Projeto EC Grupo 4
João Pereira 58189 Horas: 50 André Reis 5819 Horas: 40 Daniel Nunes 58275 Horas: 40

Explicação de algumas métricas:

- F1_macro: Calcula o f1 para cada classe e depois a média desses valores. O F1 considera a precisão e o recall, logo é um bom medidor de performance de um modelo de classificação. Possui um range de 0 a 1, quanto mais perto de 1 melhor.
- Recall: É a proporção de verdadeiros positivos em relação ao total de instâncias.
- Precisão: Mede a proporção de verdadeiros positivos em relação ao total de previsões positivas. Quanto maior melhor, a menos que seja no test uma vez que pode indicar overfit.
- R^2: Indica o quão bem os dados se ajustam a um modelo estatístico. Quanto mais próximo de 1 melhor.
- MSE: Para cada ponto de dados, calcula a diferença entre o previsto pelo modelo e o real. Quanto maior, pior o modelo.

Dataset

Foi-nos dado um dataset sobre a medição de várias características que possivelmente poderiam intervir na tiroide, assim como as hormonas que dela surgem. No dataset também existe certas condições que podem afetar a tiroide como a gravidez, a idade ou a existência de algum tumor ou doença. A coluna "Target" vai ser um derivado da coluna "Diagnoses", os valores dentro dos parênteses "()" são os que se encontram na "Diagnoses" e o seu respetivo significado: *hyperthyroid conditions (A, B, C, D), hypothyroid conditions (E, F, G, H), binding protein (I, J), general health (K), replacement therapy (L, M, N), discordant results (R), Healthy (-), Other (*|*)*

Processamento de dados

Aqui, o que vamos fazer é por os dados num dataframe e processá-los. Vamos tratar dos valores desconhecidos e remover algumas colunas (variáveis independentes) que não sejam necessárias para o problema (Neste caso serão apenas 2 colunas). Fizemos o seguinte tratamento para todos os objetivos:
Definição do target: Criamos uma coluna target, em que ao analisar o modelo, consegue dizer se possui uma das oito possíveis classes. Para tal fizemos uma função que analisa a coluna "Diagnosis" e seguidamente cria uma coluna nova com as 8 possíveis classes. No final removemos a coluna "Diagnosis", uma vez que esta passa, agora, a ser a coluna que acabamos de criar, à qual chamámos de "target", já que é esta que o nosso modelo vai prever.
Tratamento dos missing values e valores estranhos: Com a análise dos dados chegamos a conclusão de que existem valores estranhos, por exemplo idades impossíveis e pessoas gravidas sem sexo, resultando na alteração dos resultados. Nestes casos, as idades impossíveis são removidas e às pessoas grávidas sem sexo é lhes atribuída o género F indicando serem do género feminino. No caso de valores em falta, decidimos colocar alguns destes valores a "-1" para não inventarmos valores pois estamos em contexto médico.
Undersampling : Decidimos não usar undersampling porque decidimos que iria ser mais um risco de overfit aos dados e iriamos perder informação que poderia ser importante. Também poderia acontecer que a classe majoritária faça com que os dados fiquem distorcidos o que poderia levar a falsas avaliações.

O1

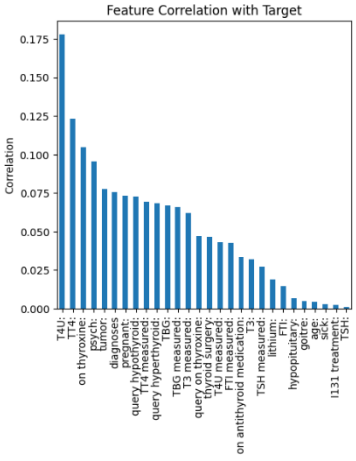
O Objetivo 1 (O1) é treinar um modelo para a avaliação clínica de um paciente com suspeita de doenças ou anomalias da tiroide. Logo estamos presentes num problema de classificação, uma vez que estamos a categorizar pessoas dentro das 8 classes possíveis relativas a saúde da tiroide.

Conclusão do processamento de dados

Após filtrarmos os dados testámos vários modelos de classificação medindo os valores "f1 avg" e "accuracy". Os modelos que obtiveram melhores resultados foram: Random Forest; Decision Tree; XGBoost;

Seleção das melhores features

Para selecionarmos as melhores features decidimos escolher aquelas que eram coerentes entre correlação, SFS e feature importances dos próprios modelos. Para isso usámos os gráficos de correlação das features e os gráficos de importância das features de cada modelo, assim como o output da classe SFS em modo forward.



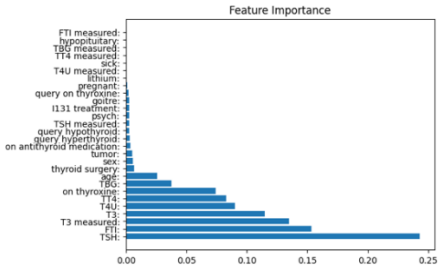
Correlação

Decidimos verificar a correlação de spearman uma vez que os dados podem não se encontrar igualmente distribuidos. Como este tipo de correlação não tem em conta a distribuição dos dados, decidimos que poderia ser mais vantajoso ter em conta esta correlação em vez da de Pearson, já que a de Pearson supõe que os dados estão bem distribuidos e é sensivel a outliers.

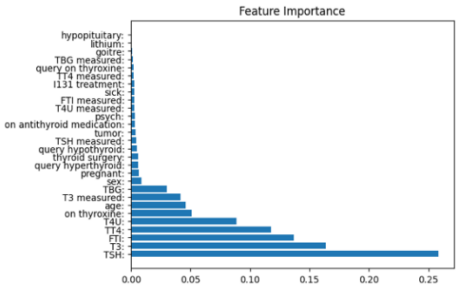
Feature Importance

Aqui podemos verificar os gráficos da importância onde cada modelo escolheu as features com mais importância para si, usando “feature importance”:

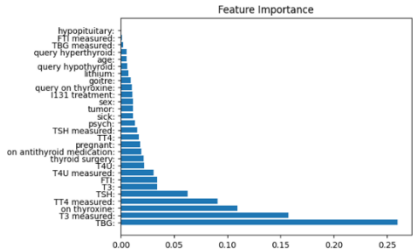
XGBoost



Random Forest



XGB



Aqui podemos verificar que as features mais importantes são: [TSH, FTI, T3 Measured, T3, T4U, TT4, on thyroxine]. Estas são, na generalidade, as mais importantes dentre os 3 modelos.

SFS

Podemos observar as melhores features na célula O1 do notebook jupyter que são:

- XGB: [‘age:’, ‘sex:’, ‘on thyroxine:’, ‘thyroid surgery:’, ‘TSH:’, ‘T3:’, ‘TT4:’, ‘T4U:’, ‘FTI:’, ‘TBG:’]
- Random Forest: [‘on thyroxine:’, ‘thyroid surgery:’, ‘TSH:’, ‘T3 measured:’, ‘T3:’, ‘TT4:’, ‘T4U measured:’, ‘T4U:’, ‘FTI:’, ‘TBG:’]
- Decision Tree: [‘age:’, ‘on thyroxine:’, ‘pregnant:’, ‘thyroid surgery:’, ‘tumor:’, ‘TSH:’, ‘T3:’, ‘TT4:’, ‘FTI:’, ‘TBG:’]

Estas estão ordenadas por ordem decrescente de importância.

Conclusão

Ao analisar as medidas, decidimos escrever hardcoded as melhores features uma vez que apenas uma medição poderia não ser a melhor ou mais importante para o problema. As features selecionadas foram:
Decision Tree: ['sex:', 'on thyroxine:','thyroid surgery:','tumor:','TSH:','T3:','TT4:','T4U:','FTI:','TBG:']
Random Forest: ['on thyroxine:', 'pregnant:', 'thyroid surgery:','TSH:','T3 measured:','T3:','TT4:','T4U:','FTI:','TBG:']
XGB: ['age:' ,'sex:', 'on thyroxine:','query on thyroxine:', 'on antithyroid medication:', 'thyroid surgery:','I131 treatment:','tumor:','psych:','TSH:','T3:','TT4:', 'T4U:','FTI:', 'TBG:']

Resultado do Modelo

Modelo	F1-macro	Train accuracy	Test Accuracy
DT	0.820	1.000	0.932
RF	0.856	1.000	0.945
XGB	0.858	1.000	0.946

Model Tuning

Verificamos que os modelos sofreram overfit uma vez que o train accuracy está a 1.0 o que indica que o modelo está a adaptar-se demasiado aos dados de treino. Para tentar remover ou suavizar o overfit vamos dar tuning dos modelos. Usámos MinMax Scaler e StandardScaler para dar tuning. No final escolhemos os 2 melhores.

XGB Tuning

Para tunar o XGB vamos mudar alguns hyperparâmetros

- ***MaxDepth:*** A max_depth define a profundidade máxima de uma árvore, no entanto, valores altos podem levar a overfit. Por isso, estamos a tentar valores menores que o default (6).
- ***N_estimators:*** O número de estimadores é o número de árvores que vão ser construídas, estamos só a testar 100 e 200 uma vez que valores muito altos podem levar ao overfit, para além de demorar muito tempo a ser avaliado.
- ***Learning_rate:*** É um campo usado para limitar a aprendizagem de cada iteração, um bom valor torna o modelo mais robusto.
- ***Subsample:*** Este campo define as amostras para cada árvore. Valores menores previnem o overfit, logo usamos valores menores que 1, que é o default, pois este tem em consideração todas as amostras para árvore.
- ***Colsample_bytree:*** Este campo define as features que vão ser consideradas em cada árvore, um valor alto pode levar a overfit, por isso estamos a usar valores menores do que o default (1).

Random Forest Tuning

Para tunar o Random Forest vamos mudar alguns hyperparâmetros

- ***Max_depth:*** A max_depth define a profundidade máxima de uma árvore, no entanto, valores altos podem levar a overfit. Por isso, estamos a tentar valores menores que o default (6).
- ***N_estimators:*** O número de estimadores é o número de árvores que vão ser construídas, estamos só a testar 100, 150 e 200 uma vez que valores muito altos podem levar ao overfit, para além de demorar muito tempo a ser avaliado.

- ***Min_Samples_Split:*** Representa o número mínimo de amostras/observações para que um nó seja considerado para divisão. Isto ajuda a controlar o overfit porque o modelo vai ser mais restrito e vai aprender relações mais especificas do conjunto de treino.
- ***min_samples_leaf:*** Determina o número mínimo de amostras/observações para estar numa folha, ajuda a controlar o overfit pelo mesmo motivo do parâmetro anterior.

É notável que os parâmetros são praticamente iguais aos de uma Decision Tree uma vez que o Random Forest usa um conjunto de Decision Tree para fazer previsões.

Decision Tree Tuning

Para tunar a DT mudamos vamos mudar alguns hyperparâmetros

- ***MaxDepth:*** Representa a profundidade máxima da árvore de procura, um valor muito grande pode tornar o modelo muito complexo o que pode levar a overfit enquanto um muito pequeno pode levar a underfit, logo é necessário encontrar um equilíbrio uma vez que, neste caso, estamos a sofrer de overfit.
- ***Min_Samples_Split:*** Representa o número mínimo de amostras/observações para que um nó seja considerado para divisão. Isto ajuda a controlar o overfit porque o modelo vai ser mais restrito e vai aprender relações mais especificas do conjunto de treino.
- ***Min_Samples_leaf:*** Determina o número mínimo de amostras/observações para estar numa folha, ajuda a controlar o overfit pelo mesmo motivo do parâmetro anterior.

Resultado do Tuning

Modelo	F1-macro	Train accuracy	Test Accuracy
DT Standard	0.805	0.971	0.915
RF Standard	0.836	0.994	0.927

Reparamos que depois do Tuning estes modelos diminuíram o overfit e ficaram com resultados até bastante favoráveis face ao objetivo.

Conclusão do Objetivo 1

Depois de todos os passos acima podemos concluir com bastante certeza se uma pessoa tem alguma doença da tiroide com base no dataset dado. Escolhemos os 2 melhores modelos, desses o melhor foi o Random Forest já resultou num F1 macro de aproximadamente 0,84. Podemos dizer então que o modelo realizado consegue prever o problema O1 até que bem.

Os melhores modelos foram então:

- Decision tree Standard: 'MaxDepth': 30, 'min_samples_leaf': 2, 'min_samples_split': 10
- Random Forest Standard: 'max_depth': 20, 'min_samples_split': 5, 'n_estimators': 250, 'random_state': 4

O2: Sex

O objetivo do O2: Sex era averiguar se era possivel fazer um modelo para prever o sexo de um individuo a partir das features presentes no dataset. Como estamos a prever o sexo de pessoas, estamos a tentar prever valores categóricos, logo é um problema de classificação.

Conclusão do processamento de dados

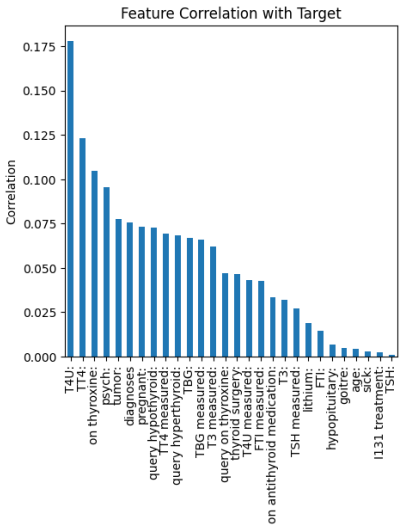
Usámos o mesmo método de avaliação do O1 para escolher os melhores modelos para trabalhar. É notável que neste havia empates entre o KNNClassifier e a Decision Tree. Decidimos usar o Decision Tree uma vez que este é mais simples de interpretar e visualizar. Os melhores foram: Decision Tree, MLP, XGBoost

Seleção das melhores features

Para selecionar as melhores features no O2 para prever o sexo usámos o mesmo método que no O1 uma vez que estes são ambos problemas de classificação.

Correlação

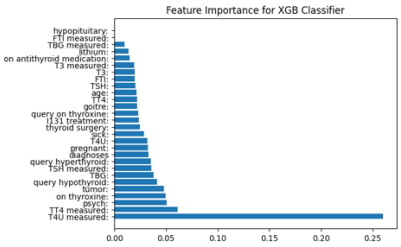
É notável que os valores da correlação são muito baixos, o que indica que cada feature não tem muita relação com o target. A correlação está dentro do intervalo [-1,1], sendo 0 as features não estarem linearmente relacionadas.

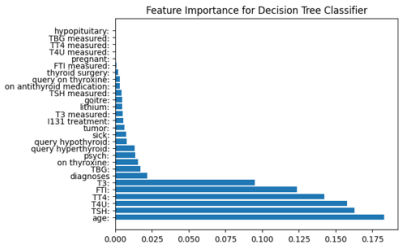


Feature Importance O2

Pela Feature Importance dos modelos temos que:

- XGB Decision Tree





As redes Neuronais não suportam a feature importance logo não foi possível avaliar dessa maneira esse algoritmo.

SFS

A lista seguinte é o output do SFS para escolher as melhores features:

- ***Decision Tree:*** [‘pregnant:’, ‘thyroid surgery:’, ‘I131 treatment:’, ‘query hyperthyroid:’, ‘lithium:’, ‘goitre:’, ‘tumor:’, ‘hypopituitary:’, ‘psych:’, ‘TSH measured:’, ‘TT4 measured:", ‘T4U measured:’, ‘T4U:’, ‘FTI measured:’, ‘TBG measured:’]
- ***MLP:*** [‘on thyroxine:’, ‘query on thyroxine:’, ‘pregnant:’, ‘thyroid surgery:’, ‘I131 treatment:’, ‘query hyperthyroid:’, ‘tumor:’, ‘hypopituitary:’, ‘psych:’, ‘TSH:’, ‘T3:’, ‘T4U measured:’, ‘T4U:’, ‘TBG measured:’, ‘TBG:’]
- ***XGBoost:*** [‘on thyroxine:’, ‘pregnant:’, ‘thyroid surgery:’, ‘query hypothyroid:’, ‘query hyperthyroid:’, ‘lithium:’, ‘goitre:’, ‘tumor:’, ‘hypopituitary:’, ‘psych:’, ‘T3 measured:’, ‘TT4 measured:’, ‘T4U measured:’, ‘FTI measured:’, ‘TBG measured:’]

Conclusão

XGB: [“T4U:”,“pregnant:”,“on thyroxine:”,“T4U measured:”,“TT4 measured:”,“on thyroxine:”,“query hyperthyroid:”,“tumor:”,“psych:”,“TBG measured:”] Decision Tree: [“on thyroxine:”,“on antithyroid medication:”,“query hyperthyroid:”,“query hypothyroid:”,“tumor”,“psych”,“TSH measured:”,“T4U measured:”,“FTI measured:”] MLP : [‘on thyroxine:’,‘query on thyroxine:’,‘thyroid surgery:’,‘tumor:’,‘hypopituitary:’,‘psych:’,‘TSH measured:’,‘T3 measured:’,‘T3:’,‘T4U:’,‘TBG measured: ’TBG:’,‘diagnoses’]

Estas foram as melhores features que escolhemos já que são as que mantêm um equilíbrio em correlação e output do SFS. Como as features não apresentam correlações muito significativas tivemos mais em conta o output do SFS, já que as redes neuronais não suportam a feature importance

Resultado do modelo

Após a feature selection reparamos que…

Modelo	F1-macro	Train accuracy	Test Accuracy	Macro
XGB	0.521	0,707	0,692	0,546
MLP	0.529	0,702	0,685	0,547
DT	0.405	0,684	0,682	0,498

Ao analisar os dados com este dataset, o método está a prever metade do que era suposto, uma vez que os valores do Recall e f1-macro estão com uma média 0,5.

Model Tuning: Com a conclusão anterior decidimos dar **Tuning**

Decision Tree Tuning

Para tunar a DT vamos mudar alguns hyperparâmetros

- ***MaxDepth:*** Representa a profundidade máxima da árvore de procura, um valor muito grande pode tornar o modelo muito complexo o que pode levar a overfit enquanto um muito pequeno pode levar a underfit. Estamos a ir para valores mais altos para tentar fazer um modelo mais complexo.
- ***Min_Samples_Split:*** Representa o número mínimo de amostras/observações para que um nó seja considerado para divisão. Isto ajuda a controlar o overfit porque o modelo vai ser mais restrito e vai aprender relações mais especificas do conjunto de treino.

- ***Min_Samples_leaf:*** Determina o número mínimo de amostras/observações para estar numa folha, ajuda a controlar o overfit pelo mesmo motivo do parâmetro anterior.

MLP Tuning

- ***hidden_layer_sizes:*** Vai definir o número de neurônios nas camadas ocultas. Estamos a treinar este parâmetro porque quanto mais neurónios, mais o modelo vai aprender padrões nos dados. O que pode melhorar o seu desempenho.
- ***activation:*** A função de ativação para as camadas ocultas. São funções que permitem que a rede neural aprende representações mais complexas de modo não linear. Faz com que a rede aprenda a partir dos erros e melhore a sua precisão.
- ***solver:*** O otimizador usado para o treinamento. É um parâmetro comum para tuning porque é o que ajusta os pesos para o treino do modelo, então um bom solver pode levar a um bom modelo.
- ***alpha:*** Controla a quantidade de regularização L2. É um parâmetro usado na regularização L2, que é uma técnica usada para evitar overfit e que funciona adicionando um erro à função de perda do modelo.
- ***learning_rate:*** A taxa de aprendizado para o optimizador. Controla o tamanho dos passos que o modelo dá a minimizar a função de perda. Um valor bom pode fazer com que o modelo convirja mais rapidamente.

XGB Tuning

Para tunar o XGB vamos mudar alguns hyperparâmetros

- ***MaxDepth:*** A max_depth define a profundidade máxima de uma árvore, no entanto, valores altos podem levar a overfit. Por isso, estamos a tentar valores maiores que o default (6) para tentar fazer um modelo mais complexo.
- ***N_estimators:*** O número de estimadores é o número de árvores que vão ser construídas, estamos só a testar 100 e 200 uma vez que valores muito altos podem levar ao overfit, para além de demorar muito tempo a ser avaliado.
- ***Learning_rate:*** É um campo usado para limitar a aprendizagem de cada iteração, um bom valor torna o modelo mais robusto.
- ***Subsample:*** Este campo define as amostras para cada árvore. Valores menores previnem o overfit.
- ***Colsample_bytree:*** Este campo define as features que vão ser consideradas em cada árvore, um valor alto pode levar a overfit.

Resultado do Tuning

Para manter um equilíbrio entre overfit e complexidade escolhemos ranges mais altos e mais baixos para equilibrar.

Modelo	F1-macro	Train accuracy	Test Accuracy	Macro
DT Standard	0.400	0.691	0.656	0.500
MLP MinMax	0.577	0.709	0.676	0.582

Depois do Tuning, o MLP melhorou o seu f1-macro e é esse que vamos usar para o mock test

Conclusão do O2: Sex

DT Standard: {'max_depth': 5, 'min_samples_split': 10} MLP MinMax: {'alpha': 0.05, 'hidden_layer_sizes': (50, 50, 50), 'learning_rate': 'adaptive', 'max_iter': 1000}

Perante a este problema de classificação verificamos que os modelos não se saem de forma excelente para prever os sexos a partir do dataset. Na qual podemos concluir que problemas da tiroide podem atingir a qualquer pessoa independentemente do sexo. Os modelos mostram uma accuracy overall de aproximadamente 50% o que pode indicar que o modelo está a acertar metade das previsões feitas, como é mostrado no Macro.

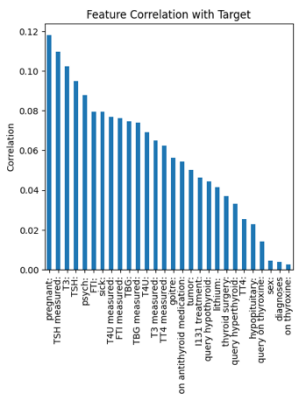
O2: Age

O objetivo do O2: Age era averiguar se era possível fazer um modelo para prever a idade de um individuo a partir das features presentes no dataset. Logo estamos num problema de regressão uma vez que estamos a prever valores contínuos.

Conclusão do processamento de dados

Depois do processamento de dados, que foi igual ao feito em O1, decidimos em usar os modelos: SVR, Ridge, Linear Regression

Seleção de Features



Correlação

Feature importance

Os modelos selecionados não suportam feature importance logo não tomamos esta medida em conta.

SFS

- SVR: ['on antithyroid medication:', 'sick:', 'pregnant:', 'thyroid surgery:', 'I131 treatment:', 'goitre:', 'psych:', 'TSH measured:', 'T3:', 'T4U:']
- Ridge: ['sick:', 'pregnant:', 'I131 treatment:', 'psych:', 'TSH measured:', 'T3 measured:', 'T3:', 'TT4:', 'T4U:', 'FTI:']
- Linear Regression: ['sick:', 'pregnant:', 'I131 treatment:', 'psych:', 'TSH measured:', 'T3 measured:', 'T3:', 'TT4:', 'T4U:', 'FTI:']

Conclusão: Ao analisarmos as correlações notamos que as correlações são muito baixas e que não há muita relação entre as features com o target. Como também não temos métricas de feature_importance, decidimos usar as best features do output do SFS.
SVR: ['query on thyroxine:', 'on antithyroid medication:', 'sick:', 'pregnant:', 'thyroid surgery:', 'I131 treatment:', 'lithium:', 'goitre:', 'tumor:', 'hypopituitary:', 'psych:', 'TSH measured:', 'T3:','T4U:', 'TBG measured:']
Ridge: ['on antithyroid medication:', 'sick:', 'pregnant:', 'thyroid surgery:','I131 treatment:', 'lithium:', 'goitre:', 'psych:', 'TSH measured:','T3 measured:', 'T3:', 'TT4:', 'T4U:', 'FTI:', 'TBG:']
Linear Regression: ['on antithyroid medication:', 'sick:', 'pregnant:', 'thyroid surgery:','I131 treatment:', 'lithium:', 'goitre:', 'psych:', 'TSH measured:','T3 measured:', 'T3:', 'TT4:', 'T4U:', 'FTI:', 'TBG:']

Resultado do Modelo

Modelo	MSE	R^2
SVR	321.8	0.085
LR	321.5	0.086
Ridge	321.5	0.086

Os valores do MSE encontram-se muito altos o que indica que o modelo não está a prever bem o target. O R2 próximo de 0 indica que o modelo não explica nenhuma variabilidade, ou seja, estes modelos estão a funcionar ao mesmo nível de modelos de previsões aleatórias.

Model Tuning

Para melhorar os modelos, decidimos que vamos dar Tuning

Ridge Model Tuning

- *alpha:* É um parâmetro que controla a regularização, com valores mais altos, mais robusto o modelo vai ser, mas demsisado pode provocar overfit.
- *solver:* Especifica o algoritmo a ser usado no cálculo do coeficiente dos modelos, então podemos testar vários.
- *fit_intercept:* Permite verificar e controlar se um termo de intersecção deve ser incluido no modelo.
- **copy_X:** Vai determinar se os dados de entrada devem ser copiados antes de serem usados.
- **max_iter:** Testamos este parâmetro para definir o número da solução ótima.

SVR Model Tuning

- *kernel:* É útil verificar o kernel para testar relações complexas ou lineares.
- *C:* Parâmetro de regularização, valores menores faz com que o modelo não dê overfit.
- *gamma*: Define o coeficiente de kernel para os dados de entrada.

Decidimos verificar estes pois são os dados nas aulas TP

Linear Regression Tuning

- *fit_intercept*: Mesmo que Ridge
- *copy_X:* Mesmo que Ridge
- *positive:* Mesmo que Ridge

Resultado do Tuning

Modelo	MSE	R^2
LR Standard	318.3	0.062
Ridge Standard	318.2	0.063

Depois do Tuning os modelos ficaram praticamente iguais com resultados fracos

Conclusão do O2: Age

- Linear Regression ()
- Ridge: {'alpha': 10, 'solver': 'lsqr'}

Para concluir, este não é capaz de realizar um modelo que consiga prever, com uma boa accuracy, a idade de pacientes relacionados à tiroide. Isto pode indicar que as condições da tiroide pode atingir a qualquer pessoa em qualquer idade.
Nota: Depois de analisarmos o modelo chegámos a conclusão que para os modelos de regressão poderia valer mais a pena o uso de OneHotEncoders, este faz com que as features categóricas se transformem em x colunas numéricas de acordo com o número de valores presentes nessa coluna. Como só chegámos a esta conclusão mais tarde, decidimos não mexer no modelo e deixar com o LabelEncoder normal.

Nota: No ZIP do trabalho foi enviado um PDF com todos os gráficos referenciados acima no caso de ser necessário uma melhor análise.