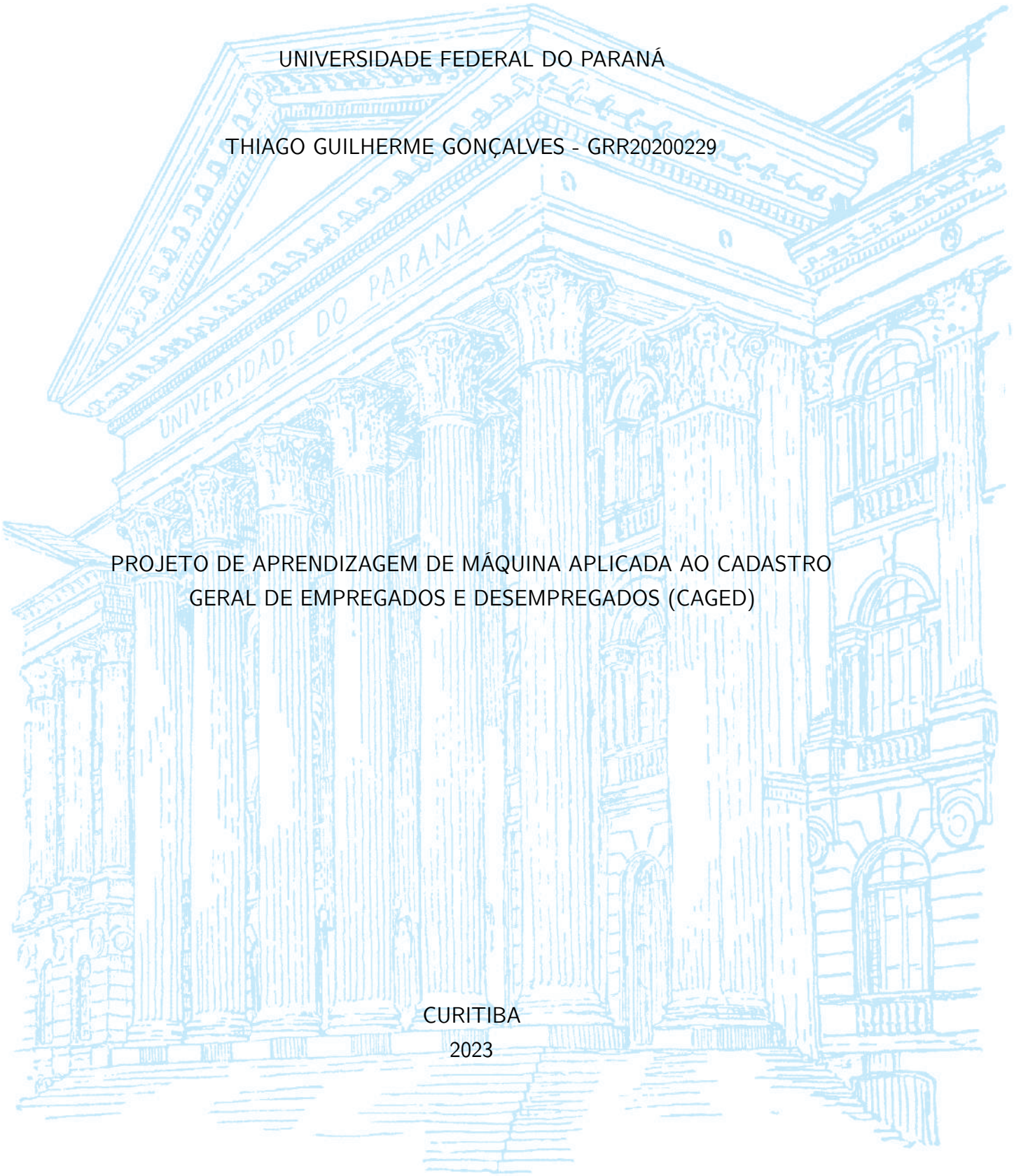
UNIVERSIDADE FEDERAL DO PARANÁ

THIAGO GUILHERME GONÇALVES - GRR20200229



PROJETO DE APRENDIZAGEM DE MÁQUINA APLICADA AO CADASTRO GERAL DE EMPREGADOS E DESEMPREGADOS (CAGED)

CURITIBA

2023

THIAGO GUILHERME GONÇALVES - GRR20200229

PROJETO DE APRENDIZAGEM DE MÁQUINA APLICADA AO CADASTRO GERAL DE EMPREGADOS E DESEMPREGADOS (CAGED)

Projeto de Aprendizagem de Máquina aplicada à base de dados governamental do Ministério da Economia: Cadastro Geral de Empregados e Desempregados (CAGED) realizado como requisito parcial para aprovação na disciplina de Machine Learning.Departamento de Matemática - Setor de Ciências Exatas da Universidade Federal do Paraná (UFPR).

Orientador: Professor Lucas Pedroso e Eduardo Vargas

CURITIBA

2023

RESUMO

O Cadastro Geral de Empregados e Desempregados (CAGED) é uma atribuição do Ministério da Economia criado com o objetivo de auxiliar no rastreio de desempregados com o objetivo de permitir o auxílio governamental, por exemplo com benefícios de seguro desemprego. Atualmente pode ser usado como fiscalização e como instrumento em auxílio de realocação do trabalhador no mercado de trabalho, segundo o Ministério da Economia. Estes dados contém, dentre outras 40 colunas, a quantidade de meses que um empregado atuou no cargo antes de ser desligado. Estas informações serão alvo de estudo deste trabalho por meio da aprendizagem de máquina supervisionada, com objetivo de prever o grau de instrução de um trabalhador baseado nos registros existentes para os anos de 2010, 2011 e 2012. Esta informação é sim relevante para as instituições administrativas do estado brasileiro, visto que políticas públicas podem ser tomadas ao aferir-se que um maior grau de instrução/escolaridade leva a maiores salários e/ou maior estabilidade no emprego . O dataset pode ser encontrado neste link e pode ser baixado no formato CSV acessando o banco de dados BigQuery disponibilizado pelo Google fazendo a consulta SQL filtrando pelos anos de 2010,2011 e 2012. Por limitações de processamento computacional, apenas uma parcela dos dados disponibilizados em cada tabela foi baixado para a realização das análises e, futuramente, previsões. O código pode ser acessado através deste repositório do GitHub.

Palavras-chave: Desemprego; Ministério da Economia; Economia; Aprendizado de Máquina.

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1 INTRODUÇÃO

1.1 INTRODUÇÃO

O Cadastro Geral de Empregados e Desempregados (CAGED) é uma atribuição do Ministério da Economia criado com o objetivo de auxiliar no rastreio de desempregados com o objetivo de permitir o auxílio governamental, por exemplo com benefícios de seguro desemprego. Atualmente pode ser usado como fiscalização e como instrumento em auxílio de realocação do trabalhador no mercado de trabalho, segundo o Ministério da Economia. Estes dados contém, dentre outras 40 colunas, a quantidade de meses que um empregado atuou no cargo antes de ser desligado. Estas informações serão alvo de estudo deste trabalho por meio da aprendizagem de máquina supervisionada, com objetivo de prever o grau de instrução de um trabalhador baseado nos registros existentes para os anos de 2010, 2011 e 2012. Esta informação é sim relevante para as instituições administrativas do estado brasileiro, visto que políticas públicas podem ser tomadas ao aferir-se que um maior grau de instrução/escolaridade leva a maiores salários e/ou maior estabilidade no emprego. O dataset pode ser encontrado neste link e pode ser baixado no formato CSV acessando o banco de dados BigQuery disponibilizado pelo Google fazendo a consulta SQL filtrando pelos anos de 2010,2011 e 2012. Por limitações de processamento computacional, apenas uma parcela dos dados disponibilizados em cada tabela foi baixado para a realização das análises e, futuramente, previsões.

2 MATERIAIS E MÉTODOS

2.0.1 O CONTEÚDO DAS COLUNAS

Aqui apresentamos o significado de algumas colunas menos óbvias (deixando de lado colunas como idade, sexo, ano, ...) mas que são importantes e constam-se presentes na nossa tabela:

- admitidos_desligados: 1 para admissão e 2 para desligamento
- saldo_movimentacao: diferença entre os admitidos e os desligamentos totais, 1 para admissão e -1 para desligamento
- indicador_portador_deficiencia: 0 - Não, 1 - Sim
- tipo_deficiencia: 0 - Nenhuma, 1 – Física, 2 – Auditiva, 3 – Visual, 4 – Intelectual (Mental), 5 – Múltipla, 6 – Reabilitado, 99 - Outro
- sexo: 1 - Masculino, 2 - Feminino
- tipo_estabelecimento: 1 - CNPJ, 3 - CAEPF (Cadastro de Atividade Econômica de Pessoa Física)
- cnae_2: Classificação Nacional de Atividades Econômicas (CNAE)
- grau_instrucao: Grau de instrução ou escolaridade (target)

FIGURA 1 – SIGNIFICADO DOS GRAUS DE INSTRUÇÃO

RAIS		IPEA
Descrição	Grau de Instrução	Nível de Escolaridade
Analfabeto	1	Nível 1
Até 5° Ano Incompleto	2	
5° Ano Completo	3	
6° ao 9° Ano do Fundamental	4	
Fundamental Completo	5	Nível 2
Médio Incompleto	6	
Médio Completo	7	Nível 3
Superior Incompleto	8	
Superior Completo	9	Nível 4
Mestrado	10	Nível 5
Doutorado	11	
Ignorado	-1	Sem Nível

Fonte: <https://www.ipea.gov.br/atlasestado/arquivos/rmd/4874-conjunto4v10.html>.

2.0.2 DADOS FALTANTES

Algumas das colunas apresentavam uma quantidade relativamente grande em relação aos demais quanto aos dados dados faltantes (Tabela 2), dessa forma, o melhor a se fazer foi remover essas colunas.

2.0.3 RECURSOS COMPUTACIONAIS E MÉTODOS UTILIZADOS

Ocorre que haviam, ao todo, disponibilizados pelo BigQuery da Base dos Dados mais de 400 milhões de registros, e desses registros, cerca de 198 milhões se concentravam nos anos de 2010-2012.

Aleatoriamente, decidiu-se analizar somente uma amostra de um período de tempo não tão disperso, e estimar que essa amostra aleatória represente toda a população daquele período. No caso, foi selecionado um período já bem documentado que foi o de 2010 até 2012 (3 anos), com amostras de mesmo tamanho de cada ano, todas na faixa de 15500 registros, aleatoriamente selecionadas na hora da extração via consulta SQL no BigQuery.

Visto que a biblioteca do scikit-learn utiliza, somente, para quaisquer cálculos, somente recursos do processador, e o processamento de dados ocorre através da RAM, uma das limitações em tempo e capacidade de processamento era um computador local com um i7 de décima geração e 8GB de RAM. A análise foi realizada através do Jupyter Notebook utilizando a linguagem Python com bilbiotecas como sklearn, seaborn, matplotlib e pandas.

Tabela 1 – Dados fornecidos pelo Base Dos Dados: uso, colunas e tipos. Fonte: Autor, 2023.

Nome	**Tipo**
ano	int64
mes	int64
sigla_uf	object
id_municipio	int64
id_municipio_6	int64
admitidos_desligados	int64
tipo_estabelecimento	int64
tipo_movimentacao_desagregado	int64
faixa_emprego_inicio_janeiro	int64
tempo_emprego	float64
quantidade_horas_contratadas	int64
salario_mensal	float64
saldo_movimentacao	int64
indicador_aprendiz	int64
indicador_trabalho_intermitente	float64
indicador_trabalho_parcial	float64
indicador_portador_deficiencia	int64
tipo_deficiencia	float64
cbo_2002	int64
cnae_1	object
cnae_2	int64
cnae_2_subclasse	object
grau_instrucao	int64
idade	int64
sexo	int64
raca_cor	float64
subsetor_ibge	int64
bairros_sp	object
bairros_fortaleza	object
bairros_rj	object
distritos_sp	object
regiao_administrativas_df	df
regiao_administrativas_rj	object
regiao_administrativas_sp	int64
regiao_corede	object
regiao_corede_04	int64
regiao_gov_sp	object
regiao_senac_pr	int64
regiao_senai_pr	object
regiao_senai_sp	object
subregiao_senai_pr	object

Tabela 2 – Percentual de dados faltantes de acordo com o nome da coluna. Fonte: Autor, 2023.

Coluna	**Dados NaN (%)**
indicador_trabalho_intermitente	98.189
indicador_trabalho_parcial	98.189
tipo_deficiencia	11.250
raca_cor	0.0129
bairros_sp	63.591
bairros_fortaleza	71.828
bairros_rj	73.684
distritos_sp	63.267
regiao_administrativas_df	0.012
regiao_administrativas_rj	98.377
regiao_corede	98.422
regiao_senai_pr	97.601
regiao_senai_sp	0.853
subregiao_senai_pr	0.853

3 RESULTADOS E DISCUSSÕES

3.0.1 CORRELAÇÕES

Em geral, os dados observados eram pouco correlacionados. As colunas com maiores e menores correlações, através do método de Pearson, com a variável "tempo_emprego"foram:

- admitidos_desligados: 0.357979
- idade: 0.200028
- salario_mensal: 0.153840
- saldo_movimentacao: -0.357979

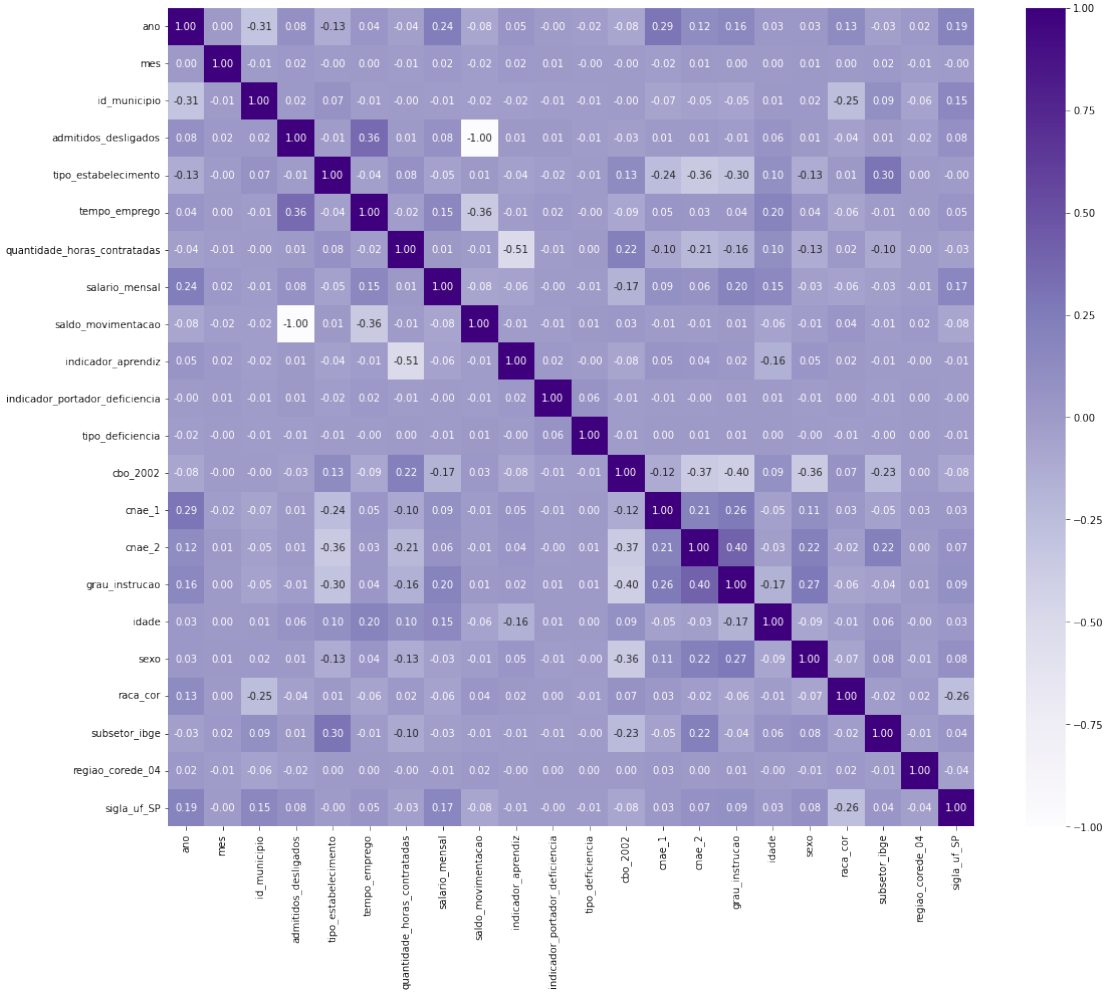
da mesma forma, avariável "salario_mensal"teve uma amplitude de correlação com as demais variáveis ainda menor. Essas duas variáveis foram colocadas em destaque pois são as que mais são de interesse em nossa análise.

Os Mapas de Calor (Heatmaps) expressam entre -1 (muito anticorrelacionado) e +1 (muito positivamente correlacionado) as correlações entre diferentes variáveis. Podemos observar esse mapa de calor, utilizando o método de Pearson, conforme apresentado pela Figura 2.

Na Figura 3, Figura 4, Figura 5, Figura 6 e Figura 7, podemos observar algumas das relações interessantes que podemos obter do CAGED. Na Figura 4, considerando 1 para admissão e 2 para desligamento, observa-se uma diferença pouco significativa entre cada tipo de deficiência e suas chances de ser admitido ou desligado, visto que os dados apresentaram média em torno de 1,5. Assim como na Figura 3, pode-se observar baixa diferença entre a proporção de admissões e desligamentos entre Pessoas Com Deficiência (PCD) e não portadores de deficiência.

Como pode-se observar na Figura 6, as pessoas inclusas na pesquisa avançaram mais em seu ensino fundamental, perceptível pela ausência de grau de ensino "1"em 2019 e por um perfil mais grosso dos graus 4, 5 e 6 no gráfico violino com os dados de 2019. Ademais, é notável que nos dados de 2019 há indicação de que os trabalhadores entre 30 e 50 anos estão mais frequentemente

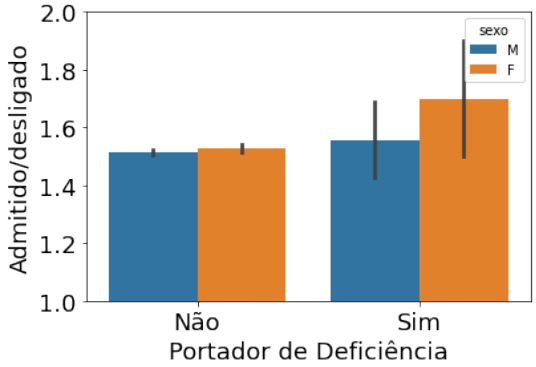
FIGURA 2 – HEATMAP DE ALGUMAS COLUNAS DA NOSSA TABELA SELECIONADA ("MICRODADOS_ANTIGOS")



voltando aos estudos acadêmicos. Isto é percepitível por esta faixa etária estar mais grossa no gráfico, enquanto que, nos dados de 2007, poucas pessoas de idade estavam estudando academicamente. Na Figura 5, esta relação não fica tão clara, por apresentar todos os dados na tabela: de 2007 a 2019.

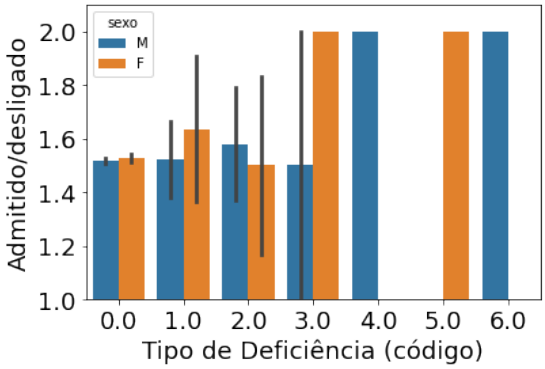
Como pode-se observar na Figura 7, a maioria dos estados apresentaram mais admissões que demissões durante os anos estudados neste trabalho. Os estados que apresentaram mais desligamentos que admissões foram: PE, CE, RJ, BA, AM, PB, GO, PI, AP.

FIGURA 3 – RELAÇÃO ENTRE PORTAR DEFICIÊNCIA E DESEMPREGO



Fonte: Autor, 2023.

FIGURA 4 – RELAÇÃO ENTRE TIPO DE DEFICIÊNCIA E DESLIGAMENTO



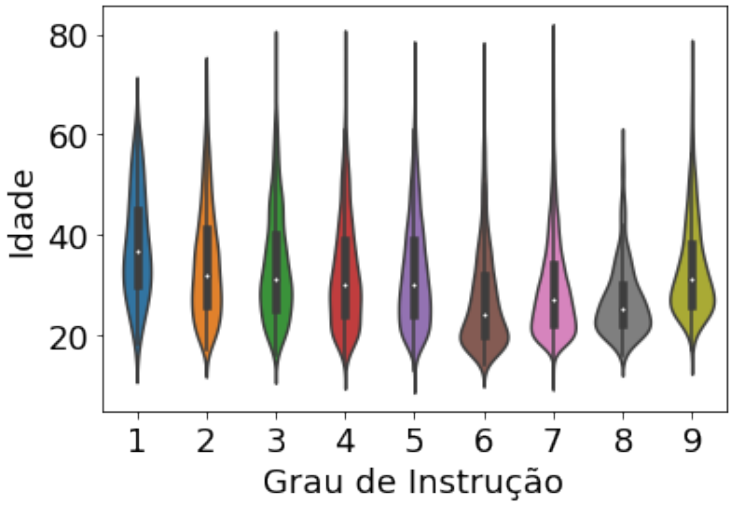
Fonte: Autor, 2023.

Na Figura 8, está a relação entre o salário e o tempo no qual o funcionário permaneceu no emprego.

3.1 PRÉ-PROCESSAMENTO

Na etapa de pré-processamento, foi analisado a natureza dos dados presentes em cada coluna, a quantidade de dados faltantes em cada coluna e avaliou-se a correlação dessas colunas com nossas variáveis de interesse (neste caso, **tempo_emprego** e **salario_mensal**) através do método de Pearson, onde podemos encontrar visualizações para este feito através do Mapa de Calor). Algumas

FIGURA 5 – RELAÇÃO ENTRE GRAU DE ESCOLARIDADE E IDADE



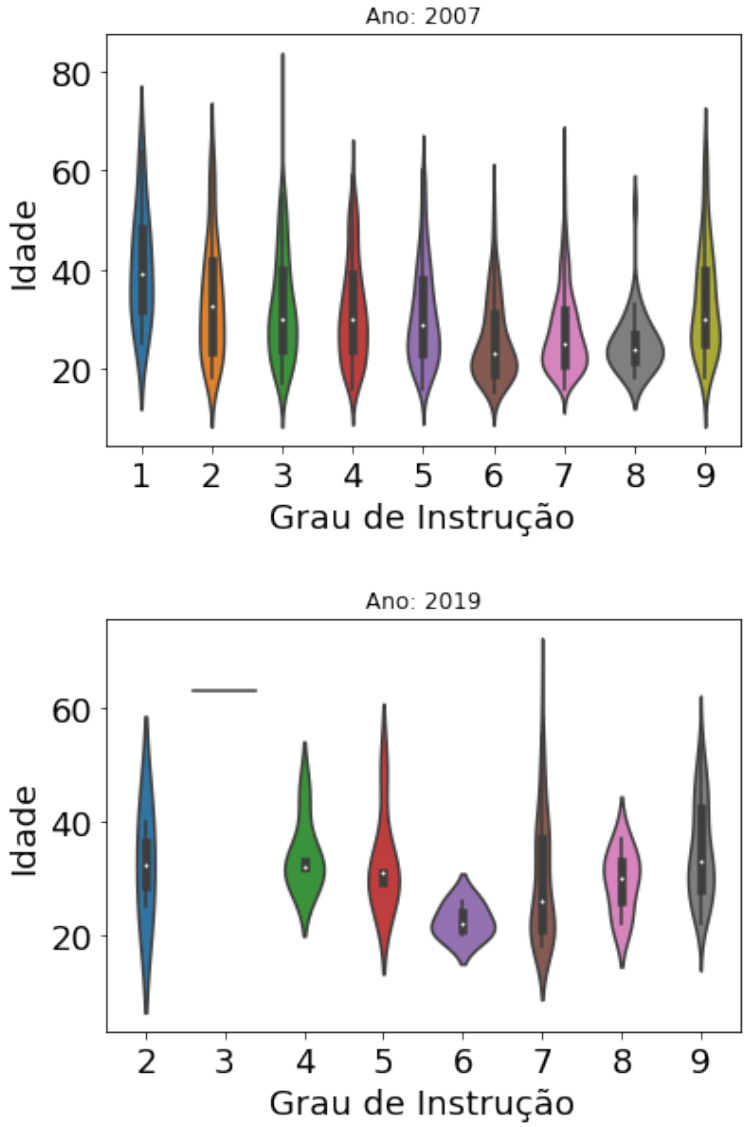
Fonte: Autor, 2023.

colunas, conforme apresentado na **??**, apresentavam quantidades inaceitáveis de dados faltantes, sendo assim, pouco relevantes para a construção de um modelo preditivo quando levamos em consideração o trabalho necessário com feature engineering para "torná-las úteis", e portanto, estas foram deletadas e não serão usadas no modelo.

Nas colunas cujo tipo era "*objeto*", foi realizado um processo de Encoding dos dados para o tipo *inteiro*. Espeficamente nas colunas de siglas de estado (26 colunas no total), foi feito *One Hot Encoding*, já que a quantidade de colunas adicionais não iria mudar muito após excluir as colunas indesejadas. O restante das colunas em que foi realizado *encoding* foi feito *Ordinal Encoding*, essa necessidade vinha do fato de que essas colunas vinham na tipagem de *string* e apresentavam muitos tipos de valores diferentes, alguns destes se repetiam, mas poucas vezes (verificação feita através do método *value_counts()* do pandas. Algumas colunas traziam dados com erros de digitação, e essas correções/limpezas foram feitas antes de realizar o processo de Ordinal Encoding.

As colunas a seguir foram removidas da tabela seja por terem pouca variação (essa verificação foi feita utilizando o método de seleção de variáveis *GenericUnivariateSelect*) e/ou serem pouco relevantes numa futura aborda-

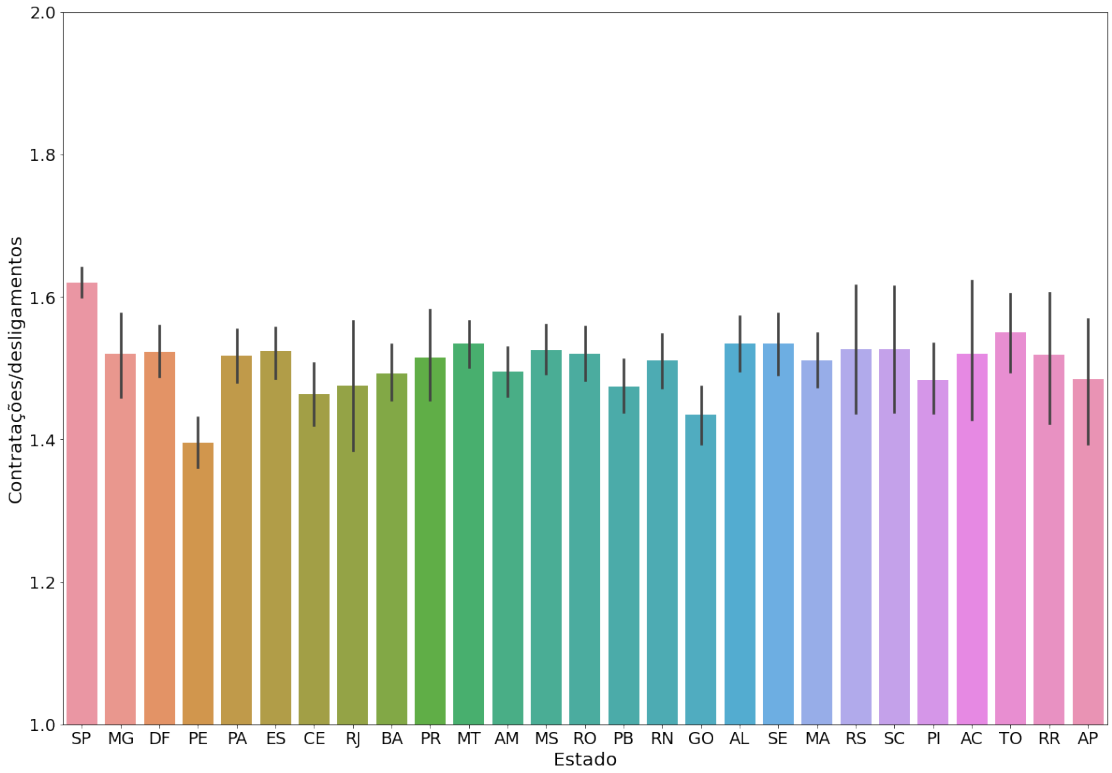
FIGURA 6 – DIFERENÇA ENTRE GRAUS DE INSTRUÇÃO EM 2007 E 2019



Fonte: Autor, 2023.

gem preditiva: "regiao_administrativas_df", "regiao_administrativas_sp", "regiao_gov_sp", "regiao_senac_pr", "id_municipio_6", "regiao_senai_sp", "subregiao_senai_pr"e "faixa_emprego_inicio_janeiro". Ademais, o restante das colunas que foram deletadas foi em razão do alto percentual de dados faltantes, neste caso, as colunas: "indicador_trabalho_intermitente", "indica-

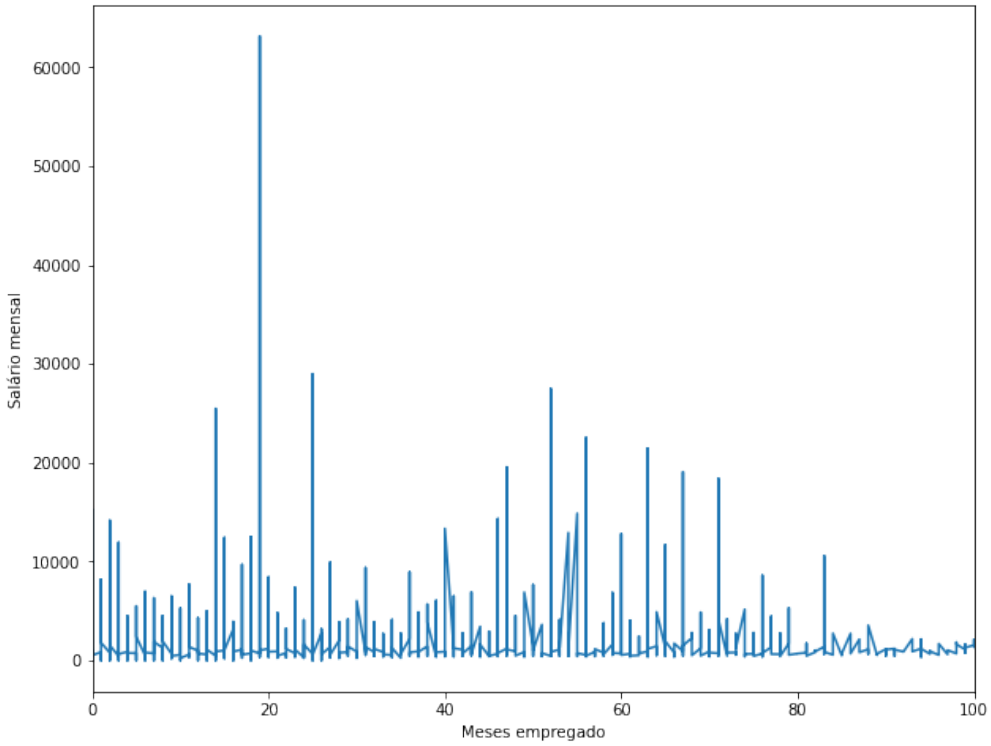
FIGURA 7 – RELAÇÃO ENTRE ADMISSÕES E DESLIGAMENTOS EM CADA ESTADO BRASILEIRO ENTRE 2007 E 2019



dor_trabalho_parcial", "bairros_sp", "bairros_fortaleza", "bairros_rj", "distritos_sp", "regiao_administrativas_rj", "regiao_corede"e "regiao_senai_pr".

No geral, grande parte das tipagens/estrutura de dados das variáveis vindas dos datasets chegavam num formato "legível"para a maioria dos modelos (dados numéricos ou facilmente conversíveis, seja por encoding ou simplesmente alterando a tipagem na própria linguagem de programação), e portanto, facilitou bastante a etapa de pré-processamento (o que é sabido que nem sempre acontece). Isso se deve, muito provavelmente, ao fato de que os dados já estavam armazenados em um banco de dados estruturado (data warehouse) antes de exportarmos eles para o formato *.csv*.

FIGURA 8 – RELAÇÃO ENTRE O SALÁRIO DO FUNCIONÁRIO E O TEMPO, EM MESES, QUE PERMANECEU NO CARGO



3.2 OS MODELOS

3.2.1 KNN

O modelo KNN (K nearest neighbors) classifica o alvo baseado em seus "vizinhos", os dados mais próximos dele. Ele calcula a distância entre o ponto e os k vizinhos mais próximos deste ponto e atribui a ele a classe predominante destes k vizinhos. Por exemplo, caso haja as classes A e B e o usuário escolha $k = 5$, o modelo irá identificar os 5 pontos mais próximos e buscará a classe mais frequente nestes pontos: se há três pontos A e dois Bs, ele classificará o ponto de classe desconhecida como A.

Este modelo é rápido e não exige muito poder computacional se comparado a outros como SVM e redes neurais para otimização. Contudo, este pode apresentar resultados bem satisfatórios caso seja bem otimizado e o target em

análise seja compatível com este modelo. O modelo KNN é mais compatível com dados "organizados"em regiões identificáveis como pertencendo a uma classificação, caso as classes estejam muito dispersas, este modelo pode não apresentar resultados satisfatórios.

3.2.2 RANDOM FOREST

O *Random Forest* gera "árvores"(conjunto de features) combinando-as até encontrar o conjunto de *features* que, quando analisadas em conjunto, produzem a melhor previsão possível. O Random Forest gera uma "floresta"de "árvores"com ramificação de *features* escolhidas aleatoriamente. A seleção de *features* para criação da árvore é aleatória para reduzir o overfitting. O Random Forest é mais complexo que o subseção 3.2.1, por exemplo, mas ainda é mais rápido e exige menos poder computacional que redes neurais, mas pode ser mais pesado que SVM e SVR.

3.2.3 SVC: SUPPORT VECTOR CLASSIFICATION

Os modelos baseados em *support vectors*, como SVM (Support Vector Machine) e SVR (Support Vector Regressor), separam os *targets* com *hyperplanes*. Eles funcionam organizando os dados do *target* de forma que eles sejam separáveis por hiperplanos. Isto, contudo, não é possível com perfeição, isto é: sempre há uma região onde a incerteza em relação à classificação de um ponto ocorre. Esta região busca ser minimizada pelo modelo, a fim de tornar as classificações precisas. O modelo é otimizado quando sua região de incerteza é a menor possível, tornando a classificação mais precisa e ágil. A distância entre as margens desta região de incerteza e o hiperplano não necessariamente será simétrica, já que o tamanho desta região para cada classe dependerá do quão "organizado"são os pontos e da facilidade para classificação de cada um. Esta região se torna maior quando um ponto de classificação "A"está presente na região que o algorítimo identificou como pertencente à classe "B", por exemplo. Este modelo pode exigir poder computacional e normalmente demora bastante para ser otimizado.

3.2.4 REGRESSÃO LOGÍSTICA

A regressão logística é uma técnica de classificação binária: classifica um *target* em apenas duas classes possíveis. É um modelo bastante eficiente por se tratar de classificação binária e um modelo matemático relativamente simples. Ele funciona calculando a probabilidade do ponto analisado pertencer a cada uma das classes e o classifica com a classe que apresentar maior probabilidade.

3.2.5 GAUSSIAN BAYES

Baseado no teorema de Bayes, este é o modelo mais simples possível de machine learning e normalmente apresenta os piores resultados de classificação. Ele, quando usado, costuma ter o propósito de ser uma referência para os resultados de outros modelos, servindo de métrica para avaliar o desempenho de outros modelos mais precisos e mais robustos. Ele calcula a probabilidade de um ponto analisado ser pertencente a uma classe partindo do pressuposto que o *target* obedece uma distribuição Gaussiana e que as classes do *target* são independentes, ou seja, que a probabilidade de pertencer a uma classe não afeta a probabilidade de pertencimento a outra. Ele atribui a classe com maior probabilidade de pertencimento.

3.2.6 GRADIENT BOOSTING CLASSIFIER

O Gradient Boosting Classifier é, na verdade, um grupo de algorítmos que combinam modelos de aprendizado de máquina mais imprecisos para obter uma previsão mais acertiva. É comum o uso de árvores de decisão nestes algorítmos. Ele funciona avaliando o desempenho de um algoritmo e focando seus futuros esforços nos pontos que apresentarem piores predições, otimizando o modelo e proporcionando melhores previsões e repetindo este processo várias vezes. Ele usa soma ponderada para dar uma relevância maior às árvores mais avançadas, por serem responsáveis por corrigir a fragilidade das anteriores.

3.3 ABORDAGEM DO PROBLEMA

3.3.1 O DATASET PRÉ-PROCESSADO

Ao agruparmos os datasets possíveis de se extrair do BigQuery para os anos de 2010, 2011 e 2012 (com aproximadamente 15500 linhas em cada), realizado o devido pré-processamento, foi ao final, observado, que algumas variáveis que especulava-se como sendo interessantes em prever (salario_mensal e tempo_emprego), tinham pouca variância e bastante outliers, como podemos ver pela tabela embaixo, e dessa forma, uma abordagem preditiva já não seria tão interessante de se fazer, dada a própria limitação dos variáveis disponíveis e também de que nenhuma das variáveis explicitamente davam indícios de coloaborar para a previsão de valores.

Dessa forma, uma abordagem mais interessante, seria a de prever o grau de instrução dos desempregados, através da classificação de múltiplas classes (multiclasse).

Vale salientar que algumas visualizações de importância de variáveis para os modelos base de Regressão Logística e RandomForest (para a previsão do grau de instrução) foram construídos utilizando a bilbioteca Yellowbrick, e constam em Figura 12 e Figura 11.

3.3.2 SEPARAÇÃO EM TREINO E TESTE E VALIDAÇÃO

Com o dataset final pré-processado e organizado, com 46684 registros, fizemos uma separação primeiramente em treino e teste, onde a amostra de teste contava com cerca de 20% dessa quantidade (9337 registros), e depois, com os dados restantes, foi feito a separação entre treino e validação, onde 15% dessa quantidade foi reservada para fazer a validação.

3.3.3 STACKING - UM ALGORITMO DE ALGORITMOS PARA CLASSIFICAÇÃO MULTICLASSE

Visto que agora, a nossa variável target é "grau_instrucao, na ideia de explorar vários algoritmos, veio a ideia de construir um Stacking, que é basicamente uma

FIGURA 9 – Total de registros, Média, Variância, valor mínimo, quartis de distribuição e valor máximo para as variáveis salario_mensal e tempo_emprego

	salario_mensal	tempo_emprego
count	46684.000000	46684.000000
mean	838.388291	7.278082
std	664.860840	19.682803
min	0.000000	0.000000
25%	583.000000	0.000000
50%	668.000000	0.000000
75%	865.000000	7.000000
max	20242.000000	464.000000

Fonte: Autor, 2023.

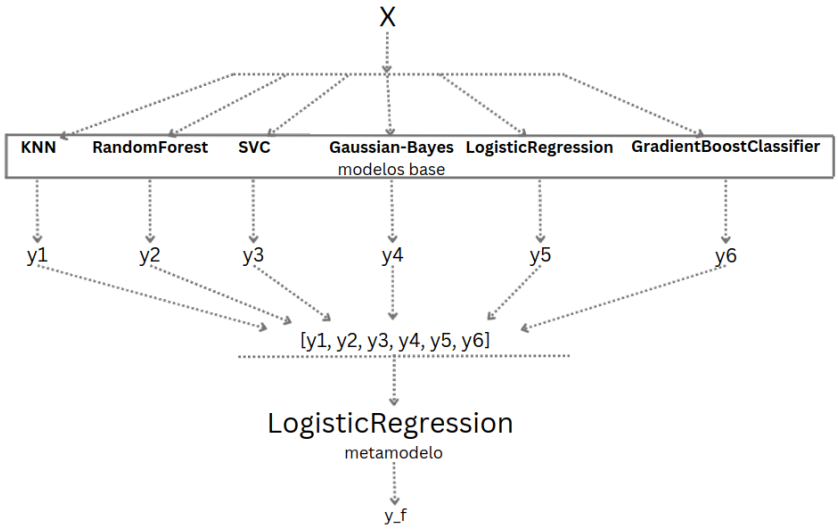
associação de modelos independentes, também chamados de modelos base, e repassar suas previsões para um modelo final, chamado de meta-modelo, este, que recebe como entrada as previsões dos modelos base e retorna uma previsão final.

Os modelos base selecionados foram KNN, Floresta Aleatória (Random Forest), SVC, Regressão Logística, Naive Bayes e Gradient Boosting. Enquanto o meta modelo selecionado foi também o de Regressão Logística. Os méritos e deméritos desses algoritmos são melhor detalhados em suas próprias subseções acima.

O fluxograma da Figura 10 ilustra justamente o que foi descrito, entretanto, vale observar que cada um dos modelos base e o próprio meta-modelo foi otimizado utilizando-se dos dados de validação.

Vale observar que, inicialmente, os dados de validação foram reservados para esse propósito de fato, porém, tendo em vista a complexidade do stacking e outros fatores na hora da execução do código, foi reservado a estes dados somente o propósito de tunar/otimizar cada um dos modelos para a obtenção dos melhores hiperparâmetros na expectativa de que eles reflitam em melhor

FIGURA 10 – Fluxograma do funcionamento da Stacking para os modelos utilizados no problema.



Fonte: Autor, 2023.

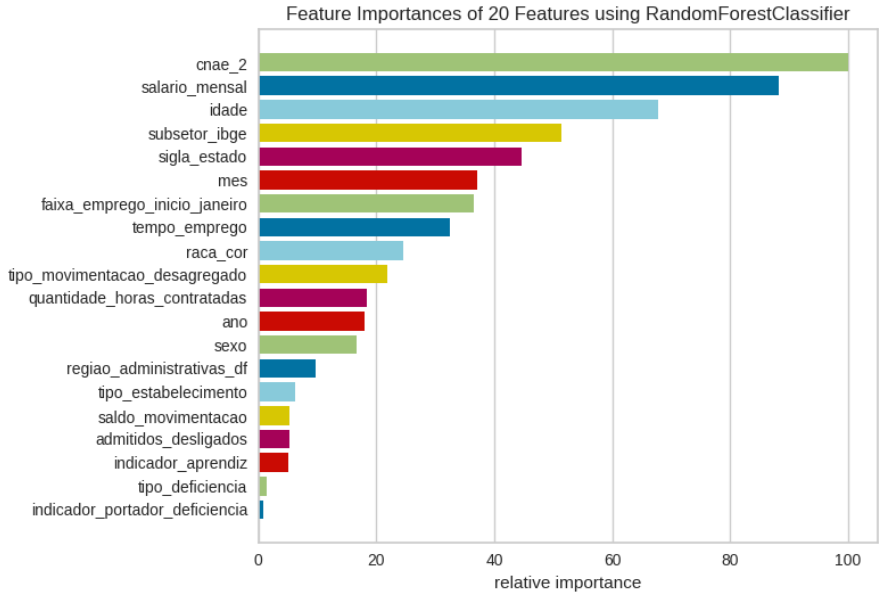
desempenho tanto para os dados de treinamento quanto teste ; esse método, evita garantidamente overfitting, porém, é sabido que não é efetivamente o propósito dos dados de validação fazer isso, mas sim verificar a performance do modelo antes de realizar o teste.

3.4 RESULTADOS FINAIS

Resumidamente, após o treinamento dos algoritmos já otimizados, ao abordarmos eles com os dados de teste, a acurácia obtida para cada um dos modelos base foi de 0.47, 0.36, 0.46, 0.47, 0.51 e 0.48 para os modelos de Regressão Logística, KNN, SVC, Naive Bayes, Random Forest e Gradient Boosting respectivamente. A acurácia do meta modelo de Regressão Logística foi de 0.47, próximo da média dos modelos base. Esse resultado tem suas peculiaridades, pois caso o stacking prevesse o grau de instrução 7 para todos os registros, o grau mais abundante entre os registros, ele teria uma acurácia de 0.45, muito próximo da média de acertos dos modelos.

Além da acurácia, outras métricas foram levadas em consideração, tais como

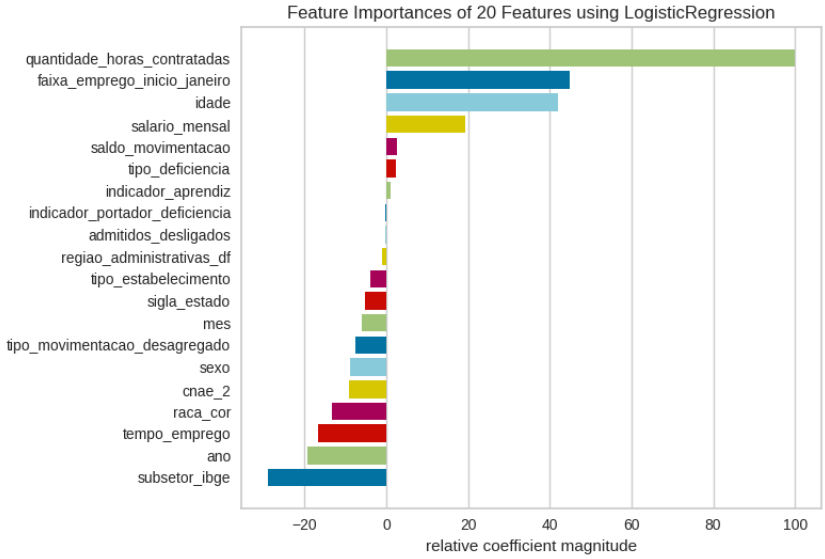
FIGURA 11 – Gráfico Importância de variáveis das 20 principais variáveis do modelo base Random Forest.



Fonte: Autor, 2023.

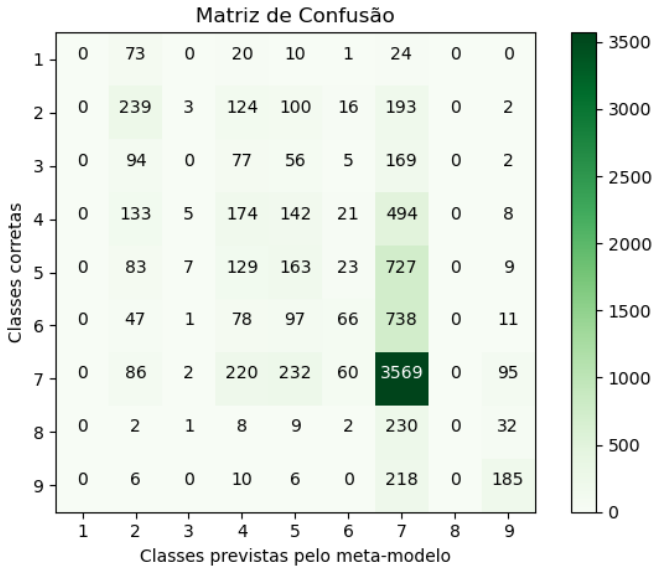
a precisão, recall e o F1-Score, porém, apenas para o modelo final e para o modelo de melhor desempenho (Random Forest). Essas visualizações podem ser consultadas através das imagens Figura 14 e Figura 15. Foi construído, da mesma forma, a matriz de confusão para o stacking multi classe, conforme a Figura 13.

FIGURA 12 – Gráfico Importância de variáveis das 20 principais variáveis do modelo base de Regressão Logística.



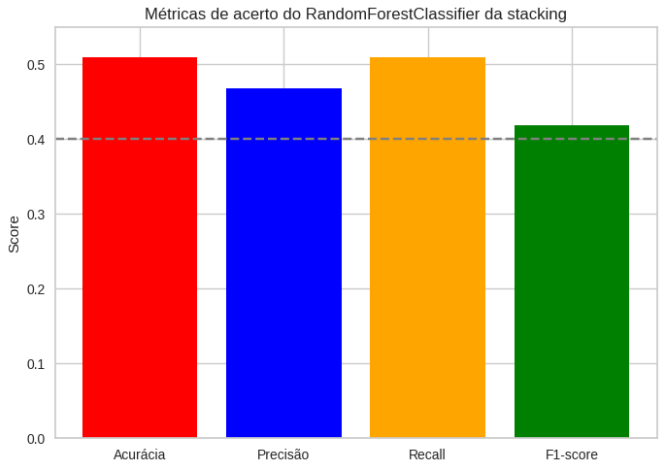
Fonte: Autor, 2023.

FIGURA 13 – Matriz de Confusão do meta modelo, a intensidade da cor verde indica a maior quantidade numérica de acertos, o ideal seria somente os termos da diagonal preenchidos.



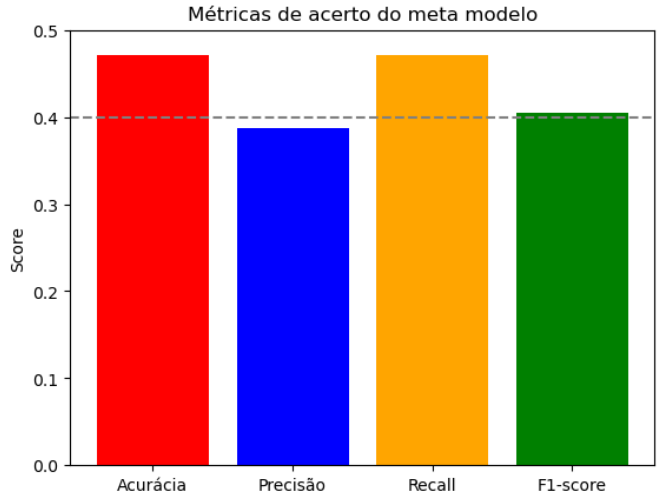
Fonte: Autor, 2023.

FIGURA 14 – Métricas de desempenho do modelo RandomForest o modelo base de melhor desempenho da stacking que supera o próprio meta modelo.



Fonte: Autor, 2023.

FIGURA 15 – Métricas de desempenho do modelo final (meta modelo), de Regressão Logística. Os valores obtidos são respectivamente 0.47 0.39 0.47 0.41)



Fonte: Autor, 2023.

4 CONCLUSÃO

Embora as métricas obtidas tanto pelos modelos base do stacking quanto do meta modelo fossem pouco assertivas e soassem, a primeira vista, um tanto decepcionantes; a construção de toda a análise, seleção de variáveis, limpeza e pré-processamento, e ainda a modelagem elaborada que utilizava-se de vários algoritmos com base somente nas poucas variáveis disponibilizadas pelo governo, tudo isso para a previsão de uma característica/classe com 9 possibilidades distintas e um valor médio para todas as métricas, no geral, próximo da faixa dos 50%, ainda soa um resultado bastante interessante.

Vale observar que, como estamos abordando os registros de pessoas ao longo de 3 anos, nossos dados não comportam-se de maneira constante no tempo, e uma abordagem ainda mais complexa, através de séries temporais, seria necessária para maior assertividade nas previsões (embora fuja do escopo dessa disciplina), e que talvez, ainda assim não seria tão eficiente em razão da escassez de variáveis que realmente se relacionariam com nosso target.

Vale lembrar que, ao longo desses 3 anos, temos ao total cerca de 198 milhões de registros, dos quais, apenas uma amostra aleatória de 46684 registros foram utilizados ao total (para teste, treino e validação), e que certamente contribuiu para o desempenho mediano obtido. Para lidar com essa quantidade massiva de dados, demandaria muito mais tempo e poder computacional que o necessário para aplicar os métodos aprendidos ao longo da disciplina, mas que segue como desafio para o futuro.

4.0.1 CÓDIGO

A consulta do código desenvolvido ao longo desse trabalho pode ser acessado através do GitHub, neste repositório.