Instituto Superior de Engenharia de Lisboa

ADEETC

Mineração de Dados em Larga Escala

Laboratório 2

Alunos

Nuno Gomes 18364
Ricardo Ramos 46638
Rafael Carvalho 47663

Professores

Artur Ferreira
Matilde Pós-de-Mina Pato
Nuno Datia

abril
2024

1.ÍNDICE

ÍNDICE DE FIGURAS

ACRÓNIMOS

FS - *Feature Selection*

FR – *Feature Reduction*

FD – *Feature Discretization*

PCA – *Principal Component Analysis*

EFB – *Equal Frequency Binning*

MM – *Mean-Median* (Média-Mediana)

SVD – *Single Value Decomposition*

CAIM - *Class-Attribute Interdependence Maximization*

2. Introdução

A análise de grandes conjuntos de dados para extrair informações valiosas e padrões significativos é o foco da disciplina de Big Data Mining. Com o aumento exponencial do volume de dados na era digital, tornou-se essencial contar com ferramentas e tecnologias especializadas para lidar eficientemente com esses vastos conjuntos de informações.
Nas tarefas de mineração de dados em larga escala, e de treino de classificadores é imperativo fazer o pré-processamento dos dados. Este processo trata-se analisar, limpar, transformar para que possa ser feito o treino de um modelo que devolva bons resultados. Pretende-se então com o pré-processamento dos dados filtrar *features* que não sejam consideradas representativas ao aplicar técnicas de seleção (*Feature Selection - FS*) ou de redução (*Feature Reduction - FR*) e consequentemente discretizar (*Feature Discretization - FD*) as *features* contínuas. A discretização apresenta vantagens no treino dos modelos dado que reduz o impacto de pequenas flutuações nos dados [1].

No segundo laboratório, são fornecidos dois *datasets*, sendo que um mostra as condições metereológicas em Lisboa no período de um mês, e o outro tem por objetivo detetar se um paciente apresenta indícios de diabetes, com dimensões conhecidas.
Numa primeira fase pretende-se estudar a ferramenta Orange e as funcionalidades disponíveis ao aplicar técnicas de seleção, redução e discretização de *features* e observar e retirar conclusões dos resultados obtidos. Na segunda fase, utiliza-se a ferramenta RStudio para novamente aplicar técnicas de seleção, redução e discretização aplicando técnicas de dados supervisionados (com classificadores) e não supervisionados (sem classificadores).

3. Estudo da ferramenta Orange

Neste capítulo aborda-se o uso da ferramenta Orange para efeitos de estudo dos exemplos fornecidos por padrão, explorando de que forma é feita a seleção de *features* e *ranking* das mesmas ao comparar resultados das pontuações dos diferentes métodos de seleção.

Na segunda parte foca-se a redução de *features* com a técnica de *Principal Component Analysis* (PCA), que projeta o *dataset* num espaço de dimensionalidade reduzida, e de seguida aplica-se o método de *Equal Frequency Binning* (EFB) para discretizar os novos dados em *bins* com a mesma frequência absoluta de amostras.

De notar que esta ferramenta apenas funciona bem em conjuntos de dados que não sejam considerados *Big Data*.

3.1 Ambiente Orange e análise de exemplos

3.1.1 Exemplo: *File and Data Table*

Neste exemplo pretende-se estudar o resumo estatístico do *dataset* 'iris' através dos *Widgets* disponíveis no Orange. O *workflow* deste exemplo é demonstrado na Figura 1 abaixo:

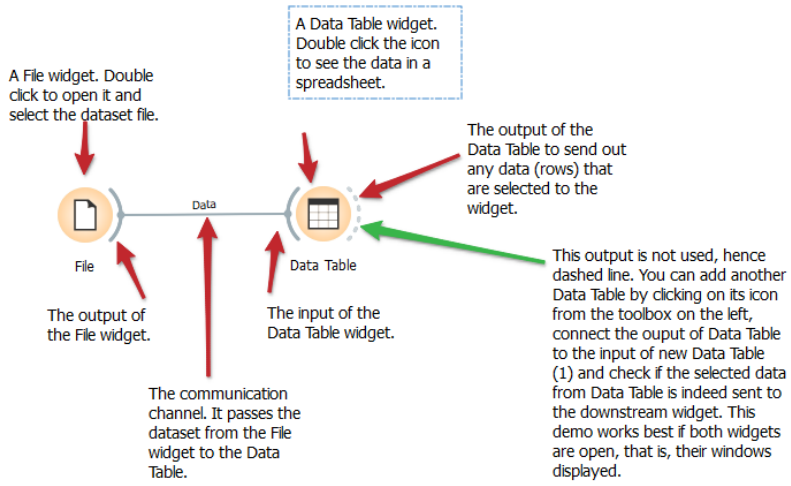

Figura 1 - Workflow do exemplo "File and Data Table"

O *Widget Feature Statistics* devolve-nos um resumo estatísticos das diferentes *features* do *dataset*, identificando a média, moda, mediana, dispersão, valores máximos e mínimos, e a quantidade de dados em falta para cada *feature*.

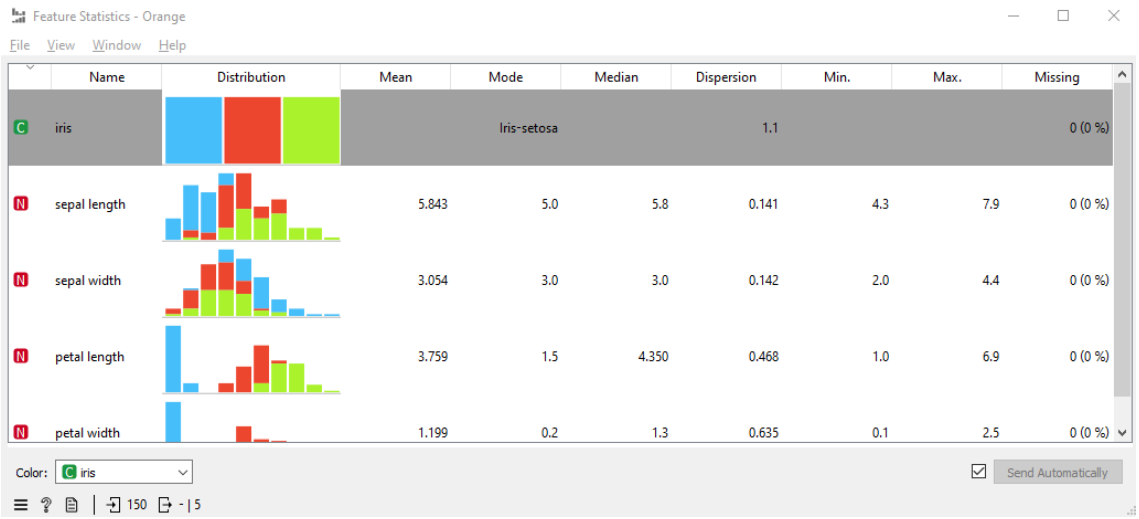

Figura 2 - Resumo estatístico das features do dataset "iris"

O resumo estatístico das *features* permite compreender melhor o tipo de dados com que estamos a trabalhar, nomeadamente se é categórico ou numérico contínuo/discreto; se é supervisionado ou não, isto é, tem uma etiqueta de classe (*class label*); quantos valores existem em falta.

Este conjunto de informação, permite inferir *à priori* que abordagens devem ser tomadas primeiro, e que tipo de técnicas de redução de dimensionalidade devem ser aplicadas, FS ou FR.

Para o *widget Data Table* são suportados diversos formatos de *datasets*, desde ficheiros *Excel* até valores separados por tabulações, como demonstrado na Figura 3.

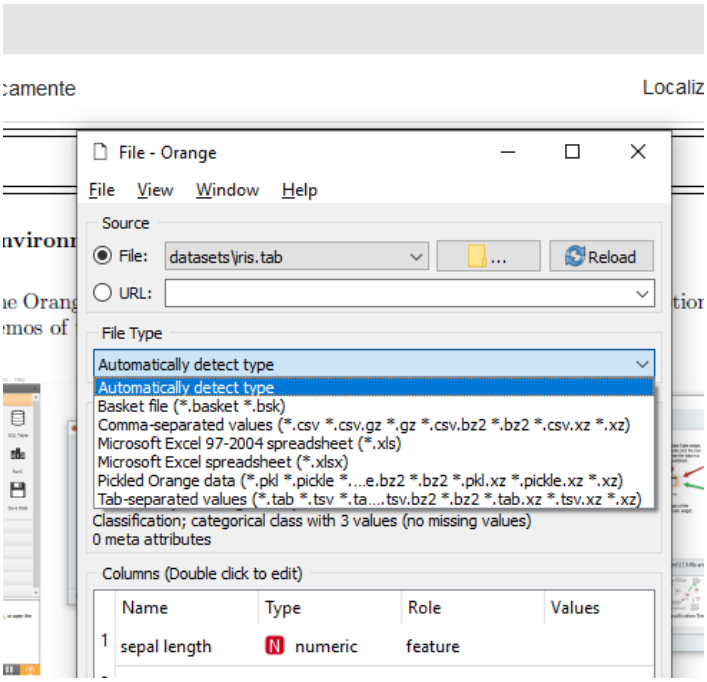

Figura 3 - Formatos suportados pelo widget Data Table

3.1.2 Exemplo: *Interactive Visualizations*

O exemplo *Interactive Visualizations* utiliza também o conjunto de dados "iris", e foi construído o *workflow* por forma a podermos observar o gráfico de dispersão das várias *features*, e consequentemente representar as que pretendemos numa tabela.

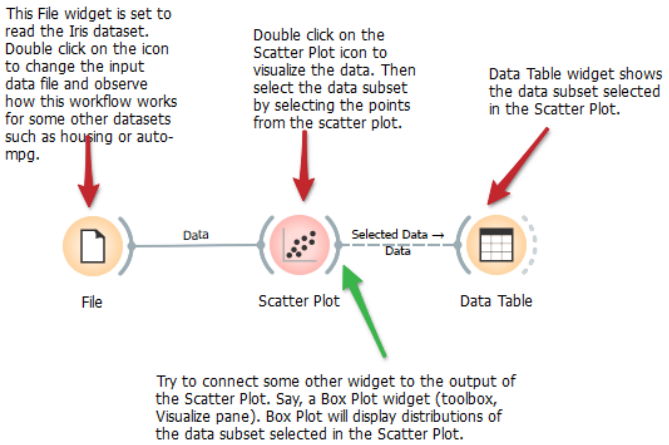

Figura 4 - Workflow do exemplo Interactive Visualizations

O *widget* do *Scatter Plot* mostra-nos um gráfico de dispersão entre duas *features*. Sendo um *dataset* com *class* labels deste gráfico é possível inferir algum tipo de correlação ou relacionamento entre as *features* de diferentes classes, seja ele linear ou não linear; bem como identificar *outliers*.Na ferramenta Orange existe uma opção de procurar projeções consideradas informativas, isto é, projeções que permitam ao utilizador compreender o comportamento das duas *features* de forma mais concreta, identificando os padrões ou correlacionamento.

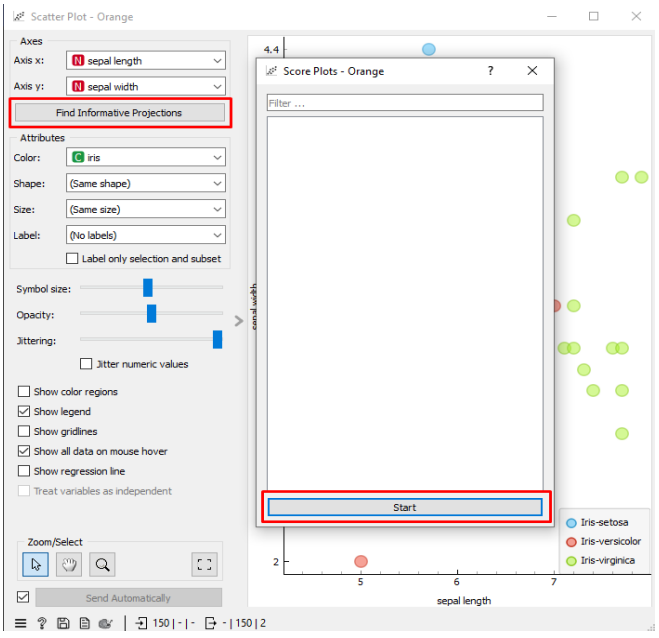

Figura 5 - Widget Scatter Plot e projeção de features

A melhor projeção é obtida das *features* "*petal.length"* e *"petal.width"*

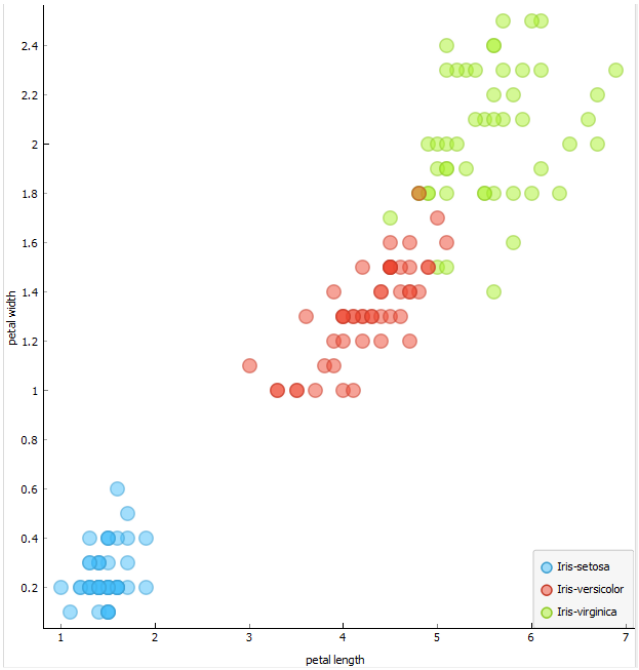

Figura 6 - Projeção das features "petal.length" e "petal.width"

Por observação, podemos dizer que esta é considerada a melhor projeção dado que existe a maior distância entre as classes, representadas cada uma com cores diferentes, e parece existir uma relação entre "petal.width" e "petal.length" consoante a classe, desta forma, treinar um classificador com estas *features* facilitaria o processo classificar novos dados com base nestas duas características.

Ao adicionar o *widget Data Info* após selecionar todos os pontos do gráfico de dispersão acima, podemos obter a informação do conjunto de dados selecionado, nomeadamente a dimensão, conjunto de *features* e meta atributos.

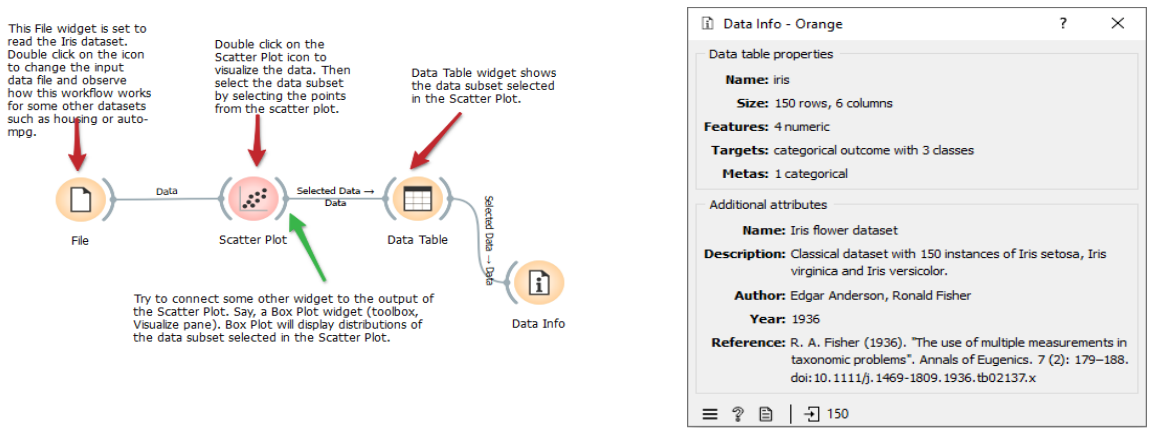

Figura 7 - Novo Workflow, e dados do widget Data Info

3.1.3 FS – Exemplo: *Feature Ranking*

Neste exemplo estudam-se as técnicas de *Feature Selection* com base na pontuação em diferentes critérios.

O *workflow* é constituído por um *widget* responsável por imputar valores em dados que estejam em falta no *dataset*. O método padrão de imputação deste *widget* é a média, para atributos contínuos, ou o valor mais frequente para atributos discretos. Este método de imputação pode ter consequências nos métodos de classificação para *Feature Selection*, nomeadamente introduzir *bias* nos dados e consequentemente piorar o desempenho do classificador. No entanto, a forma como é feita a imputação dos dados por agora não é pertinente.

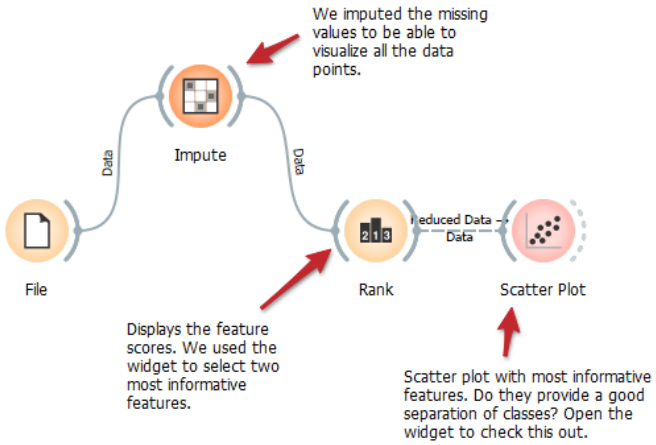

Figura 8 - Workflow do exemplo Feature Ranking

De seguida, o *widget Rank* classifica os dados com diferentes métricas:

Scoring Methods
- Information Gain
- Information Gain Ratio
- Gini Decrease
- ANOVA
- χ^2
- ReliefF
- FCBF

	#	Info. gain	Gain ratio	Gini	ANOVA	χ^2	ReliefF	FCBF
1	diau f	0.748	0.374	0.268	573.676	101.287	0.365	0.838
2	spo- early	0.720	0.360	0.276	422.389	92.854	0.379	0.782
3	diau g	0.714	0.357	0.259	595.005	94.962	0.353	0.771
4	heat 20	0.692	0.346	0.252	512.213	99.608	0.313	0.729
5	spo5 11	0.688	0.344	0.257	426.438	94.251	0.406	0.721
6	spo 2	0.672	0.336	0.256	227.500	92.258	0.308	0.693
7	Elu 0	0.669	0.334	0.257	173.768	94.272	0.255	0.688

Figura 9 - Classificação das features para diferentes métodos

Ao observar os resultados obtidos, podemos inferir que a mesma *feature* não é classificada sempre da mesma forma para os diferentes métodos, pelo que podemos concluir que é necessário ser feito o estudo de diferentes métodos de seleção de *features* por forma a poder chegar a um consenso entre os melhores atributos.

Ao alterar a ordenação por método, podemos deduzir que o atributo "diau f" está consistentemente nos melhores resultados para grande parte dos métodos, pelo que podemos deduzir que será a *feature* mais relevante do *dataset*.

Semelhante ao exemplo anterior, podemos também utilizar o *widget* do *Scatter Plot* para indicar-nos as melhores projeções entre os atributos, de onde obtemos a mais bem classificada:

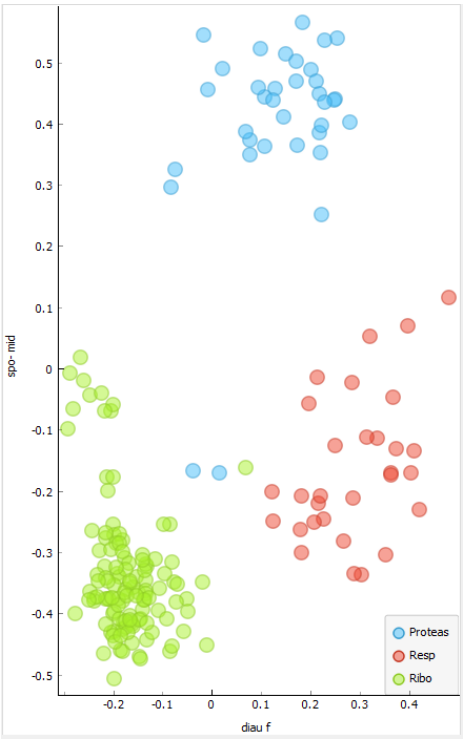

Figura 11 - Melhor projeção de features (X: diau f, Y: spo-mid)

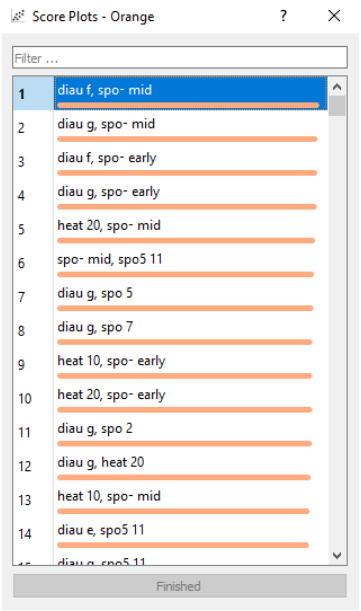

Figura 10 - Pontuação das melhores projeções

Evidentemente, pelas pontuações das melhores projeções, podemos novamente deduzir que a *feature* "diau f" será a mais relevante neste conjunto de dados visto que é a aparece mais vezes no topo da classificação.

A Figura 11 mostra a melhor projeção, obtida pelas *features* "diau f" e "spo-mid", onde vemos um forte afastamento entre as classes, com poucos *outliers*.

3.1.4 FR – Exemplo: *Principal Component Analysis*

O último exemplo da ferramenta Orange aborda o tópico de *Feature Reduction* através do método PCA.

Este método é utilizado para reduzir grandes conjuntos de dados ao aplicar uma transformação das *features* num espaço de dimensionalidade menor. No entanto, é importante mencionar que ao contrário dos métodos de *Feature Selection*, o PCA altera os valores dos atributos do conjunto de dados original, tentando preservar ao máximo a variância original em cada uma das componentes. Começa por calcular a matriz de covariância para identificar correlações entre atributos e, de seguida, calcula os vetores e valores próprios da matriz de covariância para identificar as componentes principais dos dados.

Este método é tipicamente utilizado em conjuntos de dados onde existe forte correlação entre as *features* e reduz a dimensionalidade do mesmo, preservando o máximo de informação possível em cada componente.

O *workflow* deste exemplo é descrito na Figura 12:

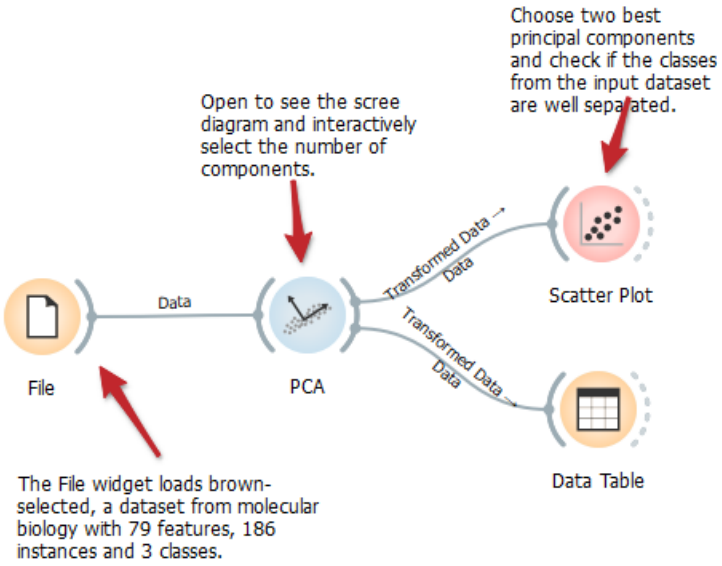

Figura 12 - Workflow do exemplo Principal Component Analysis

O *widget PCA* permite-nos observar as componentes obtidas de aplicar o método ao *dataset*. Neste gráfico é mostrada a contribuição de cada componente para a variância dos dados, e a soma acumulada da variância das componentes. Num cenário típico procura-se uma preservação de 80-90% da variância dos dados originais. Desta forma, por análise à Figura 13 podemos determinar que é possível reduzir o conjunto de dados original em 25 componentes.

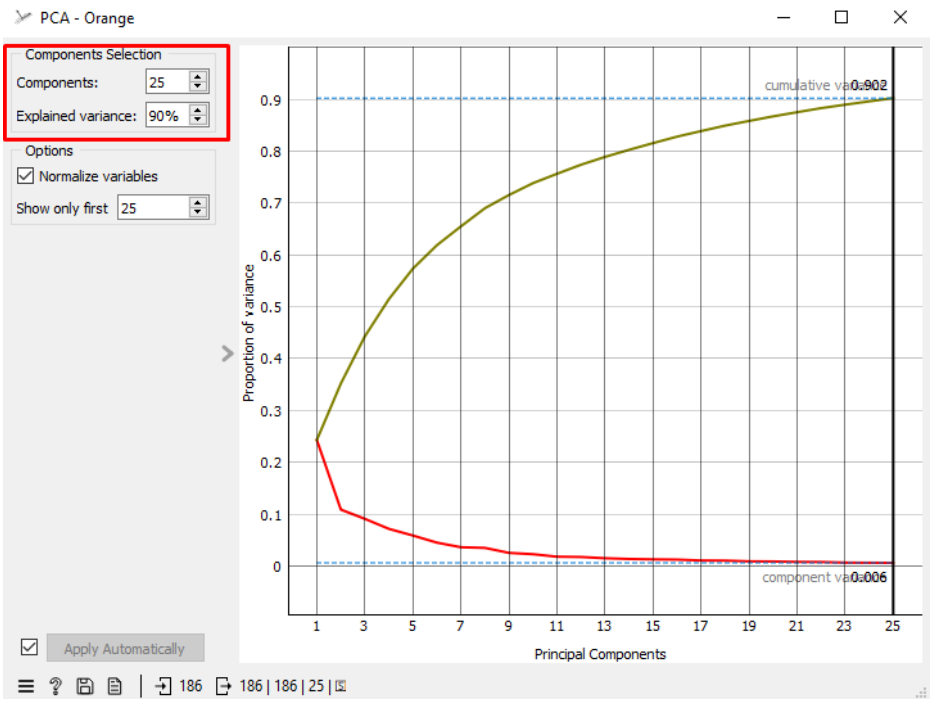

Figura 13 - Resultado do widget PCA

Assim sendo, o conjunto de dados transformado foi reduzido de 81 *features* para 25 componentes, como demonstrado na Figura 14.

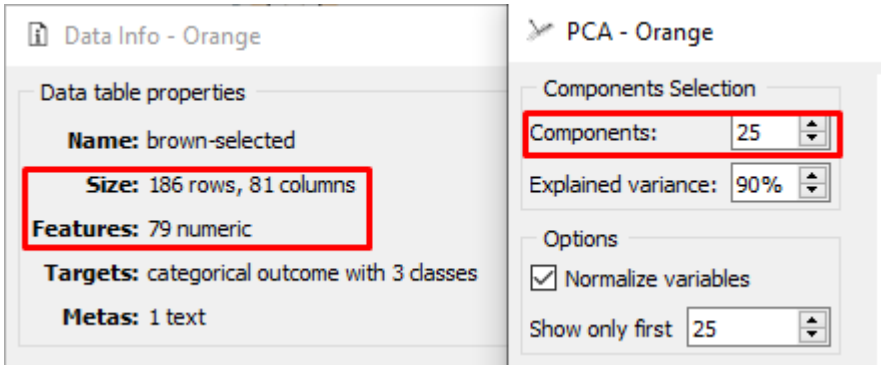

Figura 14 - Redução do dataset original para 25 componentes

Ao observar o gráfico de dispersão, a projeção com melhor pontuação é entre a primeira e terceira componente (PC1 e PC3), onde podemos observar um forte afastamento entre as classes e poucos *outliers* (figura).

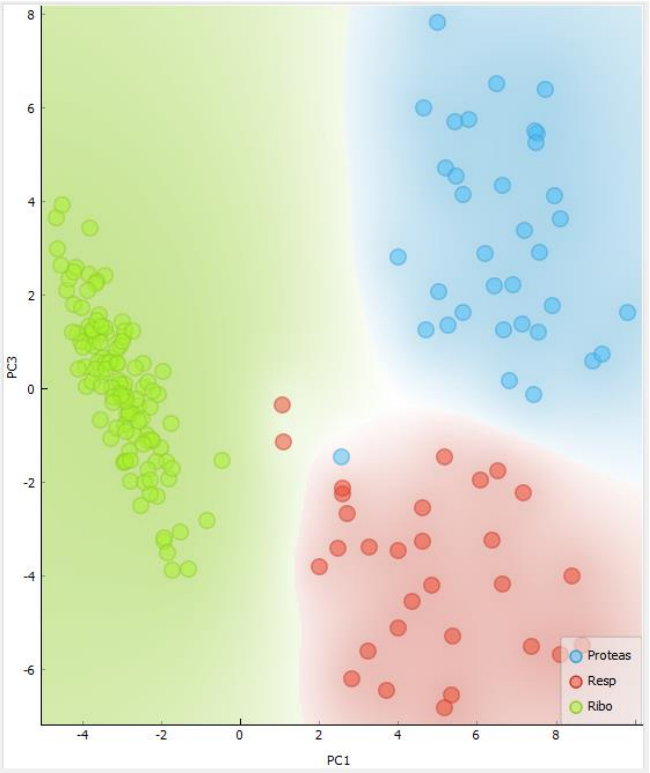

Figura 15 - Gráfico de dispersão da melhor projeção do PCA (X: PC1, Y:PC3)

Reduzido o *dataset*, podemos então aplicar uma técnica de *Feature Discretization* conhecida por *Equal Frequency Binning*, que, tal como mencionada anteriormente, procura construir *bins* de forma que o número de amostras por cada um seja igual para todos. Este é considerado um método não supervisionado, onde é feita uma quantização não uniforme.
pelo que é necessário aplicar a seguinte alteração ao *workflow*:

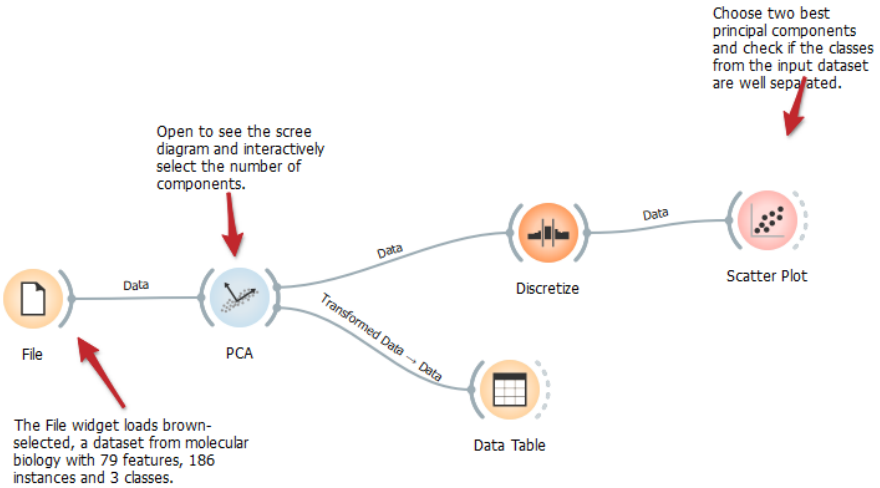

Figura 16 - Alteração ao workflow para realizar FD com o método EFB

Definiu-se uma frequência absoluta de 10 *bins*, e observando novamente a melhor projeção:

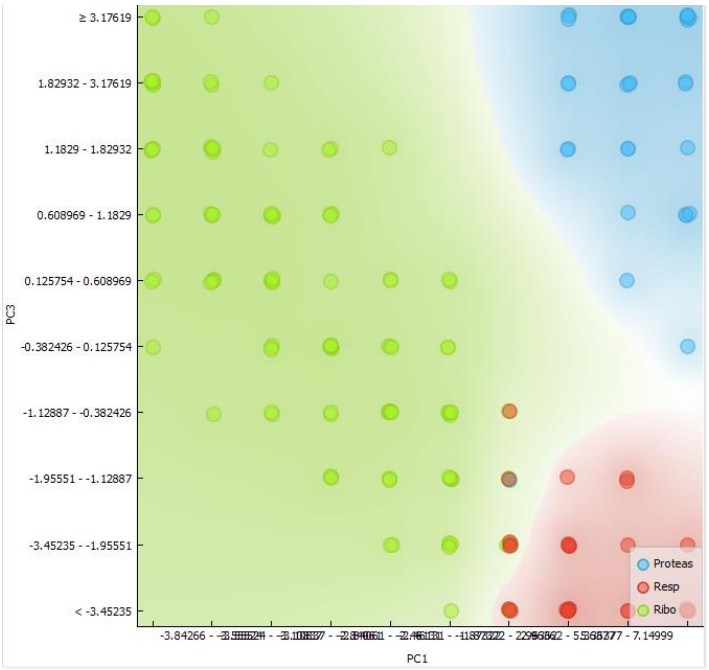

Figura 17 - Discretização da melhor projeção de componentes (PC1 vs PC3)

Com os dados discretizados, torna-se mais simples para o classificador treinar e produzir melhores resultados.

4. RStudio

Neste capítulo pretende-se fazer o estudo dos métodos de redução de dimensionalidade, tanto *Feature Selection* como *Feature Reduction*, e depois aplicar métodos de discretização supervisionados e não supervisionados. Os conjuntos de dados a utilizar são os mencionados anteriormente; condições metereológicas em Lisboa, e deteção de indícios de diabetes (pima). Após redução de dimensionalidade, é estudada a discretização, utilizando um método supervisionado e outro não supervisionado, à escolha do grupo.

4.1 Feature Selection

Pretende-se com a *Feature Selection* calcular a relevância de cada atributo, de ambos os *datasets* com base na variância e na média-mediana.

A variância indica o quão espalhados estão os conjuntos de valores para cada atributo, e a média-mediana (MM) indica a relevância por assimetria, isto é, quanto maior a distância entre a média e a mediana, mais relevante é a feature.

Elaborou-se o *script R*, entregue em anexo, que permitiu traçar as *features* não normalizada, ordenadas de mais relevante para menos relevante, representado na Figura 18.

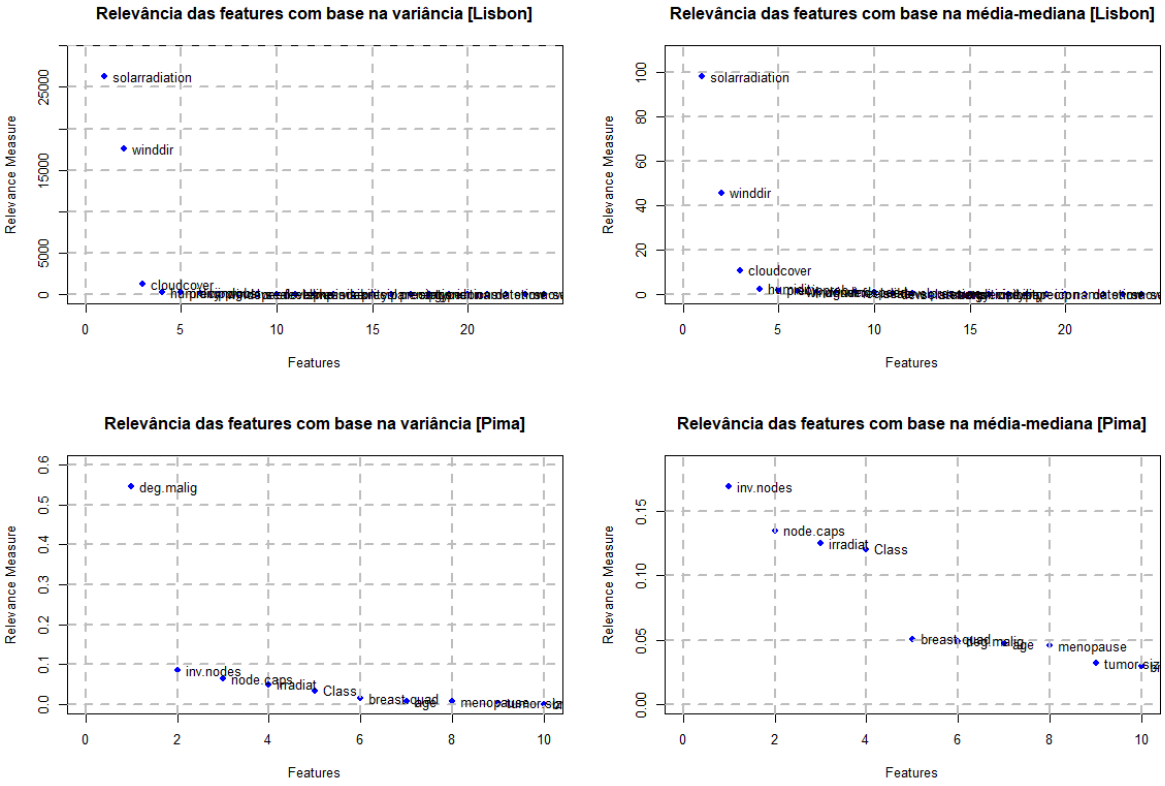

Figura 18 - Ordenação das features por relevância (variância e média-mediana)

Ao observar os resultados obtidos, no conjunto de dados de Lisboa podemos inferir que com base nas duas métricas mencionadas, as *features* mais relevantes são, respetivamente, *"solarradiation", "winddir", "cloudcover"*, isto é, são os atributos que mais contribuem para a informação do *dataset*.

Para o *dataset* "pima" obtêm-se resultados distintos. Enquanto que na relevância calculada pela variância a *feature* mais relevante é "*deg.malig"*, o mesmo não se observa quando utilizamos a métrica da média-mediana. Em simultâneo, a *feature* "*inv.nodes"* aparece em primeiro e segundo lugar, das métricas, pelo que podemos dizer que é a mais relevante segundo estes critérios.

Pretende-se agora, com estas medidas de relevância, fazer a redução de dimensionalidade, pelo que é necessário inferir quantos atributos devem ser selecionados, com base nos resultados obtidos, para diferentes níveis de confiança.

Estabeleceram-se então *thresholds* de 75%, 85% e 90%, para comparar o número de *features* necessárias.

Para o *dataset* de Lisboa, obtemos o seguinte número de *features* necessárias para ambas as métricas:

- **Medida de relevância:** Variância

	Threshold [%]	Features adequadas
1	75 %	2
2	85 %	2
3	95 %	2

```
For values of variances threshold 75 % we need features: solarradiation, winddir
For values of variances threshold 85 % we need features: solarradiation, winddir
For values of variances threshold 95 % we need features: solarradiation, winddir
```

Figura 19 - Features adequadas para o dataset lisboa com base na variância

- **Medida de Relevância:** Média-Mediana

	Threshold [%]	Features adequadas
1	75 %	2
2	85 %	2
3	95 %	4

```
For values of mean-median threshold 75 %  we need features: solarradiation, winddir
For values of mean-median threshold 85 %  we need features: solarradiation, winddir
For values of mean-median threshold 95 %  we need features: solarradiation, winddir, cloudcover, humidity
```

Figura 20 - Features adequadas para o dataset lisboa com base na média-mediana

Para o *dataset* Pima obtemos:

- **Medida de relevância:** Variância

	Threshold [%]	Features adequadas
1	75 %	2
2	85 %	3
3	95 %	5

```
For values of variances threshold 75 %  we need features: deg.malig, inv.nodes
For values of variances threshold 85 %  we need features: deg.malig, inv.nodes, node.caps
For values of variances threshold 95 %  we need features: deg.malig, inv.nodes, node.caps, irradiat, Class
```

Figura 21 - Features adequadas para o dataset pima com base na variância

- **Medida de relevância:** Média-Mediana

	Threshold [%]	Features adequadas
1	75 %	6
2	85 %	7
3	95 %	9

```
For values of mean-median threshold 75 %  we need features: inv.nodes, node.caps, irradiat, Class, breast.quad, deg.malig
For values of mean-median threshold 85 %  we need features: inv.nodes, node.caps, irradiat, Class, breast.quad, deg.malig, age
For values of mean-median threshold 95 %  we need features: inv.nodes, node.caps, irradiat, Class, breast.quad, deg.malig, age,
menopause, tumor.size
```

Figura 22 - Features adequadas para o dataset pima com base na média-mediana

Por observação destes resultados, podemos concluir que diferentes métricas produzem resultados diferentes, desta forma, é pertinente testar diferentes métodos para que possamos ter uma análise mais robusta e melhor compreensão dos dados.

Ambas as métricas mencionadas anteriormente tratam de *datasets* onde os dados não são classificados, isto é, não possuem *class labels.* No entanto, seguindo o guia de laboratório, pretende-se aplicar um teste estatístico com base no *Fisher's Ratio*, que implica um conjunto de dados classificado. Desta forma, o grupo escolheu como *class label* no *dataset* de Lisboa a *feature* "*conditions*" uma vez que esta é categórica e representativa das condições metereológicas gerais aquando da recolha da informação, isto é, se está nublado, parcialmente nublado, chuvoso ou céu limpo.

Para o *dataset* "pima", o atributo escolhido como *class label* foi o "*Class"* por se tratar de uma *feature* binária, com dois valores possíveis: "*recurrence-events"* ou "*no-recurrence-events"*.

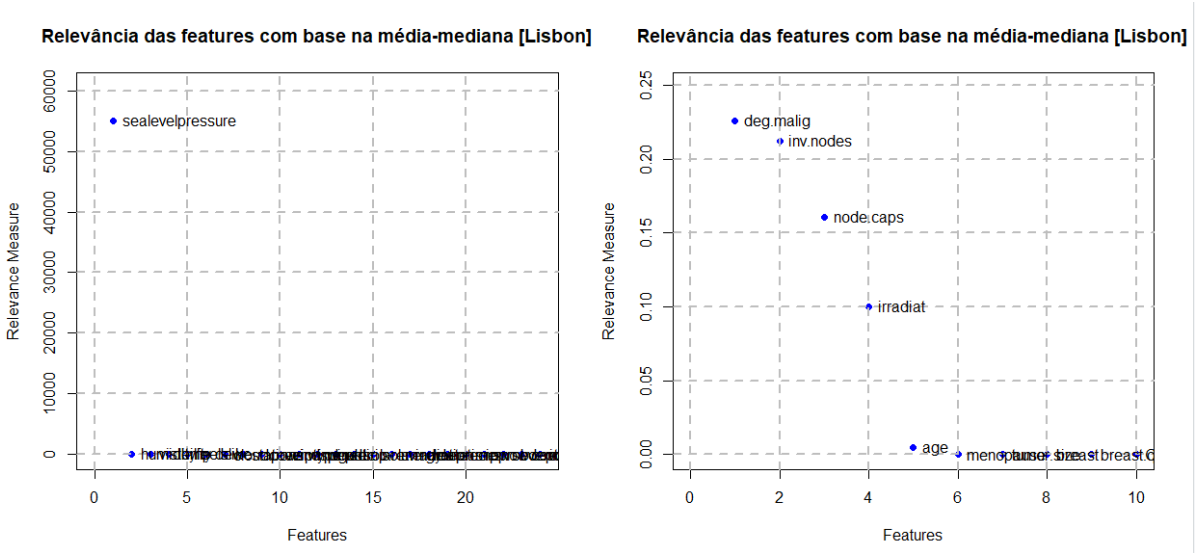

Figura 23 - Ordenação das features classificadas por ordem do Fisher's Ratio

Atendendo ao resultado obtido nas *features* mais relevantes pelo *Fisher's Ratio*, o grupo optou por testar também a *feature* "*icons"* contudo o resultado obtido foi o mesmo, apenas a *feature "sealevelpressure"* é considerada relevante para qualquer um dos *thresholds* definida anteriormente.

Já para o *dataset "*pima" os resultados obtidos são semelhantes aos anteriores quando não se utilizou *class label* no entanto, para o mesmo valor de relevância dos diferentes *thresholds* são necessárias menos *features*.

	Threshold [%]	Features adequadas
1	75 %	3
2	85 %	3
3	95 %	4

```
For values of FiRi threshold 75 %  we need features: deg.malig, inv.nodes, node.caps
For values of FiRi threshold 85 %  we need features: deg.malig, inv.nodes, node.caps
For values of FiRi threshold 95 %  we need features: deg.malig, inv.nodes, node.caps, irradiat
```

Figura 24 - Features adequadas para o dataset pima com base no Fisher's Ratio

Podemos concluir que para o *dataset "*pima", a atribuição de uma *class-label* produz uma maior redução de dimensionalidade, em que apenas são necessárias 4 *features* para preservar 95% da relevância do conjunto de dados original.

4.2 Feature Reduction

Tal como a *Feature Selection*, a *Feature Reduction* é um conjunto de etapas que visa reduzir a dimensionalidade do conjunto de dados originais, extraindo apenas informação relevante acerca do mesmo. No entanto, este tipo de métodos aplica uma transformação dos dados de um espaço com grande dimensão para um com representação em baixa dimensão, no entanto, mantendo algumas propriedades significativas com base na variância mas alterando os valores originais.

Dois dos métodos mais conhecidos para aplicar *Feature Reduction* são a *Principal Component Analysis* (PCA) e *Singular Value Decomposition* (SVD).

4.2.1 Decomposição PCA

A técnica de PCA tem por objetivo aplicar uma transformação linear nos dados originais com base na matriz de covariância dos dados, também chamada de matriz de rotação; nos valores próprios dessa matriz, e nos vetores próprios que permitem fazer a reconstrução de uma grande parte da variância do conjunto de dados original.

O PCA decompõe o *dataset* original em diferentes componentes perpendiculares entre si num espaço n-dimensional, e sabe-se que o quadrado dos valores próprios de cada componente representa a variância dos dados originais que preserva. Podemos observar a contribuição de cada componente na preservação da informação inicial, para o *dataset* de Lisboa e o pima, respetivamente, nas Figura 25 e Figura 26:

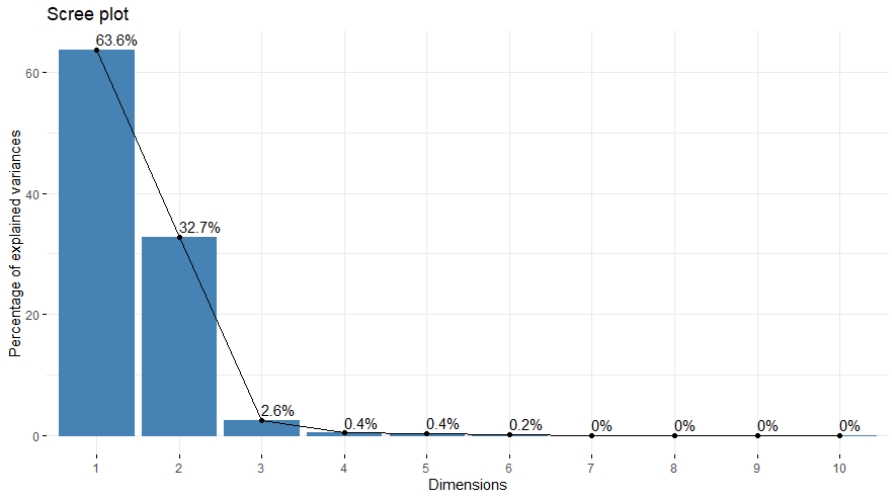

Figura 25 - Contribuição das componentes do PCA na variância, dataset Lisboa

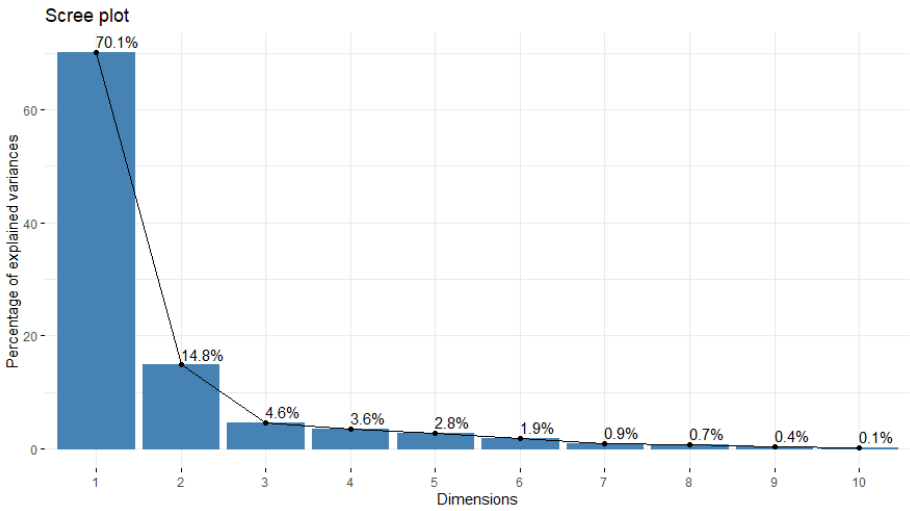

Figura 26 - Contribuição das componentes do PCA na variância, dataset pima

Podemos observar que para ambos os casos, a grande parte da variância dos dados originais está presente nas primeiras componentes, sendo estas as mais relevantes para a redução de dimensionalidade. Esta redução é então feita através do produto da matriz original com a matriz de 'n' componentes, que preservam o valor desejado 'y' da variância dos dados.

4.2.2 Decomposição SVD

A decomposição SVD permite desconstruir a matriz do *dataset* original numa composição de três matrizes, com os vetores próprios esquerdos e direitos, e a matriz diagonal. Semelhante ao PCA, os valores da matriz diagonal, quando elevados à potência de dois, permitem inferir a contribuição para a preservação da variância dos dados originais, pelo que podemos determinar quantos vetores da decomposição são relevantes para a redução de dimensionalidade. Durante a elaboração da decomposição, notou-se que os *datasets* estavam a ser reduzidos a apenas um vetor, o que indica que uma *feature* está a sobrepor demasiado a sua contribuição, pelo que foi necessário normalizar os valores das mesmas.

Aplicando a decomposição SVD, obtemos as seguintes contribuições de cada vetor:

Determinada a dimensão recomendada para a variância preservada, a nova matriz reduzida é dada pelo produto dos 'n' vetores das três matrizes resultantes da decomposição SVD. Isto é, para um exemplo onde temos 5 vetores relevantes, o resultado é dado pelo produto das 5 primeiras colunas de cada matriz:

$$X = UDV^T$$

Em que U e V são as matrizes de vetores próprios, esquerda e direita, respetivamente, e D a matriz diagonal.

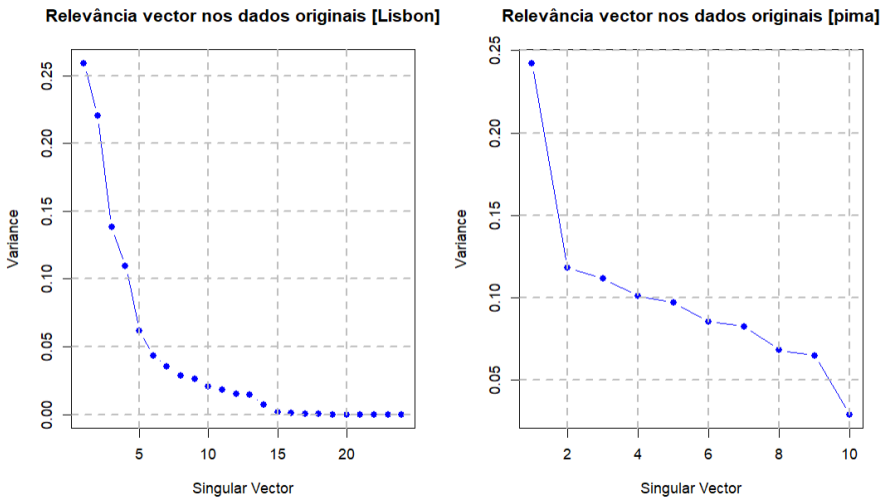

Figura 27 - Relevância de cada componente, decomposição SVD

4.2.3 Redução da dimensionalidade com PCA e SVD.

- **Redução com PCA:**

Tal como mencionado anteriormente, a redução é feita pelo produto das componentes relevantes com a matriz original. Da Figura 25 e Figura 26 concluímos que para preservar 90% da relevância dos dados originais para ambos os *datasets* são necessárias 2 e 4 componentes, para o *dataset* Lisboa e pima, respetivamente. Obtemos então dois novos *datasets* com dimensionalidade reduzida:

```
> summary(rd_pca_lisbon)
      PC1                PC2
 Min.   :-160.064   Min.   :-571.480
 1st Qu.:-129.293   1st Qu.:-295.457
 Median : -38.159   Median :-242.513
 Mean   :   1.905   Mean   :-215.284
 3rd Qu.:  43.240   3rd Qu.:-118.917
 Max.   : 506.686   Max.   :   5.522
> summary(rd_pca_pima)
      PC1               PC2              PC3               PC4
 Min.   :-2.8649   Min.   :0.3765   Min.   :-0.1731   Min.   :0.1311
 1st Qu.:-2.6063   1st Qu.:1.3172   1st Qu.: 0.3114   1st Qu.:0.3786
 Median :-1.6661   Median :1.5434   Median : 0.3252   Median :0.5750
 Mean   :-1.7415   Mean   :1.5168   Mean   : 0.2884   Mean   :0.5112
 3rd Qu.:-1.6287   3rd Qu.:1.7606   3rd Qu.: 0.3382   3rd Qu.:0.6197
 Max.   :-0.6537   Max.   :1.9848   Max.   : 0.6668   Max.   :0.8333
> dim(rd_pca_lisbon)
[1] 744   2
> dim(rd_pca_pima)
[1] 286   4
```

Figura 28 - Redução de dimensionalidade com PCA

- **Redução com SVD:**

Na decomposição SVD, reitera-se que a matriz com a dimensionalidade reduzida é dada pela reconstrução de X, com apenas as 'n' componentes consideradas relevantes para cada matriz de vetores próprios, esquerda e direita, e a matriz diagonal D.

Observando as matrizes de correlação, obtemos um novo *dataset* transformado com forte correlação entre as componentes:

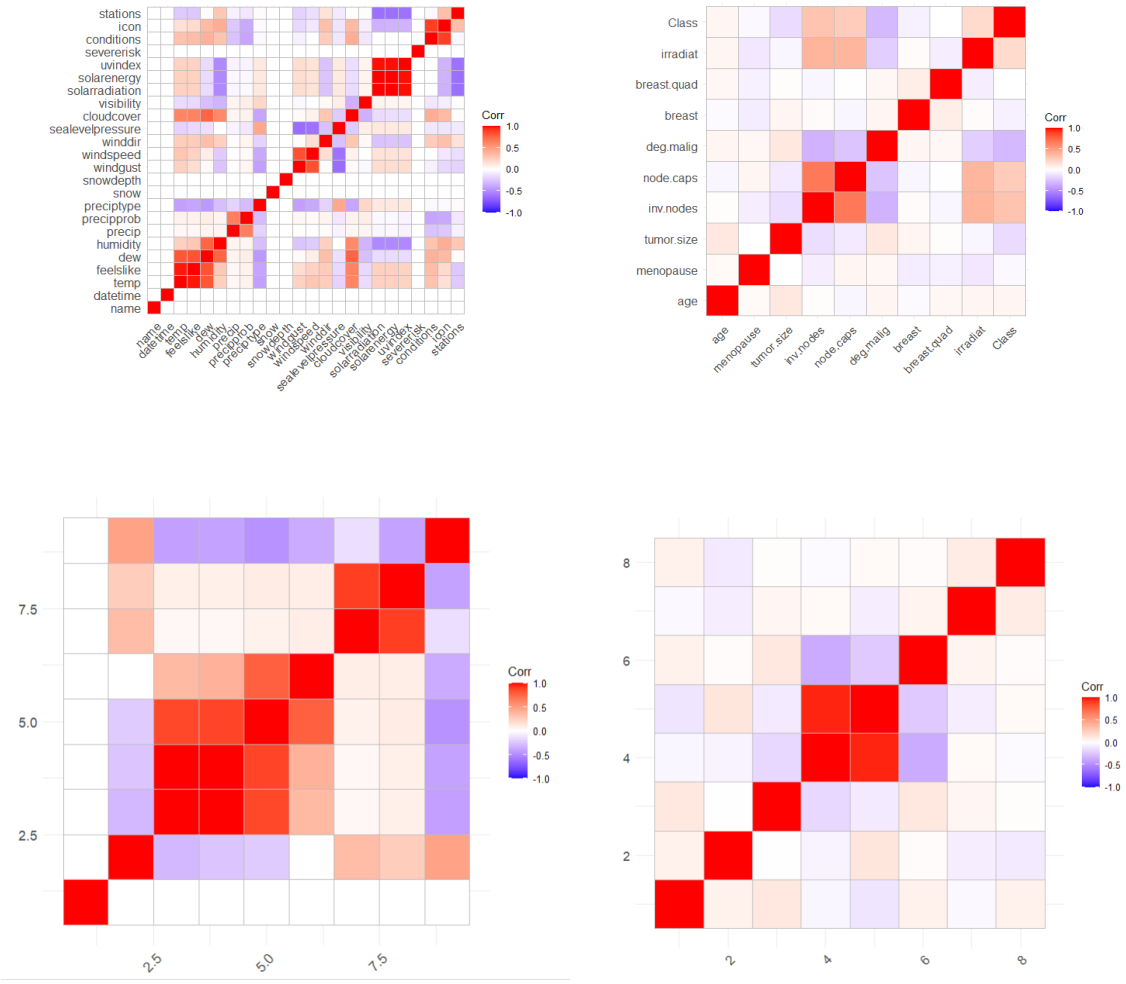

Figura 29 - Transformação SVD do dataset Lisboa e Pima, respetivamente

Podemos observar que são aproximadamente mantidas as mesma correlações entre os atributos originais e as respetivas matrizes reduzidas, pelo que a redução de dimensionalidade é pertinente para preparar os dados que posteriormente sirvam para o treino de modelos de *machine learning*.

4.3 *Feature Discretization*

A *Feature Discretization* é uma técnica de pré-processamento e torna possível reduzir a complexidade dos dados. Os dados numéricos são divididos em grupos (*bins*) o que reduz a complexidade e retira vários problemas associados à alimentação dos dados a sistemas de aprendizagem automática, por exemplo problemas de *overfitting*.

Esta técnica também tem a desvantagem de reduzir a informação presente nos dados. Como os dados são agrupados, deixa de ser guardado cada valor individual e apenas é considerado o grupo a que o valor está associado.

4.3.1 Discretização não supervisionada

O método de discretização não supervionada escolhido foi o Equal Frequency Binning (EFB) e foi efetuado no *dataset* de Lisboa. Este método tem como objetivo efetuar a discretização dos dados mantendo um número semelhante de ocorrências em cada um dos bins criados. Para o uso deste método existe a necessidade de definir o número de *bins* para os dados que vão ser discretizados o que pode levar a *bins* a mais ou *bins* a menos que influencia a eficiência do algoritmo e a utilidade dos dados resultantes.

```
> head(data_lisbon$temp, 20)
 [1] 17.0 17.0 16.2 17.1 17.2 17.0 17.1 17.2 17.2 17.1 18.0 18.2 18.1 18.0 18.2 15.0 15.2 15.2 14.1 13.2
> head(unsupervised_data_lisbon$temp, 20)
 [1] (14.2,18.2] (14.2,18.2] (14.2,18.2] (14.2,18.2] (14.2,18.2] (14.2,18.2] (14.2,18.2] (14.2,18.2] (14.2,18.2]
[10] (14.2,18.2] (14.2,18.2] (14.2,18.2] (14.2,18.2] (14.2,18.2] (14.2,18.2] (14.2,18.2] (14.2,18.2] (14.2,18.2]
[19] (13,14.2]   (13,14.2]
Levels: (-Inf,7.48] (7.48,9.2] (9.2,11.2] (11.2,13] (13,14.2] (14.2,18.2]
```

Figura 30 - Comparação entre os dados originais e os dados discretizados da coluna de temperatura com discretização não supervisionada

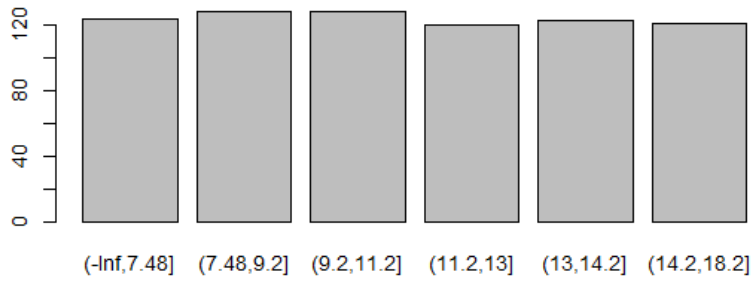

Figura 31 - Plot dos dados de temperatura discretizados com EFB

Observando a Figura 32, é possível observar que as ocorrências nos vários bins encontra-se equilibrado devido ao uso do EFB.

4.3.2 Discretização supervisionada

O método de discretização supervisionada escolhido foi o Class-Attribute Interdependence Maximization (CAIM) e foi efetuado no *dataset* de Lisboa. O CAIM tem como objetivo efetuar a discretização dos dados com base na coluna da classe do *dataset* o que aumenta a relação entre os dados discretizados e a classe. O CAIM efetua automaticamente a separação dos dados num número de *bins* descoberto pelo próprio algoritmo.

```
> head(data_lisbon$temp, 20)
 [1] 17.0 17.0 16.2 17.1 17.2 17.0 17.1 17.2 17.2 17.1 18.0 18.2 18.1 18.0 18.2 15.0 15.2 15.2 14.1 13.2
> head(supervised_data_lisbon$temp, 20)
 [1] [15.7,18.2) [15.7,18.2) [15.7,18.2) [15.7,18.2) [15.7,18.2) [15.7,18.2) [15.7,18.2) [15.7,18.2) [15.7,18.2)
[10] [15.7,18.2) [15.7,18.2) [18.2, Inf] [15.7,18.2) [15.7,18.2) [18.2, Inf] [2.9,15.7)  [2.9,15.7)  [2.9,15.7)
[19] [2.9,15.7)  [2.9,15.7)
Levels: [-Inf,2.9) [2.9,15.7) [15.7,18.2) [18.2, Inf]
```

Figura 32 - Comparação entre os dados originais e os dados discretizados da coluna de temperatura com discretização supervisionada

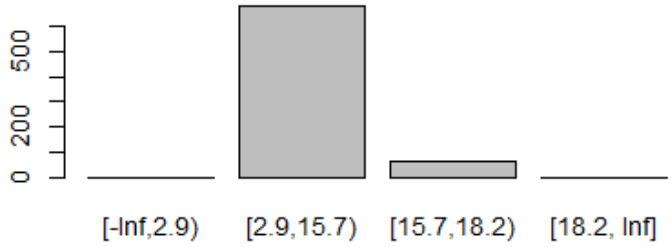

Figura 33 - Plot dos dados de temperatura discretizados com CAIM

Observando a Figura 34 e comparando com a Figura 32, verifica-se que as ocorrências de dados nos vários bins do CAIM não estão equilibradas e isto deve-se à forma de como é efetuada a discretização dos dados. Também se verifica que o número de *bins* na discretização do CAIM é muito inferior ao número de *bins* no EFB.

5. Conclusão

A elaboração do segundo laboratório tinha por objetivo compreender e aplicar os conceitos de *Feature Selection, Feature Reduction* e *Feature Discretization* por forma a combater os problemas causados por conjuntos de dados de elevada dimensionalidade, que dificultam o treino de modelos de *machine learning*.

Este problema causado pela dimensionalidade, apelidade de *Curse of Dimensionality* (maldição da dimensionalidade) comprova que à medida que o conjunto de *features* do *dataset* aumenta, a quantidade de dados necessários para treino de modelos robustos aumenta exponencialmente.

Com os objetivos de trabalho definidos, começou-se por revisitar e analisar conjuntos de dados de dimensionalidade mais reduzida com a ferramenta *Orange*, previamente utilizada na unidade curricular de Aprendizagem e Mineração de Dados (AMD). Observou-se de forma mais didática os valores obtidos das diferentes métricas para classificação de *features*, e as melhores relações entre atributos no caso da *Feature Selection.* Para a *Feature Reduction* compreendeu-se os objetivos traçados pela técnica PCA, ao reduzir um espaço n-dimensional em componentes com m-dimensões, com $m < n$. Por fim, observou-se a influência da discretização nas várias características do *dataset.*

Atendendo às limitações implicadas pela infraestrutura em que opera esta ferramenta *Orange*, para os conjuntos de dados fornecidos em anexo ao enunciado, que apresentam uma dimensão superior à suportada pela aplicação, seguiu-se para uma implementação em R.

Nesta segunda parte, desenvolveram-se um conjunto de *scripts* baseados na linguagem R para mais uma vez verificar as técnicas de *Feature Selection* supervisionadas e não supervisionadas, como a Média-Mediana, e o *Fisher's Ratio*, respetivamente, e consequentemente reduzir os conjuntos de dados originais em *features* bem selecionadas, que preservem um determinado *threshold* da métrica em avaliação.

Segue-se a *Feature Reduction* onde se abordaram as técnicas PCA e SVD, e chegou-se à conclusão da forte influência de *features* com níveis de grandeza distintas, e a necessidade de fazer o *scaling* para obter resultados pertinentes.

Para terminar estudaram-se e compreenderam-se as técnicas de discretização *Equal Frequency Binning* para combater eventual ruído que possa acontecer devido a variáveis contínuas aquando do treino dos modelos.

Desta forma, o grupo considera os objetivos do laboratório cumpridos, compreendendo a importância da redução ou transformação de *features* para combater a maldição da dimensionalidade e treinar modelos mais robustos.

Este laboratório serve também de base para seguir para a primeira fase do trabalho prático,

6. Referências

[1] https://datascience.stackexchange.com/a/23860, 1/04/2024, 13:30