Ciência de Dados

Projeto

André Patrício
87631 - MEIC-T
Instituto Superior Técnico
Torres Vedras - Lisboa
andremppatricio@gmail.com

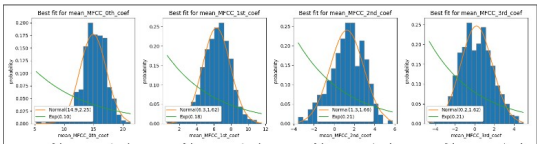
Bernardo Santos
87635 - MEIC-T
Instituto Superior Técnico
Cacém - Lisboa
bsantostecnico@gmail.com

Diogo Viegas
87649 - MEIC-T
Instituto Superior Técnico
Cacém - Lisboa
diogo.viegas@tecnico.ulisboa.pt

1. Statistical Description

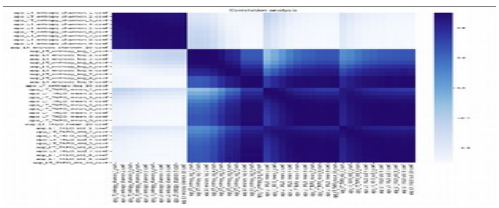
No que toca ao primeiro dataset, podemos descrevê-lo da seguinte maneira: 755 variáveis e 756 observações. Em relação às variáveis, sabe-se que 749 são do tipo float e apenas 6 do tipo int. Analisando mais atentamente as variáveis inteiras, chegamos à conclusão que apenas as variáveis “gender”, “id” e “class” são categóricas, sendo “class” a variável a prever. Cada paciente tem 3 observações no dataset, todas com o mesmo id. Em relação às classes, o balanceamento é 75% de doentes e 25% de não doentes. Podemos, por fim afirmar que o dataset não tem null values.

Analisando por alto o nome das restantes variáveis, reparamos que poderá haver correlação entre elas, sobretudo nas terminadas em “th” (mensais) ou terminadas em números (coeficientes). A título de exemplo, vimos as distribuições das variáveis acabadas em “th” e obtivémos o seguinte histrograma:

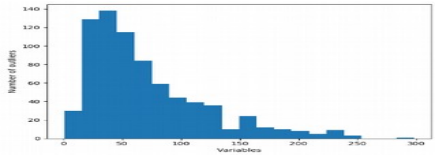


Podemos então observar que a distribuição não é igual entre elas, logo não existe elevada correlação entre as variáveis.

No entanto, para outro subset de dados, podemos observar que existem elevadas correlações entre variáveis no nosso dataset através do seguinte heatmap.



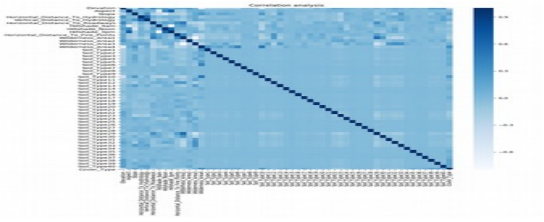
Finalmente, em relação aos outliers, fizémos ainda uma análise utilizando quartis e a fórmula A <= Q1 – 1.5 * IQR e A >= Q3 + 1.5 * IQR em que A é o outlier e IQR é Q3-Q1.



Este gráfico representa a frequência (eixo dos xx) do número de outliers (eixo dos yy) para cada variável.

Em relação ao segundo dataset, podemos afirmar que existem 55 variáveis e 581012 observações sendo que a variável a classificar é “Cover_Type” (7 valores possiveis) que representa o tipo de floresta a considerar. Sabe-se ainda que 48% das observações são do tipo 2 (máximo) e cerca de 0.4% das observações são do tipo 4 (mínimo). Todas as variáveis “Soil_Type” e “Wilderness Area” são categóricas (44 ao todo).

Acerca das correlações entre variáveis, podemos afirmar que não existe um nível elevado de correlações como havia no primeiro dataset.



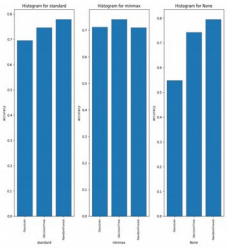
2. Preprocessing

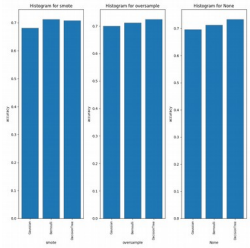
Em relação ao pré-processamento do primeiro dataset, decidimos remover a variável “id” porque não adianta informação relevante e como representa o mesmo paciente pode ser usado pelo classificador para inferir a classe.

Normalizamos os dados para ajudar na classificação. Balanceamos os dados pois o rácio da variável de classificação é 75-25, logo os classificadores teriam tendência para classificar cada observação de acordo com a classificação com maior percentagem.

Visto que o dataset é pequeno optámos por usar K-Fold Cross Validation (StratifiedKFold) com 10 folds como estratégia de treino, de forma a obtermos resultados mais precisos possiveis. Para além disso, temos o cuidado de balancear apenas o conjunto de treino obtido a cada iteração e evitar que observações com o mesmo id estejam em folds diferentes. Para além disso, garantimos que a normalização do conjunto de treino é feita em separado da do conjunto de teste, fazendo fit dos parâmetros da normalização no conjunto de treino e aplicando ao conjunto de teste.

No que toca à normalização e ao balanceamento, decidimos escolher as opções que nos ofereciam melhores resultados ao nível da accuracy e precision, ou seja, Standard e Smote respetivamente.



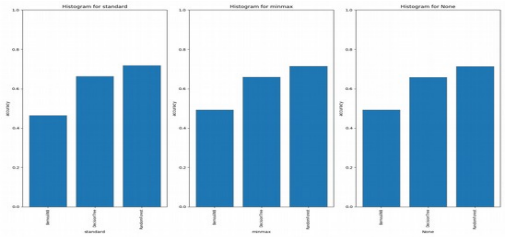


Para todo o pŕe-processamento atrás mencionado usámos classificadores com parâmetros default para inferir o desempenho do dataset de modo geral, dos quais, GaussianModel, BernoulliModel , DecisionTrees e RandomForests.

Finalmente, é ainda relevante explicar que ao nível da redução de variáveis temos uma função que dado um determinado threshold, reduz todas as variáveis relacionadas entre si com um nível de correlação acima do threshold, transformando o conjunto destas variáveis em 3 novas variáveis que representam a média, desvio-padrão e mediana. De realçar que tanto esta função de redução como outros tipos de algoritmos de feature selection são adaptados a cada classificador.

Para o segundo dataset, devido ao seu elevado tamanho, é necessário remover observações seguindo a técnica de sub-sample, ou seja, analisamos qual a classe com menos observações e fazemos sampling dessa quantidade para todas as classes. Esta técnica de balancing é apenas aplicada ao training set. Em relação ao test-set, este é sampled do dataset original sem qualquer tipo de alteração sem ser a normalização com os parâmetros do training set. Usámos também um validation set para o tuning de parâmetros dos classificadores. Testámos ainda outras técnicas como reduzir o número de observações de cada classe para um determinado valor (inferior ao mencionado anteriormente), no entanto, o valor da accuracy não melhora.

Em relação à normalização do segundo dataset, decidimos aplicar a técnica de standard. Reparámos que normalização não melhora consideravelmente a accuracy porque o dataset tem 44 variáveis categóricas que não são normalizadas.

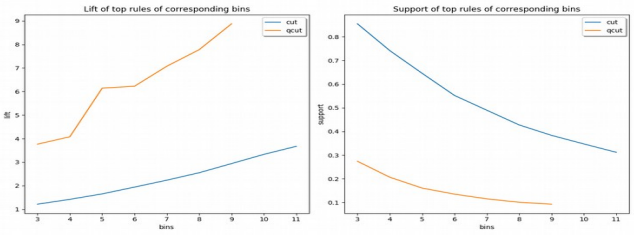


3. Unsupervised

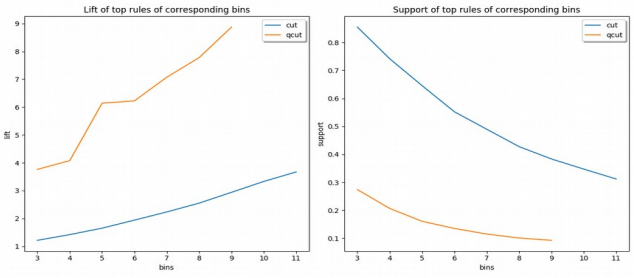
3.1 Association Rules

Os únicos métodos de pré-processamento aplicado é redução de variáveis para 10 através do selectkbest e usando a função f_classif e a dumification das variáveis não categóricas.

Usamos o algoritmo Apriori para calcular os frequent itemsets e de seguida inferimos as association rules. Usamos as top20 regras com melhor lift e support para desenhar o seguinte gráfico:



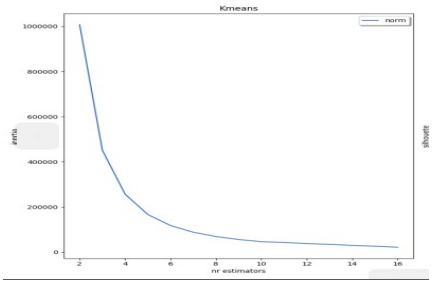
Para o segundo dataset, fazemos o mesmo processo e obtemos o seguinte gráfico:



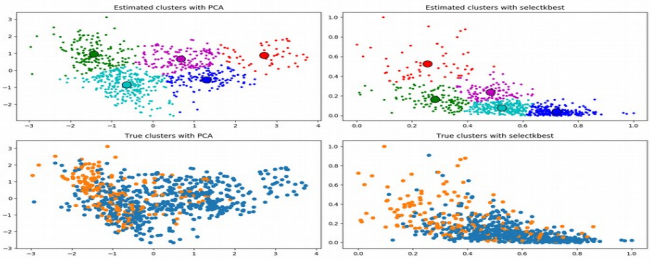
3.2 Clustering

Para determinar qual o número de clusters ideal para segmentar os nossos datasets, variamos o número de clusters e observamos a inércia obtida (utilizando o algoritmo kmeans). Escolhemos o número de clusters que representa o trade-off ideal.

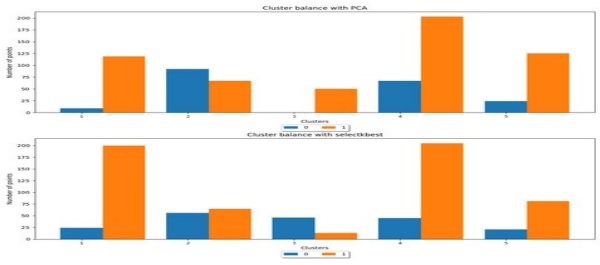
Para o primeiro dataset chegámos ao valor ideal de 5 clusters com silhueta de 0.53924. Gráfico observável em baixo.



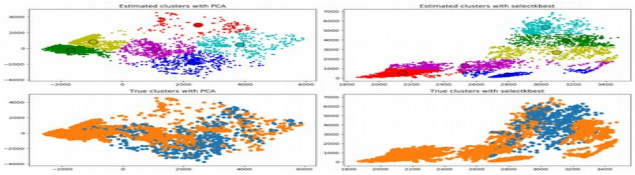
De seguida, reduzimos o dataset para dimensionalidade 2 de forma a facilitar visualizações. Esta redução é feita tanto com SelectKbest como com PCA. Em cima, estão os clusters formados. Em baixo, os pontos correspondentes à classe real.



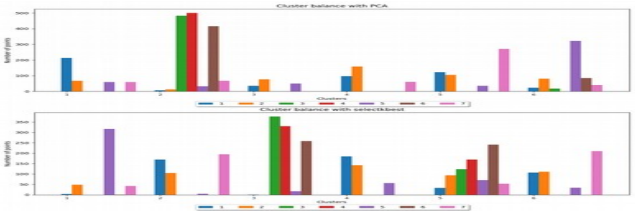
Finalmente, para cada cluster, mostramos o número de observações com a classificação real.



Para o primeiro dataset chegámos ao valor ideal de 6 clusters com silhueta de 0.431035. Gráfico observável em baixo. No entanto, em vez realizarmos sub-sample com o sampling igual ao valor da classe com menos observações, realizamos sub-sample de cada classe para 500 observações.



Por fim, para cada cluster, mostramos o número de observações com a classificação real.



4. Classification

Daqui em diante, assumimos que os dados do primeiro dataset estão normalizados seguindo a técnica standard e balanceados de acordo com o SMOTE.

Em relação aos dados do segundo dataset, assumimos que o dataset já está sub-sampled seguindo a técnica descrita no pré-processamento, e treinamos com o training set, durante a escolha de parâmetros avaliamos com o validation set e os valores finais são calculados no test set.

Em relação aos métodos de avaliação dos classificadores do primeiro dataset, decidimos optar pela accuracy e sensitivity, dando mais foco à sensitivity visto que se trata de um diagnóstico de uma doença. Sobre a avaliação do segundo dataset, usámos apenas a accuracy.

Finalmente, para cada um dos classificadores do primeiro dataset, deduzimos (através do cálculo da accuracy e sensitivity) qual a técnica de feature selection (SelectKBest ou PCA) que melhor se ajusta ao mesmo. Para além disso, aplicmos ainda a nossa função de redução de variáveis correlacionadas acima de um determinado threshold, threshold esse que também calculamos para cada classificador.

4.1 Naive Bayes

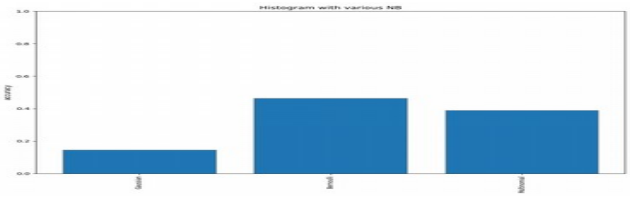
Calculámos então que o primeiro dataset aplicaria, idealmente, a técnica de SelectKBest com k=1 coma um threshold para a função de correlações de 0.9. Após feature selection, ficámos com 699 variáveis.

De entre os modelos de naive-bayes testados (Gaussian e Bernoulli), o que apresenta melhores resultados é o Gaussian, o que faz sentido, tendo em conta que o Bernoulli apresenta melhores resultados com variáveis binárias.

Em relação a aplicar alterar a variável smoothing do classificador, verificámos que não havia diferença nos resultados.

Finalmente, conseguimos os seguintes resultados: 0.728 de precisão com desvio-padrão de 0.019 e 0.959 de sensibilidade com desvio-padrão de 0.028

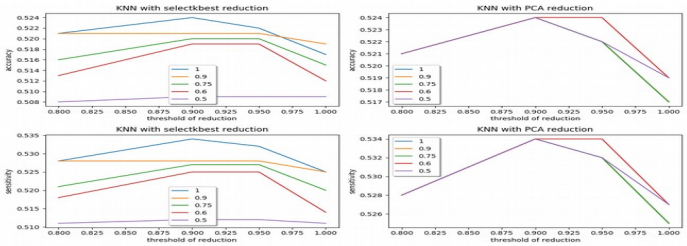
Relativamente ao segundo dataset, o modelo que apresenta melhores resultados é o Bernoulli visto que a maioria dos dados do dataset é binário.



Em relação aos resultados, conseguimos um valor de precisão de 0.46.

4.2 Instance-Based Learning

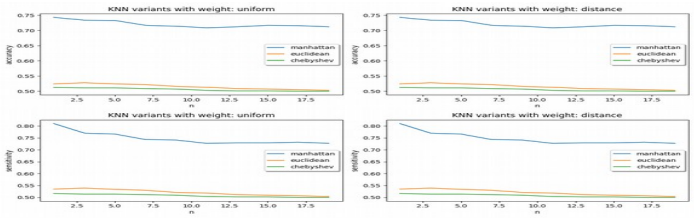
Calculámos então que o primeiro dataset aplicaria, idealmente, a técnica de SelectKBest com k=1 e ainda um threshold para a função de correlações de 0.9. Após feature selection, ficámos com 699 variáveis.



Determinámos de seguida que o número de vizinhos que origina melhores valores é 1. Este valor faz sentido porque como este dataset é relativo a um

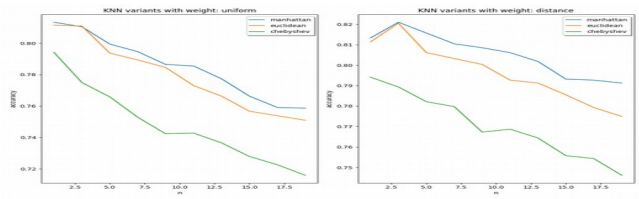
diagnóstico de uma doença, o caso perfeito de deteção de doença é ter valores próximos de outra observação com a mesma classificação.

Relativamente aos restantes atributos do classificador, observamos que a melhor medida para determinar os vizinhos é a distância de Manhattan. No entanto, alterar a forma de pesagem dos vários vizinhos não influencia a precisão do classificador.



Em relação aos resultados, obtivemos 0.743 de precisão com desvio-padrão de 0.067 e sensibilidade de 0.81 com desvio-padrão de 0.173

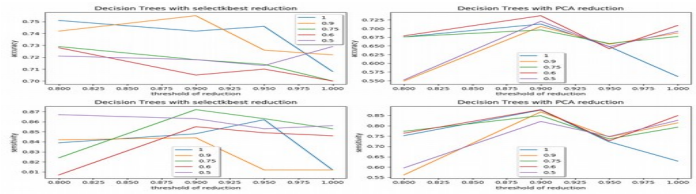
Sobre o segundo dataset, reparámos que o melhor valor para o número de vizinhos ótimo é 3. Para além disso, a melhor medida para determinar a distância de vizinhos continua a ser Manhttan, mas no caso deste dataset, a função de pesagem na escolha de vizinhos é importante, sendo que a função distance obtém melhores resultados.



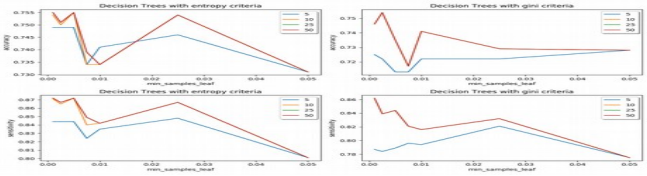
Os resultados do segundo dataset são os seguintes: precisão de 0.66745.

4.3 Decision Trees

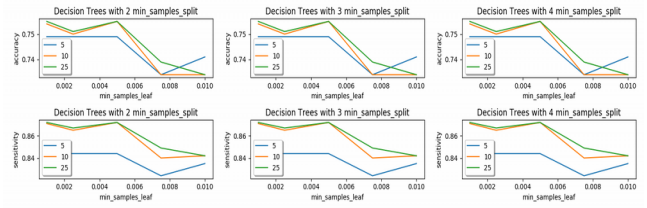
Calculámos então que o primeiro dataset aplicaria, idealmente, a técnica de SelectKBest com k=1 e ainda um threshold para a função de correlações de 0.95. Após feature selection, ficámos com 706 variáveis.



No que toca à função usada para medir a qualidade do split da árvore, a que nos dá melhores resultados é a entropy.



No que toca aos atributos de prunning utilizados para evitar que a árvore entre em overfitting, decidimos escolher 25 como o máximo de profundidade da árvore (post-prunning) e 0.5% de samples como o mínimo de samples de cada folha (pre-prunning). Já em relação ao método de split, optámos pelo best de modo a escolher a variável mais relevante.



Por fim, ao nível de resultados, obtivémos uma precisão de 0.755 com desvio-padrão 0.021 e sensibilidade de 0.872 com desvio-padrão 0.08

Para o segundo dataset, testando o critério de split, optámos pelo gini index. Em relação aos atributos de prunning, continuámos a escolher 25 como máxima profundidade, mas neste dataset escolhemos apenas 0.1% de samples do dataset como valor mínimo de samples para cada folha.

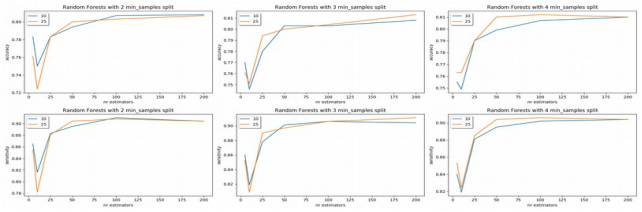
A precisão do segundo dataset situa-se em 0.6231

4.4 Random Forests

Calculámos então que o primeiro dataset aplicaria, idealmente, a técnica de SelectKBest com k=0.6 e ainda um threshold para a função de correlações de 0.95. Após feature selection, ficámos com 424 variáveis.

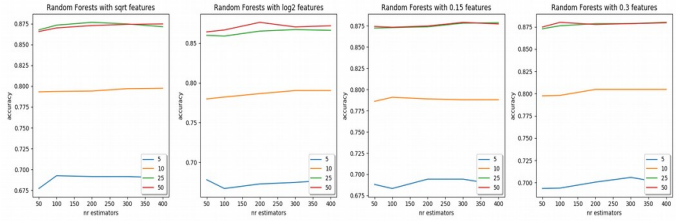
No que toca ao número de árvores, decidimos optar pelo conjunto de 50 árvores e função logaritmo como função aplicada ao número máximo de variáveis para decidir o melhor split. Split esse cuja qualidade será também decidida através do critério de entropy.

Em relação aos atributos de prunning decidimos optar por 25 como profundidade máxima das árvores e 4 mínimo de samples por folha.

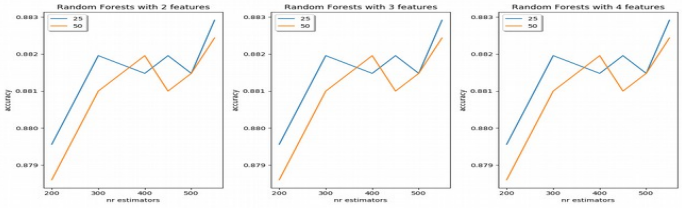


A nível de resultados, obtivémos uma precisão de 0.81 com desvio-padrão de 0.03 e sensibilidade de 0.904 com desvio-padrão 0.07.

Em relação ao segundo dataset, decidimos optar pelo conjunto de 550 árvores e 0.3 variáveis para decidir o melhor split. Split esse cuja qualidade será também decidida através do critério de entropy.



No que toca aos atributos do prunning, decidimos optar pela profunidade de 25 por árvore e 3 mínimo de samples por folha.

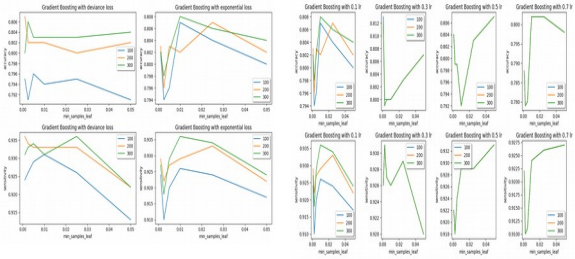


No que toca a resultados, obtivémos uma precisão de 0.76475.

4.5 GBoost

Calculámos então que o primeiro dataset aplicaria, idealmente, a técnica de SelectKBest com k=0.75 e ainda um threshold para a função de correlações de 0.9. Após feature selection, ficámos com 524 variáveis.

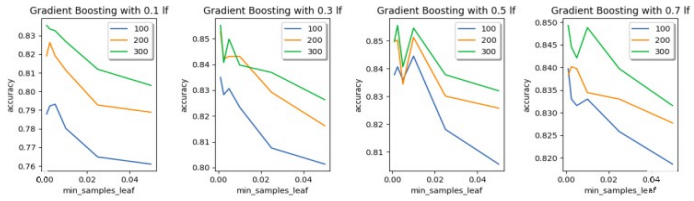
Como função diferenciável de loss, optámos pela exponencial porque apresentava melhores resultados. Já como learning rate, o valor que apresentava melhores resultados era 0.3.



Relativamente aos restantes parâmetros do classificador, verificámos que os melhores resultados surgiam com os seguintes valores: número de estimadores a 100, máximo de profundidade a 3 e mínimo de samples por folha a 1%.

Por fim, os resultados para este classificador são 0.8 de precisão com 0.024 de desvio-padrão e 0.92 de sensibilidade com 0.093 de desvio-padrão.

Para o segundo dataset, optámos pela função de loss deviance com learning-rate igual a 0.5.



Relativamente aos restantes parâmetros do classificador, verificámos que os melhores resultados surgiam com os seguintes valores: número de estimadores a 300, máximo de

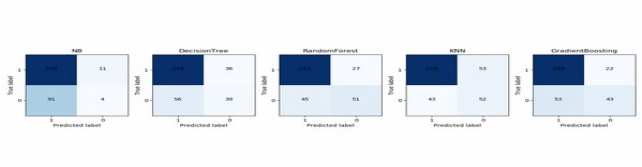
profundidade a 25 e mínimo de samples por folha a 0.1%.

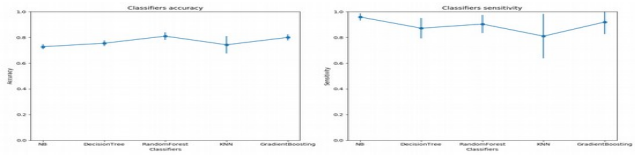
Finalmente, para este classificador em especifico, obitvémos o valor de precisão de 0.76945.

5. Evaluation and Critical Analysis

Apesar de ao longo deste relatório termos apresentado explicitamente quais os valores obtidos em cada passo, decidimos representar esses valores num gráfico e também numa matriz de confusão.

Para o primeiro dataset:





Para o segundo dataset: