Universidade do Minho
Escola de Engenharia
Mestrado em Engenharia Informática

Unidade Curricular de Dados de Aprendizagem Automática

Ano Letivo de 2024/2025

Diogo Rafael dos Santos Barros (a100600)
Pedro Emanuel Organista Silva (a100745)
Norberto Miguel Luzes Pais Pinto (pg55907)

21 de janeiro de 2025

DAA

Índice

1 Introdução

Com este trabalho prático, pretende-se desenvolver modelos de **Machine Learning**, capazes de prever a progressão de um **Comprometimento Cognitivo Leve (MCI)** para uma **Doença de Alzheimer (AD).**

Para isso, foram fornecidos dois datasets, pelos docentes, que contêm dados com um potencial para analisar e explorar o contexto proposto. O primeiro dataset, denominado **DShippo**, contém dados do **hipocampo**, uma região cerebral fundamental para a **memória** e altamente associada ao **desenvolvimento da AD.** O segundo dataset, **DSocc**, contém dados do **lobo occipital**, que não é tipicamente associado ao desenvolvimento de **demências.**

O resultado final esperado para este projeto é que os resultados obtidos com estes datasets nos confirmem, que as características da região do **hipocampos** são de uma maior importância na previsão da progressão de **MCI para AD**, quando comparadas com a região do **lobo occipital.**

2 Metodologia Aplicada

Neste projeto, utilizamos a metodologia **SEMMA**, para orientar o nosso desenvolvimento e a validação dos modelos de ML. Aplicamos a mesma metodologia da seguinte forma:

Sample:

- Os datasets **DShippo** e **DSocc** foram estruturados e fornecidos pelos docentes, com o **DShippo** dividido em datasets de treino e teste para o **KAGGLE** e o **DSocc**, só com um dataset de treino para comprovar o objetivo do projeto.

Explore:

- Efetuamos análises estatísticas e visualizações de alguns atributos, como **Age**, **Sex** e **Transitions**, para entender a distribuição dos dados, identificar, avaliar e perceber as diferenças nas características de ambos os datasets, para melhorar a nossa capacidade de prever a progressão de **MCI** para **AD.**

Modify:

- Os dados de todos os datasets foram **transformados e ajustados**, aplicando métodos de **normalização**, **seleção de atributos** e etc..., para garantir que apenas as características relevantes fossem usadas na aplicação aos modelos, para otimizar as suas performances.

Model:

- Foram aplicados múltiplos algoritmos de ML, incluindo os ensembles de **Max Voting** e **Stacking.** Todos os hiperparâmetros, de todos os modelos, foram testados e ajustados, com a ajuda da ferramenta de **GridSearchCV**, com o objetivo de **maximizar** o desempenho das métricas como, por exemplo, o **F1-Score Macro** e a **Accuracy.**

Assess:

- Os modelos, no fim, foram sujeitos a uma avaliação, com base nos resultados das métricas obtidos. Foram analisados os resultados obtidos em ambos os datasets, com o objetivo de **validar** a hipótese inicial proposta sobre as importâncias das regiões do **hipocampos** e do **occipital lobe** na previsão de **MCI** para **AD**,

Esta metodologia, aplicada de forma **iterativa**, permitiu que fossem efetuados **ajustes constantes** aos nossos modelos, para obtermos resultados mais confiáveis e alinhados com os objetivos do projeto.

3 Datasets do Projeto

O dataset **train_radiomics_occipital_CONTROL** possui **305 linhas** e **2181 atributos**, que representam as características radiómicas extraídas do **lobo occipital.** Este foi escolhido como um dataset de **controlo**, por se tratar de uma região que não está tipicamente associada à demência. O principal objetivo é comparar os resultados dos modelos aplicados a este dataset, com os obtidos na análise do hipocampo (DShippo), e pretende-se avaliar se os padrões identificados no **DSocc** são insuficientes para aprender e prever a progressão de **demências**, como o Alzheimer.

	ID	Image	Mask	diagnostics_Versions_PyRadiomics	diagnostics_Versions_Numpy	diagnostics_Versions_SimpleITK	diagnostics_Versions_PyWavelet	diagnostics_Versions_Python
0	006_S_0681	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_006_S_068...	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_006_S_068...	2.2.0	1.18.5	1.2.4	1.1.1	3.7.7
1	941_S_1203	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_941_S_120...	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_941_S_120...	2.2.0	1.18.5	1.2.4	1.1.1	3.7.7
2	011_S_0003	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_011_S_000...	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_011_S_000...	2.2.0	1.18.5	1.2.4	1.1.1	3.7.7
3	057_S_0779	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_057_S_077...	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_057_S_077...	2.2.0	1.18.5	1.2.4	1.1.1	3.7.7
4	033_S_0920	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_033_S_092...	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_033_S_092...	2.2.0	1.18.5	1.2.4	1.1.1	3.7.7

Figura 3.1: Header do dataset de controlo, DSocc

Nos datasets, **train_radiomics_hipocamp** e **test_radiomics_hipocamp**, existem um total de **305 linhas** e **2181 atributos** no dataset de **treino**, e **100 linhas** e **2180 atributos** no dataset de **teste.** Representam as características radiómicas extraídas da região do **hipocampo**. O objetivo destes datasets **(DShippo)** é verificar, se os padrões identificados nesta região, são específicos e cientificamente relevantes para a aprendizagem e a previsão da evolução de demências.

	ID	Image	Mask	diagnostics_Versions_PyRadiomics	diagnostics_Versions_Numpy	diagnostics_Versions_SimpleITK	diagnostics_Versions_PyWavelet	diagnostics_Versions_Python
0	006_S_0681	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_006_S_068...	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_006_S_068...	2.2.0	1.18.5	1.2.4	1.1.1	3.7.7
1	941_S_1203	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_941_S_120...	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_941_S_120...	2.2.0	1.18.5	1.2.4	1.1.1	3.7.7
2	011_S_0003	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_011_S_000...	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_011_S_000...	2.2.0	1.18.5	1.2.4	1.1.1	3.7.7
3	057_S_0779	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_057_S_077...	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_057_S_077...	2.2.0	1.18.5	1.2.4	1.1.1	3.7.7
4	033_S_0920	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_033_S_092...	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_033_S_092...	2.2.0	1.18.5	1.2.4	1.1.1	3.7.7

Figura 3.2: Header do dataset de treino, DShippo

	ID	Image	Mask	diagnostics_Versions_PyRadiomics	diagnostics_Versions_Numpy	diagnostics_Versions_SimpleITK	diagnostics_Versions_PyWavelet	diagnostics_Versions_Python
0	941_S_1194	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_941_S_119...	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_941_S_119...	2.2.0	1.18.5	1.2.4	1.1.1	3.7.7
1	036_S_0945	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_036_S_094...	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_036_S_094...	2.2.0	1.18.5	1.2.4	1.1.1	3.7.7
2	024_S_1171	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_024_S_117...	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_024_S_117...	2.2.0	1.18.5	1.2.4	1.1.1	3.7.7
3	035_S_0555	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_035_S_055...	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_035_S_055...	2.2.0	1.18.5	1.2.4	1.1.1	3.7.7
4	023_S_0081	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_023_S_008...	/notebooks/disk2/ DS2_FreeSurfer/ ADNI_023_S_008...	2.2.0	1.18.5	1.2.4	1.1.1	3.7.7

Figura 3.3: Header do dataset de teste, DShippo

4 Análise dos Dados

4.1 Compreensão dos Dados

Para compreender melhor os dados presentes nos datasets do **DSocc** e **DShippo**, analisamos as suas estruturas em relação aos tipos de **atributos (numéricos ou categóricos)**, à presença de **missing values**, **linhas duplicadas (duplicated values)**, **colunas duplicadas** e **colunas com valores únicos.** A partir desta análise, verificamos os seguintes factos para todos os datasets:

- Não apresentam quaisquer **missing values.**
- Não possuem **linhas duplicadas (duplicated values).**

Todos os datasets contêm os dois tipos principais de atributos: **numéricos** e **categóricos.** No dataset do **DSocc**, 2014 colunas são do tipo numérico **float64**, 147 colunas do tipo numérico **int64** e 20 colunas do tipo categórico **object.** Já no dataset de **treino do DShippo**, há 2014 colunas do tipo numérico **float64**, 147 do tipo numérico **int64** e 20 colunas do tipo categórico **object**, enquanto no dataset de **teste** há 2011 colunas do tipo numérico **float64**, 150 do tipo numérico **int64** e 19 do tipo categórico **object.**

Para terminar, durante a nossa análise dos datasets, identificamos ainda a presença de **colunas duplicadas** e **colunas com valores únicos.** A **remoção** destas colunas, mais à frente, vai ser importante para otimizar o uso de **recursos computacionais**, reduzir o **tempo de execução** e melhorar a **eficiência da análise de dados e resultados.**

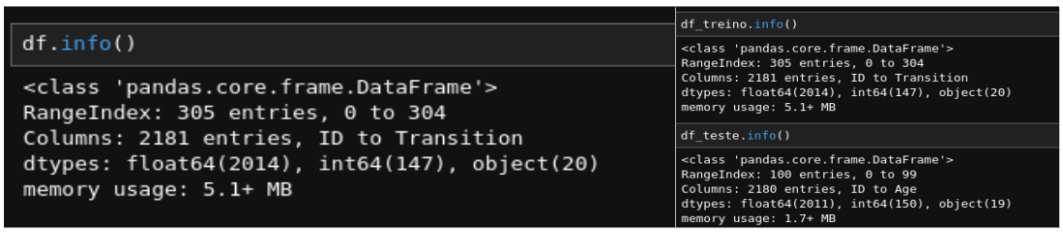

Figura 4.1: Comando **info()**, aplicado nos datasets

4.2 Visualização de Dados

Com o uso de **histogramas** e **boxplots**, foi possível explorar e compreender melhor alguns dos principais atributos de todos os datasets.

A análise dos dados, revelou que os atributos **Sex**, **Age** e **Transition** são idênticos nos datasets de treino do **DShippo** e do **DSocc**, visto que os exames de ressonância magnética foram realizados nos **mesmos pacientes.** A única diferença entre os datasets está nos dados capturados, já que os MRIs foram aplicados em **regiões diferentes do cérebro.**

Consequentemente, a análise dos histogramas e boxplots destes atributos para os dois datasets apresentam os mesmos resultados.

Age e Transition:

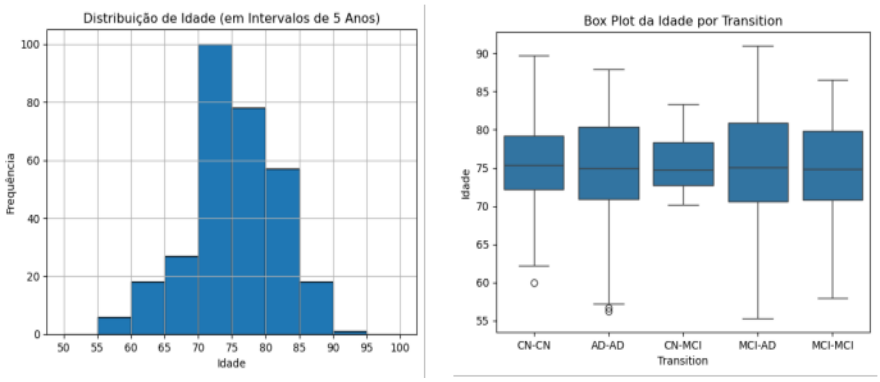

Figura 4.2: Frequência de Age e a sua relação com cada Transition

Com o **histograma**, na esquerda, conseguimos perceber que os intervalos de idade com mais **MRI's** efetuados são os intervalos de **[70-80].** Já o **box plot**, na direita, ajuda a visualizar a variação significativa da **Age** com os diferentes grupos de **Transition**, o que pode ser relevante em análises sobre a progressão ou prevalência de condições de saúde.

Sex e Transition:

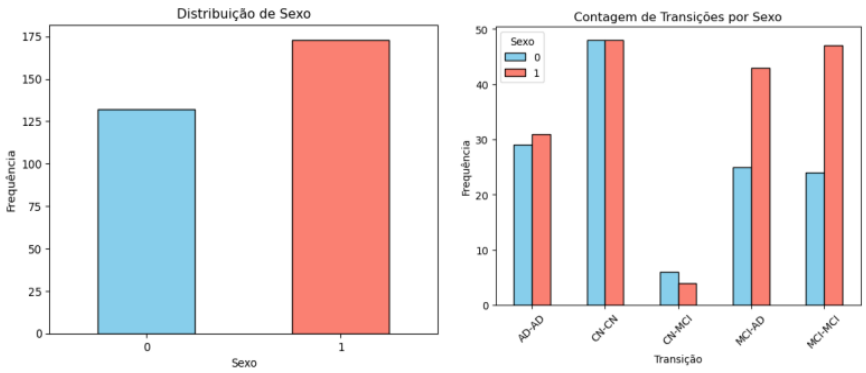

Figura 4.3: Distribuição do Sex e a sua frequência com cada Transition

Com o histograma à esquerda, é possível observar que há uma frequência mais elevada para o valor **1** na variável **Sex**. Já no histograma à direita, conseguimos visualizar as frequências de cada tipo de **Sex**, para cada tipo de **Transition.** Além disso, verificamos que em 4 dos 5 tipos de **Transition**, o Sex do tipo **1**, é superior ao seu oposto.

Transition:

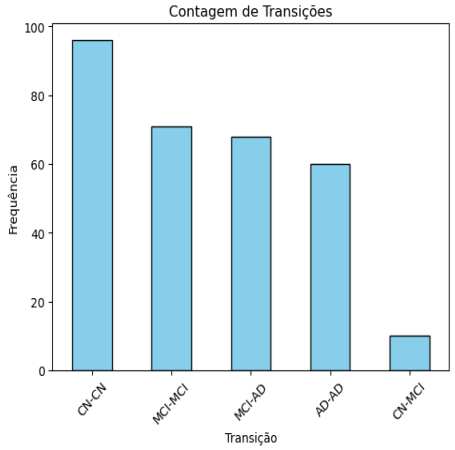

Figura 4.4: Frequência de cada Transition

Através deste histograma, é possível observar uma grande discrepância nas frequências entre as categorias de Transition. A **CN-CN** apresenta o **maior nº de ocorrências (96)**, enquanto **CN-MCI** tem a **menor frequência**, com apenas **10 ocorrências.** As outras categorias, como **MCI-MCI (71), MCI-AD (68) e AD-AD (60)**, encontram-se em níveis intermediários. Isto reflete um desequilíbrio nos dados, que pode influenciar o desempenho dos modelos de aprendizagem.

Sex e Age:

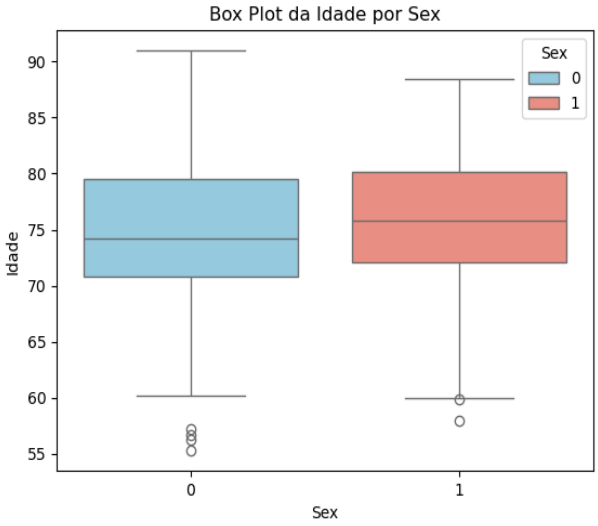

Figura 4.5: Distribuição da Age pelo Sex

Com este box plot, conseguimos visualizar as distribuições de idade semelhantes entre os sexos, com **medianas próximas** e outliers mais evidentes no Sex do tipo **0**, apesar de também existir no tipo **1**.

Por fim, para concluir a visualização e análise de alguns atributos dos nossos datasets, iremos examinar esses mesmos atributos no dataset de **teste do DShippo.**

Age:

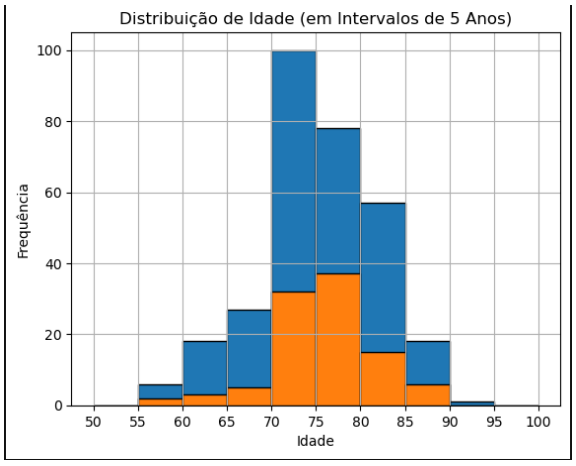

Figura 4.6: Frequência das Idades (Laranja)

Com este histograma, conseguimos observar que os intervalos de idade com **maior número de MRIs** realizados continuam a ser os intervalos de **[70-80].**

Sex:

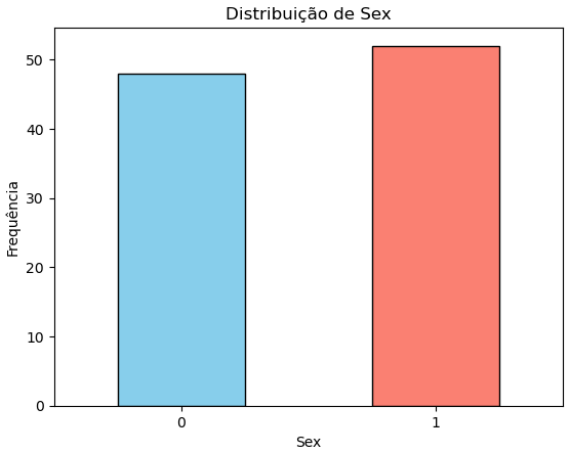

Figura 4.7: Frequência de cada tipo de Sex

Este histograma revela que, neste dataset de teste, a distribuição do atributo **Sex** está **mais equilibrada** entre os dois tipos de sexo: o tipo **1** possui **52 ocorrências**, enquanto o tipo **0** apresenta **48 ocorrências**.

Sex e Age:

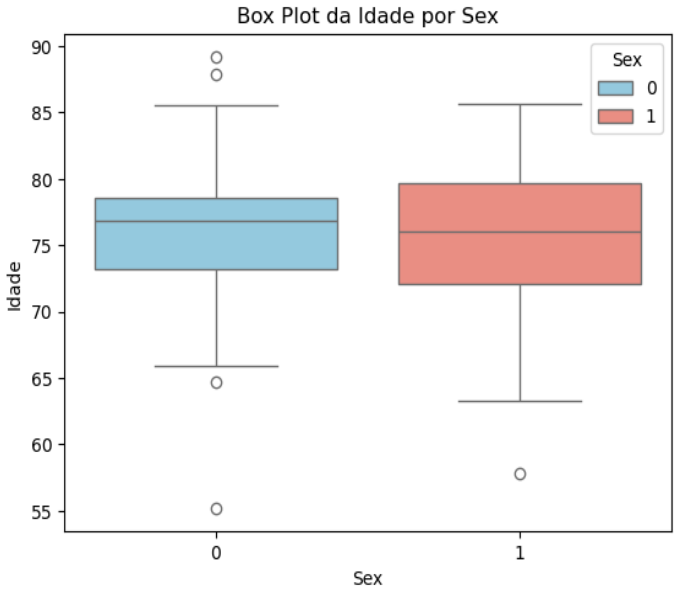

Figura 4.8: Distribuição da Age pelo Sex

O boxplot apresenta a distribuição da **Age** em função do **Sex**. Observa-se que ambos os **tipos de Sex** possuem idades predominantemente concentradas entre **70 e 80 anos.** Além disso, há **outliers** em ambos os tipos, indicando pacientes significativamente **mais jovens** ou **mais velhos** em relação à maioria.

5 Tratamento dos Dados

Antes de iniciar a construção dos modelos de **ML**, foi aplicado um processo de **tratamento de dados** igual para todos os datasets, garantindo a **consistência** e a **qualidade** nas análises subsequentes.

5.1 Remoção de Colunas de Valor Único

Colunas com valores únicos, ou seja, aquelas que possuem o **mesmo valor** em todas as **305 observações**, não contribuem para a diferenciação entre os dados nem para a construção de modelos de previsão. Estas colunas são **redundantes** e ocupam **espaço desnecessário** e aumentam também o **tempo de processamento.** Por isso, realizamos a identificação e remoção dessas colunas dos datasets:

- **DSocc:** 147 colunas com valores únicos removidas;
- **DShippo:** 159 colunas com valores únicos removidas nos dois datasets **(treino e teste)**.

5.2 Tratamento das Colunas do tipo Object

Durante a análise das colunas do tipo **object**, identificamos algumas colunas que se verificaram pouco relevantes para a previsão do atributo-alvo, **Transition**. As colunas identificadas são:

- **ID:** representa apenas o identificador das imagens obtidas.
- **Image e Mask:** referem-se apenas à localização dos diferentes scans, não fornecendo informações úteis para análises quantitativas.
- **diagnostics_Image-original_Hash:** representa um hash único da imagem original, usado para verificar sua integridade.
- **diagnostics_Mask-original_Hash:** similar à coluna anterior, este hash verifica a integridade da máscara original.

Após identificar essas colunas, procedemos com as suas eliminações, com o objetivo de otimizar o processamento e concentrar os recursos em dados mais relevantes para análise e aplicação em modelos de previsão.

Em seguida, identificamos também as seguintes colunas:

- **diagnostics_Mask-original_BoundingBox**: esta coluna contém tuplos de coordenadas dos vértices da ***bounding box***, que envolvem a região de interesse identificada pela máscara.
- **diagnostics_Mask-original_CenterOfMassIndex**: esta coluna contém tuplos de coordenadas do ***center of mass*** da região da máscara.

As colunas identificadas foram submetidas a processos de **transformação**, cujos detalhes e demonstrações serão apresentados a seguir:

- **Separação dos elementos dos tuplos:** as informações contidas nos tuplos de ambas as colunas foram separadas em colunas individuais, garantindo maior flexibilidade para processos de análise e previsão.
- **Remoção das colunas originais:** após a transformação, as colunas **diagnostics_Mask-original_BoundingBox** e **diagnostics_Mask-original_CenterOfMassIndex** foram removidas, visto que os seus conteúdos já tinham sido desmembrados e estavam presentes nas novas colunas criadas.

```python
# Separar os elementos dos tuplos em colunas individuais
bbox_cols = ['bbox_x1', 'bbox_y1', 'bbox_x2', 'bbox_y2', 'bbox_x3', 'bbox_y3']
com_cols = ['com_x', 'com_y', 'com_z']

df_teste[bbox_cols] = pd.DataFrame(df_teste['diagnostics_Mask-original_BoundingBox'].apply(eval).tolist(), index=df_teste.index)
df_teste[com_cols] = pd.DataFrame(df_teste['diagnostics_Mask-original_CenterOfMassIndex'].apply(eval).tolist(), index=df_teste.index)

# Remover as colunas originais
df_teste = df_teste.drop('diagnostics_Mask-original_BoundingBox', axis = 1)
df_teste = df_teste.drop('diagnostics_Mask-original_CenterOfMassIndex', axis=1)

# Explicação das colunas
# diagnostics_Mask-original_BoundingBox: Contém as coordenadas dos vértices da caixa delimitadora (bounding box).
# diagnostics_Mask-original_CenterOfMassIndex: Contém as coordenadas do centro de massa.

# Verificar a existência de valores ausentes
missing_values = df_teste.isnull().sum()

if missing_values.any():
    print('Sim')

    # Imprimir os nomes das colunas com valores ausentes
    cols_with_missing_values = missing_values[missing_values > 0].index.tolist()
    print("Colunas com valores ausentes:", cols_with_missing_values)
else:
    print('Não')
```

Figura 5.1: Transformações feitas nas colunas de tuplos

Resultado Final das Transformações:

```python
# Imprimir as primeiras linhas das colunas especificadas
columns = ['bbox_x1', 'bbox_y1', 'bbox_x2', 'bbox_y2', 'bbox_x3', 'bbox_y3', 'com_x', 'com_y', 'com_z']
print(df_teste[columns].head())
```

```
   bbox_x1  bbox_y1  bbox_x2  bbox_y2  bbox_x3  bbox_y3       com_x  \
0       92      123       90       42       29       76  114.350133
1       79      135       90       48       16       78  105.563676
2      105      112       87       45       27       80  128.066967
3       88      110       88       42       30       84  110.519840
4       88      145       94       45       21       71  112.503376

        com_y       com_z
0  137.932404  127.331456
1  144.513444  127.574921
2  125.648422  125.565615
3  125.623101  130.657365
4  155.283864  129.348767
```

Figura 5.2: Resultados das transformações feitas

5.3 Remoção das Colunas Duplicadas

No processo de pré-processamento dos dados, foi feita uma análise para identificar e eliminar colunas duplicadas em todos os conjuntos de dados.

DShippo:

- **Identificação de Colunas Duplicadas:** a identificação revelou que existiam **163** colunas duplicadas em **df_treino** e **168** colunas duplicadas em **df_teste.**
- **Identificação de Colunas Duplicadas Comuns:** em seguida, foi realizada uma análise para verificar quais das colunas duplicadas eram comuns nos **dois datasets.** Encontraram-se **163 colunas duplicadas comuns** entre os datasets.
- **Remoção das Colunas Duplicadas Comuns:** foi criada uma função, para remover as colunas duplicadas comuns dos dois datasets, preservando a **primeira ocorrência** de cada grupo de duplicadas. Após a aplicação da função, um total de **224** colunas duplicadas comuns foram **removidas em ambos os datasets.**

A interseção entre as colunas duplicadas de **df_treino e df_teste** foi realizada porque, ao remover as colunas duplicadas comuns, garantiu-se que ambos os conjuntos de dados ficassem **consistentes.** Isto é crucial para modelos de ML, já que os dados de treino e teste devem ter as mesmas **características** para garantir que o modelo seja avaliado corretamente.

DSocc:

- **Identificação de Colunas Duplicadas:** a identificação revelou que existiam **166** colunas duplicadas no dataset de controlo.
- **Remoção das Colunas Duplicadas** foi criada uma função, para remover as colunas duplicadas do dataset, preservando a **primeira ocorrência** de cada grupo de duplicadas. Após a aplicação da função, um total de **115** colunas duplicadas foram **removidas.**

O objetivo principal dessa etapa foi melhorar a **qualidade dos dados** e reduzir **redundâncias.** Colunas duplicadas podem afetar **negativamente** a performance do modelo, porque elas não têm **informações adicionais**, mas podem aumentar o **tempo de aprendizagem** e consumir **mais recursos.** A remoção destas colunas ajuda a garantir que o modelo se concentre nas variáveis **relevantes e únicas**, além de otimizar os **recursos computacionais.**

5.4 Normalização dos Dados

Nesta etapa, realizamos a **normalização** dos conjuntos de dados de todos os datasets. Isto é fundamental para garantir que os algoritmos utilizados, especialmente em técnicas de **ensemble learning**, operem de forma **eficiente** e **consistente.**

Para isso, utilizou-se o método **MinMaxScaler** da biblioteca **Scikit-learn**, que converte os valores dos atributos numéricos num intervalo uniforme compreendido entre **0** e **1**.

A normalização foi realizada principalmente porque, no processo de **ensemble learning**, serão testados modelos que necessitam de dados normalizados para operar corretamente. Técnicas como **Redes Neurais** e **SVM**, que frequentemente fazem parte de ensembles, têm o seu desempenho otimizado quando os dados estão normalizados.

5.5 Feature Selection e Feature Importance

No processo de feature selection, foram utilizadas três abordagens distintas para identificar **atributos com importância nula**, com a ajuda de um modelo de **Decision Tree Classifier.** O objetivo principal foi reduzir a **dimensionalidade dos dados**, eliminando atributos que não contribuem para o desempenho do modelo, tornando possível melhorar a **eficiência computacional** e a **performance do modelo final.**

A primeira abordagem aplicada foi o método de **Mean Decrease in Impurity (MDI)**, que avalia a importância dos atributos com base na **redução da impureza**, como o índice de Gini ou entropia.

A segunda abordagem utilizou a **Permutation Importance**, que mede a importância de cada **atributo** ao embaralhar os seus valores e a observar a variação das métricas de desempenho dos modelos. Este método avalia o **impacto direto** de cada atributo no **desempenho do modelo** de maneira robusta, com várias repetições para se obter maior confiabilidade nos resultados.

A terceira abordagem utilizou os valores do **SHAP (SHapley Additive exPlanations)**, que explicam o impacto de cada atributo individual nas previsões do modelo, utilizando os conceitos das **teoria dos jogos.**

Após identificar os atributos com importância nula em cada abordagem, foi realizada a **interseção dos resultados**, para identificar os atributos comuns considerados irrelevantes por todas as **três metodologias.** Estes atributos comuns foram, então, removidos dos datasets, reduzindo significativamente a dimensionalidade dos dados.

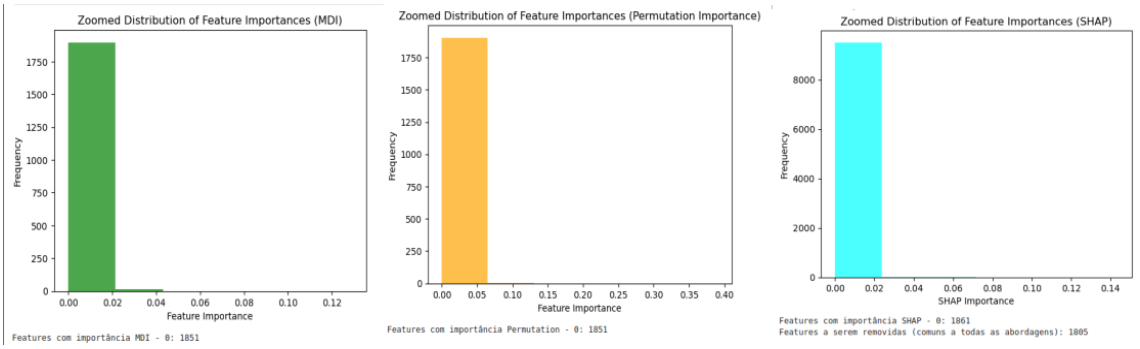

Figura 5.3: Distribuição das Importâncias das Features

A análise de importância de features usando **MDI**, **Permutation Importance** e **SHAP** revelou que a maioria das features tem relevância **próxima de zero**, com mais de **1800** identificadas como irrelevantes em todas as abordagens. A interseção das análises indicou **1805** features redundantes, às quais foram aplicados processos de remoção, em todos os datasets. Esta seleção reduz a **dimensionalidade**, melhora a **interpretabilidade**, otimiza o **custo computacional** e pode aumentar o **desempenho dos modelos.**

6 Modelação dos Dados

Para o desenvolvimento de modelos capazes de prever a evolução de demências, explorámos diversos algoritmos de **aprendizagem supervisionada.** Os métodos utilizados incluem: **Decision Tree**, **Random Forest**, **Support Vector Machine**, **Multilayer Perceptron**, **Gradient Boosted Trees**, **Stacking**, **Max Voting** e **XGBoost**. Cada modelo foi cuidadosamente configurado e avaliado para identificar a abordagem mais eficaz e robusta, considerando o desempenho em métricas como precisão, recall e F1-score macro.

6.1 Decision Tree Classifier

Este modelo opera através da divisão dos dados em subconjuntos baseados em critérios definidos pelos **atributos**, criando uma estrutura composta por nodos e ramos, até este alcançar os resultados finais nas folhas da árvore.

Para este modelo, foram definidos os seguintes **hiperparâmetros:**

- **max_depth:** *5*
- **min_samples_split:** *5*
- **min_samples_leaf:** *20*
- **max_leaf_nodes:** *10*

Com esta configuração, o modelo alcançou um desempenho avaliado por meio da score pública de **F1-macro**, obtendo um valor de **0.34888.**

6.2 Random Forest Classifier

Este método de aprendizagem de **ensemble** utiliza várias árvores de decisão independentes para alcançar uma classificação **mais robusta e precisa**. A ideia central do modelo é combinar os resultados de todas as suas árvores, aumentando a **capacidade de generalização** e reduzindo a suscetibilidade **ao sobreajuste (overfitting)**, em comparação com os modelos isolados.

Para este modelo, foram definidos os seguintes **hiperparâmetros:**

- **bootstrap:** *False*
- **min_samples_split:** *15*
- **n_estimators:** *100*
- **min_samples_leaf:** *1*
- **max_depth:** *10*

Com esta configuração, o modelo alcançou um desempenho avaliado por meio da score pública de **F1-macro**, obtendo um valor de **0.36888.**

6.3 Gradient Boosting Classifier

Este método de aprendizagem de **ensemble** constrói modelos de forma sequencial, com o objetivo de corrigir os erros cometidos pelos **estimadores anteriores.** O Gradient Boosting combina árvores de decisão de maneira iterativa, onde cada árvore é treinada para **minimizar os resíduos/erros** do modelo anterior.

Embora seja altamente eficaz, este modelo exige cuidados adicionais com **hiperparâmetros**, como a **profundidade máxima** das árvores e a **taxa de aprendizagem**, para evitar **sobreajuste (overfit)** e para garantir um bom desempenho com dados desconhecidos e desbalanceados, como é o caso dos datasets deste projeto.

Para este modelo, foram definidos os seguintes **hiperparâmetros:**

- **learning_rate:** *0.1*
- **n_estimators:** *100*
- **max_depth:** *3*
- **min_samples_split:** *2*
- **min_samples_leaf:** *1*
- **max_leaf_nodes:** *None*

Com esta configuração, o modelo alcançou um desempenho avaliado por meio da score pública de **F1-macro**, obtendo um valor de **0.38000.**

6.4 XGBoosting Classifier

O **XGBoost Classifier (Extreme Gradient Boosting)** é uma versão do método de **Gradient Boosting**, projetada para ser altamente eficiente. Este modelo utiliza árvores de decisão, como estimadores base e usa **otimizações adicionais**, para trabalhar com grandes conjuntos de dados.

Para utilizar este modelo, foi necessário aplicar o **LabelEncoder**, que converte os dados dos conjuntos de **treino** e **teste (y_treino/y_teste)** de **valores categóricos** para **valores numéricos**, para garantir a compatibilidade com o modelo **XGBoost.**

Para este modelo, foram definidos os seguintes **hiperparâmetros:**

- **learning_rate:** *0.5*
- **n_estimators:** *100*
- **max_depth:** *5*
- **gamma:** *0.2*
- **min_child_weight:** *2*
- **colsample_bytree:** *0.8*

Com esta configuração, o modelo alcançou um desempenho avaliado por meio da score pública de **F1-macro**, obtendo um valor de **0.33992.**

6.5 Support Vector Machine

O SVM é um modelo de aprendizagem supervisionado, cujo o seu objetivo principal, é encontrar o **hiperplano** que melhor separa as **classes** no espaço de **características**, maximizando a margem entre os pontos **mais próximos** de cada classe, conhecidos como os **vetores de suporte.**

Para este modelo, foram definidos os seguintes **hiperparâmetros:**

- **C:** *1*
- **gamma:** *0.1*
- **kernel:** *'linear'*

Com esta configuração, o modelo alcançou um desempenho avaliado por meio da score pública de **F1-macro**, obtendo um valor de **0.37128.**

6.6 Multilayer Perceptron Classifier

O **MLP** é um modelo de aprendizagem supervisionado baseado em **redes neurais artificiais.** Este utiliza vários **neurónios conectados**, organizados numa estrutura de camadas de entrada, ocultas e de saída.

Cada **neurónio** aplica uma função de ativação, onde captura relações entre os dados. O treino do modelo é realizado por meio de um algoritmo, que ajusta os **pesos das conexões** para minimizar uma função de **perda.**

Para este modelo, foram definidos os seguintes **hiperparâmetros:**

- **hidden_layer_sizes:** *(100,50)*
- **activation:** *'tanh'*
- **solver:** *'lbfgs'*
- **alpha:** *0.001*
- **max_iter:** *1000*
- **early_stopping:** *True*

Com esta configuração, o modelo alcançou um desempenho avaliado por meio da score pública de **F1-macro**, obtendo um valor de **0.36455.**

6.7 Stacking Classifier

Este método de aprendizagem por ensemble combina diversos modelos para criar um classificador final **mais robusto** e **preciso**. Para isso, foram utilizados alguns dos modelos previamente analisados, mantendo os mesmos hiperparâmetros configurados:

- **Gradient Boosting Classifier:** com os hiperparâmetros da score de *0.38000.*
- **Support Vector Machine:** com os hiperparâmetros da score de *0.37128.*
- **Multilayer Perceptron Classifier:** com os hiperparâmetros da score de *0.36455.*

O modelo final foi treinado através do modelo de **Random Forest Classifier (RFC)**, com a score pública de **0.36888**, que combina as previsões dos estimadores base para tomar a decisão final.

Este método aproveita as forças individuais de cada modelo, permitindo capturar diferentes aspectos dos dados e melhorar a capacidade de **generalização de classes.** Com esta configuração, o **Stacking Classifier** alcançou uma pontuação de score pública de F1-macro de **0.36761**, demonstrando um desempenho competitivo ao integrar as previsões dos modelos analisados.

6.8 Max Voting Classifier

Este método de aprendizagem por ensemble combina previsões de múltiplos estimadores base para obter uma classificação final mais confiável. Neste projeto, utilizamos os seguintes modelos como estimadores:

- **Gradient Boosting Classifier:** com os hiperparâmetros da score de *0.38000.*
- **Support Vector Machine:** com os hiperparâmetros da score de *0.37128.*
- **Multilayer Perceptron Classifier:** com os hiperparâmetros da score de *0.36455.*

O modelo foi configurado para utilizar a votação do tipo **hard**, em que a classe final é determinada pela **maioria das previsões feitas** pelos estimadores base. Neste processo, cada modelo **"vota"** numa classe, e a classe que receber o **maior número de votos** é selecionada como o **resultado final.** Além disso, foram atribuídos **pesos diferentes** aos modelos para refletir a sua importância relativa: **GBM** e **SVM** receberam um **peso de 2**, enquanto o **MLP** recebeu um **peso de 1.**

Esta abordagem alcançou uma pontuação de score pública de F1-macro de **0.39836**, demonstrando um desempenho competitivo ao combinar a força dos modelos base de forma balanceada.

O **MVC** destacou-se de todos os modelos **treinados** e **testados**, visto que , alcançou a melhor pontuação de score pública. Este resultado demonstra a sua maior eficácia na generalização de **diferentes perspectivas dos dados**, melhorando a **robustez** e a **confiabilidade** do resultado final.

7 Resultados do DShippo

Modelo	Accuracy	AD-AD PR	CN-CN PR	CN-MCI PR	MCI-AD PR	MCI-MCI PR	Macro AVG
DTC	0.56	0.58	0.62	0.00	0.75	0.31	0.44
GBC	0.57	0.56	0.64	0.00	0.40	0.62	0.44
RFC	0.51	0.57	0.53	0.00	0.36	0.50	0.38
XGB	0.52	0.50	0.61	0.00	0.38	0.55	0.41
SVC	0.41	0.39	0.52	0.00	0.22	0.36	0.31
MLP	0.46	0.32	0.65	0.00	0.14	0.59	0.35
STC	0.54	0.35	0.68	0.00	0.40	0.83	0.41
MVC	0.49	0.38	0.64	0.00	0.29	0.55	0.37

Tabela 7.1: Resultados retirados dos Classification Reports dos Modelos

Modelo	AD-AD	CN-CN	CN-MCI	MCI-AD	MCI-MCI
DTC	Acertos: 11 Falhas: 1	Acertos: 13 Falhas: 6	Acertos: 0 Falhas: 2	Acertos: 6 Falhas: 8	Acertos: 4 Falhas: 10
GBC	Acertos: 10 Falhas: 2	Acertos: 16 Falhas: 3	Acertos: 0 Falhas: 2	Acertos: 4 Falhas: 10	Acertos: 5 Falhas: 9
RFC	Acertos: 8 Falhas: 4	Acertos: 16 Falhas: 3	Acertos: 0 Falhas: 2	Acertos: 4 Falhas: 10	Acertos: 3 Falhas: 11
XGB	Acertos: 7 Falhas: 5	Acertos: 14 Falhas: 5	Acertos: 0 Falhas: 2	Acertos: 5 Falhas: 9	Acertos: 6 Falhas: 8
SVC	Acertos: 7 Falhas: 5	Acertos: 12 Falhas: 7	Acertos: 0 Falhas: 2	Acertos: 2 Falhas: 12	Acertos: 4 Falhas: 10
MLP	Acertos: 6 Falhas: 6	Acertos: 11 Falhas: 8	Acertos: 0 Falhas: 2	Acertos: 1 Falhas: 13	Acertos: 10 Falhas: 4
STC	Acertos: 7 Falhas: 5	Acertos: 17 Falhas: 2	Acertos: 0 Falhas: 2	Acertos: 4 Falhas: 10	Acertos: 5 Falhas: 9
MVC	Acertos: 8 Falhas: 4	Acertos: 14 Falhas: 5	Acertos: 0 Falhas: 2	Acertos: 2 Falhas: 12	Acertos: 6 Falhas: 8
Total do Teste	12	19	2	14	14

Tabela 7.2: Quantidades de Acertos e Falhas por Modelo para as Transições

8 Resultados do DSocc

Modelo	Accuracy	AD-AD PR	CN-CN PR	CN-MCI PR	MCI-AD PR	MCI-MCI PR	Macro AVG
DTC	0.33	0.32	0.44	0.00	0.30	0.21	0.26
GBC	0.31	0.44	0.28	0.00	0.21	0.38	0.26
RFC	0.33	0.40	0.31	0.00	0.25	0.36	0.25
XGB	0.25	0.36	0.23	0.00	0.14	0.21	0.19
SVC	0.31	0.42	0.37	0.00	0.00	0.33	0.23
MLP	0.20	0.27	0.33	0.00	0.00	0.12	0.15
STC	0.38	0.44	0.36	0.00	0.40	0.33	0.29
MVC	0.30	0.38	0.35	0.00	0.00	0.25	0.20

Tabela 8.1: Resultados retirados dos Classification Reports dos Modelos

Modelo	AD-AD	CN-CN	CN-MCI	MCI-AD	MCI-MCI
DTC	Acertos: 6 Falhas: 6	Acertos: 8 Falhas: 11	Acertos: 0 Falhas: 2	Acertos: 3 Falhas: 11	Acertos: 3 Falhas: 11
GBC	Acertos: 4 Falhas: 8	Acertos: 7 Falhas: 12	Acertos: 0 Falhas: 2	Acertos: 3 Falhas: 11	Acertos: 5 Falhas: 9
RFC	Acertos: 4 Falhas: 8	Acertos: 10 Falhas: 9	Acertos: 0 Falhas: 2	Acertos: 2 Falhas: 12	Acertos: 4 Falhas: 10
XGB	Acertos: 5 Falhas: 7	Acertos: 6 Falhas: 13	Acertos: 0 Falhas: 2	Acertos: 1 Falhas: 13	Acertos: 3 Falhas: 11
SVC	Acertos: 5 Falhas: 7	Acertos: 11 Falhas: 8	Acertos: 0 Falhas: 2	Acertos: 0 Falhas: 14	Acertos: 3 Falhas: 11
MLP	Acertos: 3 Falhas: 9	Acertos: 7 Falhas: 12	Acertos: 0 Falhas: 2	Acertos: 0 Falhas: 14	Acertos: 2 Falhas: 12
STC	Acertos: 4 Falhas: 8	Acertos: 12 Falhas: 7	Acertos: 0 Falhas: 2	Acertos: 4 Falhas: 10	Acertos: 3 Falhas: 11
MVC	Acertos: 5 Falhas: 7	Acertos: 11 Falhas: 8	Acertos: 0 Falhas: 2	Acertos: 0 Falhas: 14	Acertos: 2 Falhas: 12
Total do Teste	12	19	2	14	14

Tabela 8.2: Quantidades de Acertos e Falhas por Modelo para as Transições

9 Análise Crítica do Projeto

Este estudo teve como objetivo, avaliar a capacidade de prever a evolução de **demências**, com foco na **Transition** de **MCI** para **AD**, utilizando as características extraídas dos datasets da região do **hipocampo (DShippo)** e da região do **lobo occipital (DSocc).** Para isso, foram aplicadas várias técnicas de **limpeza** e **aprendizagem**, como descritas em cima, nos dois conjuntos de dados. Além disso, as métricas obtidas em submissões no **Kaggle**, especialmente a pontuação **F1-macro**, foram utilizadas para complementar as análises e validar os **modelos desenvolvidos.**

Para avaliar o desempenho, foram utilizadas métricas como a **precision (PR)** e o **Macro Average**, que foi testado localmente. Adicionalmente, o **F1-macro** obtido pelo **Kaggle** permitiu comparar a generalização dos modelos num ambiente externo.

Os resultados obtidos confirmam a importância do **hipocampo** como um biomarcador crucial na previsão da evolução das **demências**, particularmente na progressão de **MCI** para **AD** e na estabilidade de condições como **AD-AD** e **CN-CN**, destacando o seu papel essencial em pesquisas sobre a evolução de demências. No entanto, mesmo os melhores modelos enfrentaram dificuldades em prever **Transitions**, menos representadas, como **MCI-AD** e **CN-MCI**, devido ao desbalanceamento dos dados, o que limitou a capacidade de **generalização** dos padrões associados a essas **transições críticas.**

Este desbalanceamento, com o maior número de amostras de estados a ser as transições, **AD-AD** e **CN-CN**, impactou diretamente os resultados, levando os modelos a priorizarem os **padrões mais frequentes.** Isto resultou em baixos valores de precisão nas transições mais importantes para o estudo, como a **MCI-AD.** Nem com a aplicação de técnicas de geração de **dados sintéticos (SMOTE)**, estes problemas conseguiram ser mitigados.

Embora o **DSocc** ou a região do **lobo occipital**, tenha apresentado um desempenho **inferior**, demonstrou alguma utilidade em transições cognitivamente mais estáveis, como a **CN-CN**, sugerindo que pode desempenhar um papel secundário num estudo desta natureza. No entanto, a sua relevância direta na previsão da progressão de demências é muito **limitada**, em alguns modelos, até mesmo nula, o que reforça a sua posição como uma região de controlo ou dataset de controlo neste estudo, com pouca capacidade de previsão de demências.

Para concluir, as submissões no **Kaggle** e as análises dos resultados obtidos destacaram o **MVC** e o **GBC**, como os modelos mais promissores em termos de generalização, validando parcialmente os resultados obtidos no projeto.

10 Conclusão

O projeto cumpriu o objetivo de realizar uma análise de previsão ao utilizar um conjunto de dados complexo e técnicas de ML. Durante o desenvolvimento, foram aplicadas diversas etapas fundamentais, como a **seleção de features**, o **pré-processamento de dados**, a **construção** e a **avaliação** de modelos de previsão. Estes processos não só permitiram identificar os atributos mais relevantes para o problema, como também permitiram compreender o **desempenho** e as **limitações** de diferentes abordagens.

No geral, os resultados obtidos demonstram a eficácia dos modelos construídos e o uso de estratégias de seleção de features que reduziram significativamente a **dimensionalidade do conjunto de dados.** Ainda assim, os resultados finais mostram que há espaço para melhorias, especialmente na otimização do desempenho de previsão.

Recomendações e Sugestões:

- **Exploração e Valorização de Modelos Ensembles:** os modelos de ensembles apresentaram **resultados promissores**, mostrando que são capazes de capturar interações complexas. Recomenda-se ajustar/testar novos **hiperparâmetros** e até mesmo, explorar novas abordagens, como a **AdaBoost** ou **CatBoost**, para melhorar os resultados obtidos.
- **Técnicas de Balanceamento de Dados:** recomenda-se adicionar mais dados das classes minoritárias, compensando as transições com **menor quantidade.** Esta abordagem também pode ser combinada com técnicas como **SMOTE ou undersampling**, garantindo maior **precisão** na classificação e reduzindo o impacto do **desbalanceamento nos resultados.**
- **Validação e Generalização:** ampliar o processo de **validação cruzada** para incluir mais divisões e iterações, pode garantir uma maior **robustez** e **confiabilidade** nos modelos desenvolvidos.
- **Expansão do Projeto:** considerar a **inclusão de novas fontes de dados** ou a extração de **atributos adicionais relevantes**, que possam enriquecer o conjunto de informações, contribuindo para a melhoria da **qualidade e precisão das previsões.**

O projeto apresentou um **progresso significativo** na análise do problema e na construção de modelos de previsão, destacando a importância de uma abordagem sistemática para lidar com **dados complexos.** Embora os resultados obtenham **sucesso** em certos aspectos, as **recomendações** aqui apresentadas oferecem um caminho claro para alcançar uma **performance ainda mais robusta.** O trabalho realizado estabelece uma base sólida para **futuros estudos** e **implementações práticas.**