

Universidade do Minho
Escola de Engenharia
Mestrado em Engenharia informática

Dados e Aprendizagem Automática

Ano lectivo 2024/2025

Grupo 14

Diogo Pontes Pereira, PG55885

Ruben Miguel Ferreira Magalhães, PG56008

Miguel Gonzaga Morais de Magalhães, PG54099

João Pedro Alves Couto de Abreu, PG55895

Índice

1. Introdução	4
2. Metodologia utilizada	4
3. Descrição dos <i>Datasets</i>	5
3.1. <i>Dataset DShippo</i>	5
3.2. <i>Dataset DSocc</i>	5
3.3. Relação entre <i>datasets</i>	5
4. Preparação dos dados	6
4.1. Compreensão dos Dados	6
4.2. Análise e visualização dos dados	7
4.2.1. Exploração dos dados	7
4.3. Importância dos atributos	10
4.4. Tratamento dos dados	11
4.4.1. Feature Engineering	13
4.5. Segregação dos dados	13
5. Modelos e o seu treino	13
5.1. Técnicas de refinamento de Parâmetros	14
5.2. Modelos utilizados e descrição	14
5.2.1. <i>Random Forest</i>	14
5.2.2. <i>SVM</i>	15
5.2.3. <i>XGBoost</i>	15
5.2.4. <i>Ensemble</i>	16
5.3. Aplicação do dataset de controlo	17
5.3.1. Exemplo da utilização do dataset de controlo	17
6. Resultados obtidos e análise crítica	17
6.1. Tabela dos modelos	18
7. Conclusões e Reflexão Final	19

Índice de Figuras

Figure 1: Metodologia Utilizada	4
Figure 2: Informação sobre o <i>dataset</i>	7
Figure 3: Relação de Features	8
Figure 4: Distribuição de pessoas por faixa etária	9
Figure 5: Distribuição das <i>transactions</i>	9
Figure 6: Relação entre <i>transiction</i> e o sexo das pessoas	10
Figure 7: Relação entre <i>transiction</i> e a idade das pessoas	10
Figure 8: Importância dados Face uma RF	11
Figure 9: Análise SHAP com o modelo RF da 1ª submissão ..	11
Figure 10: Mapeamento da coluna dependente “Transition” ..	12
Figure 11: Equilíbrio das classes “Transition”	13
Figure 12: Resultados do <i>Random Forest</i>	15
Figure 13: Resultados do <i>SVM</i>	15
Figure 14: Resultados do <i>XGBoost</i>	16
Figure 15: Exemplo no modelo Smart Ensemble.	17

1. Introdução

Este relatório surge no âmbito do trabalho prático da UC de Dados e Aprendizagem Automática, com o objetivo de aprofundar os conhecimentos relativos ao Machine Learning adquiridos ao longo do semestre.

Este projeto insere-se num challenge do *Kaggle* com o propósito de através de várias imagens de MRI, analisar correlações dependendo do estado de desenvolvimento do paciente (CN-MCI-AD). Assim, através de uma metodologia estruturada, serão analisados, explorados e preparados os dados do conjunto fornecido, com o intuito de extrair informações relevantes sobre o problema proposto. Este processo permitirá conceber e otimizar múltiplos modelos de Machine Learning. No final deste projeto, o objetivo é desenvolver um modelo equilibrado que aproveite as correlações identificadas e seja capaz de prever, de forma precisa, o estado de desenvolvimento da doença de *Alzheimer* num paciente.

2. Metodologia utilizada

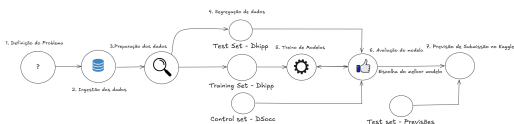


Figure 1: Metodologia Utilizada

A metodologia utilizada ao longo deste projeto foi uma adaptação à realizada nas aulas práticas. No **capítulo 1**. Realizamos a definição do problema: “Definir correlações nas características radiômicas do hipocampo que permitam prever a progressão e o estado da doença de *Alzheimer*.”

No **capítulo 2**. o processo foi relativamente simples, uma vez que os datasets foram fornecidos pela equipa docente. Não foi necessário implementar um processo complexo de ingestão de dados, uma vez que a nossa tarefa se limitou à leitura de ficheiros CSV. No **capítulo 3**. foi realizada a análise, exploração e tratamento dos dados, assim como o Feature engineering.

No **capítulo 4**. foi realizada uma segregação dos dados de treino classificados em dois sets. Um para realizar o treino do modelo descrito no **capítulo 5**. e outro para realizar os testes de previsão descrito de cada modelo no **capítulo 6**.

Também, utilizamos um dataset de controlo que serviu para garantir a fiabilidade do modelo na sua validação. No **capítulo 7**. com o dataset de test, realizamos previsões com o nosso melhor modelo para realizar submissões no desafio do *Kaggle*.

3. Descrição dos *Datasets*

O projeto baseia-se na utilização de dois *datasets* fornecidos pela equipa docente, cada uma a representar extrações de características radiômicas das diferentes áreas do cérebro – hipocampo e lobo occipital : *DShippo* e *DSocc*, respectivamente.

3.1. *Dataset DShippo*

O dataset *DShippo* cobre os casos de Alzheimer, capturando os 305 pacientes que se situam entre os 55 e 91 anos. Este dataset é composto por imagens de ressonância magnética (MRIs) de vários pacientes, sendo constituído por imagens do cérebro em que o hipocampo é destacado através de uma máscara. Esta máscara permite focar a análise nas características específicas dessa área, como a textura, a forma, a intensidade, entre outros parâmetros, possibilitando um estudo mais detalhado das correlações entre o estado do hipocampo e a progressão da doença de Alzheimer.

Como o hipocampo é estrutura cerebral fundamental para a formação de memórias, para pacientes com Alzheimer é das primeiras áreas do cérebro a ser afetada pela degeneração neuronal. Portanto, neste *dataset* as observações feitas ao hipocampo, como na diminuição do hipotálamo, o alargamento das regiões ocupadas por fluidos e o uso reduzido de glicose, entre outros fatores, mesmo que minuciosas podem fornecer insights importantes sobre a evolução do Alzheimer.

Devido a esta Natureza, o dataset apresenta uma elevadíssima dimensionalidade devido à quantidade de características necessárias para descrever um MRI (conjunto de 256 imagens bidimensionais, que empilhadas descrevem um objeto tridimensional). Além disso, este dataset é limitado pelos casos de uso presentes, com um relativo número reduzido de casos devidamente classificados. Esta interpretação do dataset será de uma maior importância para o seguimento do nosso projeto.

3.2. *Dataset DSocc*

O dataset *DSocc* é um dataset de controlo. Isto deve-se ao facto de a área de extração do lobo occipital ter sido selecionada com o propósito de utilização da region ROI, puramente de controlo, uma vez que essa área do cérebro não está associada à demência de Alzheimer. O lobo occipital é principalmente responsável pelo processamento visual e não apresenta as alterações patológicas características da doença de Alzheimer, como o acúmulo de proteínas beta-amiloide e tau.

3.3. Relação entre *datasets*

O objetivo de análise destes *datasets* é validar a hipótese de que as características radiômicas extraídas do hipocampo apresentam diferenças significativas, capazes de distinguir pacientes com *MCI* que evoluirão para Alzheimer daqueles que não evoluirão.

Neste projeto, tal como descrito, o *dataset* do hipocampo irá ser o dataset que vamos realizar o treino de modelos, já que esta é a mais relevante para a previsão do Alzheimer.

Além dos datasets relacionados com o treino, teste e validação do modelo. Foi-nos fornecido um dataset com 100 casos não classificados. O objetivo deste dataset, é realizar a previsão através dos nossos modelos finais.

4. Preparação dos dados

Antes de qualquer alteração ao *dataset*, é essencial compreender os seus atributos. Contudo, esta tarefa não é simples, uma vez que exige conhecimentos numa área especializada, que a maioria dos estudantes não domina, e requer um estudo adicional nesta fase. Além disso, o dataset apresenta uma elevada dimensionalidade, contendo um grande número de colunas. Esta característica, aliada ao facto de os dados estarem relacionados com ressonâncias magnéticas de pacientes na área da saúde, torna o dataset particularmente difícil de interpretar e compreender.

4.1. Compreensão dos Dados

Os diversos *features* do nosso *dataset* de treino são sub-divididos nas seguintes categorias descritas na tabela:

Categoria	Descrição
First Order statistics	Estas funcionalidades resumem a distribuição das intensidades dos voxéis dentro da região de interesse (ROI). Exemplos incluem média, mediana, desvio padrão, skewness e curtose.
Shape-Based Features	Estas funcionalidades capturam as propriedades geométricas tridimensionais (3D) da ROI, como volume, área de superfície, esfericidade e compacidade.
Shape-Based (3D)	Estas funcionalidades capturam as propriedades geométricas tridimensionais (3D) da ROI, como volume, área de superfície, esfericidade e compacidade.
Shape-Based (2D)	Estas funcionalidades descrevem propriedades geométricas em 2D com base nas projeções slice-a-slice da ROI. Exemplos incluem a área, perímetro e circularidade de cada fatia 2D.

Categoria	Descrição
Gray Level Co-occurrence Matrix (GLCM)	Descreve as relações espaciais entre pares de intensidades de píxeis. Funcionalidades comuns incluem contraste, correlação, energia e homogeneidade.
Gray Level Run Length Matrix (GLRLM)	Mede o comprimento de sequências consecutivas de níveis de cinza semelhantes numa dada direção, capturando a rugosidade e os padrões de textura.
Gray Level Size Zone Matrix (GLSZM)	Quantifica o tamanho das zonas homogêneas de níveis de cinza, focando-se na distribuição de tamanhos e na uniformidade de intensidade dentro da ROI.
Neighbouring Gray Tone Difference Matrix (NGTDM)	Mede a diferença entre a intensidade de um voxel e a intensidade média dos seus vizinhos, enfatizando a textura local.
Gray Level Dependence Matrix (GLDM)	Avalia a dependência das intensidades dos voxéis em relação aos seus vizinhos numa distância definida, capturando a complexidade das relações.

Tabela com as Categorias de *features radiômicas* e suas descrições

4.2. Análise e visualização dos dados

Nesta secção vamos entrar num contexto mais prático e começar a explorar mais informações do nosso *dataset* com a utilização de ferramentas de visualização com as várias bibliotecas disponíveis, como o *pandas*, o *numpy* *seaborn* e o *matplotlib* através do *Jupyter Notebook*.

4.2.1. Exploração dos dados

Inicialmente, verificamos quantas colunas e linhas o nosso *dataset* continha, sendo que tinha cerca de 2181 colunas e 305 linhas.

```
RangeIndex: 305 entries, 0 to 304
Columns: 2181 entries, ID to Transition
dtypes: float64(2014), int64(147), object(20)
memory usage: 5.1+ MB
```

Figure 2: Informação sobre o *dataset*

Exploramos relações entre a variável Transition e outros features no nosso dataset tal como 'diagnostics_Image-original_Mean', 'original_glm_Contrast', 'original_firstorder_Mean' etc. O diagnostics_Image-original_Mean vimos que há um maior *spread* no 'AD-AD' que sugere que consegue distinguir 'AD-AD' dos outros, mas por exemplo a 'original_glm_Contrast', as classes de transição sobrepõem, por isso esta feature pode não ser muito boa a prever.

Decidimos explorar a utilização do *ANOVA* (Analysis of Variance), um método estatístico amplamente utilizado para identificar diferenças significativas entre as médias de dois ou mais grupos. Este método é especialmente útil para analisar a relação entre variáveis categóricas e numéricas, como no nosso caso, onde queremos avaliar se os valores médios de determinadas features numéricas, como 'diagnostics_Image-original_Mean' e 'original_glm_Contrast', variam significativamente entre as diferentes classes da nossa label. Essa análise nos ajudará a identificar quais features possuem maior impacto na distinção entre as categorias da variável de interesse.

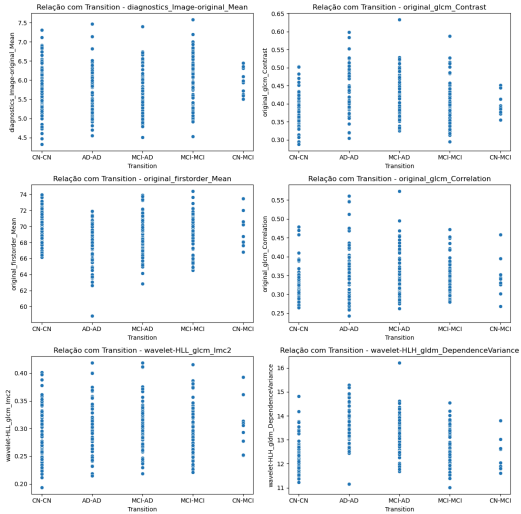


Figure 3: Relação de Features

Verificamos o número de pessoas por cada faixa etária e descobrimos que entre os 70-75 anos é a faixa com o número mais abundante de pacientes.

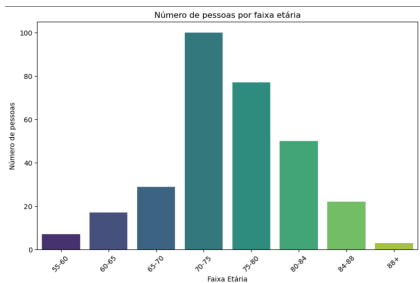


Figure 4: Distribuição de pessoas por faixa etária

Verificamos o número por cada categoria de *transactions* para verificar que tipo de transação é mais comum entre os pacientes e verificamos que o estado 'CN-CN' é o mais comum entre pacientes e o 'CN-MCT' mais incomum. Com isto podíamos ajustar o peso, por exemplo dar mais importância a CN-CN do que a CN-MCI

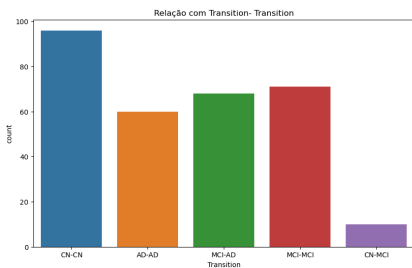


Figure 5: Distribuição das *transactions*

Depois de realizar as transformações dos nossos dados, decidimos fazer uma análise dos nossos dados. Verificamos o número de cada tipo de "Transaction" de acordo com cada faixa etária e com o sexo.

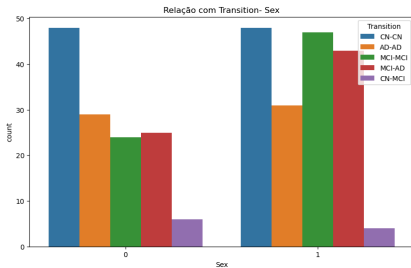


Figure 6: Relação entre *transition* e o sexo das pessoas

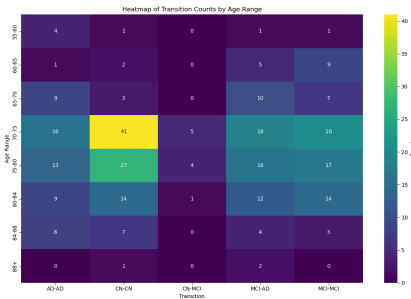


Figure 7: Relação entre *transition* e a idade das pessoas

4.3. Importância dos atributos

Devido à elevada dimensionalidade, é difícil exibir uma matriz de correlação, ou outro gráfico que compare os diversos atributos porque reduz a intuitividade.

No gráfico seguinte estabelecemos a importância dos atributos através do modelo *Random Forest*, que é uma métrica crucial para avaliar quais características do conjunto de dados têm maior impacto na predição do modelo.

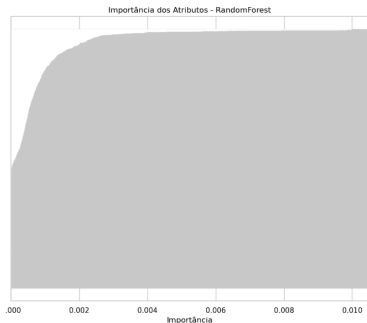


Figure 8: Importância dados Face uma RF

É possível visualizar que para o modelo RF, pelo menos metade das Features não tem qualquer importância. Isto significa que, se este fosse o modelo utilizado, aquelas *Features* poderiam ser retiradas.

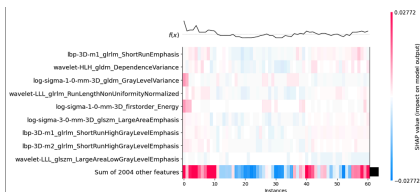


Figure 9: Análise SHAP com o modelo RF da 1ª submissão

Este gráfico apresenta os atributos mais relevantes, ordenados pela sua importância média, com as colunas a representar diferentes instâncias. As cores indicam o impacto das características: vermelho para impacto positivo e azul para negativo, com a intensidade refletindo a força da contribuição. Na linha superior, é exibida a previsão do modelo para cada instância. Aqui conseguimos realizar uma análise das características mais importantes, considerando a relevância entre elas.

4.4. Tratamento dos dados

Na análise dos dados do nosso *dataset*, identificamos diversos *features* que não serão considerados para o treinamento do modelo.

Inicialmente decidimos excluir *features* categóricas que não entregam nada de relevante para o modelo. Também removemos os *features* com valores únicos (nunique), que não contribuem para a previsão, e optamos por removê-los do *dataset*.

Aplicamos *label-encoding* no nosso label, sendo o “*Transition*”, para transformar os dados categóricos para dados numéricos ordenados de acordo com o estado do alzheimer de cada transação:

```
# Mapear a coluna 'Transition' para valores numéricos
mapping = {
    'CN-CN': 0, # Estado Normal
    'CN-MCI': 1, # Estado Intermediário
    'MCI-MCI': 2, # Estado Intermediário
    'MCI-AD': 3, # Demência
    'AD-AD': 4 # Demência
}
```

Figure 10: Mapeamento da coluna dependente “*Transition*”

Essa abordagem foi escolhida porque os valores categóricos da coluna possuem significado ordinal, representando diferentes estados de transação da doença e preserva a relação hierárquica sem criar várias colunas como no *one-hot* encoding. Além disso, vários modelos que pretendemos utilizar lidam melhor com representações numéricas, como o *XGBoost*, modelos lineares e *Support Vector Machines*.

Optámos por não remover os outliers (apesar de não serem muitos), pois não queríamos perder informação. Pelo mesmo motivo, apesar de no Capítulo **Tratamento de Dados** termos observado a existência de muitas *features* sem relevância, para vários modelos decidimos manter todas as *features* úteis, de forma a não perder informação.

Dentro das diversas implementações, das que obtiveram os melhores resultados no privado, utilizamos algumas técnicas como o *PCA*, *RFE*, *ANOVA* e o *SMOTE*.

O *PCA* permitia reduzir a elevada dimensionalidade dos dados, mantendo boa parte da informação mais relevante, o que facilitava o treinamento de modelos.

O *RFE* identifica os atributos mais importantes para um modelo já previamente treinado eliminando as menos significativas.

Também foi experimentado o *ANOVA* que é um método usado para determinar a diferença significativa entre as médias entre dois ou mais grupos, entre *features* categóricas e numéricas. Permitiu determinar se, por exemplo, os nossos *features* ‘*diagnostics_Image-original_Mean*’ e ‘*original_glm_Contrast numéricos*’, ao determinar se a média dos *features* numéricos varia muito de acordo com as classes da nossa *label*, como referido na Figura 3.

Como visto na Figura 5, existe um desequilíbrio entre as diversas classes, e para isso utilizamos o *SMOTE* para tentar equilibrar os datasets, apesar de localmente não apresentar muito sucesso.

4.4.1. Feature Engineering

Devido à incapacidade de manipular as várias Features relativas aos MRIs, ficamos limitados apenas ao Feature Engineering nos atributos “Age” e “Sex”. Como ilustrado na Figura 4, observa-se um desbalançamento na distribuição etária dos casos. Para mitigar este problema, aplicamos a técnica de *binning*, que em vez de armazenar a idade como um valor único, categorizamos os indivíduos em intervalos de idade adaptados à densidade de casos (intervalos menores para faixas com mais casos). Esta transformação resultou na criação de várias novas colunas, representando os bins definidos, enquanto a coluna original *Age* foi removida do *dataset*.

Em relação ao *Sex* não separamos apesar de ser interessante devido às pequenas diferenças biológicas entre os sexos, por alguns motivos. Primeiro, separar casos pelo sexo não seria muito inteligente porque estamos limitados a poucos casos, e isso iria reduzir ainda mais. Além disso, o *Age* parece mais relevante como visto na Figura 7, e queríamos que o modelo capturasse padrões gerais aplicáveis a toda a população.

4.5. Segregação dos dados

Para garantir uma avaliação adequada do modelo, os dados foram divididos em conjuntos de treino e teste com a função `train_test_split`. O conjunto de treino, que representa 80% dos dados, é utilizado para treinar o modelo, enquanto os 20% restantes, correspondentes ao conjunto de teste, são reservados para avaliar o desempenho do modelo com dados não vistos.

Em alguns modelos, utilizamos o parâmetro `stratify=y_train` para garantir que os dados de treino e teste mantenham a mesma proporção de classes como visto na Figura 11.

```

Distribuição de Classes no Treino:
transition
0    0.24756
1    0.24756
2    0.22993
3    0.22993
4    0.26721
1    0.002787
name: proportion, dtype: float64
Distribuição de Classes no Teste:
transition
0    0.24969
1    0.24969
2    0.26299
3    0.26299
4    0.21315
1    0.002787
name: proportion, dtype: float64

```

Figure 11: Equilíbrio das classes “Transition”

5. Modelos e o seu treino

O desafio que nos foi proposto é um problema de aprendizagem supervisionada já que o modelo é treinado com dados de entrada (*features*) e as suas saídas (*labels*). Trata-se, portanto, de um problema de classificação multiclasse, com o objetivo de atribuir os indivíduos a uma das categorias possíveis da label “Transaction”. Com base nisso, implementamos nos

seguintes modelos, o que acreditamos ser eficaz para esta tipologia de problema em questão.

5.1. Técnicas de refinamento de Parâmetros

Com a utilização dos métodos *GridSearch*, *RandomSearch* e métodos *bayesianos* foi possível identificar os valores ideais para os hiperparâmetros dos modelos, permitindo otimizar o desempenho e alcançar uma maior eficiência nas previsões.

O *GridSearch* é uma técnica de busca sistemática que avalia diferentes combinações de hiperparâmetros predefinidos. Ao testar cada combinação, o método seleciona aquela que resulta no melhor desempenho com base numa métrica de avaliação como o *F1-score*.

Apesar do *GridSearch* revelar os melhores valores para os principais hiperparâmetros, é extremamente custoso a nível computacional, principalmente quando não realizamos a limpeza de *features*.

Para tentar resolver este problema optámos pela utilização do *RandomSearch* que realiza uma busca aleatória entre os hiperparâmetros, selecionando combinações de forma não sistemática, o que o torna mais eficiente do que o *GridSearch* em cenários de alta dimensionalidade.

Já os métodos *bayesianos* utilizam uma abordagem probabilística para ajustar os hiperparâmetros. Com base num modelo probabilístico, estes métodos combinam os resultados anteriores para prever de forma mais informada as melhores combinações de hiperparâmetros.

Esta etapa de otimização é crucial, pois um ajuste incorreto dos hiperparâmetros pode levar ao overfitting ou underfitting, afetando a capacidade preditiva do modelo. Com os hiperparâmetros ideais, conseguimos maximizar a eficiência do modelo, equilibrando precisão, robustez e capacidade preditiva.

5.2. Modelos utilizados e descrição

Ao longo deste projeto, tentamos diversos modelos. Sempre com o objetivo de torná-los os mais equilibrados possíveis.

5.2.1. *Random Forest*

Random Forest é um algoritmo de aprendizagem supervisionada usado para problemas de classificação e/ou de regressão. Baseia-se em coleções de árvores de decisão que agrega as suas previsões num output final. Inicialmente testamos com hiperparâmetros base, mas com a utilização do *GridSearch* descobrimos que os melhores hiperparâmetros são:

- *n_estimators*: 75
- *max_depth*: None
- *min_samples_split*: 10
- *min_samples_leaf*: 4

Resultado:

	precision	recall	f1-score	support
CN-CN	0.46	0.76	0.58	17
CN-MCI	0.00	0.00	0.00	2
MCI-MCI	0.33	0.32	0.33	16
MCI-AD	0.31	0.38	0.34	13
AD-AD	0.55	0.46	0.50	13
accuracy			0.43	61
macro avg	0.33	0.35	0.32	61
weighted avg	0.40	0.43	0.39	61

Accuracy do modelo Random Forest: 0.4262295081967213

Figure 12: Resultados do *Random Forest*

5.2.2. SVM

Support Vector Machine (SVM) é uma algoritmo de aprendizagem supervisionada que classifica os dados ao encontrar a linha *optimal* ou o *hyperplane* que maximiza a distância entre as classes.

O *SVM* é ideal para problemas onde é crucial diferenciar estados, neste caso, são estados clínicos associados com o Alzheimer e é bastante eficaz com dados de alta dimensionalidade como no nosso caso.

Com a utilização do *GridSearch* estes foram os melhores hiperparâmetros que encontramos:

Melhores Hiperparâmetros

- C: 35
- kernel: 'rbf'
- probability: True

Nota: Depois de revelado os resultados privados, muito provavelmente no nosso Ensemble, o C foi o hiperparâmetro que apresentou Overfitting e prejudicou os nossos modelos finais. Apesar de sabido que um C elevado apresenta overfitting, este foi escolhidos por dois motivos: 1: Foi o melhor encontrado pelo *GridSearch*. 2: A elevada dimensionalidade do problema poderia trazer uma maior necessidade de ajustamento.

Resultado:

	precision	recall	f1-score	support
CN-CN	0.44	0.82	0.57	17
CN-MCI	0.00	0.00	0.00	2
MCI-MCI	0.00	0.00	0.00	16
MCI-AD	0.32	0.46	0.38	13
AD-AD	0.67	0.46	0.55	13
accuracy			0.43	61
macro avg	0.28	0.35	0.30	61
weighted avg	0.33	0.43	0.36	61

Accuracy do modelo SVM (RBF): 0.4262295081967213

Figure 13: Resultados do *SVM*

5.2.3. XGBoost

XGBoost (Extreme Gradient Boosting) é uma implementação otimizada do *Gradient Boosting*. O *XGBoost* constrói um conjunto de árvores de decisão

de forma sequencial, em que cada nova árvore corrige os erros cometidos pelas anteriores.

A utilização do *XGboost* deve se ao facto de ser excelente em capturar relações não-lineares entre as variáveis.

Com a utilização do *GridSearch* descobrimos estes valores como os melhores:

- *booster*: 'dart'
- *colsample_bytree*: 0.6
- *gamma*: 0.1
- *learning_rate*: 0.2
- *max_delta_step*: 1
- *max_depth*: 3
- *min_child_weight*: 3
- *n_estimators*: 97
- *reg_alpha*: 0.01
- *reg_lambda*: 10
- *subsample*: 0.6

Além do *dart*, outros boosters como o *glinear* e o *gtree* apresentaram resultados muito bons. Este modelo foi o modelo mais sólido em termos de resultados privados, sendo aquele que teve dos maiores valores.

Resultado:

	precision	recall	f1-score	support
ON-ON	0.45	0.76	0.57	17
ON-MCI	0.60	0.80	0.68	2
MCI-MCI	0.30	0.19	0.23	16
MCI-AD	0.33	0.38	0.36	13
AD-AD	0.71	0.38	0.50	13
accuracy			0.43	61
macro avg	0.36	0.34	0.33	61
weighted avg	0.43	0.43	0.40	61

Accuracy do modelo XGB (G): 0.4262295081967213

Figure 14: Resultados do *XGBoost*

5.2.4. Ensemble

Face ao modelos previamente criados, decidimos especializar cada modelo na classe do Transition que obtém mais sucessos na previsão, assim como um *f1_score*. Ao refinar os nossos modelos *Support Vector Machines*, *Random Forest* e *XGBoost*, tentamos por *VotingClassifier* como pela combinação ponderada de previsões onde a decisão final sobre a classe é tomada com base nas probabilidades obtidas de cada modelo, seguindo a regra de escolher a classe com a maior probabilidade.

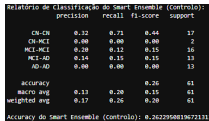
Alguns Ensembles obtiveram bons resultados no privado, porém outros obtiveram um Overfitting muito grande, principalmente aqueles que utilizavam o SVM na previsão da maioria das classes.

5.3. Aplicação do dataset de controlo

Para garantir que o modelo seja avaliado corretamente, podemos utilizar o *dataset* de controlo. Tal como já foi descrito na Descrição dos Datasets, este *dataset* é focado na região do lobo occipital por isso pode ser utilizado como controlo.

5.3.1. Exemplo da utilização do dataset de controlo

Para utilizar o *dataset* de Controlo fizemos exatamente o mesmo processo de tratamento de dados e treinamento de modelos que no *dataset* de treino.



	precision	recall	f1-score	support
CN-CN	0.12	0.71	0.44	17
CN-MCI	0.00	0.00	0.00	2
MCI-MCI	0.28	0.11	0.15	16
MCI-AD	0.14	0.15	0.15	13
AD-AD	0.00	0.00	0.00	13
accuracy			0.26	61
macro avg	0.11	0.28	0.15	41
weighted avg	0.17	0.26	0.20	61

Accuracy do Smart Ensemble (controlo): 0.2629568106721114

Figure 15: Exemplo no modelo Smart Ensemble.

Na Figura 15, o modelo apenas prevê o estado CN-CN. Isto provavelmente ocorre porque esse é o estado normal, que não apresenta diferenças do dataset de Controlo. Isto permite concluir que o modelo está a analisar as características necessárias para um modelo correto.

6. Resultados obtidos e análise crítica

Ao longo do projeto, desenvolvemos uma série de métodos e implementações com o objetivo na procura do melhor modelo. No início, começámos com um modelo simples de *Random Forest (RF)*, sem realizar alterações significativas nos dados.

Esta fase, foi uma fase caracteriza pela exploração de modelos. Procurámos optar por modelos que fossem os mais adequados face aos dados fornecidos, com especial atenção para *Random Forest*, *XGBoost (XGB)* e *Support Vector Machines (SVM)*.

Em seguida, obtivemos as submissões mais sólidas, baseadas na *XGB*, com estratégias semelhantes de exploração dos dados. As principais diferenças surgiram no refinamento dos parâmetros, com a utilização do *GridSearch*, de *métodos bayesianos*, *RandomSearch*, entre outros. Esta secção foi a mais demorada, porque exigiu tempos muito demorados de refinamento de parâmetros.

Depois de esgotarmos as possibilidades de otimização com o *XGB*, procedemos ao desenvolvimento do *Ensemble*. Devido à elevada dimensionalidade dos dados, recorremos ao uso de *SVM*, dado que este modelo apresentou melhores resultados na previsão de alguns estados de desenvolvimento da doença de Alzheimer, em comparação com o *XGB*.

O desenvolvimento dos *Ensembles* foi compreendido em duas duas fases:

1. Na primeira recorremos aos *VotingClassifier* com o *hard* e *soft* vote.
2. Na segunda tentamos combinar os melhores modelos, especializando cada modelo para a previsão de uma classe diferente do pacientes com Alzheimer.

Nesta segunda parte do *Ensemble*, cometemos os maiores erros neste projeto, porque incorretamente, começamos a priorizar demasiado o *f1_score* local o que levou a escolhas menos acertadas e comprometeu a obtenção de melhores resultados no score *privado*. Como consequência, os modelos apresentaram um elevado grau de *overfitting* face aos antecessores.

6.1. Tabela dos modelos

Na seguinte tabela, é apresentado as diversas submissões e respetivos resultados públicos e privados.

Submissão .csv	Resultado Público	Resultado Privado
EnsembleRS21	0.41397	0.30737
Smart-Ensemble-Weighted-Class	0.38714	0.36958
SVM-Submission	0.41111	0.30827
SVM-Ensemble	0.36289	0.39288
Smart-Ensemble-SequentialXGBSVM	0.40522	0.37483
XGBNewFeature	0.39297	0.36188
ModeloOverfitting	0.22344	0.40135
XGBN-FANOVAPCA	0.35212	0.39137
SVM-Ensemble2	0.30232	0.40277
SVM-XGB-Ensemble	0.32470	0.40666
XGB0.39Refinado	0.36349	0.38537
submissionXGB-booster	0.39111	0.40893

Submissão .csv	Resultado Público	Resultado Privado
submissionXGB-Bayesiana	0.32537	0.35904
submissionRFMDI	0.27426	0.38944
submissionRF	0.35040	0.37023

Resultados público/privados no Kaggle. Como podemos observar, obtivemos uma série de resultados muito positivos devido a uma série de escolhas. Infelizmente na fase final do projeto, optámos por decisões não tão corretas que resultaram num *overfitting*. A submissão *EnsembleRS21*, que foi a melhor no dataset público, foi relativamente fraca no privado.

7. Conclusões e Reflexão Final

Ao longo deste projeto, a equipa demonstrou grande interesse em experimentar novas técnicas, sempre com o objetivo de melhorar o que já estava a ser desenvolvido. Os dados com que trabalhámos apresentavam uma elevada complexidade e dimensionalidade, além de abordarem uma área na qual a equipa não tinha conhecimentos especializados, como em MRIs. Este desafio proporcionou à equipa uma oportunidade de ser proativa, permitindo-nos expandir os nossos conhecimentos para áreas que inicialmente nos eram desconhecidas.

Apesar de termos alcançado resultados satisfatórios no conjunto de dados privado ao longo do projeto, a nossa vontade de alcançar um desempenho ainda melhor levou-nos a experimentar novas abordagens, o que, por vezes, foi feito de forma um pouco impetuosa. No entanto, essa atitude de exploração foi importante para o nosso crescimento enquanto equipa e para o desenvolvimento de uma solução mais robusta.

No final, a combinação que gerou o melhor desempenho foi o modelo XGBoost (XGB), acompanhado dos tratamentos de dados descritos, como o PCA e o SMOTE, e refinado por uma busca de hiperparâmetros com GridSearch. Os melhores parâmetros encontrados para este modelo foram: {booster: gbtree, colsample_bytree: 0.6, gamma: 0.1, learning_rate: 0.2, max_delta_step: 1, max_depth: 3, min_child_weight: 3, n_estimators: 97, reg_alpha: 0.01, reg_lambda: 10, subsample: 0.6}

Em suma, o projeto proporcionou uma grande aprendizagem de técnicas com recurso a *Machine Learning*, e no final, acreditamos que contribuiu significativamente para a melhoria das competências em trabalhar com dados reais sobre um problema real.