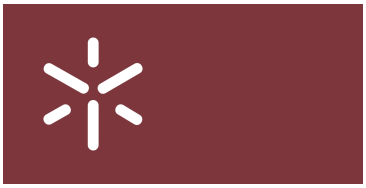

Universidade do Minho
Escola de Engenharia
Mestrado em Engenharia Informática

Unidade Curricular de Dados e Aprendizagem Automática

Ano Letivo de 2024/2025

Design and optimization of Machine Learning models

Grupo 2

Carlos Ribeiro	**Diogo Matos**	**Júlio Pinto**	**Lara Regina**
PG55926	PG55934	PG57883	PG57884

January 22, 2025

DAA

Índice

Lista de Figuras

Lista de Tabelas

1. Introdução

A Organização Mundial de Saúde (OMS) publicou um relatório denominado "Demência: Uma Prioridade de Saúde Pública", que apresenta dados sobre o estado da demência no mundo. Segundo este relatório, um novo caso de Demência é diagnosticado a cada 4 segundos. Em todo o mundo, cerca de 35,6 milhões de pessoas vivem com demência. Este número deverá duplicar até 2030 (65,7 milhões) e mais que triplicar em 2050 (115,4 milhões). A demência afeta pessoas em todos os países, com mais de metade dos casos (58%) em países desenvolvidos. Em 2050, este número é suscetível de aumentar para mais de 70%. Cuidar de pessoas com demência tem um custo atualmente estimado em mais de 604 mil milhões de dólares por ano [1]. A Doença de Alzheimer assume, neste âmbito, uma posição de destaque, representando 60-70% de todos os casos de demência [2].

Perante este cenário, é urgente o foco na garantia de uma resposta da saúde pública à demência, que passa pelo incentivo à investigação e inovação no diagnóstico, tratamento e cuidados. Neste projeto, pretendemos determinar se características de regiões cerebrais específicas podem ser utilizadas para prever a progressão de declínio cognitivo para Doença de Alzheimer. A investigação das diferenças entre o hipocampo e o lobo occipital, permite-nos validar cientificamente a relevância do estudo do hipocampo para esta tarefa. Para além disso, pretendemos explorar como modelos de aprendizagem automática permitem a extração de conclusões a partir dos dados fornecidos.

2. Contextualização e Objetivos

2.1. A Doença de Alzheimer e a importância de um diagnóstico precoce

A Doença de Alzheimer (em inglês, *Alzheimer's Disease*, AD) é o tipo de demência mais comum, constituindo cerca de 50% a 70% dos casos. Provoca uma deterioração global, progressiva e irreversível de diversas funções cognitivas (memória, atenção, concentração, linguagem, pensamento, entre outras). Esta deterioração tem como consequências alterações no comportamento, na personalidade e na capacidade funcional da pessoa, dificultando a realização das suas atividades de vida diária [3].

Vários estudos comprovam que um antecessor comum da Doença de Alzheimer é o Declínio Cognitivo Leve (em inglês, *Mild Congnitive Impairment*, MCI), como podemos verificar na Figura 1. Esta condição também se caracteriza por défices de memória, dificuldades na aprendizagem e na concentração, porém, não é grave o suficiente para ser caracterizado como demência. Estudos demonstraram que as pessoas com MCI e perda de memória têm maior probabilidade de desenvolver demência por doença de Alzheimer (cerca de 10% a 15% dos casos por ano) do que as pessoas sem MCI (1% a 2% por ano)[4].

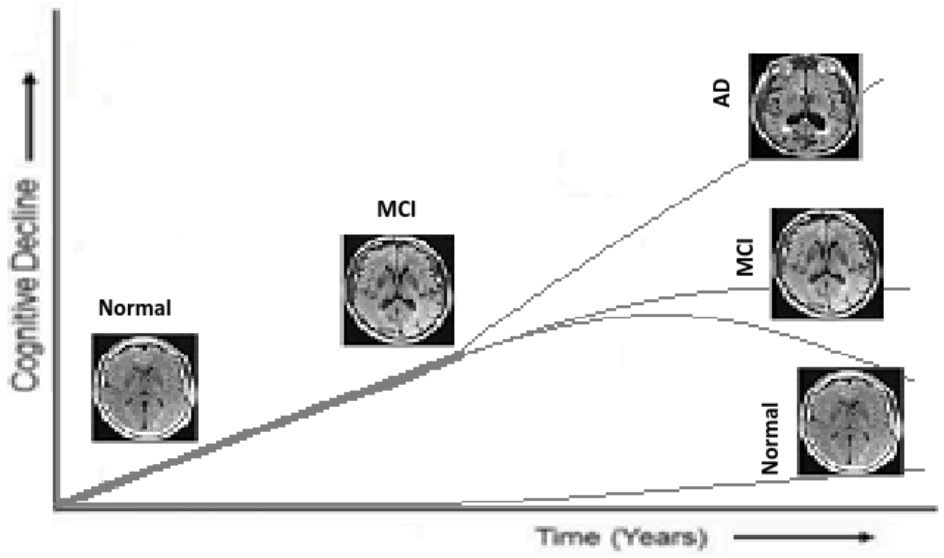

Figura 1: Progressão de um estado Normal, para Declínio Cognitivo Leve (MCI) ou Doença de Alzheimer (AD), com o avançar da idade de um indivíduo

O diagnóstico precoce de declínio cognitivo é crucial para que seja possível receber um tratamento precoce que previna ou atrase a maior degradação das capacidades cognitivas. Deste modo, para além de ser essencial a investigação da Doença de Alzheimer, é de extrema importância o estudo da previsão do seu aparecimento, a tempo de permitir um eventual tratamento, e assim reduzir o número de pessoas com demência a longo prazo.

2.2. Análise de dados no diagnóstico do declínio cognitivo

Atualmente, a forma mais fiável de diagnosticar a Doença de Alzheimer consiste no uso da ressonância magnética (em inglês, *Magnetic Resonance Imaging*, MRI). Este procedimento permite detetar alterações nas estruturas cerebrais que indicam o início da Doença de Alzheimer, nomeadamente a diminuição do hipocampo, uma região essencial do cérebro responsável pela memória (Figura 2).

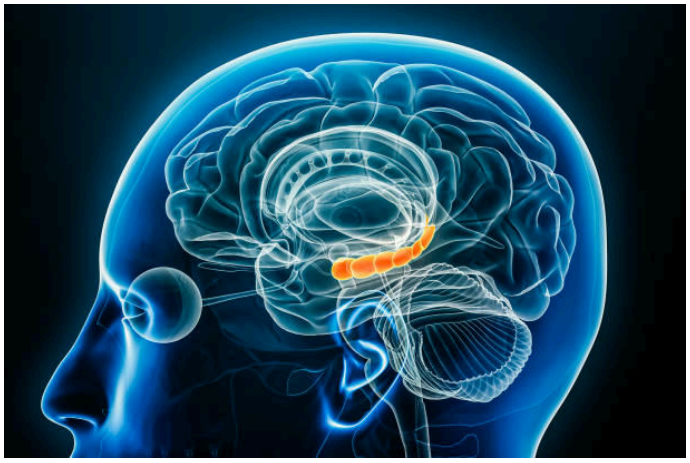

Figura 2: Hipocampo - Região do cérebro responsável pela memória

Para análise dos dados extraídos através da MRI, podemos usar *radiomics*, uma abordagem quantitativa à imagiologia médica que fornece informações textuais através da extração matemática da distribuição espacial das intensidades de sinal e das relações entre pixeis.

Desta forma, a partir das características extraídas via *radiomics*, optámos por definir como objetivo principal deste projeto a utilização de métodos de aprendizagem automática (em inglês, *Machine Learning*, ML) para a análise destes dados e consequente previsão da progressão de MCI para Doença de Alzheimer.

2.3. Objetivos do projeto

Com este projeto, propomo-nos a alcançar os seguintes objetivos:

- Explorar e analisar os conjuntos de dados fornecidos: `DShippo`, com informação relativa ao hipocampo, uma área com relevância conhecida na investigação sobre Alzheimer; e `DSocc`, com informação relativa ao lobo occipital, é um *dataset* de controlo uma vez que esta região não é tipicamente associada à demência.
- Extrair informações relevantes de ambos os *datasets*, que permitam compreender a relação entre as características *radiomics* e a progressão do MCI para AD.
- Desenvolver e otimizar modelos de *Machine Learning*, com o propósito de prever quais os pacientes com défice cognitivo (MCI) que têm a maior probabilidade de progressão para Alzheimer (AD).
- Testar a hipótese científica de que as características *radiomics* do hipocampo apresentam diferenças significativas entre pacientes que evoluem para AD e os que não evoluem, enquanto o lobo occipital não apresenta tal associação.
- Utilizar métricas para avaliar o desempenho dos modelos aplicados a ambos os *datasets*, permitindo comparar os resultados.
- Analisar criticamente e interpretar os resultados, com base nas métricas calculadas. Determinar a utilidade prática dos resultados no contexto clínico e identificar as características *radiomics* mais relevantes para prever a evolução da demência.

3. Metodologia

Para o desenvolvimento deste projeto, adotamos uma metodologia baseada na *Data Science Pipeline*. Esta escolha visa melhorar a estruturação e organização do projeto, a promoção de boas práticas que maximizem a qualidade dos resultados obtidos. Para além disto, uma abordagem metodológica é essencial para garantir a replicabilidade do projeto.

Optámos pela metodologia CRISP-DM (*Cross Industry Standard Process for Data Mining*), publicada em 1999 e amplamente reconhecida como uma referência nos campos da ciência de dados e aprendizagem automática [5]. O nosso processo compreendeu seis fases, representadas na Figura 3 e detalhadas de seguida.

Figura 3: Metodologia CRISP-DM

4. Análise e exploração dos dados

Como os datasets têm um número demasiado grande de colunas para avaliar cada uma individualmente, recorremos a uma *approach* mais holística, usando estatísticas sobre as colunas em geral para as caracterizar, e ajudar a tomar decisões.

4.1. *Dataset* `DShippo`

4.1.1. Caracterização dos atributos

Como já referido anteriormente, este *dataset* apresenta as características *radiomics* do hipocampo, a região do cérebro responsável pela memória e cuja diminuição no tamanho está relacionada, segundo vários estudos científicos, com o declínio da função cognitiva [6].

Este *dataset* tem 2181 colunas e a porção de treino disponibilizada pelos docentes tem 305 linhas. Destas 2181 colunas 2161 são numéricas (2014 números reais e 147 inteiros) e as restantes 20 são categóricas.

Do total de 2181 colunas, 159 possuem valores sempre repetidos (148 numéricas, 11 categóricas) e 113 são idênticas a pelo menos uma outra coluna (112 numéricas, 1 categórica), não trazendo, por isso, qualquer mais valia para o treinamento do modelo.

Das 8 colunas categóricas restantes, todas apresentam valores sempre únicos, o que as torna, à partida, inúteis no contexto de aprendizagem automática. No entanto, 2 destas são na realidade tuplos correspondentes a coordenadas e que poderiam ser desdobrados em várias *features* numéricas:

diagnostics_Mask-original_BoundingBox	**diagnostics_Mask-original_CenterOfMass**
(103, 113, 93, 36, 30, 71)	(121.94230227976358, 129.27272727272728, 128.40402476780187)

Tabela 1: Excerto de valores relativos às colunas mencionadas

A coluna **`diagnostics_Mask-original_BoundingBox`** representa as coordenadas da caixa que delimita o hipocampo, podendo estes valores ser utilizados no cálculo aproximado do volume do hipocampo que, como explorado na Secção 2, pode ser um dado promissor na previsão da progressão de MCI para AD e é, por isso, um dado mais útil ao modelo de ML. No entanto, a amplitude dos dados de *radiomics* já abrange esta nuance, existindo já colunas com informação sobre o volume da área analisada e com valores mais precisos que aqueles que seriam possíveis extrapolar com base nestas coordenadas. Assim, esta coluna representa informação redundante que já é representada de forma mais útil ao modelo por outras, não sendo por isso uma mais valia para o mesmo e não se justificando o seu desdobramento.

De forma semelhante, as coordenadas do centro de massa, representadas pela segunda coluna, constituem dados essenciais para o cálculo de métricas que permitem definir a localização espacial e a forma do hipocampo, entre outras características. Assim, esta coluna serve de matéria prima para a obtenção de dados mais úteis e possivelmente importantes na previsão, tendo em conta que descrevem diretamente características do hipocampo. No entanto, de forma isolada, não é necessariamente uma mais valia, acabando por ser também redundante, não se justificando o seu desdobramento.

O *dataset* não apresenta *missing values*, nem linhas repetidas.

4.1.2. Nomes das colunas

Numa tentativa de melhor compreender a semântica dos dados, olhamos atentamente para os nomes das colunas e começamos a identificar um padrão, a maior parte das colunas tem um nome no formato X_Y_Z (por exemplo: `wavelet-LLH_shape_Maximum3DDiameter`), o primeiro componente do nome refere-se a um filtro ou tratamento, o segundo, de acordo com a documentação da biblioteca *pyradiomics* [7], refere-se à classe da *feature* e o terceiro à *feature* específica.

4.1.3. Coluna *target*: "*Transition*"

O que pretendemos estimar com os nosso modelos é a transição de estado de um utente no declínio das suas faculdades cognitivas no curso de dois anos. Esta coluna está *encoded* em 5 categorias de transição:

- CN-CN
- CN-MCI
- MCI-MCI
- MCI-AD
- AD-AD

Onde:

- CN = *Cognitively Normal*
- MCI = *Mild Cognitive Impairment*
- AD = *Alzheimer's Desease*

Uma análise da sua distribuição segue-se no capítulo seguinte.

4.1.4. Distribuição das *features*

4.1.4.1. *Skewness* & *Kurtosis*

342 das 2013 colunas numéricas relevantes apresentam ***skewness*** negativa com 99% de certeza (`p_value` de $.01$), não existem colunas com *skewness* positiva. 152 colunas numéricas apresentam ***kurtosis*** negativa com 99% de certeza (`p_value` de $.01$), não existem colunas com *skewness* positiva.

Há 32 colunas que apresentam *skewness* e *kurtosis*, isto significa que a grande maioria das *features* são aproximadamente normalmente distribuídas.

4.1.4.2. Normalidade

Testamos a normalidade dos dados com o teste de normalidade de D'Agostino-Pearson, este disse-nos que 1477 destas *features* são normalmente distribuídas, um pouco menos do que o que estimamos recorrendo a *kurtosis* e *skewness* ($2013 - 342 - 152 + 32 = 1551$).

4.1.4.3. *Range*

Avaliamos a diferença entre o mínimo e máximo de cada coluna.

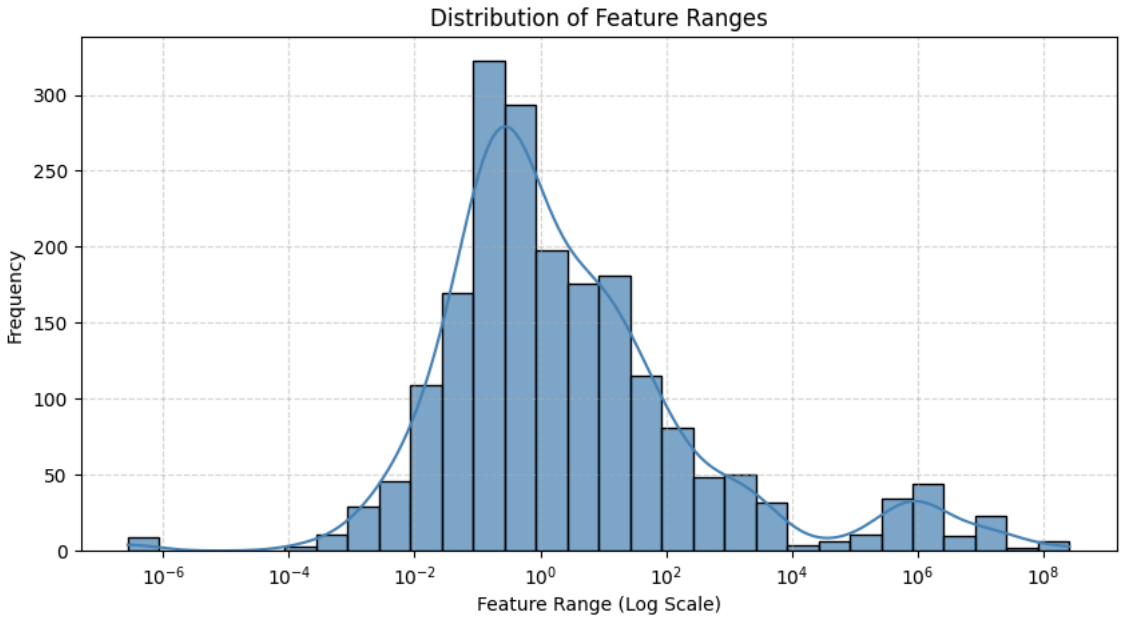

Figura 4: *Ranges* das colunas - Escala logarítmica

Percebemos que, efetivamente, há bastante variação no *range* das colunas, pelo que isso foi tido em consideração durante o tratamento.

4.1.4.4. Média

Avaliamos também a distribuição dos valores médios dos *features*, tendo verificado que esta é também bastante diversa.

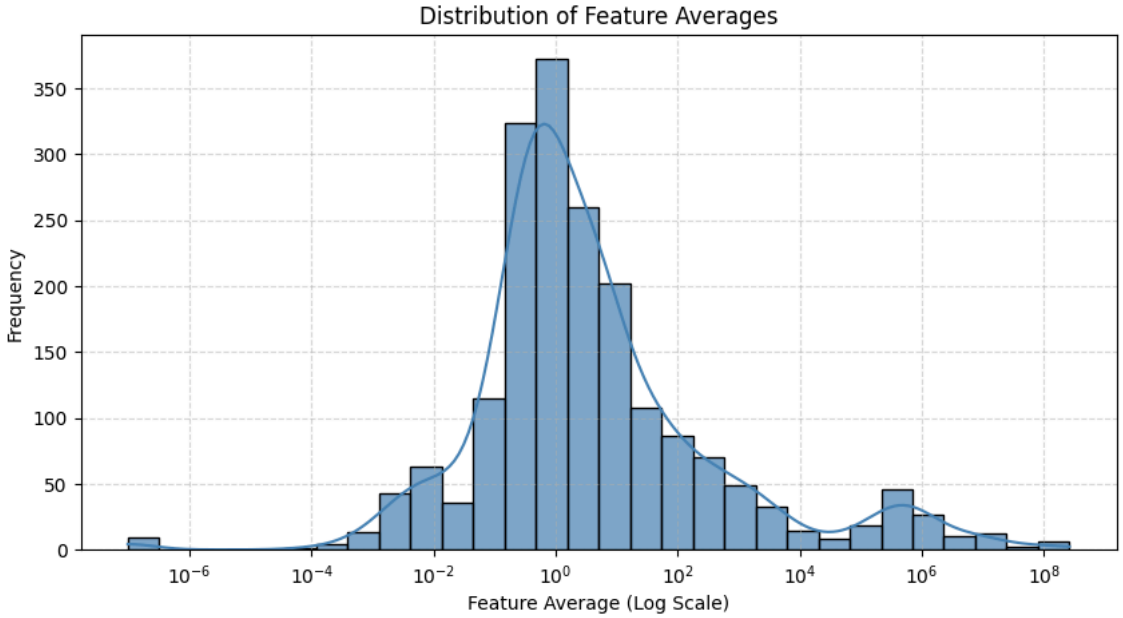

Figura 5: Médias das colunas - escala logarítmica

De forma análoga à análise anterior, esta diversidade de distribuição foi tida em consideração no tratamento.

4.1.4.5. Idade

Nesta secção apresentamos a análise da distribuição da idade, do sexo e da classe e de como estes se relacionam.

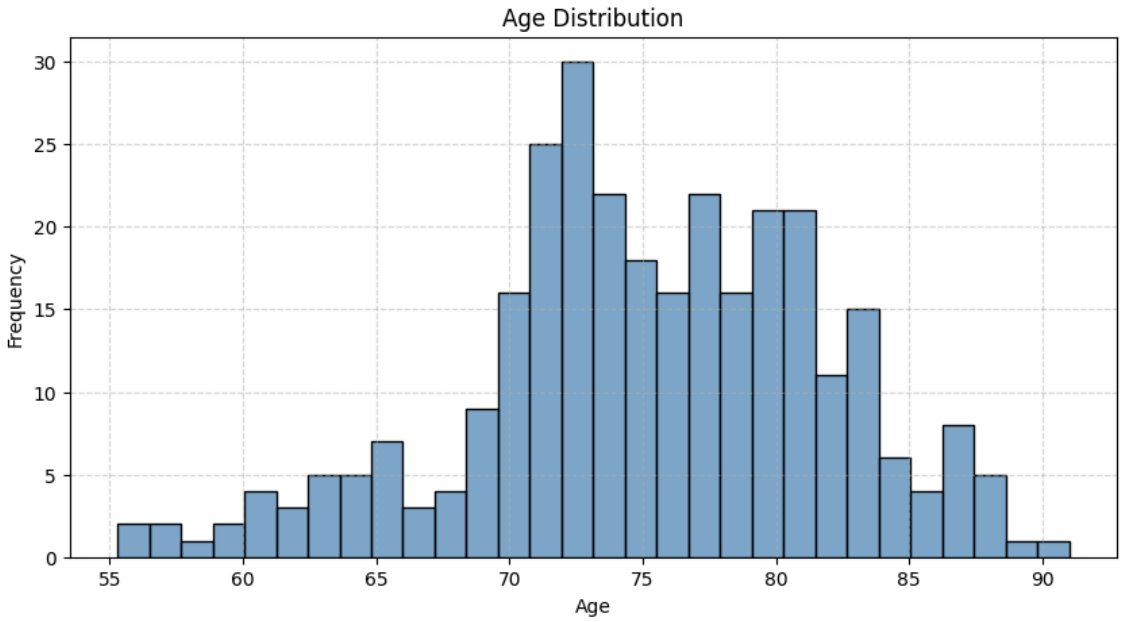

Figura 6: Distribuição de Idades

Daqui verifica-se que a maioria dos utentes têm 70 a 85 anos.

4.1.4.6. Sexo

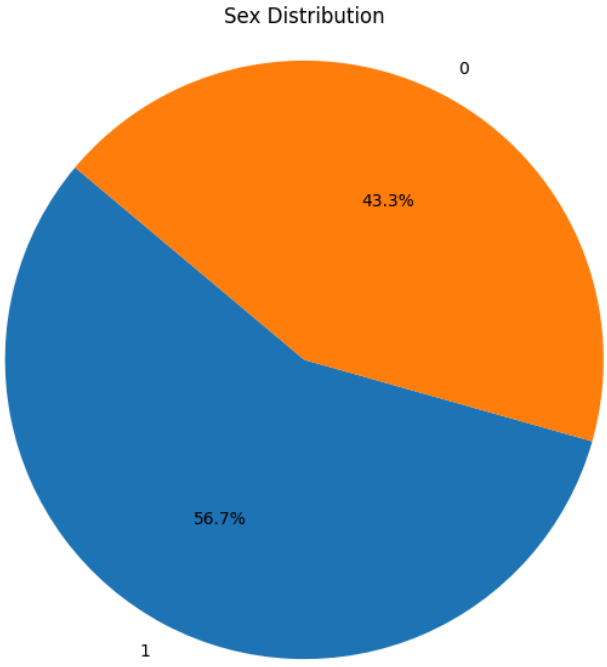

Figura 7: Distribuição de Sexo

O *dataset* está relativamente balanceado no que toca ao sexo dos utentes, no entanto constata-se uma predominância de homens (valor 1) em comparação com mulheres (valor 0).

4.1.4.7. Classe

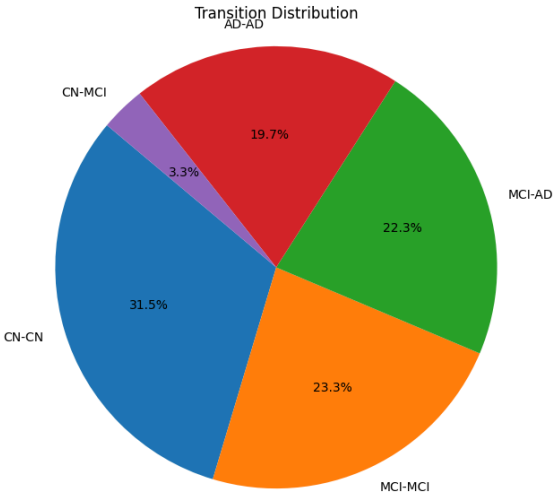

Figura 8: Distribuição de Classes

O *dataset* mostra-se relativamente balanceado quanto à classe, excetuando a transição de condição normal para declínio cognitivo leve, que é bastante infrequente em relação às outras, o que faz algum sentido semântico. Outro detalhe que fica claro é que a norma é que o utente não sofra de um significativo, além disso, pelo menos neste *dataset*, não há casos de recuperação, o que segundo [8] corresponde à realidade.

4.1.4.8. Relação da classe com o sexo

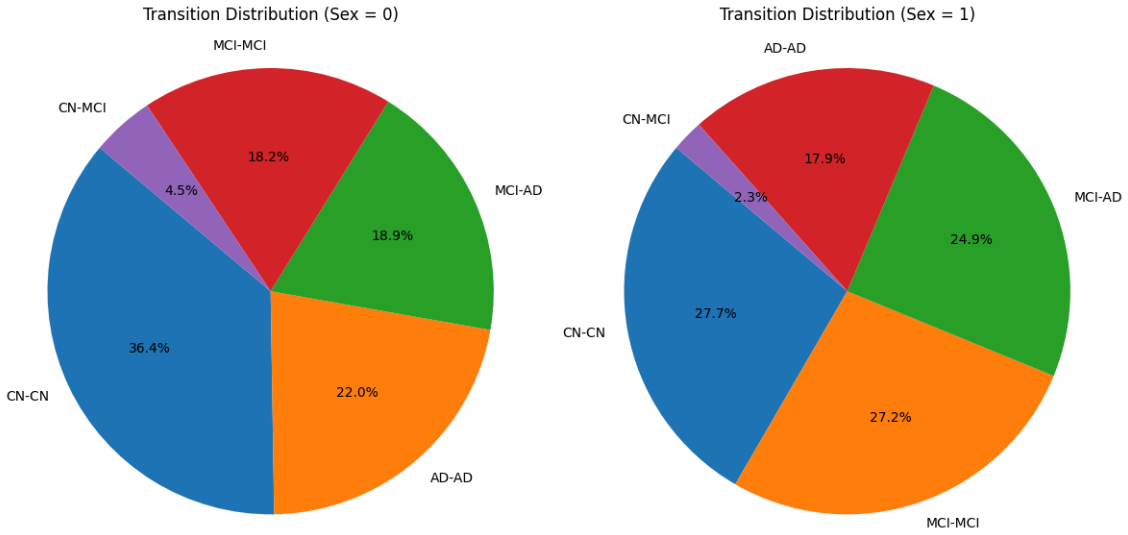

Figura 9: Distribuição de Classes por Sexo

Com base nestes gráficos parece que o sexo não tem uma grande influência na coluna *target*, no entanto realizamos um teste chi quadrado de Pearson com um *p value* de .01 para o efeito, este teste, no entanto, disse-nos que o sexo efetivamente exerce um efeito estatisticamente significativo sobre a classe.

4.1.4.9. Relação da classe com a idade

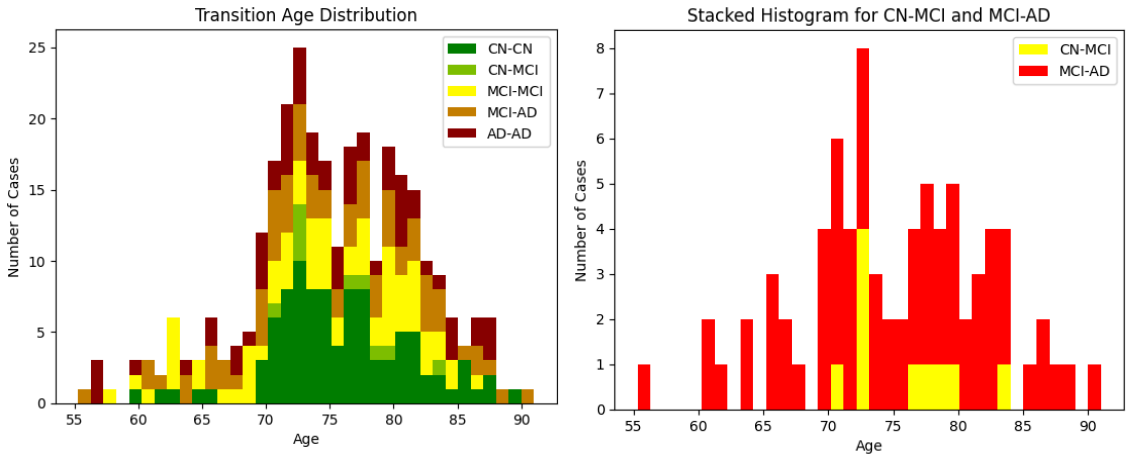

Figura 10: Distribuição de Classe por Idade e por Idade em que existe Transição

A partir destes gráficos ficamos a perceber que a idade do utente tem uma forte influência na transição.

4.2. *Dataset* DSocc

Depois de analisar o *dataset* DSocc, constatamos que ele é análogo aos dados examinados no DShippo.

5. Preparação de dados

5.1. Pré-processamento

Iniciamos o processo de preparação dos dados através de um conjunto de técnicas às quais designamos de pré-processamento dos dados, uma vez que são executadas numa primeira abordagem de preparação básica aos dados. Estas técnicas incluem:

- **Remoção das colunas que não apresentam informação**: eliminação das colunas com valores únicos ou constantes (numéricos e categóricos), uma vez que não contribuem para a previsão das diferentes classes;
- **Remoção das linhas duplicadas**: eliminação de linhas com todos os valores iguais, uma vez que não acrescentam informação ao conjunto de dados;
- **Remoção de colunas idênticas**: eliminação de colunas com valores idênticos a pelo menos uma outra coluna, uma vez que representam informação redundante.
- **Standardização de dados**: como analisado na Secção 4, o *dataset* apresenta uma distribuição bastante heterogénea, pelo que foi aplicada standardização de modo a balancear as distribuições das diferentes classes;
- ***Label Enconding***: aplicação de *Label Enconding* à variável objetivo, de forma a codificar cada classe num valor numérico, para garantir compatibilidade com todos os modelos.

Este pré-processamento varia levemente para o *dataset* de teste, uma vez que não podemos remover linhas e não existe uma coluna *target* à qual aplicar *Label Encoding*. A tabela seguinte ilustra esta diferença:

Dataset	Remoção das colunas que não apresentam informação	Remoção das linhas duplicadas	Remoção de colunas idênticas	Standardização de dados	*Label Encoding*
Treino	*	*	*	*	*
Teste	*		*	*	

Tabela 2: Variação no pré-processamento de acordo com o *dataset*

5.2. Redução da dimensionalidade

Como parte da preparação dos dados, decidimos aplicar técnicas de redução de dimensionalidade, isto é, reduzir o número de *features* do conjunto de dados. A eliminação de características tem como objetivo aumentar a precisão dos modelos de ML, reduzir a sua complexidade e tempo de treino, reduzir o ruído, torná-los mais interpretáveis, e diminuir o risco de *overfitting*.

Para além das técnicas de eliminação de atributos irrelevantes ou redundantes aplicadas no pré-processamento, implementamos *Recursive Feature Elimination* com *Cross-Validation* (RFECV) ou *Principal Component Analysis* (PCA). A aplicação, ou não, de uma destas técnicas pode ser especificada através de um ficheiro de configuração.

- ***Recursive Feature Elimination* com *Cross-Validation* (RFECV)**: combina duas técnicas de preparação de dados, resultando na seleção das características mais relevantes de um conjunto de dados, avaliando o seu impacto num modelo, até atingir um número ótimo de características. Para efetuar esta seleção, escolhemos utilizar o modelo *Random Forest*. A afinação da seleção é realizada através da técnica de validação *cross-validation*, que particiona o conjunto de dados em subconjuntos mutuamente exclusivos, alguns dos quais são usados para treino do modelo, enquanto os restantes são usados para a validação do modelo. Para evitar repetir o processo demorado de seleção dos atributos usando RFECV, as características selecionadas são armazenadas num ficheiro *cache*, carregado nas execuções subsequentes, se assim pretendido.
- ***Principal Component Analysis* (PCA)**: transforma os atributos originais em combinações lineares que capturam a maior variância possível nos dados. Esta técnica não é ideal para problemas não lineares, como é o caso deste problema, porém pode ser útil para simplificar o conjuntos de dados, que tem um elevado número de atributos, pelo que a implementamos como uma opção adicional e menos *resource intensive*.

5.3. Balanceamento de dados

Finalmente, para garantir que os modelos não sejam enviesados para as classes maioritárias, decidimos implementar *Synthetic Minority Over-sampling Technique* (SMOTE) como uma opção na preparação dos dados, o que ajuda a tornar os modelos mais precisos e generalizáveis.

O desiquilíbrio verifica-se quando uma classe contém uma quantidade de amostras muito superior à quantidade das restantes classes. Esta técnica consiste na criação de novas amostras de dados sintéticas para as classes minoritárias, de forma a equilibrar a sua representação nos modelos de ML, evitar viés em favor da classe maioritária e melhorar a sua capacidade de generalização para as classes com menos amostras.

A aplicação de SMOTE é importante após a utilização de *Recursive Feature Elimination* para garantir que as características existem todas na mesma proporção, tornando o conjunto de dados mais robusto, com uma escala uniforme e classes equilibradas.

A aplicação de SMOTE antes ou depois da utilização de PCA é alvo de debate na comunidade. No entanto, decidimos guiar a nossa decisão com base nos resultados obtidos no artigo de investigação "*Combination of PCA with SMOTE Oversampling for Classification of High-Dimensional Imbalanced Data*" [9], onde para um *dataset* com, de igual modo, grande dimensionalidade, a aplicação de SMOTE **antes** de PCA se provou benéfica.

6. Modelos de Aprendizagem Automática

Como foco central do projeto, testamos vários algoritmos de aprendizagem automática no contexto deste *dataset*. O problema a abordar é de classificação, pelo que nos focamos em modelos capazes de trabalhar sobre problemas deste tipo.

6.1. Afinamento de Hiperparâmetros

Para todos os modelos, a escolha dos hiperparâmetros pós-otimizações foi realizada utilizando *Grid Search*, de modo a garantir que a melhor combinação de valores era encontrada. É também aplicada *Cross Validation* para medir o desempenho do modelo, utilizando *F1-Score* devido a ser um indicador mais fidedigno do que precisão.

De seguida exploram-se os modelos implementados, assim como as mais valias que justificaram a sua inclusão no projeto. Na Secção 7 exploram-se os hiperparâmetros utilizados em cada um destes e os resultados obtidos.

6.2. *Decision Tree*

A *Decision Tree* constrói uma estrutura hierárquica de decisões baseada em divisões sucessivas dos dados. É fácil de interpretar e eficiente para conjuntos de dados de tamanhos variados. Utilizámos este modelo como ponto de partida para entender melhor os padrões nos dados.

6.3. *Support Vector Machine* (SVM)

As *Support Vector Machines* são modelos de aprendizagem supervisionada utilizados em problemas de regressão e classificação e que são especialmente eficazes em espaços com muitas dimensões, como é o caso do problema em questão. Por esta compatibilidade teórica com a natureza do nosso problema, implementámos SVMs na esperança de ser um dos modelos mais promissores.

6.4. Modelos de *Ensemble Learning*

6.4.1. *Bagging*

O *Bagging* treina múltiplos modelos em diferentes *subsets* dos dados, agregando as suas previsões. Este modelo é promissor na sua capacidade de melhorar a estabilidade e precisão do *base learner* que melhor se adeque ao nosso problema, como um SVM, por exemplo.

Testamos este com SVMs como *base learners* e usamos regressão logística como *meta-learner*.

6.4.2. *Random Forest*

Random Forest é um modelo supervisionado bastante poderoso que, geralmente, consegue alcançar precisões altas quando comparado com outros algoritmos. Para além disso é eficaz a lidar com *datasets* com um elevado número de colunas, resistente a *overfitting* e útil no processo de *feature selection* devido à sua medida de *feature importance*. Por todos estes motivos, não podia faltar no nosso catálogo de modelos.

6.4.3. *Gradient Boosting*

O *Gradient Boosting*, ao contrário do *Bagging*, treina diversos modelos sequencialmente e não independentemente, com a vantagem de que cada modelo tenta corrigir os erros do anterior. Este tipo de modelos costuma também trazer boas precisões, pelo que foi fundamental experimentar.

6.4.4. *XGBoost*

O *XGBoost* é uma versão otimizada e mais poderosa do *Gradient Boosting* tradicional, e foi importante utilizar de modo a poder comparar as diferenças entre os dois e tentar obter melhores resultados com este paradigma.

6.4.5. *Stacking*

O *Stacking* permite combinar múltiplos e diferentes modelos já treinados que se especializem em diferentes nuances, de modo a criar um modelo mais poderoso que tem em conta os pontos fortes dos modelos que utiliza. Este modelo é útil para a combinação dos nossos melhores e mais variados modelos de modo a tentar obter um modelo mais robusto.

6.4.6. *Custom Bagging*

Testamos também um modelo concebido por nós em que separamos as colunas por filtro (Secção 4.1.2) antes de qualquer tratamento de dados que envolva eliminação de colunas. Passa-se cada uma destas partições do *dataset* a um modelo e no fim agregamos estes resultados com um *meta-learner*.

À semelhança dos modelos de *bagging* (Secção 6.4.1) usamos SVMs como *base learners*, testamos algumas opções de *meta-learner* incluindo *random forest*, árvores de decisão , regressão logística e também o testamos com *soft voting*.

O treino e avaliação deste modelo e os seus sub-modelos trouxe-nos como subproduto um *ranking* da importância de cada filtro (Secção 9.1).

6.5. Redes Neuronais: *Multi-Layer Perceptron* (MLP)

O MLP é uma rede neural artificial capaz de aprender padrões complexos por meio de múltiplas camadas de neurónios. Escolhemos este modelo pela sua flexibilidade e potencial para capturar relações complexas nos dados.

Estes modelos são corridos sem qualquer passo de pré-processamento que implique apagar ou combinar colunas (PCA ou RFE) uma vez que o seu intuito, e principal mais valia, é encontrar padrões complexos entre os dados, no entanto, tratamentos como PCA encontram relações básicas entre as *features* e combina-as com base nisso, tarefa que o MLP já faz.

Quanto à topologia, a camada de entrada do *MLP* tem tantos nodos como *features* do *dataset*, a camada de saída tem tantos nodos como classes do *target*, testamos algumas configurações diferentes das camadas escondidas.

6.5.1. Topologia em funil

Testamos uma topologia interna em que o número de nodos nas camadas internas vai decrescendo exponencialmente (com um fator de $.5$), ou seja a primeira camada com 1024, a segunda com 512 e repetindo este padrão até 64.

Escolhemos esta topologia porque nos pareceu fazer sentido intuitivo, aparecendo em alguns exemplos de código e documentação durante a nossa pesquisa sobre o tema.

6.5.2. Outras topologias

- **Topologia em funil estendido** : `n_colunas` -> 1024 -> 512 -> 512 -> 512 -> 256 -> 128 -> `n_classes`
- **Topologia em ampulheta** : `n_colunas` -> 1024 -> 512 -> 256 -> 512 -> 1024 -> `n_classes`

6.5.3. *Overfitting*

Todas as nossas tentativas com MLPs sofrem fortemente de *overfitting*, como é claro ao observar a Secção 9.2, vendo que a curva de *accuracy* dos dados de treino se aproxima bastante de 1, ao passo que a *accuracy* dados de teste não passa de $.4$.

7. Resultados

De seguida apresenta-se uma tabela com um resumo dos resultados obtidos **localmente** para cada modelo, após todas as otimizações mencionadas anteriormente desde o processo de preparação de dados até ao processo de modelagem, ordenados por ordem decrescente de *F1-Score*. Para cada modelo são enumerados também os valores utilizados nos seus hiperparâmetros, assumindo-se o valor padrão da biblioteca *Python* para aqueles que não são mencionados.

Modelo	*F1-Score* Local	Hiperparâmetros
Random Forest	0.41	*criterion: gini; max_depth: 8; max_features: sqrt; min_samples_leaf: 1; min_samples_split: 6; n_estimators: 150*
Gradient Boosting	0.40	*criterion: squared_error; max_depth: 4; max_features: log2; n_estimators: 200*
Stacking	0.40	*estimators:* ***Random Forest, Gradient Boosting, Bagging (SVM base learner), SVM****; stack_method: auto*
Bagging	0.37	base_learner: **SVM**; *bootstrap: False; bootstrap_features: False; max_features: 0.1; max_samples: 1.0; n_estimators: 150*
XGBoost	0.37	*booster: gbtree; learning_rate: 0.3; n_estimators: 50*
SVM	0.36	*C: 5; gamma: auto; kernel: rbf*
Decision Tree	0.34	*criterion: entropy; max_depth: 8; max_features: sqrt; max_leaf_nodes: 1000; min_samples_leaf: 1; min_samples_split: 4; splitter: best*
MLP Funnel	0.42	*epochs: 50*

Tabela 3: Resultados obtidos localmente

Estes resultados têm por base um subconjunto de teste de 20% retirado do *dataset* de treino fornecido pela equipa docente, logo não são 100% fidedignos e podem ser resultado de um modelo *overfitted*. Assim, comparamos estes resultados com os resultados **privados** obtidos na competição *Kaggle*, que são calculados utilizando um *dataset* de maior dimensão e que permitem, por isso, resultados mais fidedignos da *performance* dos modelos. Comparamos também com os valores obtidos localmente para o *dataset* de controlo. A tabela seguinte ilustra essa comparação.

Modelo	**Competição** *Private Score*	**Local** *F1-Score*	***Dataset* Controlo** *F1-Score* Local
Random Forest	0.40	0.41	0.35
Stacking	0.37	0.40	0.29
SVM	0.37	0.36	0.24
XGBoost	0.37	0.37	0.18
Bagging	0.35	0.37	0.26
Decision Tree	0.34	0.34	0.26
Gradient Boosting	0.33	0.40	0.24
MLP Funnel	0.25	0.42	-

Tabela 4: Comparação entre resultados obtidos no *dataset* privado de competição e localmente

7.1. Análise dos resultados

A comparação de resultados locais com os resultados do *dataset* privado da competição (Tabela 4) permite-nos concluir que, em geral, os modelos por nós desenvolvidos apresentam **baixo *overfit*** aos dados de treino, onde, sendo a um *f1-score* da competição e b um *f1-score* local:

$$\frac{\sum |a-b|}{7} = 0.02$$

Obtemos uma diferença média 0.02, com um erro médio de:

$$\frac{\sum \frac{|a-b|}{a}}{7} \approx 0.0575 \approx 5.75\%$$

Assim, podemos com alguma confiança concluir que os modelos desenvolvidos são, em média, **bem generalizados** ao problema, sendo o modelo de *Gradient Boosting* aquele que apresentou maior *overfitting* e o modelo de *Random Forest* aquele que melhor conseguiu prever a transição do estado cognitivo dos pacientes.

A melhor *performance* do *Random Forest* pode ser atribuída à sua natureza de *ensemble* que utiliza múltiplas árvores de decisão, cada uma treinada com um subconjunto aleatório dos dados e das *features*. Esta abordagem reduz a variância e o *overfitting*, tornando o modelo mais robusto e generalizável, especialmente em *datasets* com um número elevado de colunas, como é o caso. A seleção aleatória de *features* em cada divisão da árvore também contribui para a diversidade do *ensemble*, diminuindo a correlação entre as árvores e melhorando o desempenho da previsão. Adicionalmente, o *Random Forest* é menos sensível a *outliers* e ruído nos dados, o que pode ser vantajoso em *datasets* complexos como o utilizado.

Em contraste, o *Gradient Boosting*, que também é um método de *ensemble*, constrói as árvores sequencialmente, onde cada nova árvore corrige os erros das árvores anteriores. Embora esta abordagem possa levar a uma alta precisão nos dados de treino, também a torna mais suscetível a *overfitting*, especialmente quando o modelo é complexo (profundidade das árvores e número de estimadores elevados) ou quando o *dataset* possui ruído ou *outliers*. No presente caso, apesar da otimização dos hiperparâmetros, o *Gradient Boosting* demonstrou maior *overfitting*, possivelmente devido à alta dimensionalidade do *dataset*, que pode ter permitido ao modelo memorizar detalhes específicos do treino, prejudicando a sua capacidade de generalização para dados novos. A interação complexa entre as *features*, combinada com a natureza sequencial do *Gradient Boosting*, pode ter exacerbado este efeito, resultando na discrepância observada entre os resultados locais e privados.

Apesar da boa generalização dos modelos, a sua capacidade de previsão foi baixa, o que era de esperar dada a complexidade do problema e a dimensão dos dados. Mesmo assim, foi consideravelmente acima da aleatoriedade ($\frac{1}{5} = 20\%$) e consideravelmente acima dos valores obtidos para o *dataset* de controlo ($\approx$

0.26), o que **ajuda a comprovar que, de facto, a natureza do hipocampo cerebral está relacionada com a progressão do estado cognitivo do utente, incluindo com o seu declínio.**

7.1.1. Redes Neuronais

Tendo em conta a grande complexidade do *dataset* e à pequena dimensão do *dataset* de treino, era de esperar que os modelos de redes neuronais artificiais ficassem especialmente suscetíveis a *overfitting*. A obtenção de bons resultados com este tipo de modelos requereria um maior investimento de tempo da nossa parte, tempo esse que, infelizmente, não existiu. Deste modo, os resultados obtidos com o modelo de *MLP Funneling* implementado sofreram, como era de prever, de uma quantidade bastante significativa de *overfitting*, como se observa pela grande diferença entre o resultado local e o resultado da competição da Tabela 4.

7.1.2. Importância dos Filtros

O trabalho resultante do processo de *Custom Bagging* realizado anteriormente e explorado na Secção 6.4.6 permitiu concluir também, através dos resultados disponíveis na Secção 9.1, que existem filtros específicos, resultantes dos dados *radiomics* provenientes da ressonância magnética, que fornecem dados mais úteis à previsão quando comparados com os restantes. Esta conclusão provém da maior *performance* exibida pelo modelo treinado nesses filtros, nomeadamente no filtro **`wavelet-HLH`**.

Esta conclusão pode, no futuro, ser considerada na construção de modelos mais poderosos, que foquem as sua atenção nos atributos ou conjuntos de atributos que comprovadamente mais influenciam os dados a prever.

8. Conclusão

Este problema revelou-se, como esperado, bastante difícil de resolver com recurso a técnicas de aprendizagem automática, no entanto conseguimos alguns modelos significativamente melhores do que *random chance*.

Sentimos que aprendemos bastante sobre o processo de "data-mining", nomeadamente por ter de lidar com um *dataset* tão largo como este o que nos forçou a explorar os dados de forma mais holística e automática.

8.1. Trabalho futuro

Tendo acesso ao dados de imageologia podíamos testar redes neuronais de convolução [10] e explorar redes neuronais artificial no geral em mais profundidade.

9. Anexos

9.1. Custom Bagging Ranking

Valores do `F1 Score` para as diferentes colunas utilizados no nosso *Custom Bagging*.

```
[{'filter': 'lbp-3D-m2', 'f1': 0.3373429848839685},
 {'filter': 'gradient', 'f1': 0.31394006186677353},
 {'filter': 'original', 'f1': 0.3063286620835537},
 {'filter': 'lbp-2D', 'f1': 0.20964179263359595},
 {'filter': 'wavelet-LHH', 'f1': 0.29004790291675536},
 {'filter': 'log-sigma-2-0-mm-3D', 'f1': 0.35304449648711944},
 {'filter': 'log-sigma-5-0-mm-3D', 'f1': 0.3527229108356657},
 {'filter': 'square', 'f1': 0.3725905242298685},
 {'filter': 'wavelet-LLL', 'f1': 0.4116640463041033},
 {'filter': 'log-sigma-3-0-mm-3D', 'f1': 0.3778738865782758},
 {'filter': 'wavelet-HLH', 'f1': 0.41942080310732854},
 {'filter': 'lbp-3D-m1', 'f1': 0.39526322149272974},
 {'filter': 'logarithm', 'f1': 0.30443587270973965},
 {'filter': 'squareroot', 'f1': 0.3308593046297964},
 {'filter': 'lbp-3D-k', 'f1': 0.3747072599531616},
 {'filter': 'wavelet-LHL', 'f1': 0.25898393958066646},
 {'filter': 'wavelet-HLL', 'f1': 0.3654755834905247},
 {'filter': 'wavelet-HHL', 'f1': 0.3499405728913926},
 {'filter': 'log-sigma-1-0-mm-3D', 'f1': 0.395927147449977},
 {'filter': 'exponential', 'f1': 0.2926278229775344},
 {'filter': 'wavelet-HHH', 'f1': 0.3094925039388633},
 {'filter': 'wavelet-LLH', 'f1': 0.37434286049322796},
 {'filter': 'log-sigma-4-0-mm-3D', 'f1': 0.36156915212879037},
 {'filter': 'diagnostics', 'f1': 0.12874681505696842}]
```

9.2. *Learning curves*

Topologia em ampulheta

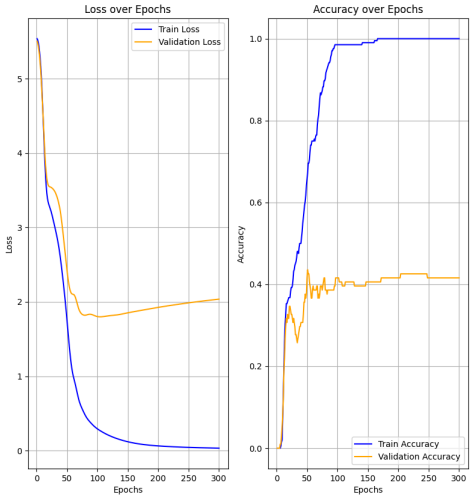

Figura 11: topologia em ampulheta
`n cols->1024->512->256->512->1024->n classes`

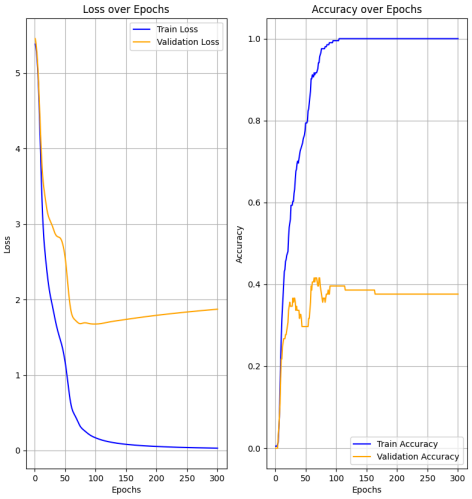

Figura 12: topologia em funil
`n_cols->1024->512->256->128->n_classes`

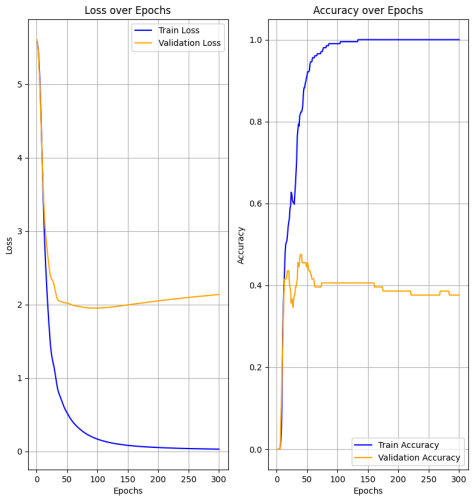

Figura 13: topologia em funil
`n_cols->512->256->128->64->n_classes`

Referências

[1] A. Portugal, «Demência: Uma Prioridade de Saúde Pública». [Online]. Disponível em: https://alzheimerportugal.org/demencia-uma-prioridade-de-saude-publica/

[2] S. G. D. Costa, «Demência: a importância do diagnóstico atempado». [Online]. Disponível em: https://www.trofasaude.pt/artigos/demencia-a-importancia-do-diagnostico-atempado/

[3] A. Portugal, «A doença de Alzheimer». [Online]. Disponível em: https://alzheimerportugal.org/a-doenca-de-alzheimer/

[4] C. E. Lombardi G Crescioli G, «How accurate is magnetic resonance imaging for the early diagnosis of dementia due to Alzheimer's disease in people with mild cognitive impairment?», 2020.

[5] D. S. PM, «What is CRISP DM?». [Online]. Disponível em: https://www.datascience-pm.com/crisp-dm-2/

[6] B. V. M. Y. Lakshmisha Rao B. Ganaraja, «Hippocampus and its involvement in Alzheimer's disease: a review». [Online]. Disponível em: https://pmc.ncbi.nlm.nih.gov/articles/PMC8807768/

[7] N. A. J.-C. F.-R. A. H. S. P. H. A. Joost van Griethuysen Andriy Fedorov, «Radiomic Features - pyradiomics». [Online]. Disponível em: https://pyradiomics.readthedocs.io/en/latest/features.html

[8] C. Clinic, «Mild Cognitive Impairment». [Online]. Disponível em: https://my.clevelandclinic.org/health/diseases/17990-mild-cognitive-impairment

[9] M. M. H. Guhdar A. A. MULLA Yıldırım DEMİR, «Combination of PCA with SMOTE Oversampling for Classification of High-Dimensional Imbalanced Data». [Online]. Disponível em: https://dergipark.org.tr/en/download/article-file/1777398

[10] [Online]. Disponível em: https://pmc.ncbi.nlm.nih.gov/articles/PMC8449698/#Sec8