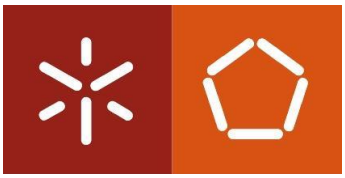


Universidade do Minho

Departamento de Informática

Trabalho Prático
Aprendizagem Automática I

Filipe Monteiro (a80229) Leonardo Silva (pg39261)

13 de Janeiro de 2020

Conteúdo

1 Introdução

Neste projeto é pretendido que, à escolha dos alunos, seja analisado um *dataset* usando as diferentes metodologias lecionadas ao longo do semestre e no fim concluir sobre esta análise e seus resultados. O *dataset* selecionado e analisado foi retirado da plataforma Kaggle - um local de partilhas dos mesmos para fins educacionais.

1.1 Caso de Estudo

O *dataset* escolhido para este projeto foi o Breast Cancer Wisconsin, um *dataset* sobre amostras recolhidas de massa mamária com o intuito de detetar se a existência de cancro será Maligna ou Benigna - um problema de **classificação**. Estes dados contêm características do núcleo das diferentes células apresentadas nas imagens. Em termos de dimensão, este é composto por 33 variáveis independentes (*features*) e são ao todo 569 observações.

Apesar da presença de 33 variáveis independentes, na realidade são apenas 10 características derivadas em diferentes formas. São então as características bases a seguir:

- *Radius*
- *Texture*
- *Perimeter*
- *Area*
- *Smoothness*
- *Compactness*
- *Concavity*
- *Concave points*
- *Symmetry*
- *Fractal dimension*

Adicionando ainda o ID, o *Diagnosis* (diagnóstico atribuído) e uma coluna X a qual provavelmente veio por engano, pois não possui valores.

Nas então 10 características base, 3 diferentes análises foram executas:

- A média do atributo X das células analisadas (*radius_mean, texture_mean, perimeter_mean, ...*)
- O *standard error* calculado da mesma forma anteriormente exposta (*radius_se, texture_se, perimeter_se, ...*)
- O pior/média dos 3 maiores valores para o atributo X (*radius_worst, texture_worst, permiter_worst, ...*)

Neste projeto iremos tentar criar um modelo capaz de classificar corretamente, com grande precisão, a variável objetivo: *diagnosis*.

2 Metodologia

Para a análise deste *dataset* iremos utilizar gráficos para a interpretação dos dados, realizar limpeza no *dataset* caso seja necessário e realizar várias técnicas de seleção das caraterísticas (*feature selection*) para utilizar no nosso modelo de regressão logística. Entre as diferentes formas de seleção das variáveis independentes, iremos fazer análises com base nos gráficos e correlações entre estas e iremos também usar *subset selection*, métodos de redução da variância e técnicas de redução da dimensão. Por fim, iremos testar os diferentes modelos criados utilizando *k-cross validation* com o intuito de descobrir o que melhor resultados apresentou.

3 Resultados

3.1 Análise dos Dados

Primeiramente, após inspecionar todo o *dataset*, verificámos que havia 1 variável com valores em falta, em todos as observações. Por isso, retirámos esta, juntamente com o *ID*, pois este também não será necessário.

Como explicado anteriormente, este *dataset* possui 3 categorias diferentes para as mesmas variáveis: *mean*, *SE* e *worst*. Devido a isto, começamos por fazer *boxplot* para cada uma das categorias (para ficar menos confusa a análise) tentando encontrar fronteiras nos dados para usar na classificação.

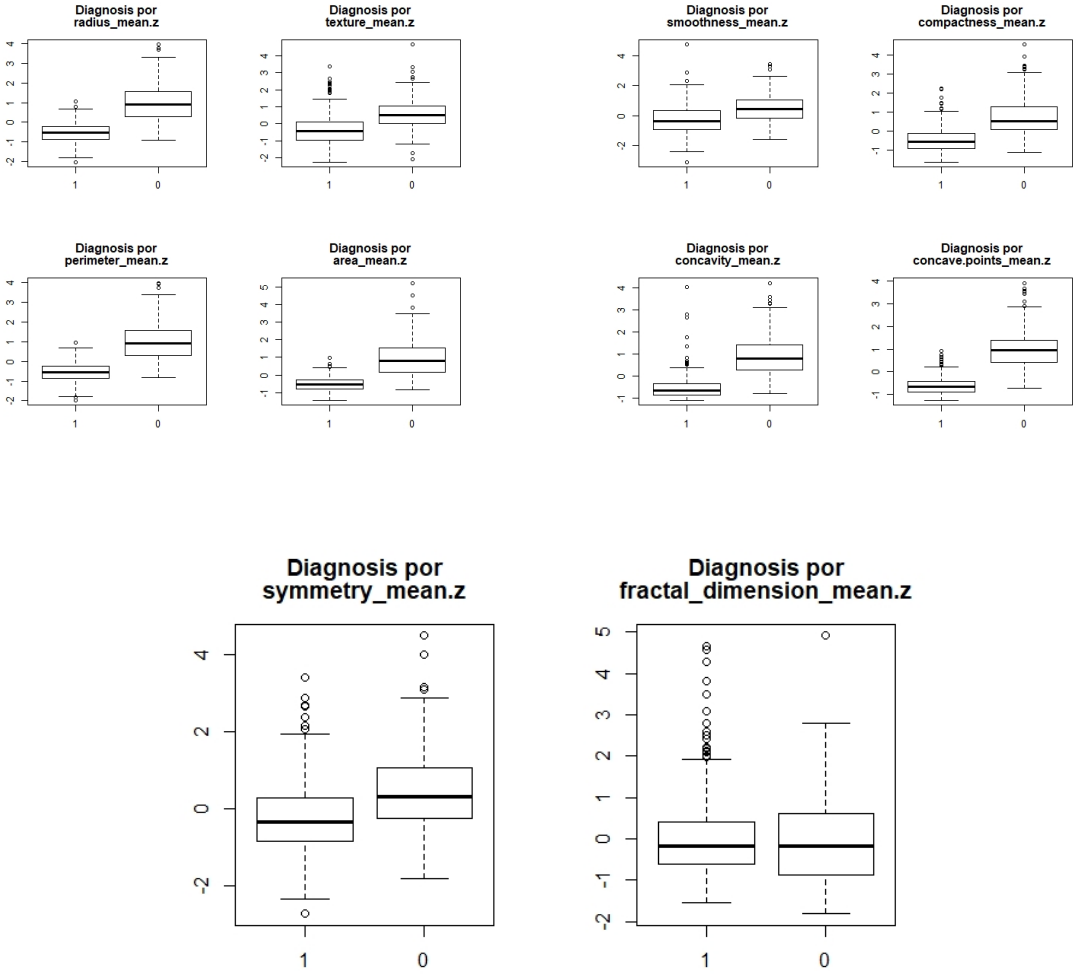


Figura 2: Categoria *mean*.

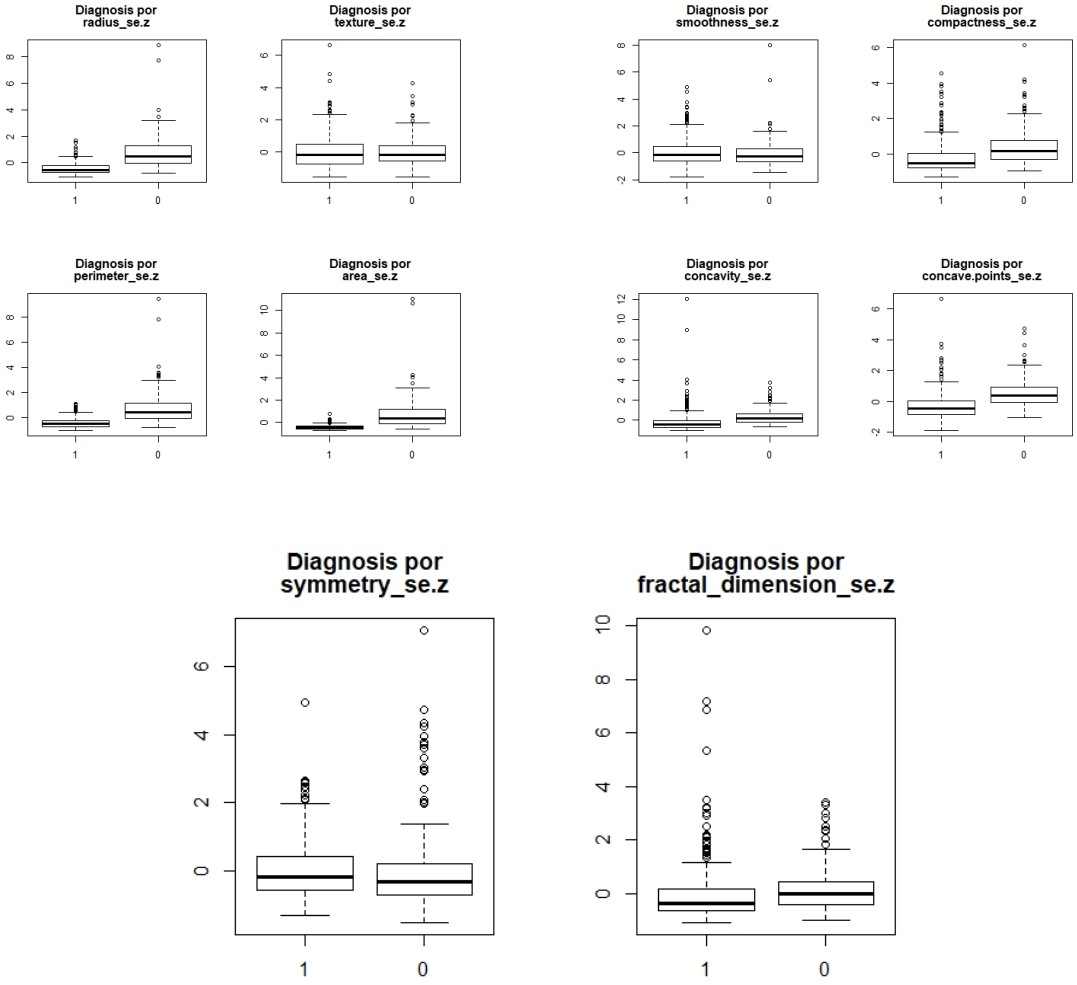
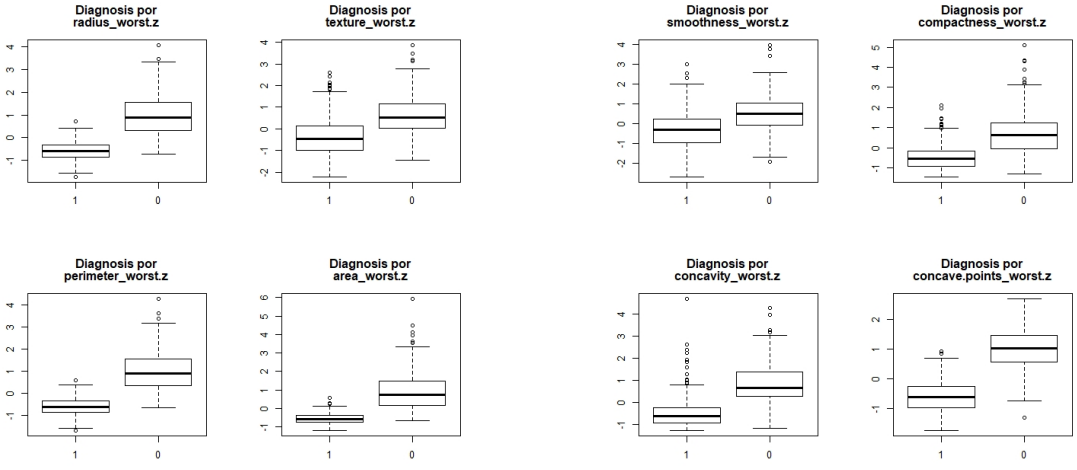


Figura 4: Categoria *se*.



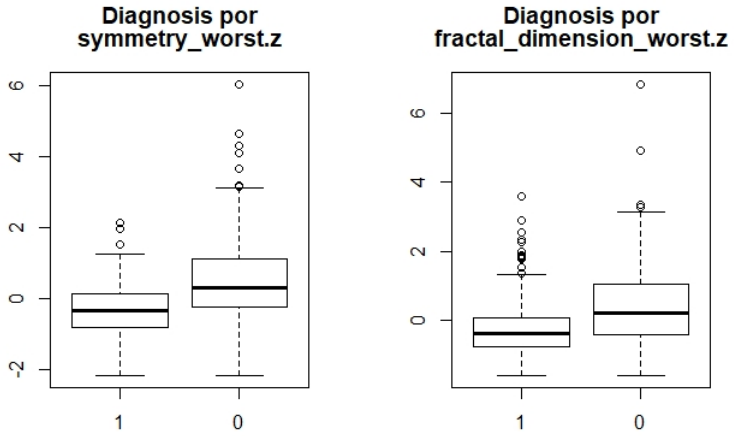


Figura 6: Categoria *worst*.

Ao analisarmos estes gráficos conseguimos perceber que certas variáveis parecem estar bastante bem separadas (como *radius_mean, area_mean, perimeter_mean* entre outras), enquanto algumas aparentam pouca separação nos dados, indicando que provavelmente não serão favoráveis na criação do modelo de regressão logística (por exemplo, *fractal_dimension_mean, texture_se*, entre outras).

Como explicado também no **livro** [1] de apoio sugerido para as aulas, um dos grandes problemas que podem surgir na criação de modelos de regressão é a existência de colinearidade - preditores os quais estão altamente correlacionados entre si. Para combater este problema, um dos métodos mais simples é a análise da matriz de correlações entre todas as variáveis de forma a detetar se existe alguma com alta correlação com outra e se sim, retirá-la do *dataset*.

Com esta matriz, verificamos que:

- *radius_mean/_worst/_se, perimeter_mean/_worst_se e area_mean/_worst_se* estão altamente relacionadas entre si;
- *concave.points_mean/_worst/_se, concavity_mean/_worst/_se e compactness_mean/_worst/_se* estão altamente relacionadas entre si;
- *texture_worst e texture_mean* estão altamente relacionadas;

Nota que, em vez da matriz de correlações, poderíamos ter feito gráficos de dispersão (*scatterplot*) entre todas as variáveis. Por exemplo:

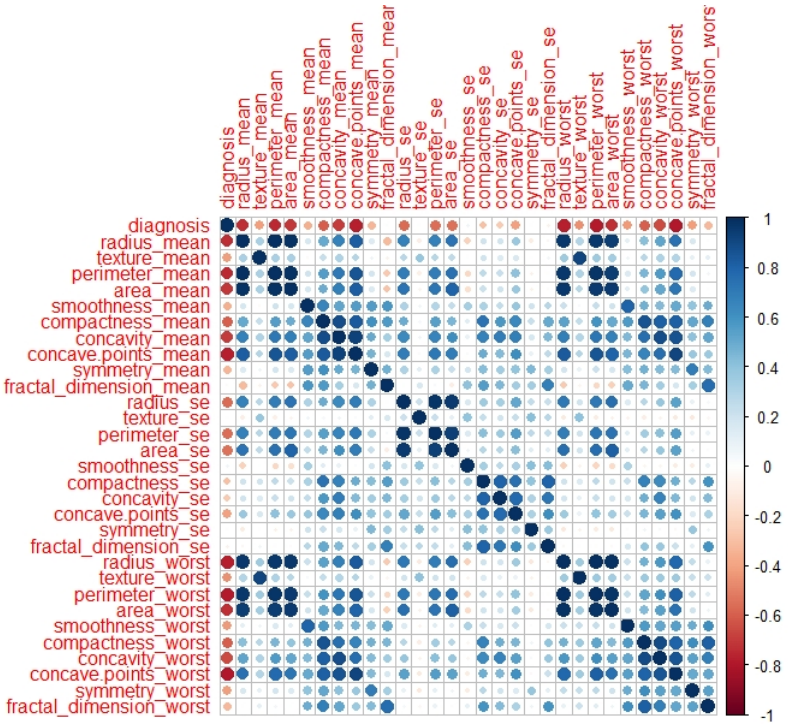
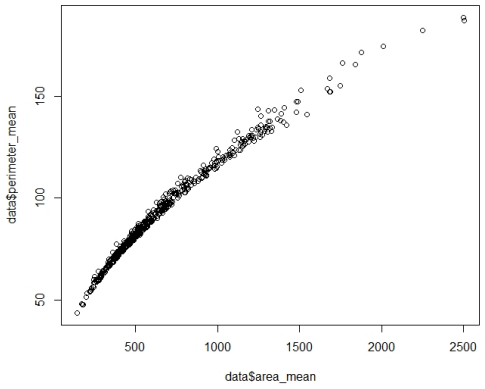
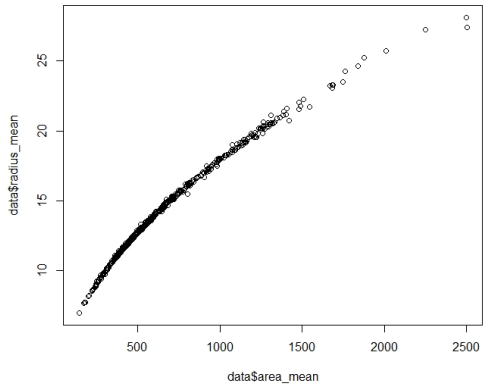


Figura 7: Gráfico com as correlações entre todas as variáveis



(a) *Scatterplot* entre area_mean e perimeter_mean.



(b) *Scatterplot* entre area_mean e radius_mean.

No fim, decidimos retirar *radius_mean, area_mean, concavity_mean, compactness_mean, radius_se, perimeter_se, compactness_se, concavity_se, perimeter_worst, area_worst, texture_worst.*

Analisando os modelos gerados com as variáveis todas e com apenas as ficaram após a limpeza, verificamos que o primeiro ao contrário do segundo, não converge, logo nem sequer ponto de partida para futuras comparações servirá. Por isso, o nosso modelo base para comparações será constituído pelas variáveis resultantes da primeira limpeza - total de 19.

```
model <- glm(diagnosis ~., newdata, family=binomial)
```

Listing 1: Código R do modelo base.

```
Call:
glm(formula = diagnosis ~ ., family = binomial, data = newdata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.3622   0.0000   0.0002   0.0086   1.4641

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)               41.94172   19.22623   2.181  0.02915 *
texture_mean              -0.55631    0.17366  -3.204  0.00136 **
perimeter_mean             0.38897    0.18611   2.090  0.03662 *
smoothness_mean           23.59874  121.92070   0.194  0.84652
concave.points_mean     -165.12538   77.38874  -2.134  0.03287 *
symmetry_mean             35.65563   35.97563   0.991  0.32163
fractal_dimension_mean   309.54479  240.09228   1.289  0.19730
texture_se                -1.26654    1.28666  -0.984  0.32494
area_se                   -0.19450    0.09675  -2.010  0.04439 *
smoothness_se           -299.19514  400.52877  -0.747  0.45506
concave.points_se       -153.85827  301.35457  -0.511  0.60966
symmetry_se              208.67820  149.56596   1.395  0.16295
fractal_dimension_se    1359.61134  848.55675   1.602  0.10910
radius_worst              -2.95476    1.25941  -2.346  0.01897 *
smoothness_worst         -26.11985   77.49930  -0.337  0.73609
compactness_worst         19.61252    9.84799   1.992  0.04642 *
concavity_worst          -11.36941    5.64401  -2.014  0.04397 *
concave.points_worst     -13.24524   46.69205  -0.284  0.77666
symmetry_worst           -49.68427   26.76414  -1.856  0.06340 .
fractal_dimension_worst -227.68141  132.42329  -1.719  0.08555 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 751.440  on 568  degrees of freedom
Residual deviance:  47.352  on 549  degrees of freedom
AIC: 87.352

Number of Fisher Scoring iterations: 11
```

Figura 9: Modelo base: sem as variáveis correlacionadas.

3.2 Otimização do modelo utilizando *Shrinking methods*

3.2.1 *Lasso Regression*

Utilizando agora um modelo baseado em *Lasso Regression* pretendemos treinar um modelo onde possa ser possivel haver variáveis com coeficiente 0 de forma a reduzir o número de variáveis do modelo. Para isto é preciso escolher um *alpha* responsável por levar certos coeficientes a chegarem a 0 (pois este atua no momento de ajuste da regressão, no cálculo do RSS). Para isto utilizamos *cross-validation* como forma de encontrar o melhor *alpha* para o nosso modelo.

```r
library(glmnet)
X <- as.matrix(newdata[,-1])
Y <- newdata$diagnosis

model.lasso <- glmnet(X,Y,family="binomial", alpha=1, lambda=0.007024236)
```

Listing 2: Código R para o modelo *Lasso Regression* com o melhor *alpha*.

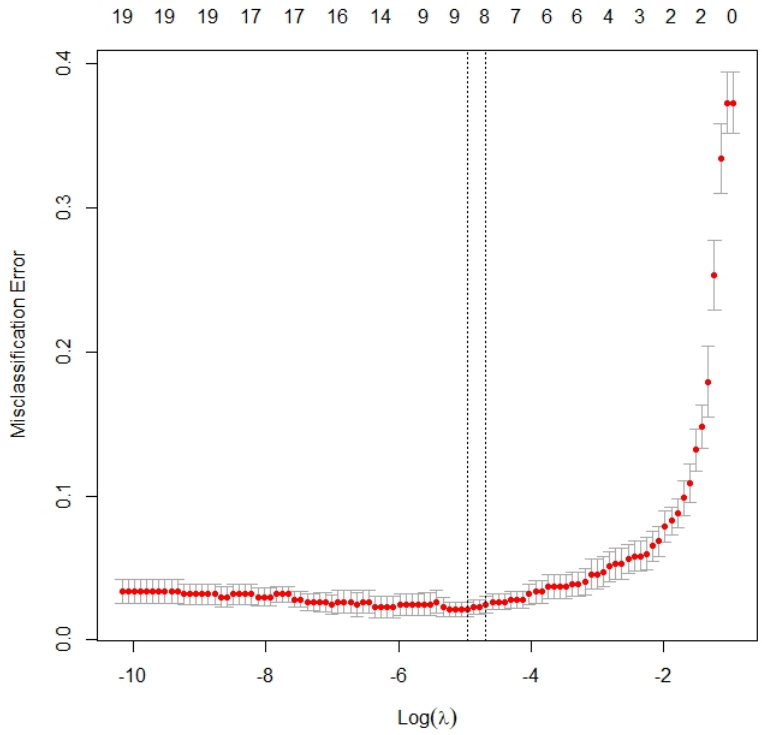


Figura 10: Gráfico dos diferentes *alphas* e o seu erro associado.

(Intercept)	23.93851252
texture_mean	-0.22746884
perimeter_mean	.
smoothness_mean	.
concave.points_mean	-13.68373537
symmetry_mean	.
fractal_dimension_mean	.
texture_se	.
area_se	-0.02320647
smoothness_se	.
concave.points_se	.
symmetry_se	.
fractal_dimension_se	51.98090873
radius_worst	-0.64647812
smoothness_worst	-25.80800475
compactness_worst	.
concavity_worst	-1.52546127
concave.points_worst	-14.96571513
symmetry_worst	-5.33035509
fractal_dimension_worst	.

Figura 11: Coeficientes do modelo baseado em *lasso regression*.

Com a indicação dos coeficientes reduzidos a 0, refazemos um modelo normal de regressão logística, com agora menos variáveis, apresentando os seguintes resultados:

```
Call:
glm(formula = diagnosis ~ . - perimeter_mean - smoothness_mean -
    symmetry_mean - fractal_dimension_mean - texture_se - smoothness_se -
    concave.points_se - symmetry_se - compactness_worst - fractal_dimension_worst,
    family = binomial, data = newdata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6464  -0.0001   0.0024   0.0296   1.3204

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)           46.6251     8.7351   5.338 9.42e-08 ***
texture_mean          -0.4844     0.1084  -4.467 7.92e-06 ***
concave.points_mean  -25.6967    34.0448  -0.755 0.450375
area_se               -0.1343     0.0439  -3.058 0.002227 **
fractal_dimension_se 675.9494   233.8765   2.890 0.003850 **
radius_worst          -1.0235     0.2897  -3.532 0.000412 ***
smoothness_worst     -65.0863    24.6456  -2.641 0.008269 **
concavity_worst       -7.1769     3.5708  -2.010 0.044440 *
concave.points_worst -26.8798    21.2204  -1.267 0.205263
symmetry_worst       -10.7258     5.8598  -1.830 0.067188 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 751.440  on 568  degrees of freedom
Residual deviance:  65.292  on 559  degrees of freedom
AIC: 85.292

Number of Fisher Scoring iterations: 10
```

Figura 12: Modelo base depois da redução de variáveis utilizando o *lasso regression*.

```r
model <- glm(diagnosis~ .-perimeter_mean-smoothness_mean-symmetry_mean -
    fractal_dimension_mean-texture_se-smoothness_se -
    concave.points_se-symmetry_se-compactness_worst -
    fractal_dimension_worst, data=newdata, family=binomial)
```

Listing 3: Código R para o modelo de regressão logística sem os coeficientes reduzidos a 0, derivados do *lasso regression*.

3.2.2 *Ridge Regression*

Seguindo a lógica anterior, refizemos o modelo agora utilizando *Ridge Regression* que resultou em resultados estranhos: para o melhor *alpha* nenhuma variável chegou a 0.

```r
library(glmnet)
X <- as.matrix(newdata[,-1])
Y <- newdata$diagnosis

model.ridge <- glmnet(X,Y,family="binomial", alpha=0, lambda=0.03836832)
```

Listing 4: Código R para o modelo *Ridge Regression* com o melhor *alpha*.

Por isso, ao contrário do anterior, que possui 9 coeficientes iguais a zero, este como apresenta 0, tem apenas coeficientes diferentes em relação ao nosso modelo base, logo não conseguimos executar nenhum tipo de redução das variáveis.

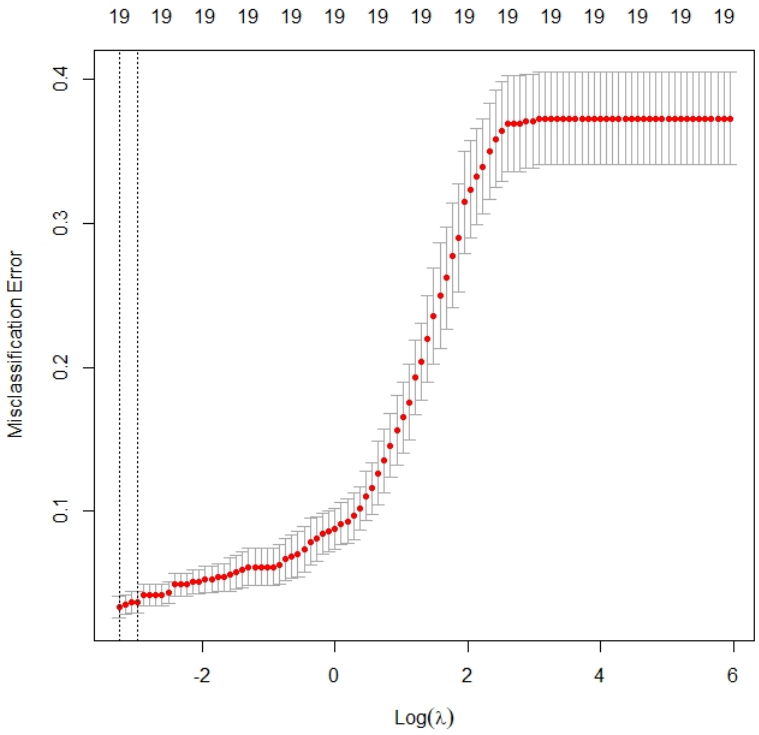


Figura 13: Gráfico dos diferentes *alphas* e o seu erro associado.

(Intercept)	13.25706499
texture_mean	-0.13132618
perimeter_mean	-0.02608043
smoothness_mean	-9.20845276
concave.points_mean	-14.53917317
symmetry_mean	-2.66549979
fractal_dimension_mean	47.08032973
texture_se	-0.15502650
area_se	-0.01139008
smoothness_se	3.17206907
concave.points_se	-22.37042714
symmetry_se	15.08944068
fractal_dimension_se	79.19882669
radius_worst	-0.15592469
smoothness_worst	-14.41401782
compactness_worst	-1.16354667
concavity_worst	-1.76067428
concave.points_worst	-8.44621678
symmetry_worst	-5.55022947
fractal_dimension_worst	-4.61253016

Figura 14: Coeficientes do modelo baseado em *ridge regression*.

3.3 Otimização do modelo utilizando *Stepwise Subset Slection*

Para testar modelos com diferentes combinações das variáveis, recorremos a um método chamado *Stepwise Selection*, onde a cada iteração se vai diminuindo ou aumentando o número de preditores e, a cada iteração, verifica-se qual o que dá melhor resultado. No nosso caso experimentamos a implementação *backwards* (vai retirando preditores), *forward* (vai adicionando preditores) e uma mistura dos dois. *Forward Stepwise Selection* forneceu piores resultados que o modelo base, por isso nem guardado para a comparação final. A mistura dos dois e o de *backwards* deram o

mesmo resultado:

```
Call:  glm(formula = diagnosis ~ texture_mean + perimeter_mean + concave.points_mean +
    fractal_dimension_mean + area_se + smoothness_se + symmetry_se +
    fractal_dimension_se + radius_worst + compactness_worst +
    concavity_worst + symmetry_worst + fractal_dimension_worst,
    family = binomial, data = newdata)

Coefficients:
            (Intercept)             texture_mean           perimeter_mean
                44.4656                  -0.6036                   0.3688
    concave.points_mean   fractal_dimension_mean                  area_se
              -164.5255                 260.2374                  -0.1716
          smoothness_se              symmetry_se     fractal_dimension_se
              -463.0830                 184.5403                1338.4731
           radius_worst        compactness_worst          concavity_worst
                -2.9000                  19.0348                  -14.2532
         symmetry_worst  fractal_dimension_worst
              -36.9264                -217.7147

Degrees of Freedom: 568 Total (i.e. Null);  555 Residual
Null Deviance:      751.4
Residual Deviance: 49.25        AIC: 77.25
```

Figura 15: Modelo conseguido usando *backward stepwise selection*.

```
model <- stepAIC(model.2, trace=FALSE, direction="backward")
```

Listing 5: Código R para determinar o melhor modelo usando *backward stepwise selection*.

3.4 Otimização do modelo utilizando *Dimension Reduction methods*

3.4.1 *PCA*

O método *Principal Component Analysis* também conhecido como **PCA**, é um dos métodos de *Dimension Reduction methods*, com o objectivo de reduzir a complexidade do modelo. Extraindo preditores significativos de um grande conjunto de variáveis presentes num determinado *dataset*, condensa assim a informação contida nessas num conjunto menor de variáveis estatísticas (componentes) com uma perda mínima desta informação relativamente ao conjunto de dados inicial.

Neste estudo o *dataset* possui uma dimensão de 569 (n) x 30 (p). O **n** representa o número de observações e o **p** representa o número de preditores. Como temos um p = 30, para efectuar uma análise por meio de gráficos de dispersão seria necessário p(p-1)/2 gráficos de dispersão, ou seja, 435 gráficos possíveis para analisar a relação entre as variáveis (apesar de termos realizado esta análise na secção 3.1, esta foi muito superficial, de modo que existirá provavelmente variáveis ainda a ser retiradas devido às suas correlações). Este modo foi uma alternativa ao processo anteriormente realizado, executado automaticamente e de forma menos trabalhosa.

Desta forma utilizou-se o **PCA** para verificar quais componentes agregariam maior valor ao nosso modelo de regressão logística para classificação. Ao efectuar o método **PCA** no *dataset* obtivemos o Gráfico 1, o qual demonstra a proporção acumulativa de variância explicativa dos componentes principais gerados. O que vai de encontro ao Gráfico 2 que exibe a importância das principais componentes em relação à variância. Analisando o Gráfico 1 com maior detalhe, percebe-se uma maior representatividade nas 6 principais componentes, os quais juntos representam mais de 88% dos dados sendo que a partir da 7º componente já não há um crescimento significativo. Desta forma escolhemos as 6 primeiras componentes para colocar como *input* do nosso modelo de regressão logística.

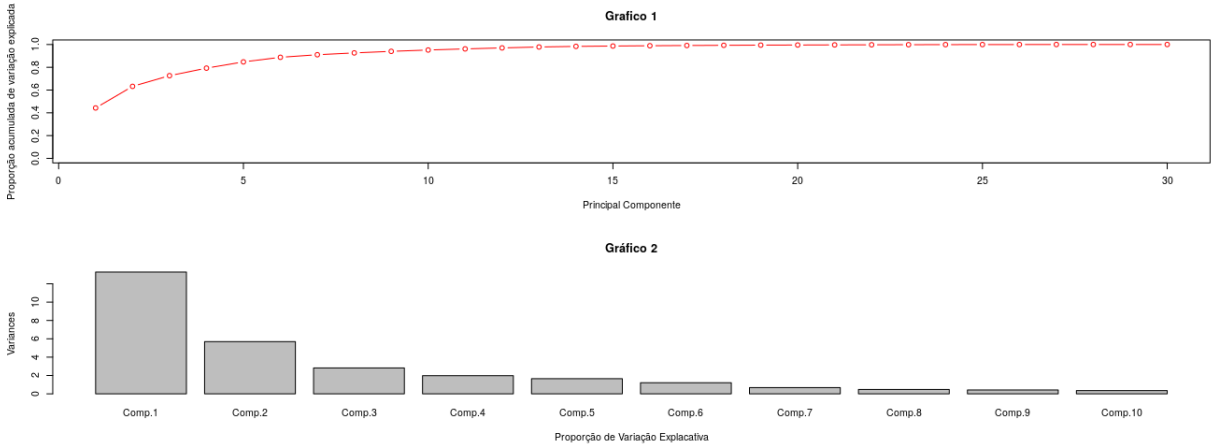


Figura 16: Analise dos principais componentes do *dataset*

4 Conclusões

Perante o *dataset* inicial, várias modelos de regressão logística foram desenvolvidos com base em diferentes conjuntos de características, obtidas através de um conjunto de técnicas pertencentes ao plano de estudos da disciplina. Na tabela seguinte encontram-se os resultados obtidos através de um *k-cross validation* (k = 10) para os 4 modelos, onde analisamos vários aspectos: média da precisão de cada modelo, média dos falsos positivos e falsos negativos e *AIC* médio.

Modelo	Precisão (média)	Falsos Positivos (média)	Falsos Negativos (média)	AIC (média)
1	0.97931	0	0.02068	85.128
2	0.98275	0.00344	0.01379	82.937
3	0.89310	0.03793	0.06896	223.103
4	0.98275	0.00344	0.01379	90.855

Tabela 1: Tabela comparando os 4 modelos gerados (4 diferentes combinações de variáveis).

1. modelo com conjunto de variáveis reduzido com base em **correlações (análise manual)** - total de 19;
2. modelo baseado no anterior, mas ainda mais reduzido utilizando **lasso regression** - total de 9;
3. modelo baseado no primeiro, mas ainda mais reduzido utilizando **stepwise subset selection** - total de 6;
4. modelo com conjunto de variáveis reduzido utilizando **PCA** - total de 6;

	Modelo 1	Modelo 2	Modelo 3
radius_mean			
texture_mean	X	X	
perimeter_mean	X		
area_mean			
smoothness_mean	X		X
compactness_mean			
concavity_mean			
concave.points_mean	X	X	
symmetry_mean	X		X
fractal_dimension_mean	X		
radius_se			
texture_se	X		X
perimeter_se			
area_se	X	X	
smoothness_se	X		
compactness_se			
concavity_se			
concave.points_se	X		X
symmetry_se	X		
fractal_dimension_se	X	X	
radius_worst	X	X	
texture_worst			
perimeter_worst			
area_worst			
smoothness_worst	X	X	X
compactness_worst	X		
concavity_worst	X	X	
concave.points_worst	X	X	X
symmetry_worst	X	X	
fractal_dimension_worst	X		

Tabela 2: Variáveis selecionadas para os 3 modelos (o 4º possui novas variáveis denominadas de componentes). A verde estão variáveis significativas a mais de 95%, a vermelho não significativas.

Analisando a tabela com as variáveis, verificamos que *texture_mean*, *area_se*, *radius_worst* e symmetry_worst parecem ser as mais significativas na classificação dos dados (como se consta vagamente nos *boxplot* em cima apresentados). Consta-se também que é normal o modelo 2 dar melhor resultados que o 1, pois é derivado deste mas com menos variáveis não significativas.

Distinguir os casos de cancro Maligno e Benigno das amostras de massa mamária era o objectivo deste projecto, sendo que este foi conseguido com uma precisão de 98% utilizando as reduzidas variáveis fornecidas pela *PCA* ou *lasso regression*, contendo um rácio de 0.3% de falsos positivos e 0.1% de falsos negativos, o que provam ser insignificantes. Foi curioso os resultados entre o modelo 2 e 4 serem iguais com excepção do *AIC*, que foi o factor de decisão do melhor modelo. Concluímos então que, o método *PCA*, com muito pouco esforço, conseguiu valores tão bons quanto os do método *lasso regression* que era mais trabalhoso - apesar dos diferentes conjunto de variáveis. Com isto percebemos que o *dataset* possui conjuntos diferentes de variáveis suficientemente capazes de responder ao problema, sendo que quanto menos estiverem presentes, mais simples este é.

Para trabalho futuro poderíamos executar diferentes algoritmos de classificação (por exemplo *LDA* - **Linear Discriminant Analysis** para comparar com estes de regressão logística.

Referências

[1] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani *An Introduction to Statistical Learning with Applications in R*