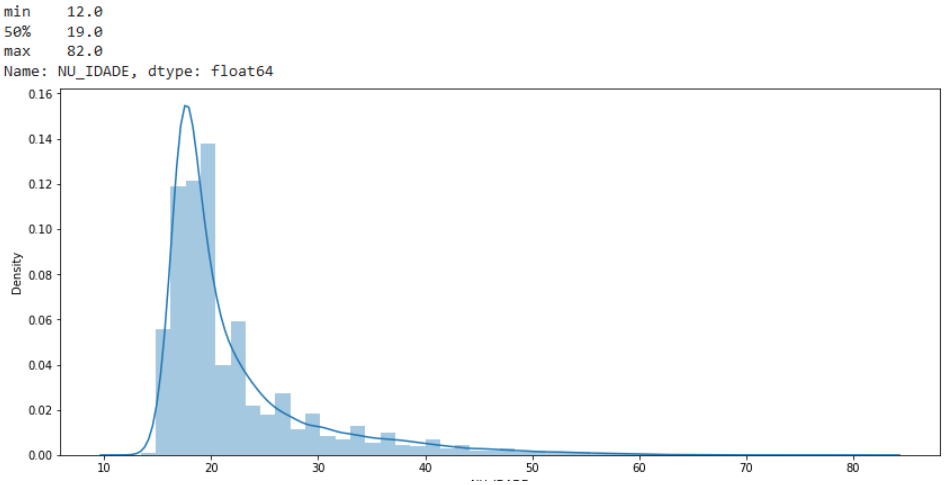

Figura 12: Min, Média, Max Idade

Como a prova realizada não possui limite mínimo de idade, alguns candidatos realizam a prova para testar seus conhecimentos e realizar treinos como forma de obter experiência e se familiarizar com a didática e conteúdos abordados durante. Notamos que há uma pequena parcela principalmente na faixa etária que antecede o ingresso na faculdade.

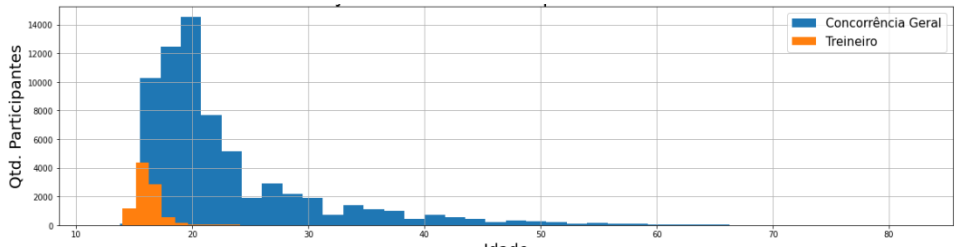

Figura 13: Distribuição dos participantes por idade entre treineiros

Temos também a proporção de participantes (Concorrência Geral X Treineiros) por renda indicando que pessoas com renda familiar superior demonstra um interesse maior para testar seus conhecimentos através da prova.

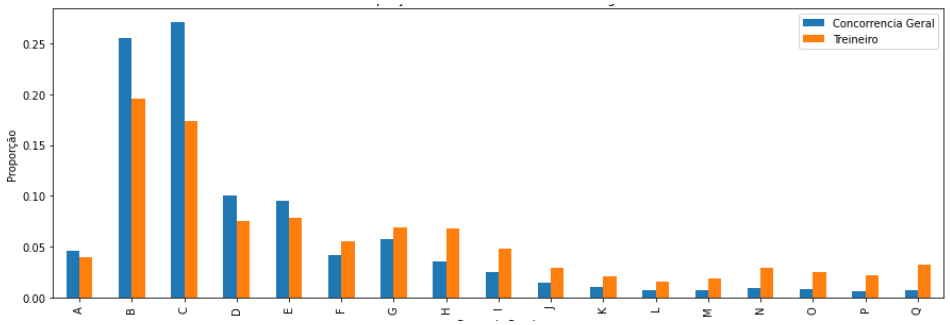

Figura 14: Proporção participantes X renda

Como análise importante destacamos a participação entre as cidades. Observamos que as TOP 5 são (SP, MG, BA, RJ, CE). A cidade de São Paulo é sem dúvidas a mais populosa com cerca de 12.325.232 habitantes com isso cresce também as chances de temos mais participantes. https://pt.wikipedia.org/wiki/Lista_de_munic%C3%ADpios_do_Brasil_acima_de_cem_mil_habitantes. Acesso em: 15 de abril 2021.

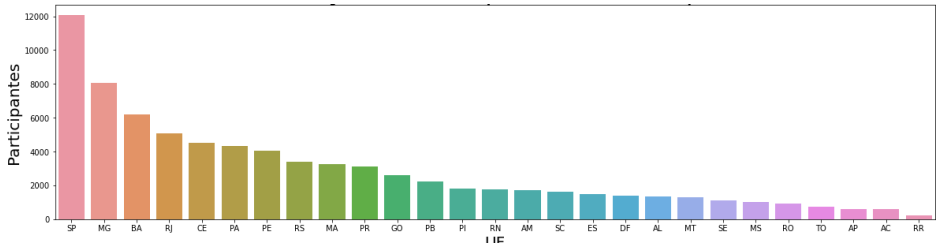

Figura 15: Proporção participantes X renda

Continuando nossa exploração de dados em relação as notas agora faremos uma visualização de notas distribuídas por municípios. O comportamento mostrado no gráfico indica que principalmente em MATEMÁTICA as regiões Nordeste, Sudeste e Sul estão com mais pontos vermelho, o que indica uma baixa nas notas. Linguagens e Códigos e Ciências Humanas predominam mais na região sul.

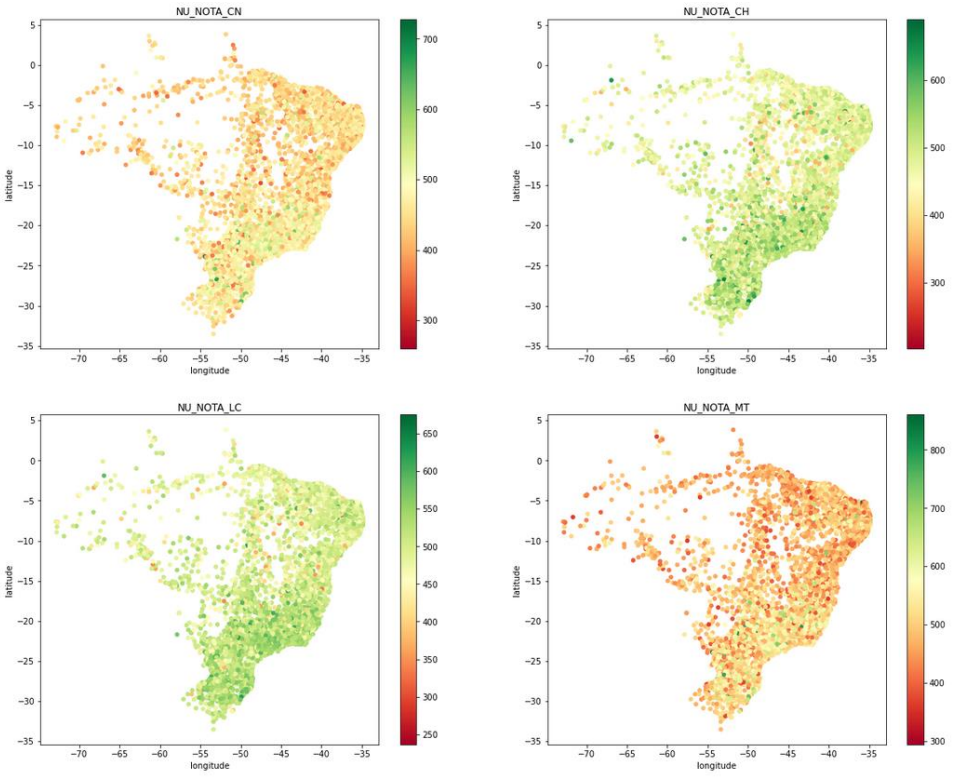

Figura 16: Distribuição de notas por município

5. Criação de Modelos de Machine Learning

Para o treino foram utilizados dois modelos de aprendizado supervisionado baseado em regressão, que foram Random Forest e Linear Regression, (a idéia é prever a nota baseada no test e comparar posteriormente com o modelo training utilizando o cross validation).

Iniciamos realizando utilizando da correlação. A correlação de Pearson mede a associação linear entre variáveis contínuas. É o valor que indica o quanto a relação entre as variáveis pode ser descrita por uma reta.

Interpretando o valor de *ρ*

- 0.9 a 1 positivo ou negativo indica uma correlação muito forte.
- 0.7 a 0.9 positivo ou negativo indica uma correlação forte.
- 0.5 a 0.7 positivo ou negativo indica uma correlação moderada.
- 0.3 a 0.5 positivo ou negativo indica uma correlação fraca.
- 0 a 0.3 positivo ou negativo indica uma correlação desprezível.

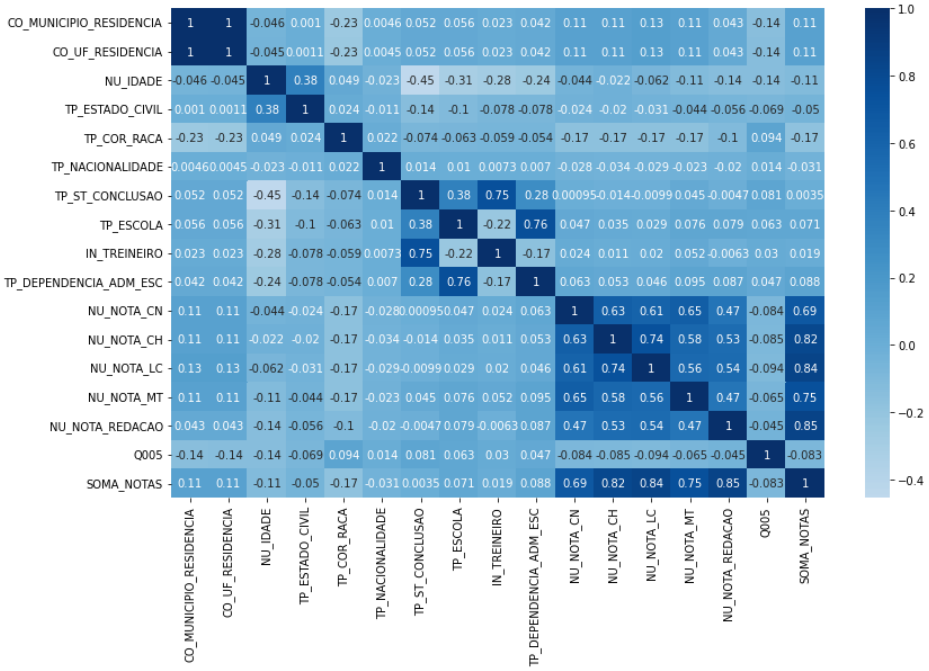

Figura 17: correlação

O resultado acima mostra a correlação entre as variáveis selecionadas. Esses valores podem variar de -1.0 até +1.0. Temos correlação FORTE DIRETA quando o valor de correlação se aproxima de +1.0. Para valores de correlação que se aproxima de -1.0, temos uma correlação FORTE INVERSA. Para valores de correlação próximos de 0.0 não há correlação.

Uma forma simples de enxergar essas correlações é através das cores mais fortes "positivamente" ou mais claras "negativamente".

Os dados de Treino e teste foram criados para as colunas presentes abaixo:

```
1 base_enem_corr.head(5)
```

	NU_NOTA_CN	NU_NOTA_CH	NU_NOTA_LC	NU_NOTA_MT	NU_NOTA_REDACAO	SOMA_NOTAS
0	560.000000	592.0	591.1	695.100000	540.0	2978.200000
1	477.665532	457.9	500.1	523.463214	560.0	2519.128746
2	396.500000	549.6	547.6	409.300000	560.0	2463.000000
3	396.600000	553.7	547.9	497.300000	800.0	2795.500000
4	511.500000	495.1	533.2	596.300000	620.0	2756.100000

Figura 18: base_enem_corr

Um conceito importante em Machine Learning é a divisão dos dados em dois grupos:

- dados de treinamento do modelo
- dados de checagem do modelo

É muito importante fazer isso antes de iniciar a construção do modelo, pois isso evita que o resultado utilizado para treinamento do modelo seja o mesmo de checagem, gerando-se assim uma métrica errada de qualidade do modelo, isto é um modelo pode parecer muito bom para dados de treinamento, porém pode ser muito ruim para dados fora da amostra de treinamento. Para isso o python possui uma biblioteca sckit-learn que facilita bastante a vida do usuário.

Definimos a variável alvo como as colunas de NOTAS e separamos a base de dados com 80% dos dados para treino e 20% para teste. Escolhemos o SEED=4321 de forma aleatória, com o fim de evitar variação de resultados da quebra dos dados para simulação. Setamos o ramdom_state com o valor de SEED=4321 para garantia de reprodutibilidade do processo.

```
1 from pandas.core.common import random_state
2 # Splitting dataset into training and testing
3 from sklearn.model_selection import train_test_split
4 seed = 4321
5 X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.20,random_state=seed)
6
```

Figura 19: train_test_split

Na Regressão Linear nosso objetivo é fazer a previsão de números. Modelagem da relação entre variáveis numéricas (variável dependente X, variável esplanatória y).

O primeiro modelo que usaremos é o de "Linear Suport Vector Machine" (LinearSVR), que é basicamente uma previsão dos dados através de uma função linear.

```
1 from sklearn.svm import LinearSVR
2 seed = 4321
3 modelo = LinearSVR(random_state=seed)
4 modelo.fit(X_train, y_train)
```

Figura 20: modelo = LinearSVR

Com os resultados de previsão disponível, podemos fazer uma comparação destes com os dados reais.

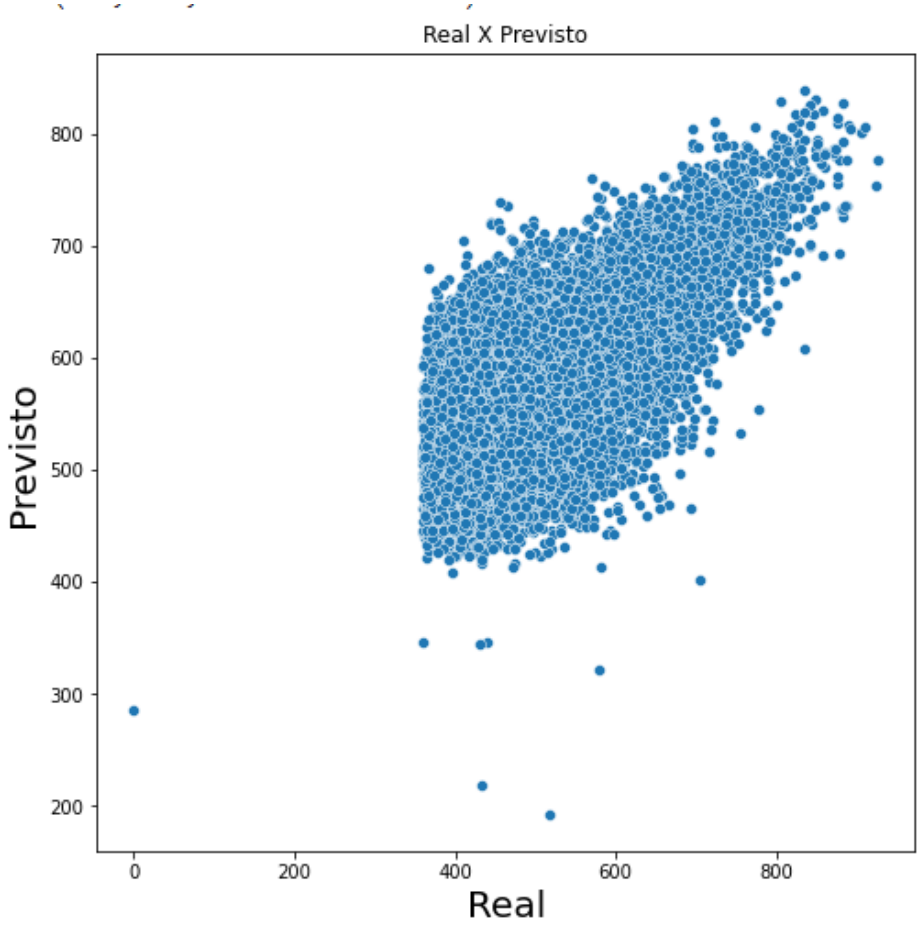

Figura 21: real x previsto

Se fizermos um "modelo simplificado" comparando a média e mediana dos dados de entrada do modelo e comparar com o resultado previsto, temos:

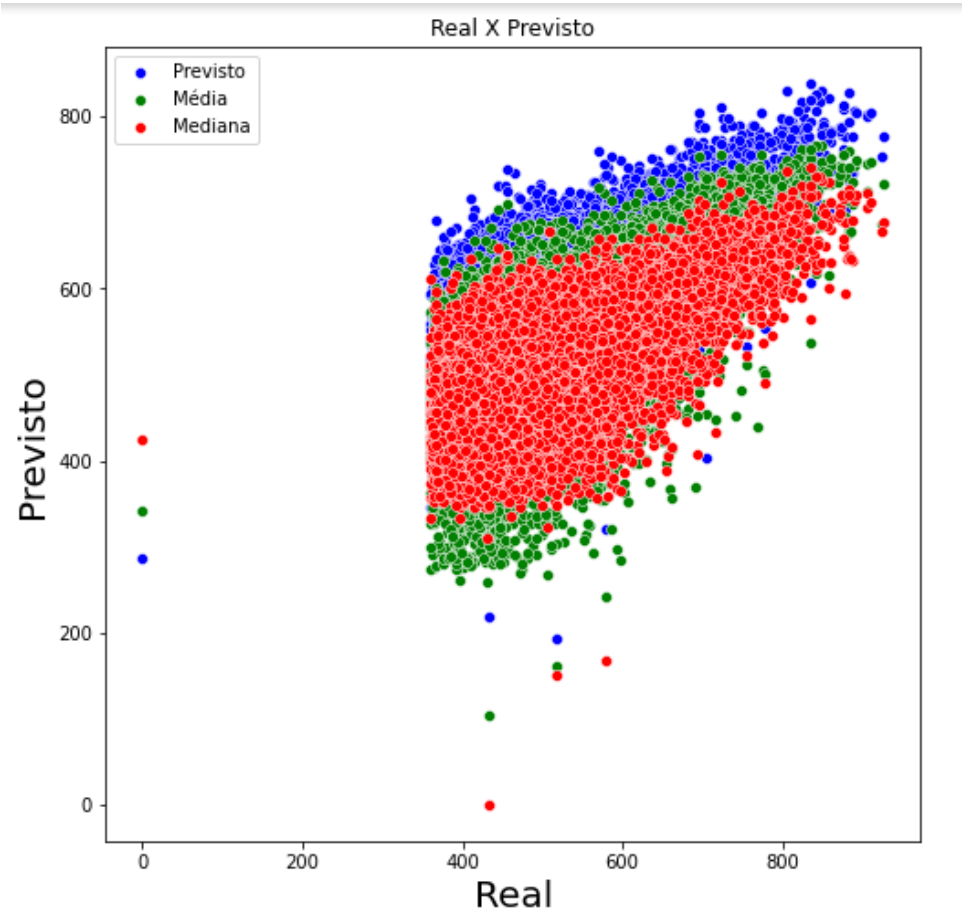

Figura 22: previsão, média, mediana

Utilizamos o “mean_squared_error” para medir a qualidade do modelo.

```
1 resultados = pd.DataFrame()
2 resultados['Real'] = y_test
3 resultados['Previsto'] = y_prev
4 resultados['Diferenca'] = y_test-y_prev
5 resultados['Diferenca_Quadrada'] = (y_test-y_prev)**2
6 resultados['Diferenca_Quadrada'].mean()
```

```
9618.331677997987
```

```
1 #medir a qualidade do modelo
2 from sklearn.metrics import mean_squared_error
3 mean_squared_error(y_test, y_prev)
```

```
9618.33167799797
```

Figura 23: mean_squared_error

Realizamos a comparação com outros modelos como DummyRegressor. Este modelo faz previsões usando regras simples. Esse modelo é útil como uma linha de base simples para comparação com outros regressores (reais).

```
]  1 #comparar com modelo Dummy
   2 from sklearn.dummy import DummyRegressor
   3 import numpy as np
   4 modelo_dummy = DummyRegressor
   5 #Treino do Modelo
   6 X = X_train
   7 y = y_train
   8
   9 dummy_regr = DummyRegressor(strategy="mean")
  10 modelo_dummy = dummy_regr.fit(X, y)
  11 modelo_dummy_prev = modelo_dummy.predict(X_test)
  12 mean_squared_error(y_test, modelo_dummy_prev)
  13 #dummy_regr.predict(X)
  14

11839.179568445132
```

Figura 24: comparação modelo Dummy

Utilizamos outra comparação com o modelo LinearRegression que ajusta um modelo linear com coeficientes w = (w1, …, wp) para minimizar a soma residual dos quadrados entre os alvos observados no conjunto de dados e os alvos previstos pela aproximação linear.

```
1 from sklearn.linear_model import LinearRegression
2
3 modelo_LinearRegression = LinearRegression()
4
5 #Treino
6 modelo_LinearRegression.fit(X_train, y_train)
7 modelo_LinearRegression_prev = modelo_LinearRegression.predict(X_test)
8 mean_squared_error(y_test, modelo_dummy_prev)

11839.179568445132
```

Figura 25: comparação modelo Dummy

Aplicamos aqui o "Linear Suport Vector Machine" (LinearSVR), como explicado anteriormente realiza uma previsão dos dados através de uma função linear.

```
1 from sklearn.svm import SVR
2 modelo_SVR = SVR()
3
4 modelo_SVR.fit(X_train, y_train)
5 modelo_SVR_prev = modelo_SVR.predict(X_test)
6 mean_squared_error(y_test, modelo_SVR_prev)
```

```
5426.078054896129
```

Figura 26: modelo SVR

Realizando o cálculo do MAE entre os modelos:

```
[ ]  1 from sklearn.metrics import mean_absolute_error as mae
     2 MAE_LinearRegression = mae(y_test, y_prev)
     3 MAE_LinearRegression
```

```
80.0707733402368
```

Figura 27: resultado MAE_LinearRegression

```
[ ]  1 MAE_LinearRegression = mae(y_test, modelo_dummy_prev)
     2 MAE_LinearRegression
```

```
90.01282369590935
```

Figura 28: resultado MAE_modelo_dummy_prev

```
[ ]  1 MAE_LinearRegression = mae(y_test, modelo_SVR_prev)
     2 MAE_LinearRegression
```

```
58.28108353807321
```

Figura 29: resultado MAE_modelo_svr_prev

Aplicamos também o modelo de Arvore de decisão e tivemos uma resposta maior em relação ao cálculo do MAE.

```python
#Divisão entre treino e teste
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.30)


modelo_DecisionTreeRegressor = DecisionTreeRegressor(max_depth=3)
#treino
modelo_DecisionTreeRegressor.fit(X_train, y_train)
#result
y_DecisionTreeRegressor = modelo_DecisionTreeRegressor.predict(X_test)
print(mae(y_test, y_DecisionTreeRegressor))
```

```
63.25632540982931
```

Figura 30: DecisionTreeRegression

Utilizamos o Cross_validade retorna um dicionário contendo 3 parâmetros:

1. fit_time: tempo de treino e teste dos dados para cada divisão.
2. score_time: tempo para calcular o score de cada divisão.
3. test_score: a medida de qualidade do modelo para cada divisão.

```python
from sklearn.model_selection import cross_validate

modelo_DecisionTreeRegressor = DecisionTreeRegressor(max_depth=2)
cross_validate(modelo_DecisionTreeRegressor, X, y)
```

```
{'fit_time': array([0.05232692, 0.09915733, 0.04631495, 0.05282879, 0.04871225]),
 'score_time': array([0.00780416, 0.00341582, 0.00298333, 0.00323462, 0.00454307]),
 'test_score': array([0.43107958, 0.44642188, 0.45567168, 0.44797982, 0.36344703])}
```

Figura 31: Cross_validate

6. Interpretação dos Resultados

Ao realizar a comparação entre os modelos, vemos que o (**LinearSVR**) não foi tão efetivo quanto o de regressão linear simples (**LinearRegression**).

O Teste no modelo **LinearSVR** apresentou o menor erro quadrático, mas precisa de um tempo maior para a previsão elevando também o custo computacional. Dependendo da aplicação, esse custo pode inviabilizar seu uso.

Além do mean_square_error, utilizamos o MAE (**Mean Absolute Error**) erro médio absoluto do modelo, significando o quanto o modelo erra em média.

7. Apresentação dos Resultados

Este estudo apresenta um grande potencial de previsões levando em consideração os diversos insights retirados do dataframe que servem de apoio para a construção de novos estudos com o aprofundamento de novos algoritmos de Machine Learning.

Realizamos a Exploração dos resultados diferenciando entre Sexo, cidade, treineiro dentre outros que foram as propriedades mais fortes contidas na análise. A relação direta aferida durante análise mostrando os segmentos de renda e notas acompanhando uma tendência.

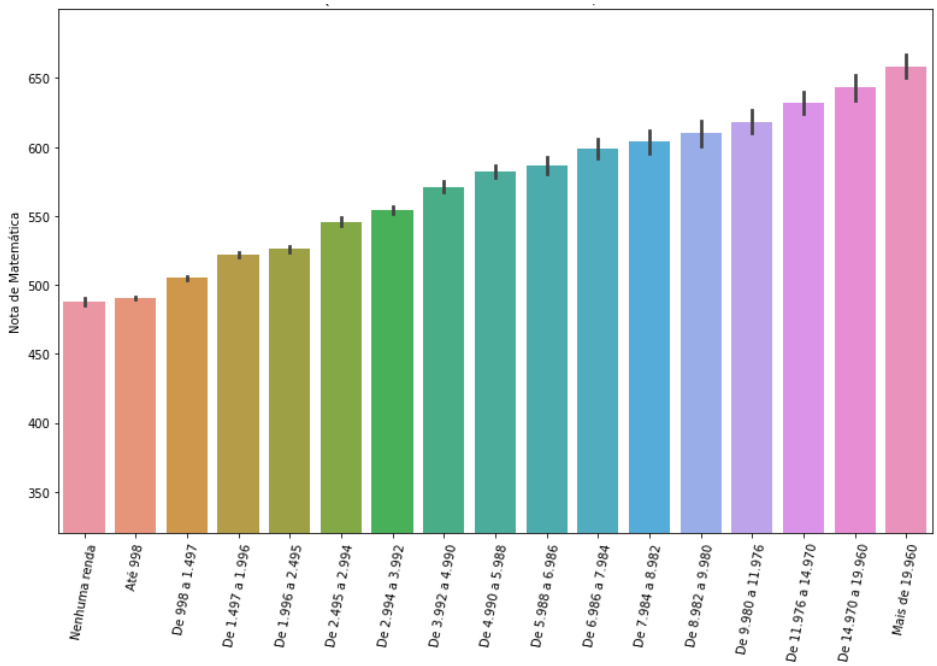

Figura 32: tendência renda x nota

É preciso ter muita cautela para não obter insights errados visto que há uma forte tendência entre renda e nota obtida. Isso pode causar um certo pré-conceito e nos induzir a navegar com a análise de forma duvidosa. Mesmo utilizando como parâmetros outros dados socioeconômicos como o IDHM.

Como sugestão para próximas etapas do projeto seria a utilização de outros modelos de aprendizagem e uma avaliação direta dos recursos de investimentos que o Governos vem realizando por Município e o quanto esse investimento tem influenciado em relação ao ranking das notas obtidas pelos candidatos.

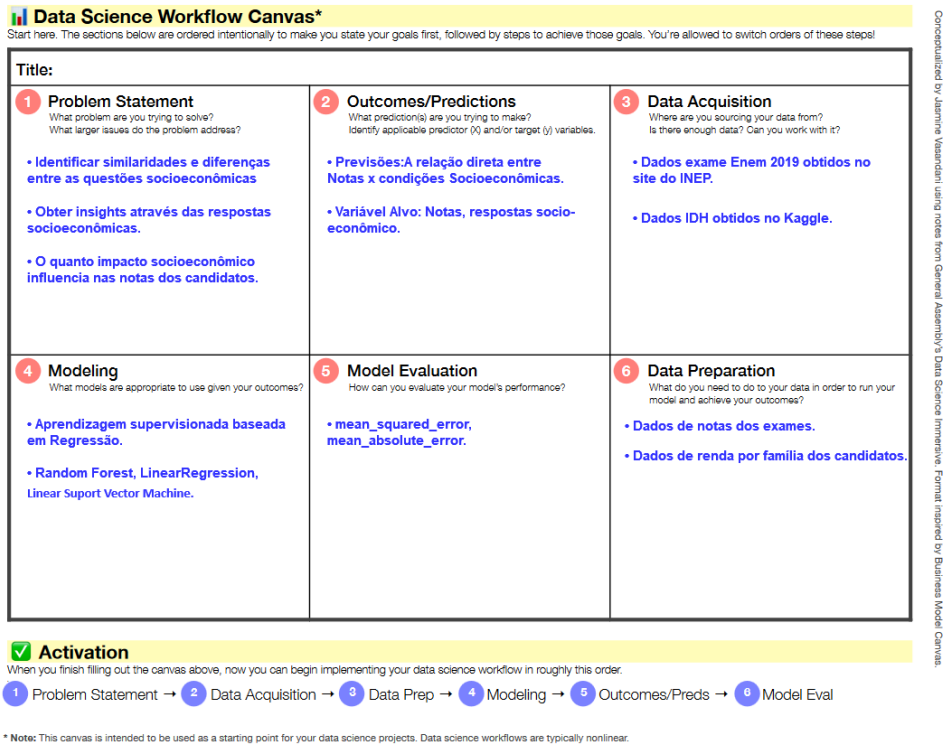

Figura 33: Workflow Canvas

O Assunto levantado cria diversas oportunidades de análise principalmente no que tange a qualidade do ensino no Brasil. Os resultados mostram com que prioridades as atenções são voltadas para a Educação e o quanto isso significa para uma parcela da população. Os dados levantados refletem as diferenças sociais de renda no Brasil levando a interpretação de quando temos renda menor tiramos notas menores no exame.

Entendemos que os comportamentos obtidos através das respostas de dado socioeconômico fazem com que o brasileiro busque cada vez mais a educação não só como forma de testar seus conhecimentos, mas também aproveitando as formas de ingresso em faculdades Públicas, Federais e Privadas. Entendo que isso também pode se tornar uma oportunidade para os órgãos voltar seus olhares principalmente para criação de programas de Educação de apoio a famílias de renda inferior.

Esta análise abre um precedente para várias frentes importantes, principalmente sobre a qualidade do ensino no Brasil entre as instituições. O quanto isso movimenta a economia atraindo cada vez mais instituições e empresários a investirem em educação e o quanto esse desafio se torna grande para o Brasil, onde os Governantes tem a missão junto aos órgãos públicos de realizar investimento de verbas orçamentárias distribuídas de forma homogênea.

8. Links

Link para o vídeo: https://youtu.be/ldYqE1j6vKo

Link para o repositório: https://github.com/eliascruzdba/Enem_2019

REFERÊNCIAS

MCKINNEY, William Wesley. Python para análise de Dados. São Paulo:

Novatec Editora , 2018.

https://www.google.com/url?sa=D&q=https://didatica.tech/como-funciona-o-algoritmo-arvore-de-decisao/&ust=1650217860000000&usg=AOvVaw0kOqsV--5Rstru3aKIGSl5&hl=pt-BR

https://didatica.tech/substituindo-dados-missing-com-machine-learning/

https://didatica.tech/o-que-e-e-como-funciona-o-algoritmo-randomforest/

https://didatica.tech/a-biblioteca-scikit-learn-pyhton-para-machine-learning/

https://matplotlib.org/

https://pandas.pydata.org/

https://scikit-learn.org/stable/

https://www.br.undp.org/content/brazil/pt/home/idh0.html. Acesso em: 24 de out. 2021.

Dados ENEM.CSV = https://drive.google.com/file/d/1hg1PKOht8TLnaiUHNFBAgQQqUBU7To-i/view?usp=sharing.

Dados IDHM = https://www.kaggle.com/BrasilComCenso/atlas-idhm-brasil-1991-2000-e-2010-lat-e-long/version/1

APÊNDICE

Programação/Scripts

Instalação dos pacotes (pandas, numpy, matplotlib) através do GoogleColab.

Gráficos

Tabelas