DATA SCIENCE PROJECT

G59

Carlota Dias
Instituto Superior Técnico
carlota.lopes.dias@tecnico.ulisboa.pt
87639

José Miguel Ramos
Instituto Superior Técnico
jose.miguel.ramos@tecnico.ulisboa.pt
87679

Maria Osório
Instituto Superior Técnico
maria.osorio@tecnico.ulisboa.pt
87682

1. Descrição Estatística

O primeiro *dataset* (*Parkinson's Disease Classification Data Set*) tem 756 instâncias e 754 atributos, não apresentando quaisquer *missing values*. Este *dataset* tem, no total, informação acerca de 252 pacientes, cada um destes pacientes tem 3 observações. Nestes 252 pacientes sabemos também que há 188 pacientes com *parkinson's disease* e 64 sem a doença.

No que concerne à correlação entre as variáveis fizemos o *heatmap* apresentado abaixo (**Fig. 1.1**), utilizando a função *k_best* (com f_classif) para selecionar as 10 variáveis mais representativas. Concluímos através da análise à **Fig. 1.1** que estas variáveis estão bastante corelacionadas.

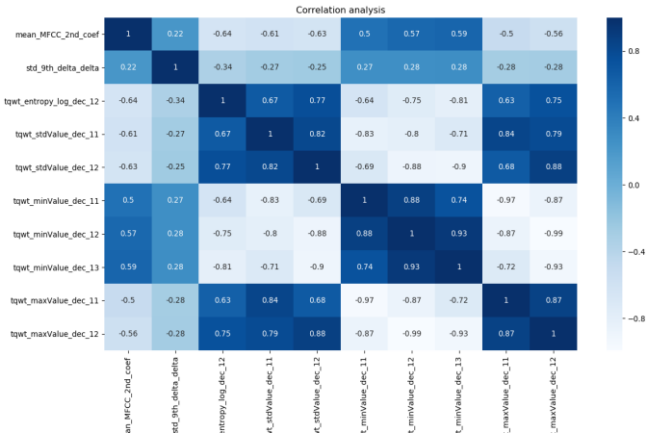

Fig. 1.1

O segundo *dataset* (*Covertype Data Set*) tem 581012 instâncias e 54 atributos, não apresentado quaisquer *missing values*. Este *dataset* prevê o *Covertype* de uma floresta, este *Covertype* pode assumir um valor entre 1 e 7. Na **Fig. 1.2** podemos ver a distribuição de observações para cada valor de *Covertype*.

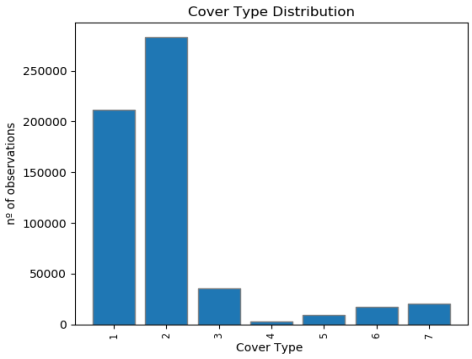

Fig. 1.2

2. Pré-Processamento

Como referido na secção anterior, no primeiro *dataset* temos 3 observações por cada indivíduo, assim, no pré processamento deste *dataset* começámos por fazer a média das 3 observações de cada pessoa para todos os atributos, com o intuito de ter apenas uma observação por pessoa.

Respetivamente à normalização, realizámo-la apenas no primeiro *dataset.* Utilizámos *Min-Max Normalization*, de forma a cada atributo ter valores compreendidos entre 0 e 1.

Para eliminar o problema de variáveis correlacionadas mencionado anteriormente, decidimos retirar atributos altamente correlacionados, fazendo uma média entre atributos com correlação acima de um certo *threshold* para cada instância. Acreditamos que se removêssemos apenas os valores correlacionados poderíamos perder alguma informação, deste modo, ao fazer a média, contornamos esse problema.

Para encontrar o *threshold* ótimo fizemos o gráfico apresentado na **Fig. 2.1**. Este gráfico mostra a *accuracy* do KNN, Decision Tree with *Entropy* e *Decision Tree* with *Gini* com a utilização dos diferentes *thresholds*.

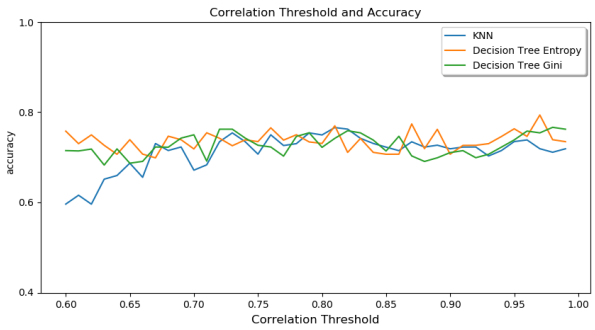

Fig. 2.1

Deste gráfico podemos retirar que não existe uma correlação muito clara entre o valor do *threshold* e a *accuracy* média, no entanto, verificámos que o melhor valor em média para os 3 modelos de classificação é 0.81.

Como forma reforçar a escolha referida acima realizamos um gráfico (**Fig. 2.2**) que compara o número de *features* removidas, tendo em conta o *threshold*. Isto ajuda-nos comprovar e validar o *threshold* escolhido ser 0.81, visto que na **Fig. 2.2** a partir deste valor começam a ser removidas demasiadas *features* para uma variação pequena de *threshold*.

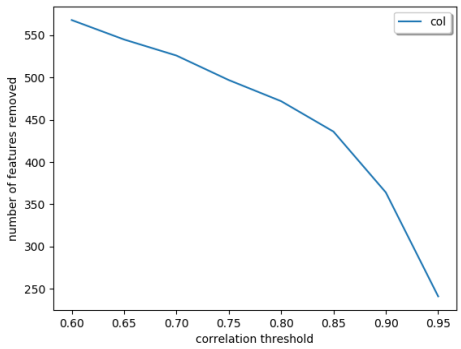

Fig. 2.2

Ainda dentro do pré processamento, fizemos o balanceamento dos dados no conjunto de treino. Este balanceamento é feito uma vez que, como referido anteriormente, o primeiro *dataset* tem uma disparidade muito elevada de classes positivas e negativas. Esta disparidade não é desejada para treinar um classificador. No caso do *Parkinson's Disease Dataset* fizemos um gráfico em que comparamos a *accuracy* máxima sem balancear os dados, utilizando a técnica de *Over Sampling* e de SMOTE. Olhando para a **Fig. 2.3** concluímos que a *accuracy* com Oversampling é superior à do SMOTE para a maioria dos casos. Embora, sem balanceamento a accuracy seja maior, acreditamos que isto acontece, pois, não usando qualquer tipo de balanceamento, estamos a dar demasiada importância à classe com maior frequência, logo, esta abordagem não deve ser a escolhida.

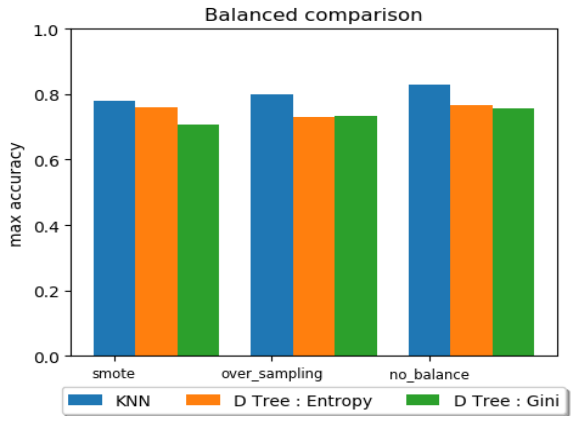

Fig. 2.3

Em relação aos *outliers* do *parkinson's Disease Dataset* decidimos não aplicar qualquer tratamento, visto que este *dataset* não tem observações suficientes para que seja razoável remover os *outliers*. Para além disso, não temos conhecimento suficiente sobre as variáveis para saber entre que valores cada um dos atributos se deve encontrar.

Para o segundo *dataset*, dado a quantidade reduzida de atributos do mesmo considerámos que não faria sentido retirar variáveis correlacionadas. No entanto, procedemos a um *subsample* do nosso *dataset* na medida em que este tem 581012 observações e para além disso se encontra altamente desbalanceado.

A primeira parte do *pre-processing* consiste em fazer *subsampling*. Desta forma, para cada classe retiramos uma sample de 2000 observações, o que nos permite manter os dados balanceados reduzindo o número de observações de 581012 para 14000. Após este processamento procedemos à remoção de *outliers*. Esta remoção foi feita através da aplicação de um threshold máximo e mínimo, todas as observações que possuíssem valores superiores a Q3 + 1.5 x IQR ou inferiores a Q1 -1.5 x IQR para uma determinada *feature* foram removidas.

3. Unsupervised Learning

Para realizar as técnicas apresentadas de seguida, no primeiro *dataset*, o único tipo de pré processamento aplicado consistiu em fazer a média das 3 observações por cada indivíduo, de forma a não considerar mais do que uma observação por indíviduo.

Para o segundo *dataset*, *o* único processamento aplicado consistiu em fazer *subsampling* dado ao elevado número de observações.

3.1 Associating Rules

Para o estudo das regras de associação, decidimos juntar as 3 observações de cada indivíduo através da média, tomámos esta opção uma vez que, como as observações em causa seriam posteriormente discretizadas, poderia acontecer que para um mesmo indivíduo uma determinada característica fosse classificada em bins diferentes.

Como referido anteriormente, para realizar discretização utilizámos duas técnicas: *cut* e *qcut*. Para isto analisámos na **Fig. 3.1.1** o *average support* em comparação com o número de *bins*. Através desta análise concluímos que utilizando a técnica de *qcut* (discretização por frequência) conseguimos um *average support* menor, o que significa que chegamos a regras mais interessantes (menos óbvias), no entanto menos representativas.

Por outro lado, na **Fig. 3.1.2** mostramos o *average lift* em comparação com o número de *bins*. Usando tanto a técnica de *qcut* como de *cut* conseguimos obter maior *lift* com maior número de *bins*. Para além disso conseguimos ver que usando a técnica de *qcut* obtemos *lifts* muito superiores isto prende-se com o facto de ao usar uma discretização por frequência nenhum bin poderá ter uma probabilidade muito elevada, ou seja tanto o consequente como antecedente nunca poderão ter uma probabilidade muito alta.

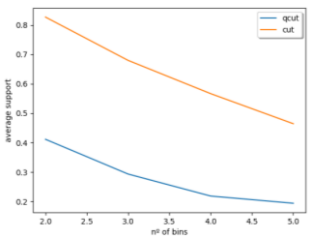

Fig 3.1.1

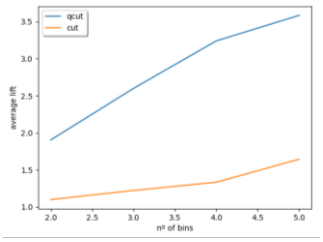

Fig 3.1.2

Na Tabela **3.1.1** apresentamos as regras de associação para o primeiro *dataset*, utilizando para a discretização dos dados a técnica de *qcut* com 5 *bins*, de forma a tentar obter regras menos triviais e com *lifts* superiores.

Antecedentes	Consequentes	Support	Confiança	Lift
{tqwt_stdValue_dec_12_high}	{tqwt_entropy_log_dec_12_high}	0.19444	0.96078	4.74740
{tqwt_minValue_dec_11_high}	{tqwt_maxValue_dec_11_low}	0.19444	0.96078	4.74740
{std_8th_delta_delta_high}	{std_8th_delta_high}	0.19048	0.94118	4.65052

Tabela 3.1.1

No Segundo *dataset* realizamos a mesma análise apresentada anteriormente nas **Fig. 3.1.1** e **Fig. 3.1.2.** obtendo resultados semelhantes.

Na Tabela **3.1.2** apresentamos as regras de associação, utilizando para a discretização dos dados a técnica de *qcut* com 3 *bins*. Olhando para a primeira regra podemos concluir que 90% das vezes em que a Horizontal_Distance_To_Fire_Points é baixa e a Elevation também é baixa, a Wilderness_Area é do tipo 4. Também conseguimos retirar que os antecedentes Horizontal_Distance_To_Fire_Points_low e Elevation_low só acontecem com o consequente Wilderness_Area_4 17% das vezes. Dado o alto valor de *lift* conseguimos perceber que a confiança é alta ainda que a probabilidade do consequente seja baixa, o que nos permite concluir que esta regra não é trivial.

Antecedentes	Consequentes	Support	Confiança	Lift
{'Horizontal_Distance_To_Fire_Points_low', 'Elevation_low'}	{'Wilderness_Area_4'}	0.17450	0.90494	3.04190
{'Horizontal_Distance_To_Hydrology_low', 'Wilderness_Area_4'}	{'Elevation_low'}	0.14081	0.99782	2.98888
{'Elevation_low'}	{'Wilderness_Area_4'}	0.29391	0.88037	2.95931
{'Elevation_medium', 'Horizontal_Distance_To_Fire_points_medium'}	{Horizontal_Distance_To_Roadways_medium}	0.12555	0.70785	2.12356
{'Elevation_medium_low'}	{'Wilderness_Area_3'}	0.23697	0.71200	1.53616

Tabela 3.1.2

Para ambos os datasets, após a *dumificação* das variáveis juntamos a classe também esta *dumificada*, na tentativa de encontrar regras cujo o consequente fosse a classe.

3.2 Clustering

Com o intuito de descobrir qual o melhor número de clusters a usar para o primeiro *dataset* fizemos dois gráficos (**Fig. 3.2.1** e **Fig. 3.2.2**) que nos mostram a variação do SSE (*error sum of square*) com o número de *clusters*. Na **Fig. 3.2.1** fazemos o gráfico sem *feature selection* enquanto que na **Fig. 3.2.2** fazemos *feature selection* com o PCA. O número de *clusters* escolhidos foram 3 visto que a partir deste ponto o valor do SSE tem uma variação muito menos significativa.

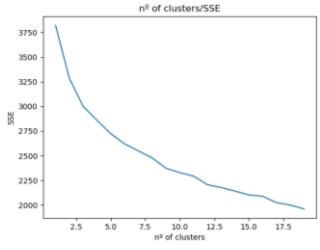

Fig. 3.2.1

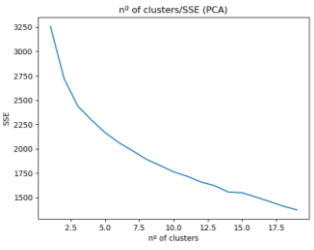

Fig. 3.2.2

Para fazer *clustering* com PCA fizemos um gráfico (**Fig. 3.2.3**) no qual vemos qual o valor mínimo de componentes que explica uma variabilidade significativa dos dados. Concluímos através da **Fig. 3.2.3** que são necessárias 40 variáveis para que sejam explicados 85% dos dados.

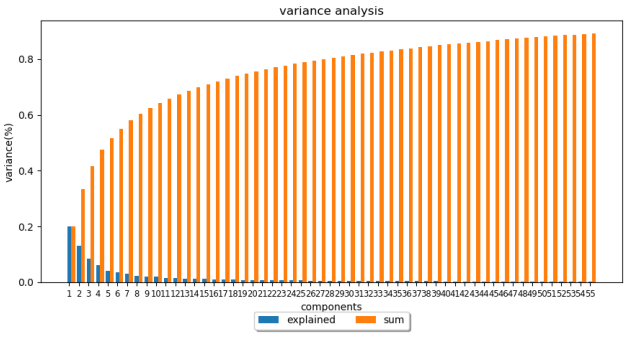

Fig. 3.2.3

Após concluir que o número ideal de clusters é 3, procedemos à análise dos mesmos em comparação com a *real distribution* dos valores.

Na **Fig 3.2.4** podemos ver uma representação gráfica dos *clusters* de acordo com todas as variáveis do *dataset*. A representação é dada segundo as 2 *features* mais representativas, escolhidas segundo *select k_best* utilizando a função *f_classif* .Utilizamos a função f_classif uma vez que estamos a lidar com dados numéricos.

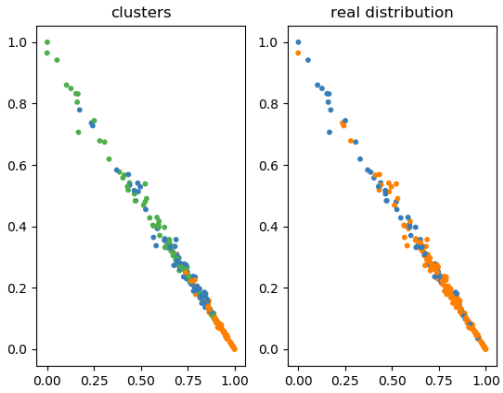

Fig. 3.2.4

Na **Fig 3.2.5** podemos ver os clusters mapeados nas duas variáveis mais representativas selecionadas pelo PCA.

Para escolher o número de componentes escolhidas pelo PCA, usados na formação dos clusters, analisámos a variabilidade dos dados explicada por cada componente, nesta análise chegámos à conclusão que 85% da variabilidade é explicada por 40 variáveis, como analisado na **Fig. 3.2.3**.

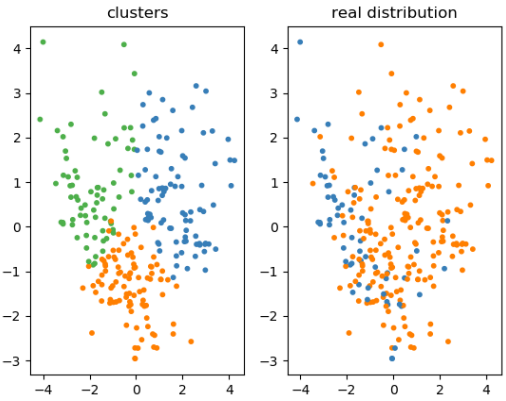

Fig 3.2.5

Ainda sobre o primeiro dataset calculámos o rand index e a silhueta. Para o clustering sem *feature selection* os valores foram os seguintes:

Rand_index: 0.11110561821650762

Silhouette: 0.11791116488344823

Quanto ao *clustering* com PCA os valores dos mesmos foram os seguintes:

Rand_index: 0.11314537255504407

Silhouette: 0.145732544120659

Em ambos os casos a *Silhoutte*, que nos permite ver tanto a coesão *intra-cluster* como o afastamento *inter-cluster*, apresentou valores muito abaixo do desejável (> 0.4).

Para além da análise anteriormente apresentada também podemos ver o número de indivíduos com *parkinson's Disease* em cada um dos 3 clusters. Na **Fig. 3.2.6** podemos observar informação acerca dos indivíduos doentes e saudáveis em cada cluster, selecionando 40 *features* com o PCA. Enquanto que na **Fig. 3.2.7** podemos observar os indivíduos doentes e saudáveis em cada cluster utilizando todo o dataset. O facto do Cluster 3, em ambos os casos ter aproximadamente o mesmo número de índividuos com e sem doença, pode explicar os valores baixos do *Rand índex*.

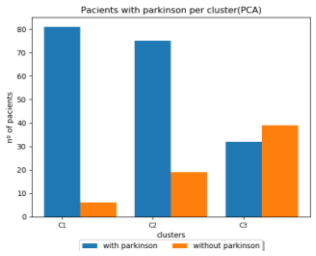

Fig. 3.2.6

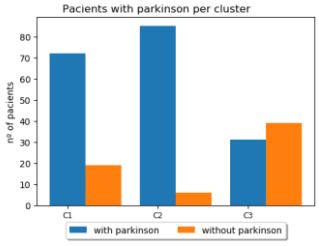

Fig. 3.2.7

No que toca ao Segundo *dataset* fizemos dois gráficos (**Fig. 3.2.8** e **Fig. 3.2.9**) que nos mostram a variação do SSE (*error sum of square*) com o número de *clusters*, tal como fizemos para o primeiro *dataset*. O número de *clusters* escolhidos, neste caso, foram 3 visto que a partir deste ponto o valor do SSE tem uma variação muito menos significativa.

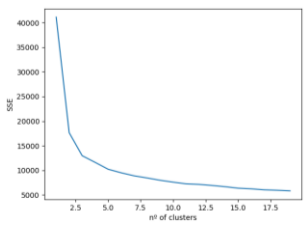

Fig. 3.2.8

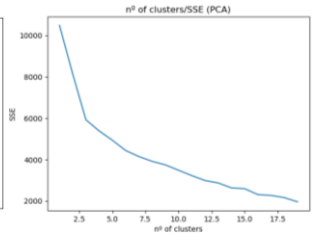

Fig. 3.2.9

Para fazer *clustering* com PCA fizemos um gráfico (**Fig. 3.2.10**) no qual vemos qual o valor mínimo de componentes que explica uma variância significativa dos dados. Concluímos através da **Fig. 3.2.10** que são necessárias 19 variáveis para que sejam explicados 85% dos dados.

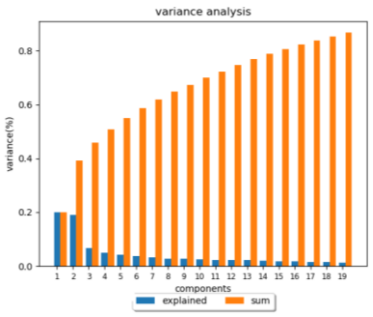

Fig 3.2.10

Após concluir que o número ideal de clusters é 3, procedemos à análise dos mesmos em comparação com a *real distribution* dos valores.

Na **Fig 3.2.11** podemos ver os clusters mapeados nas duas variáveis mais representativas selecionadas pelo PCA.

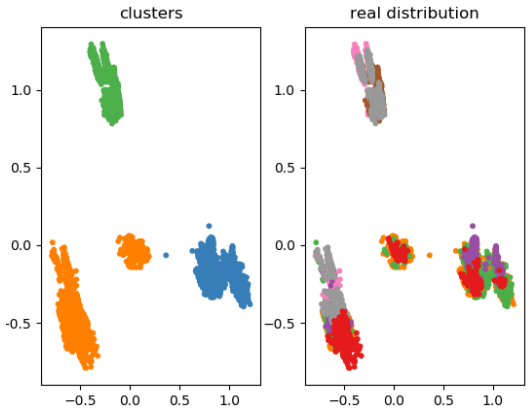

Fig. 3.2.11

Na **Fig 3.2.12** podemos ver uma representação gráfica dos *clusters* de acordo com todas as variáveis do *dataset*. A representação é dada segundo as 2 *features* mais representativas, escolhidas segundo *select k_best* utilizando a função *f_classif*.

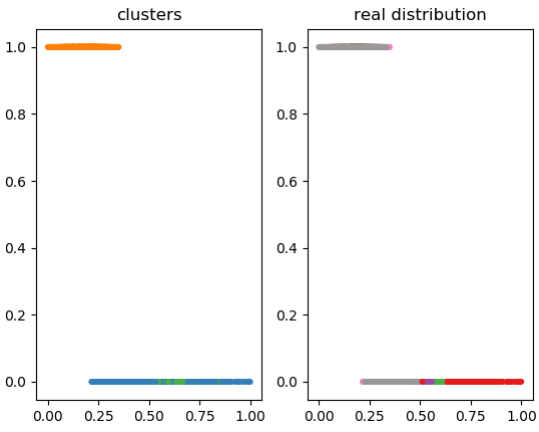

Fig. 3.2.12

Como conclusão ao estudo de *clustering* do segundo *dataset* calculamos o *rand index* e a silhueta. Para o *clustering* sem *feature selection* os valores foram os seguintes:

Rand_index: 0.16998144262373543

Silhouette: 0.2917756116914991

Quanto ao *clustering* com PCA os valores dos mesmos foram os seguintes:

Rand_index: 0.16998144262373543

Silhouette: 0.34780900596132175

4. Classification

Utilizámos ambos os *datasets* para estudar os diferentes classificadores e para cada um destes variámos os valores dos parâmetros de interesse. Posteriormente, verificámos a evolução da *accuracy* de cada classificador consoante essas variações. Os os desvios das linhas da *accuracy* nos *line charts* representam um intervalo de confiança de 95%.

No primeiro dataset, devido ao número reduzido de observações decidimos utilizar a técnica de *Cross Validation*. Esta técnica é também útil no segundo dataset, como forma de obter outras métricas para além da accuracy, como o intervalo de confiança.

4.1 Naïve Bayes

Para analisar o *Naïve Bayes*, verificámos apenas as diferentes *accuracies* para cada estimador e obtivemos os resultados mostrados na **Fig. 4.1.1** para o primeiro *dataset* e **Fig. 4.1.2** para o segundo *dataset*.

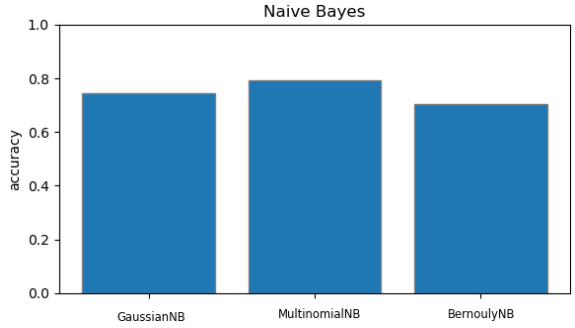

Fig. 4.1.1

Concluímos que o melhor estimador para o primeiro *dataset* é o *MultinomialNB*. com uma accuracy de **79.6±1.22%**.

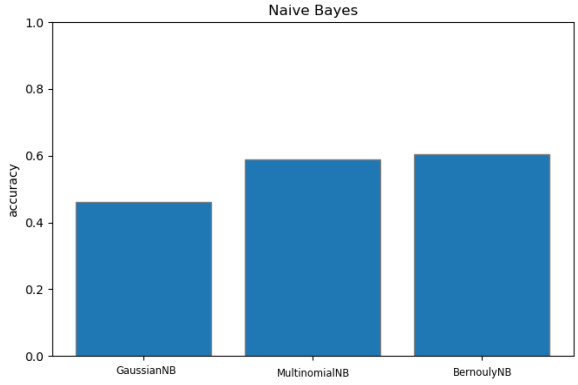

Fig. 4.1.2

Concluímos que o melhor estimador para o segundo *dataset* é o *BernoulyNB* com uma accuracy de **60.4±0.95%**.

4.2 Instance-based Learning

Para estudar o *Instance-based Learning* (ou *K-Nearest Neighbours* – KNN) decidimos verificar os resultados obtidos para diferentes números de vizinhos e diferentes medidas de distâncias. Os resultados obtidos estão descritos na **Fig. 4.2.1**.

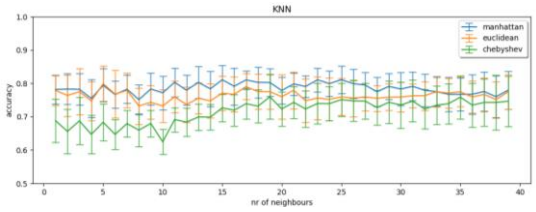

Fig. 4.2.1

Deste gráfico podemos retirar que a *manhattan distance* obteve as melhores *accuracies*. Para esta medida de distância, quanto ao número de neighbours as diferenças foram pouco significativas, no entanto, os máximos de *accuracy* encontrados foram com 15, 23 e 25 vizinhos. Destes máximos, o 23 foi o que obteve o menor intervalo de confiança, apresentando uma *accuracy* total de **81.1±3.56%**.

No que toca à *feature selection* do primeiro *dataset*, podemos observar na **Fig. 4.2.2** a accuracy e o recall da seleção de diferentes números de *features* utilizando o *select k_best*.

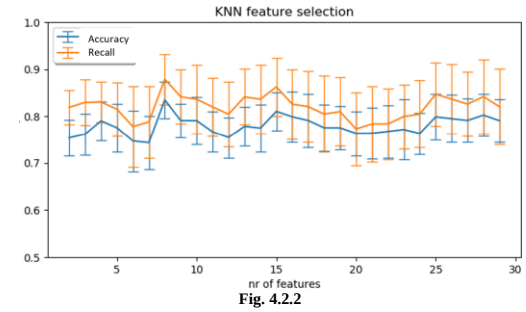

Fig. 4.2.2

Pela figura, conseguimos ver que a accuracy é maxima ao selecionarmos 8 features, atingindo um pico máximo de **83.5±3.61%**. Para este pico o valor do *recall* foi **87.9±5.73%**.

No que toca ao segundo dataset, verificámos os resultados obtidos ao estudar o *Instance-based Learning* com os dados normalizados **Fig. 4.2.3** e comparamos este resultado com o estudo do mesmo classificador, mas sem normalizar os dados **Fig. 4.2.4**

A partir da comparação destes dois gráficos chegamos à conclusão que a *accuracy* do classificador KNN é melhor quando os dados não estão normalizados.

Da **Fig. 4.2.4** podemos retirar que a *manhattan distance* obteve o máximo de *accuracy* com 1 vizinho. Para todas as medidas de distância analisadas podemos tirar a conclusão que quando o número de *neighbours* aumenta a *accuracy* diminui. A *accuracy* máxima obtida para este caso foi de **82.6±0.72%**.

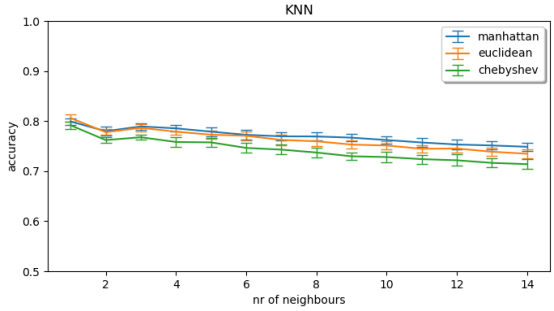

Fig. 4.2.3

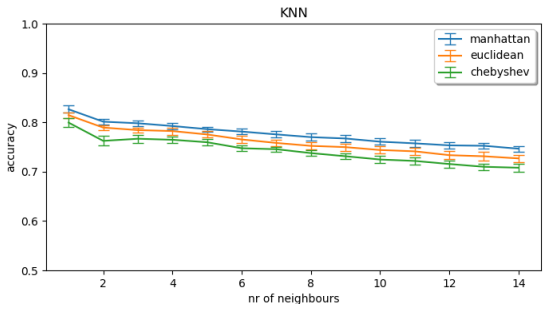

Fig. 4.2.4

Acreditamos que a razão para a melhor *accuracy* ser obtida sem normalização se deve ao facto de existirem muitas variáveis categóricas, e ao fazer a *Min-Max Normalization* os valores dos dados categóricos continuam iguais, e os restantes ficam distribuídos num intervalo de 0 a 1, logo, a distância entre variáveis categóricas é sempre 0 ou 1, e nas restantes está distribuída nesse intervalo. Assim, a distância entre variáveis categóricas terá um peso maior que as restantes, o que prejudica a classificação do KNN.

4.3 Decision Trees

Para estudar as *Decision Trees* variámos o valor das *min sample leafs* (percentagem mínima de *samples* numa folha da àrvore) e *max depths* (profundidade máxima da àrvore). Verificámos os valores anteriores para dois critérios, o *gini* e a *entropy*. Os resultados estão descritos na **Fig. 4.3.1** para o primeiro *dataset* e na **Fig. 4.3.2** para o segundo *dataset.*

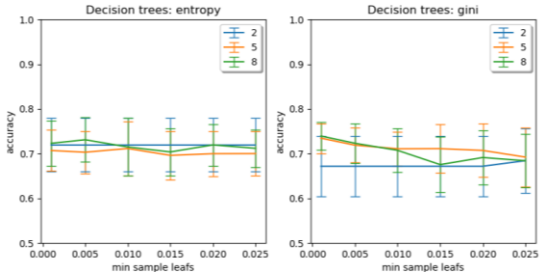

Fig. 4.3.1

Do gráfico obtido (**Fig. 4.3.1**) podemos notar uma pequena descida de *accuracy* com o aumento do *min sample leafs* para a maioria das *max depths* a partir de um certo valor. Decidimos que o valor ótimo seria com *max depth* de 8 e *min sample leafs* de 0.001% para o *gini* e 0.005% para a *entropy*. Destes dois critérios o *gini* obteve maior *accuracy* e um menor intervalo de confiança, dando uma *accuracy* total de **73.9±3.08%**.

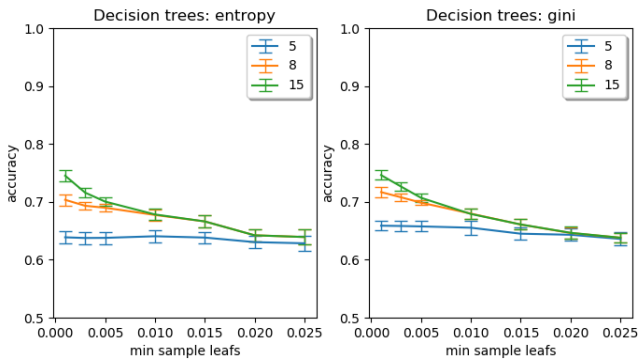

Fig. 4.3.2

Do gráfico obtido (**Fig. 4.3.2**) podemos notar uma pequena descida de *accuracy* com o aumento do *min sample leafs* para todas as *max depths* a partir de um certo valor. Decidimos que o valor otímo seria com *max depth* de 15 e *min sample leafs* de 0.001% para o *gini* e para a *entropy*. Destes dois critérios o *gini* obteve maior *accuracy* e um menor intervalo de confiança, dando uma *accuracy* total de **74.6±1.02%**.

4.4 Random Forests

Na análise das *Random Forests* utilizamos duas funções de determinação do número de *features*, raiz quardrada e logaritmo. Para ambas as funções variámos ainda o valor das *max features* e número de estimadores. Os resultados estão descritos na **Fig. 4.4.1** para o primeiro *dataset*.

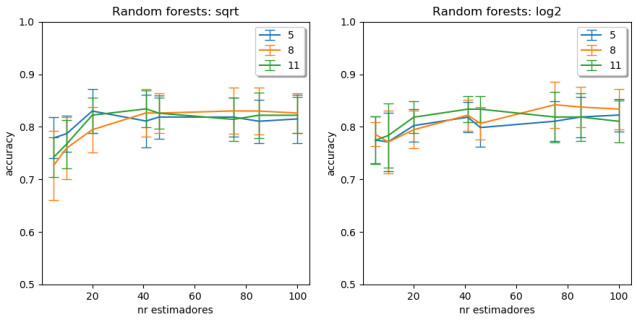

Fig. 4.4.1

Em ambos os gráficos podemos ver um aumento progressivo da *accuracy* até chegar ao valor de cerca 40 estimadores para *max depth* de 8 e 11. A escolha de número máximo de *features* com raiz quadrada (*sqrt*) atinge o seu máximo em 41 estimadores. O logaritmo atinge um máximo de accuracy maior que o da raiz quadrada, para 75 estimadores e uma *max depth* de 8, sendo a *accuracy* total de **84.2±4.36%**.

Os resultados obtidos para o segundo *dataset* estão descritos na **Fig. 4.4.2**.

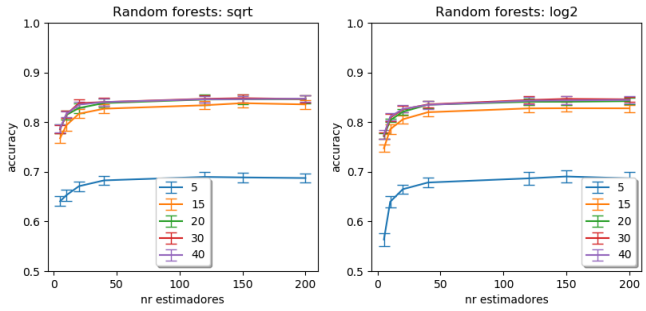

Fig. 4.4.2

Em ambos os gráficos podemos ver um aumento significativo da *accuracy*, para um número reduzido de estimadores, até por volta dos 35 estimadores a accuracy começa a ter um aumento consideravelmente menos significativo. A raiz quadrada (*sqrt*) e o logaritmo atingem um máximo de accuracy em 150 estimadores e uma *max depth* de 30, sendo a *accuracy* da raiz quadrada (sqrt) mais elevada nesse ponto (**84.8±0.78%**).

4.5 Gradient Boosting

No estudo do *Gradient Boosting* analisámos a variação da *accuracy* consoante o *learning rate* e o número de estimadores definidos. Analisámos também a variação da *max depth* e *max*

features, no entanto o *default* de 2 em ambos revelou-se sempre a melhor opção.

Através da **Fig. 4.5.1** conseguimos perceber que o *learning rate* ótimo é de 0.25 com 300 estimadores. Atingindo uma maior *accuracy* que qualquer outro classificador estudado para o primeiro dataset, sendo esta de **85.4±3.38%**.

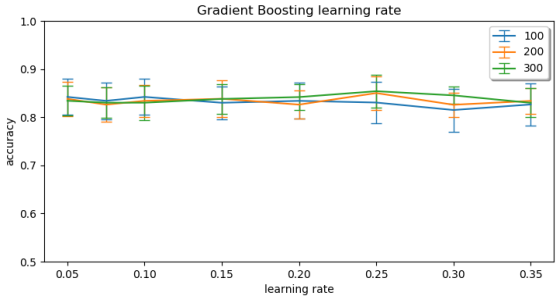

Fig. 4.5.1

Relativamente ao segundo *dataset*, através de uma análise à **Fig. 4.5.2** conseguimos retirar a informação que que o *learning rate* ótimo é de 0.4 com 500 estimadores, com uma accuracy total de **79.5±0.66%**. Este classificador não se revelou tão satisfatório neste *dataset* como no primeiro.

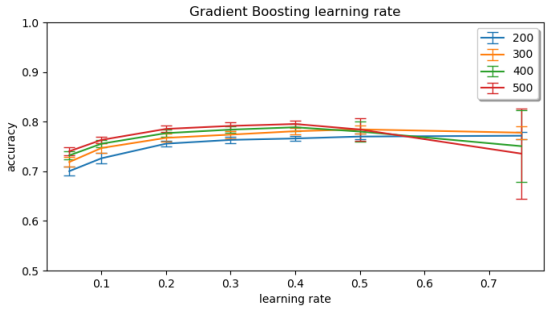

Fig. 4.5.2

5. Evaluation and critical analysis

Após a avaliação individual de cada um destes classificadores, retirámos algumas observações interessantes.

No primeiro dataset e através da classificação com KNN (**Fig. 4.2.2**) conseguimos ver que o recall é superior à accuracy. Este aspeto é importante, dado que numa avaliação médica devem ser evitados *"False Negatives"*.

Mesmo considerando a disparidade dos dois datasets, um com muitos atributos numéricos altamente correlacionados e outro com menos atributos e mais observações, o KNN e as Random Forests mostraram-se abordagens consistentes.

Em Relação ao pattern mining, no segundo data set conseguimos encontrar regras mais intuitivas do que no primeiro, em várias destas regras um atributo categórico é explicado por atributos numéricos. Consideramos que isto se deva principalmente ao facto de no segundo data set existir um elevado número de variáveis categóricas e as variáveis não serem altamente correlacionadas.

Em Relação a clusters, conseguimos ainda concluir que usando PCA com um número de componentes representativo da variabilidade dos dados conseguimos obter melhores resultados.