Mestrado em Engenharia Informática 2021/2022



Machine Learning

Avaliação prática – M2

João Fernando Malva Valente Nº 2017016033

João Pedro Neves Gonçalves Nº 2018014306

22 janeiro 2022

Sumário

Este documento apresenta uma descrição dos algoritmos de supervised e unsupervised learning, quais foram utilizados e a eficiência dos mesmos para o problema detalhado na meta 1 deste trabalho.

# Introdução

Esta segunda meta tem como objetivo mostrar vários tipos de algoritmos de machine learning, sendo esses supervised ou unsupervised.

Vai ser mostrado quais algoritmos foram utilizados, detalhar um pouco os algoritmos, os resultados dos mesmos no dataset já escolhido previamente e uma conclusão sobre esses resultados.

# Supervised Learning

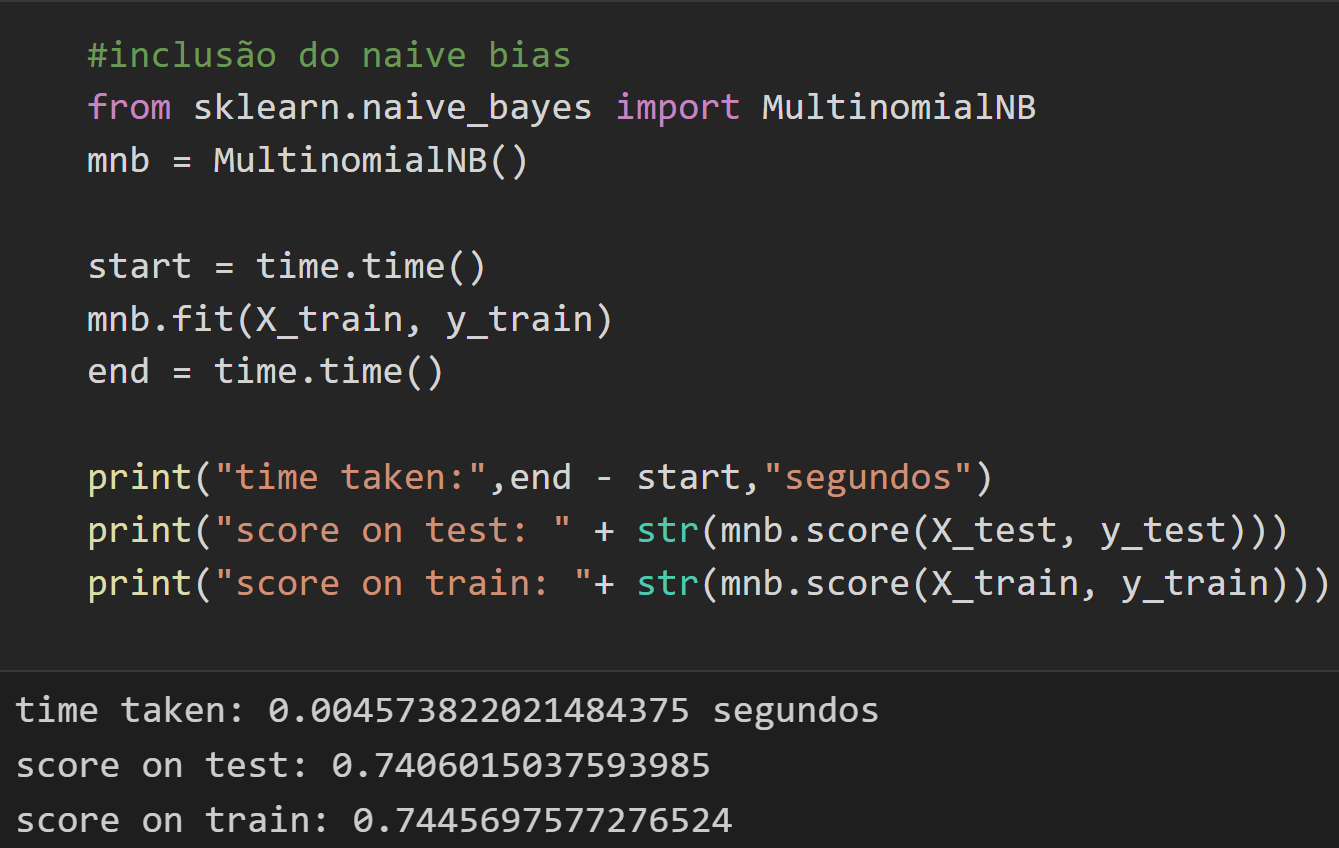
Depois de uma pesquisa sobre os métodos utilizados de supervised learning, decidiu-se utilizar o *Naive Bayes*, *Logistic Regression*, *Random Forest* e o *Voting Classifier*.

Para a análise destes algoritmos, teve-se em atenção a precisão e a velocidade em runtime dos algoritmos, sendo estes dados dependentes do volume dos dados e a qualidade dos mesmos.

Os algoritmos utilizados são dos mais comuns de se encontrar para classificação binária.

## Naive Bayes

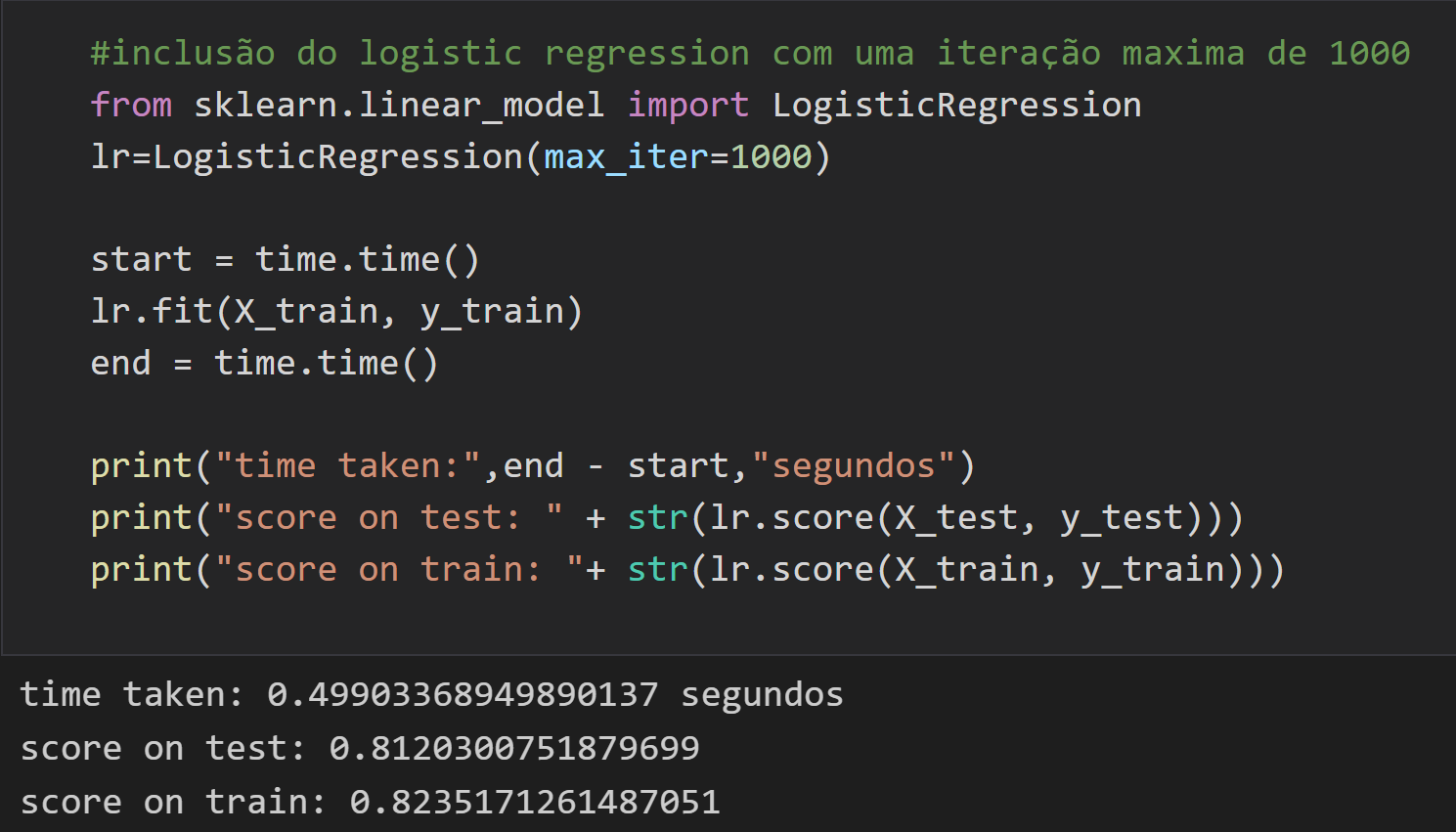
Naive Bayes é um algoritmo de supervised learning baseado na aplicação do teorema de Bayes com a suposição “ingénua” de independência condicional entre cada par de características dado o valor da variável de classe.



Como se pode ver na Figura, este algoritmo mostrou-se bastante rápido de executar e também retorna um resultado inferior a 75%, mostrando-se um algoritmo que fica aquém do esperado em questão de precisão. Dado que ambos as pontuações se encontram bastantes próximas uma das outras, pode-se concluir que não estamos a sobrecarregar o processo de treino.

## Logistic Regression

Logistic Regression é um dos mais velhos e mais básicos algoritmos para resolver um problema de classificação. É um modelo estatístico que na sua forma básica usa uma função logística para modelar uma variável binária dependente. Neste algoritmo estima-se os parâmetros de um modelo logístico (forma de regressão binária).

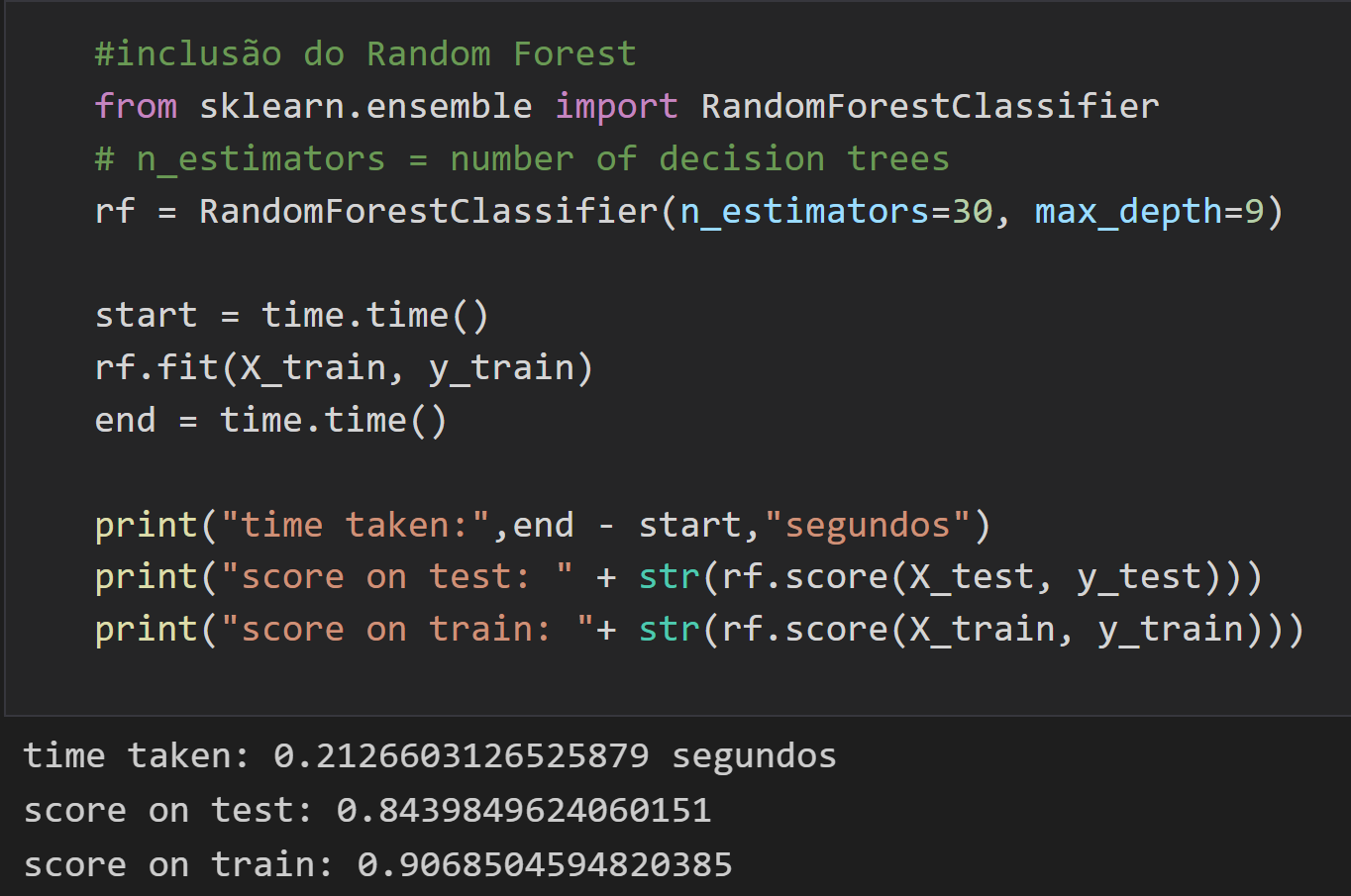


Conclui-se que este algoritmo demora um bom bocado para ser treinado, e apresenta uma precisão boa, acima de 80%, sendo o processo de treino também bastante equilibrado dado os valores próximos.

## Random Forest

Este algoritmo é outro bastante frequente de ser utilizado que usa múltiplas árvores de decisões. Basicamente é um algoritmo de ensacamento modificado de uma árvore de decisão que seleciona os subconjuntos de forma diferente.

Conclui-se que o tempo de execução é bastante aceitável e retorna valores precisos, ficando com um score acima de 90% depois do treino executado. Dada a diferença dos dados de treino, mostra-se um sinal de overfitting.



## Voting Classifier

A ideia por trás deste algoritmo é conceptualmente combinar vários algoritmos de classificação e usar um voto majoritário ou as probabilidades médias para prever os rótulos das classes. Este algoritmo pode ser útil para modelos de desempenho com uma performance bastante parecida com o objetivo de equilibrar as fraquezas individuais de cada algoritmo.



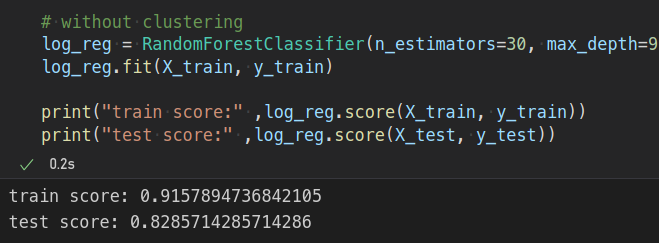
Conclui-se que este algoritmo mostrou resultados aquém do esperado para um algoritmo mais complexo, dado que demorou bastante tempo a ser executado em relação aos outros algoritmos e a precisão não foi a melhor, ficando com resultados um pouco acima dos 80%.

# Unsupervised Learning

Começou se por seguir se o documento anexo1\_M2.pdf para a realização do estudo de unsupervised learning.

# Controlo

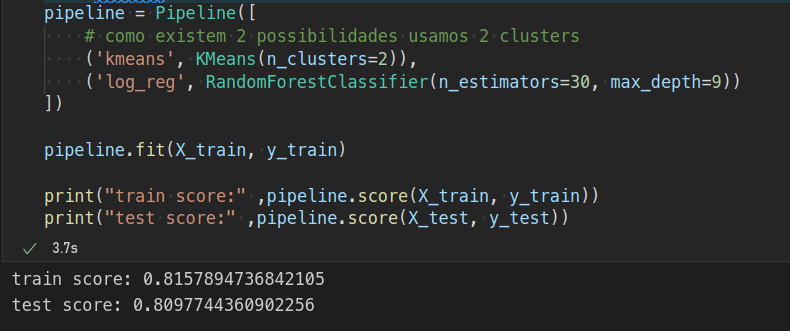
Para valores de controlo testou se sem clustering incialmente, com o melhor algoritmo usado na secção de Supervised Learning [2], o random forest classifier com o seguinte resultado:



Não foi o melhor resultado que já se obteu na area do suppervisionado

## K-Means

Foi Testado de seguida com um k-mean de 2, já que é um problema binario, um numero binario de clusters parecia fazer sentido no contexto:



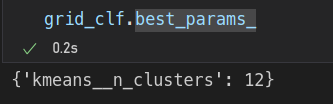
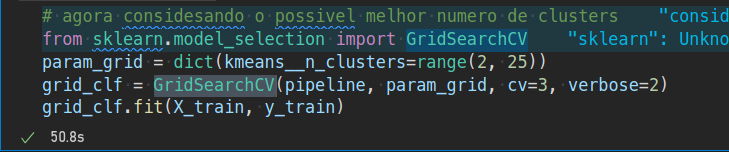
O que deu um valor mais baixo do que antes calculado.

## GridSearchCV

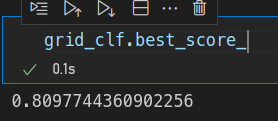
Como podemos ver os valores anteriormente vistos, não são os melhores comparados com o controlo.

Para encontrar o valor ideal usou se o GridSearchCV, este tem a capacidade de correr pipelines em batch para se encontrar valor otimos.

Testou se entre 2 a 25 clusters incialmente se iria para um extremo. Se fosse continua va se para o numero de clusters desse extremo até encontrar se um meio otimo.



No final obteu um valor bastante pequeno de clusters, 12.



O que se obteu um valor baixo tal como no com 2 clusters.