DEFAULT OF CREDIT CARD CLIENTS

Daniel Silva | Luís Henriques
2012141749 | 2012146215

Abstracto

O projeto em estudo trata de um problema de classificação de dados pertencentes a um banco sediado na Tailândia e que descrevem a situação financeira de cada um dos clientes aí fidelizados. A resolução do mesmo passa pelo recurso a uma grande variedade de métodos de **classificação supervisionada** e **não supervisionada**, que por sua vez são auxiliados de outras ferramentas capazes de simplificar toda a informação disponível. O objetivo principal centra-se na avaliação de cada um destes algoritmos de classificação aqui inseridos de forma a realizar a separação mais eficaz possível entre cliente "capaz" e "não capaz" de pagar as suas dívidas em crédito no mês seguinte.

Para tal, utilizam-se métricas quantitativas, como por exemplo a *accuracy*, para a qual se obteve taxas de sucesso aproximadamente entre os 41% e os 79%, destacando-se entre os algoritmos testados o **K-NN**, o que demonstra em termos gerais a má performance obtida.

Introdução

Atualmente, um dos problemas que enfrentamos no que diz respeito à **modelação e previsão das classes de *datasets*** é a existência de uma grande quantidade de informação (variáveis) que descreve o sistema, pelo que a sua grande dimensão torna difícil a deteção de padrões e consequente classificação dos dados. Daqui segue-se a necessidade imediata de reduzir ao máximo essa informação sem, no entanto, pôr em causa a fiabilidade da representação do sistema.

Baseando-nos nesta necessidade foi realizado este trabalho. Mais especificamente, dada a situação de um importante banco de Taiwan, com base no historial de cada cliente, será instigado se este mesmo banco deverá confiar nesse cliente, de modo a prever se o dinheiro outrora emprestado será retornado. No entanto, como anteriormente mencionado, existem um grande número de variáveis a considerar, pelo que numa primeira fase serão postos em prática métodos (atualmente utilizados) que nos permitam **reduzir esta dimensionalidade** desnecessariamente elevada, e numa segunda fase será realizada uma classificação dos dados de forma a reunir esta informação descomplexada de forma útil.

Métodos

Descrição das *Features*

De modo a perceber a natureza das *features* do *dataset* é necessário realizar uma rápida visualização da distribuição de cada uma das *features* pelas classes. Para tal geram-se ***boxplots***, para as *features* contínuas, e **histogramas** para as *features* nominais.

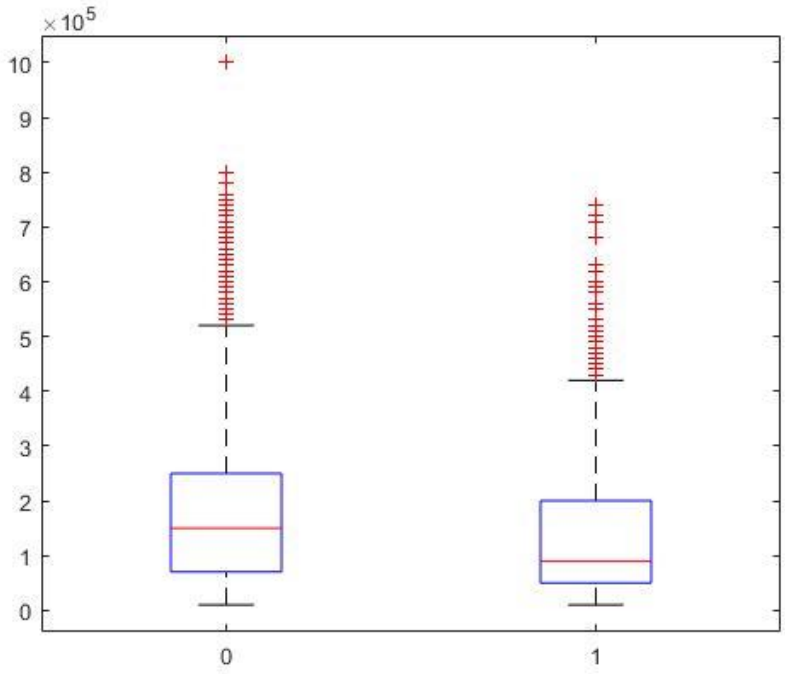

Figura 1 – Boxplot of the variable X1, LIMIT_BAL, for each class;

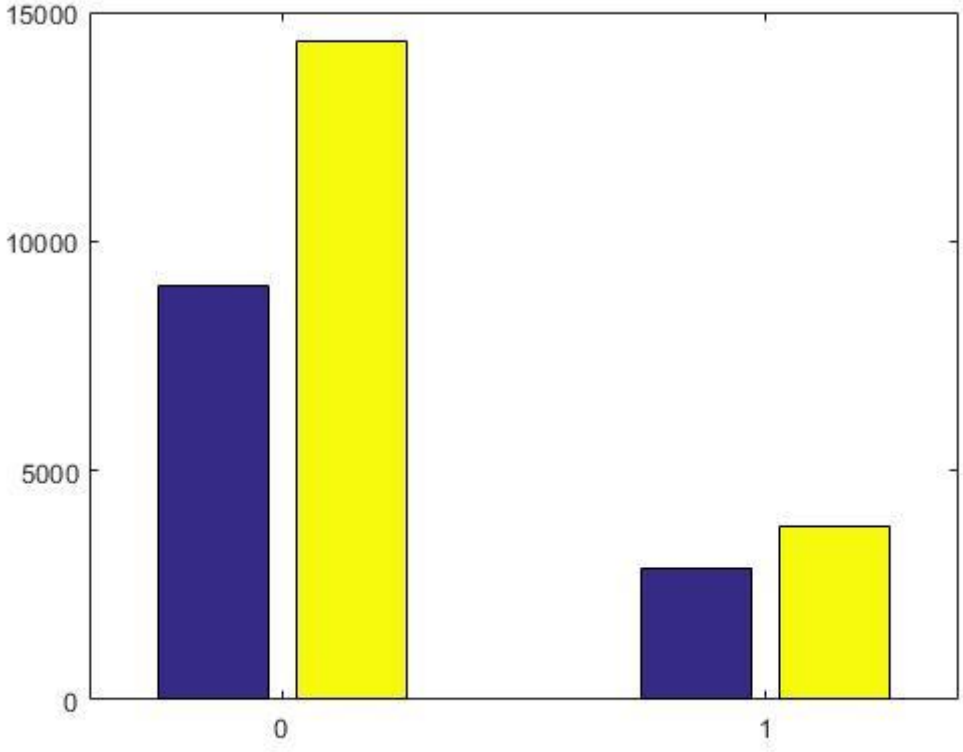

Figura 2 – Histogram of the variable X2, GENDER, distributions for each class;

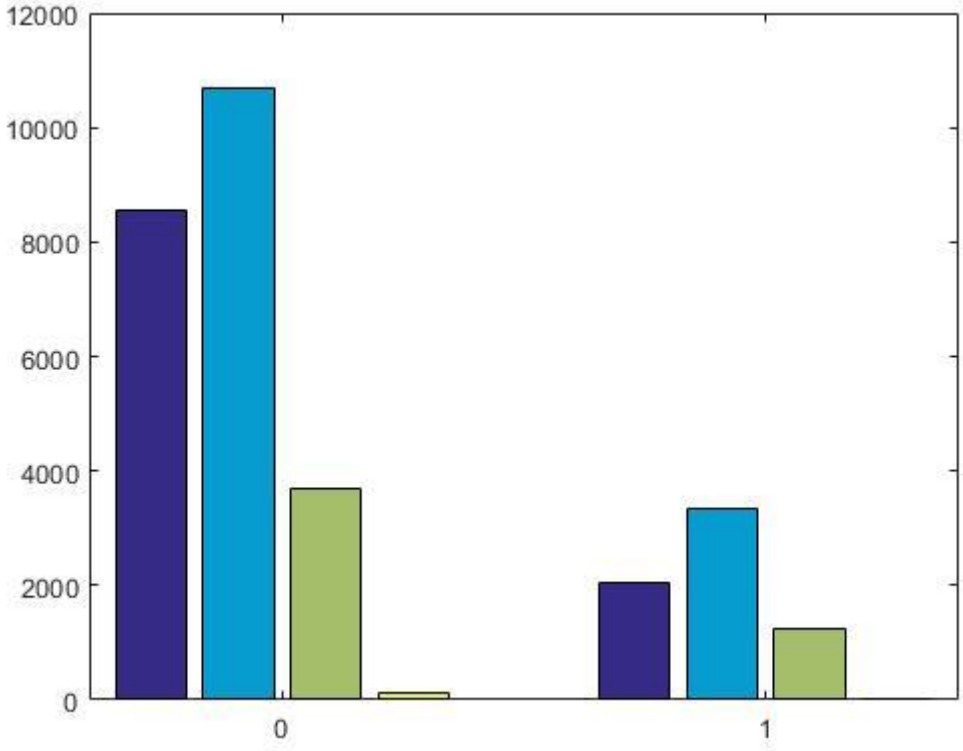

Figura 3 – Histogram of the variable X3, Education, distributions for each class;

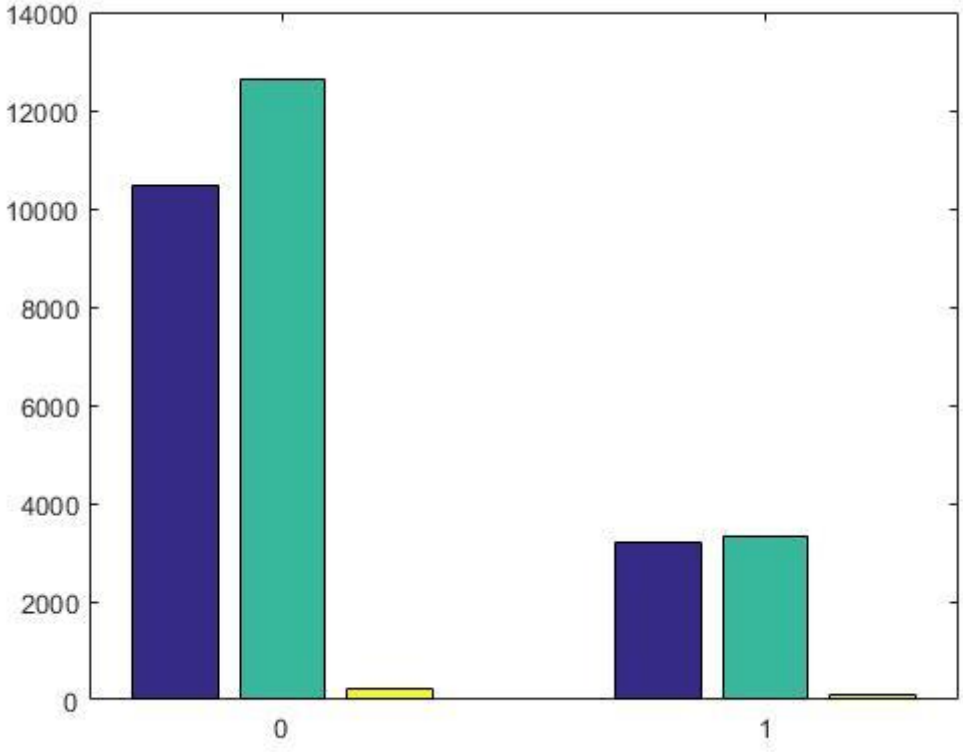

Figura 4 – Histogram of the variable X4, Marital, distributions for each class;

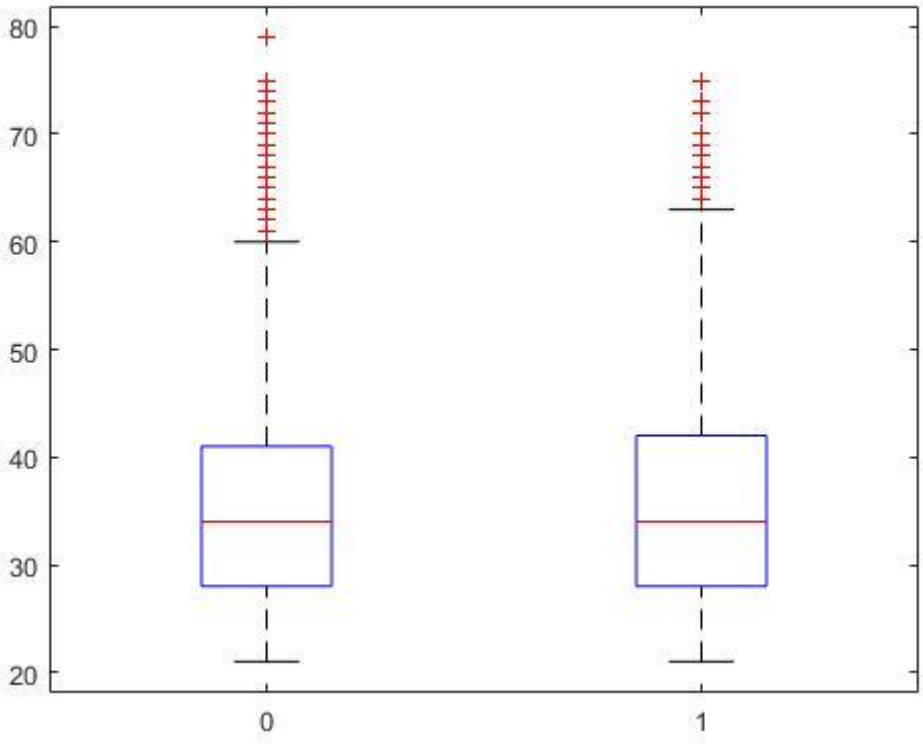

Figura 5 – Boxplot of the variable X6, AGE, for each class;

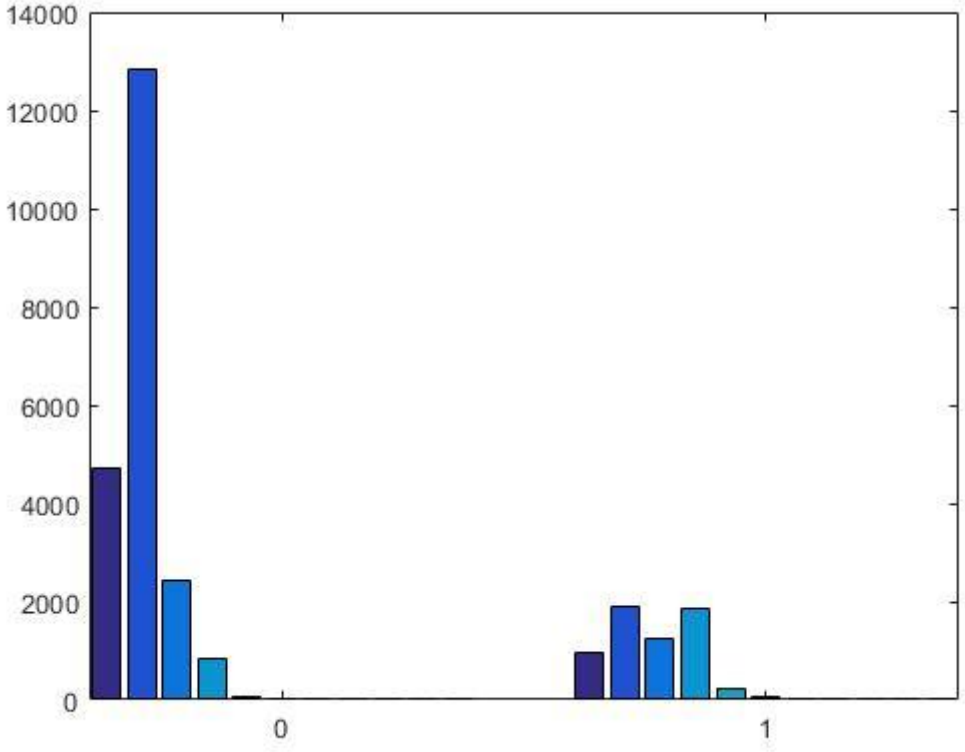

Figura 6 – Histogram of the variable X6, September 2005, distributions for each class;

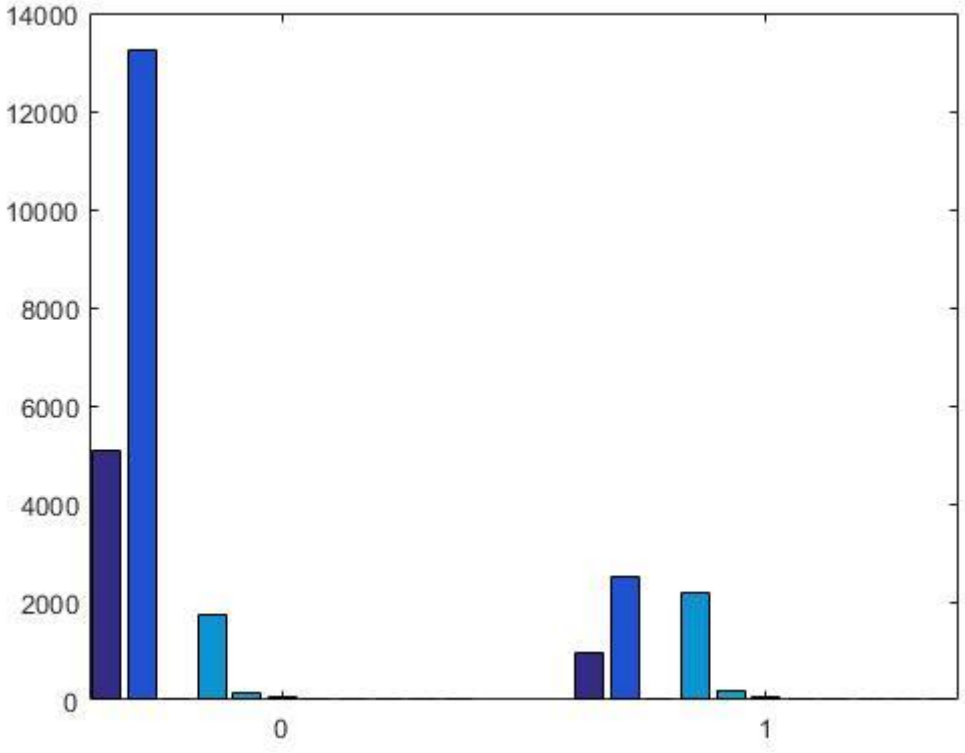

Figura 7 – Histogram of the variable X7, August 2005, distributions for each class;

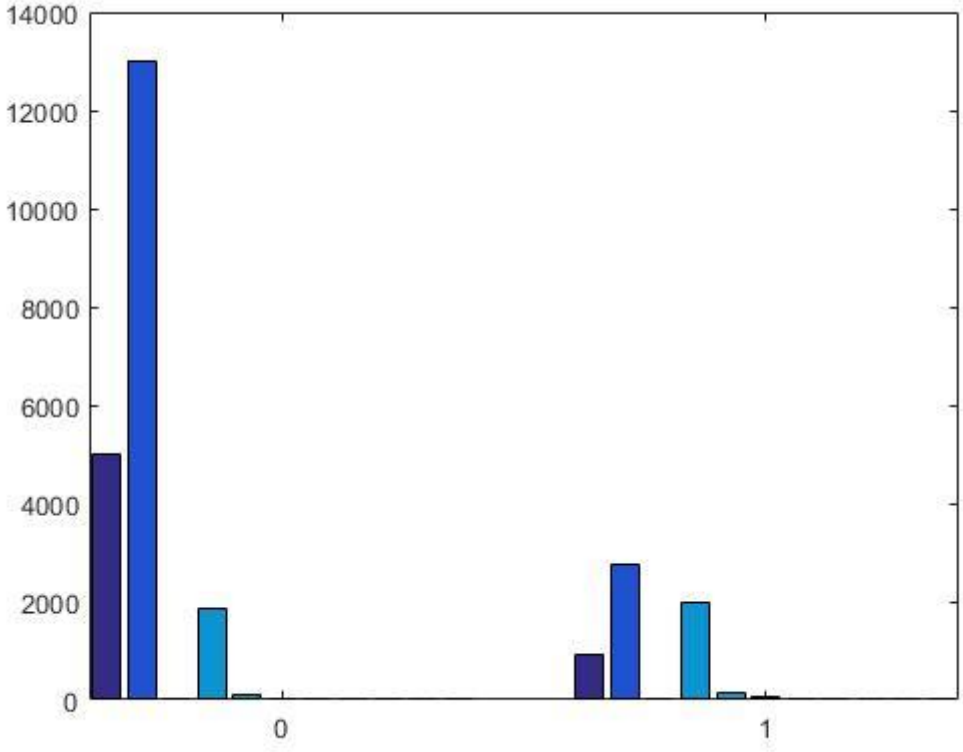

Figura 8 – Histogram of the variable X8, July 2005, distributions for each class;

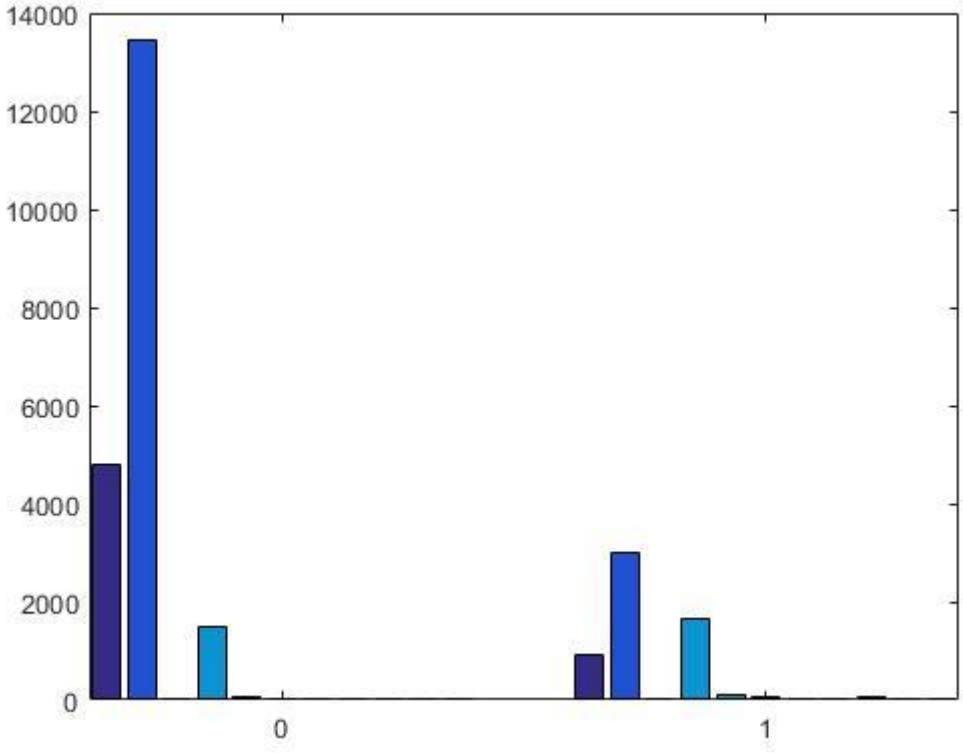

Figura 9 – Histogram of the variable X9, June 2005, distributions for each class;

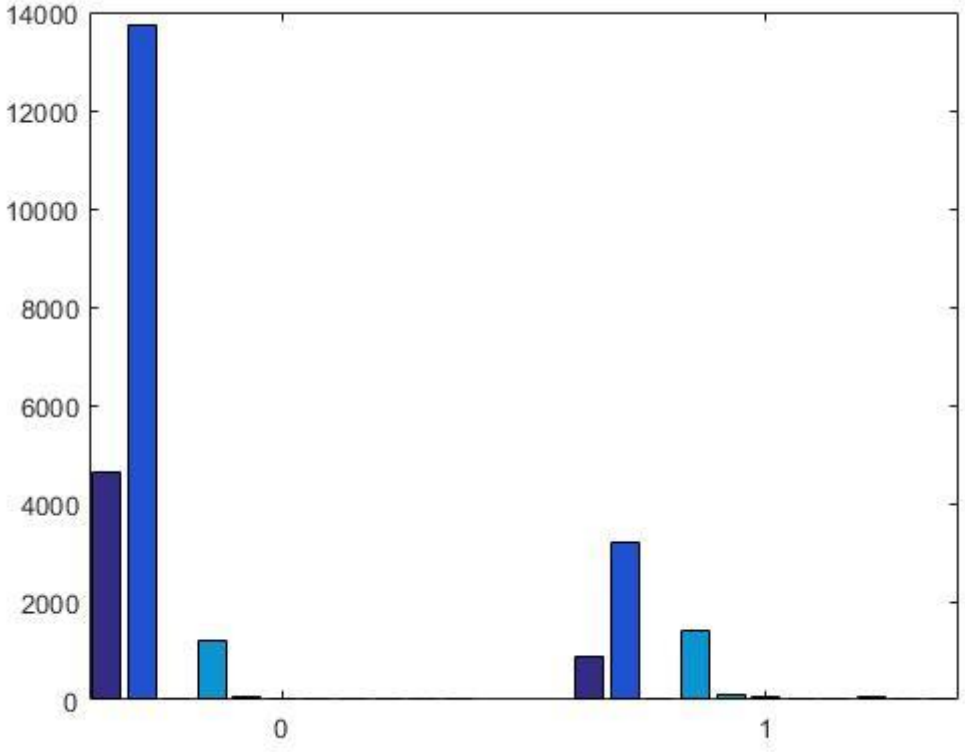

Figura 10 – Histogram of the variable X10, May 2005, distributions for each class;

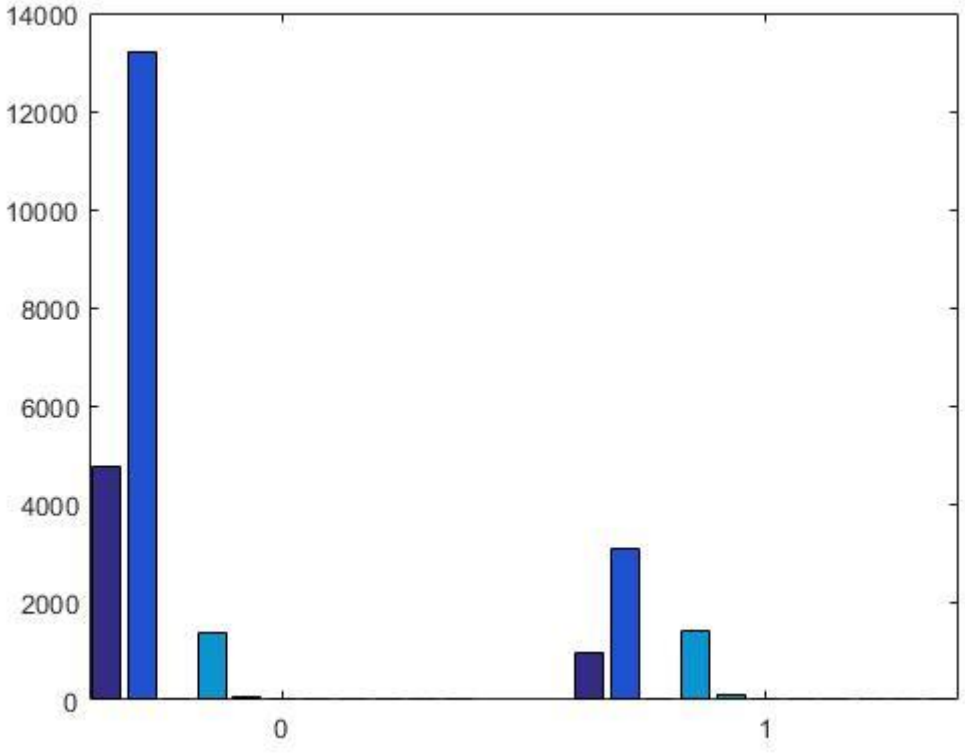

Figura 11 – Histogram of the variable X11, April 2005, distributions for each class;

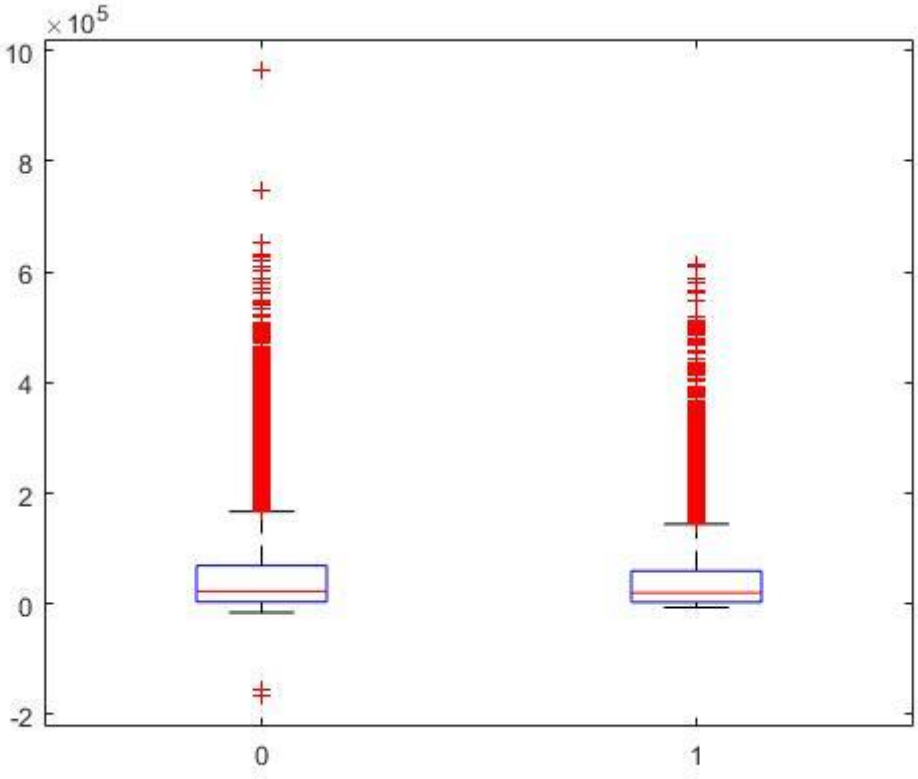

Figura 12 - Boxplot of the variable X12, September 2005 Bill State, for each class;

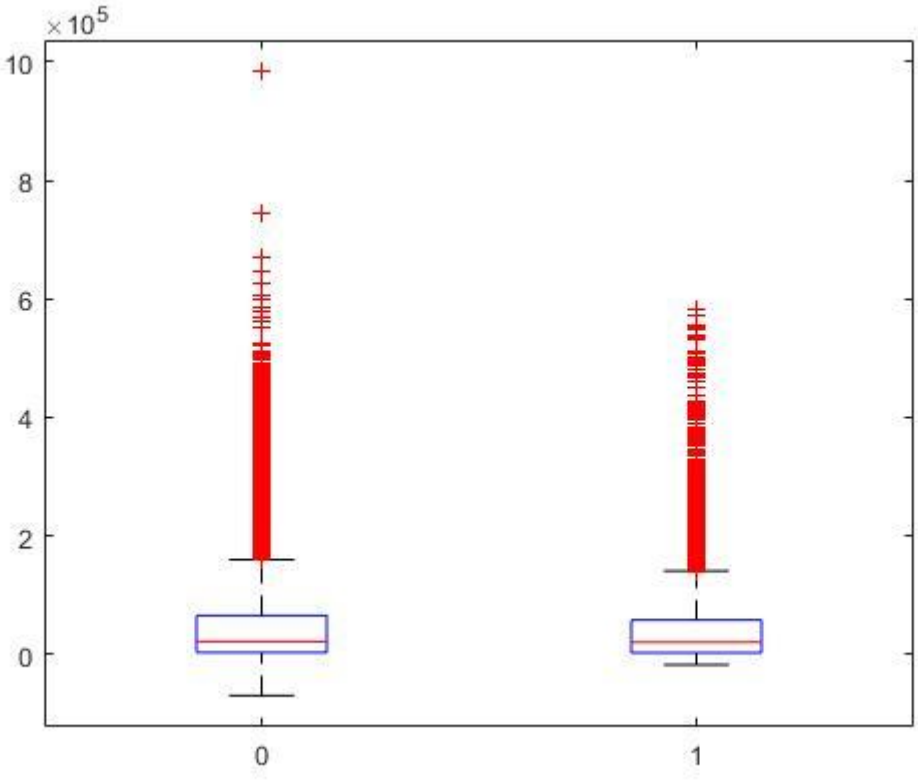

Figura 13 - Boxplot of the variable X13, August 2005 Bill State, for each class;

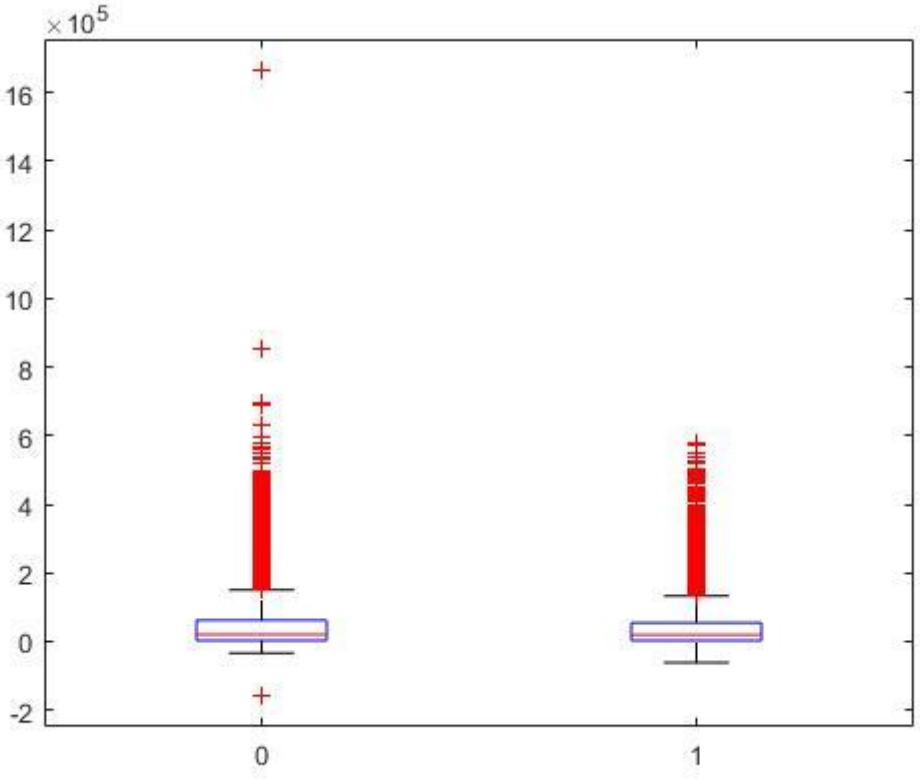

Figura 14 - Boxplot of the variable X14, July 2005 Bill State, for each class;

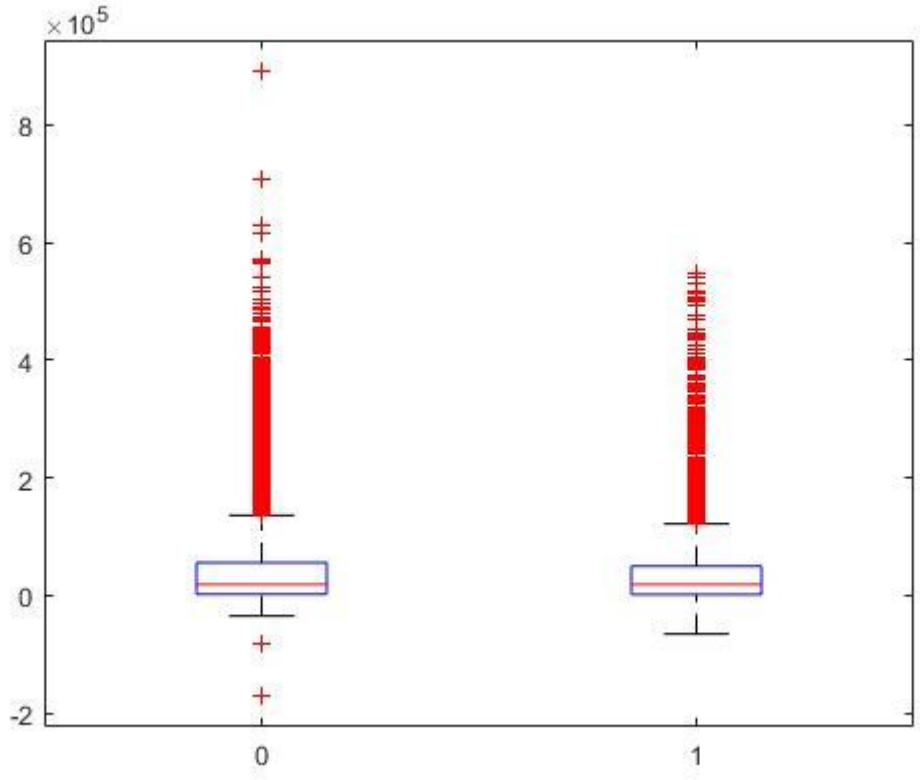

Figura 15 – Boxplot of the variable X15, June 2005 Bill State, for each class;

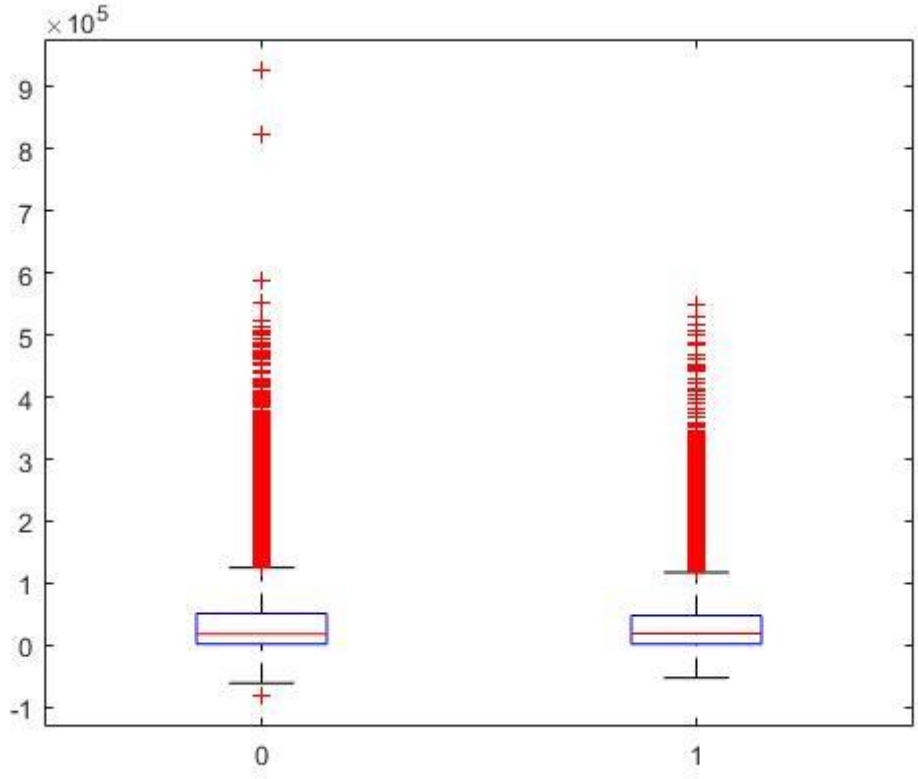

Figura 16 - Boxplot of the variable X16, May 2005 Bill State, for each class;

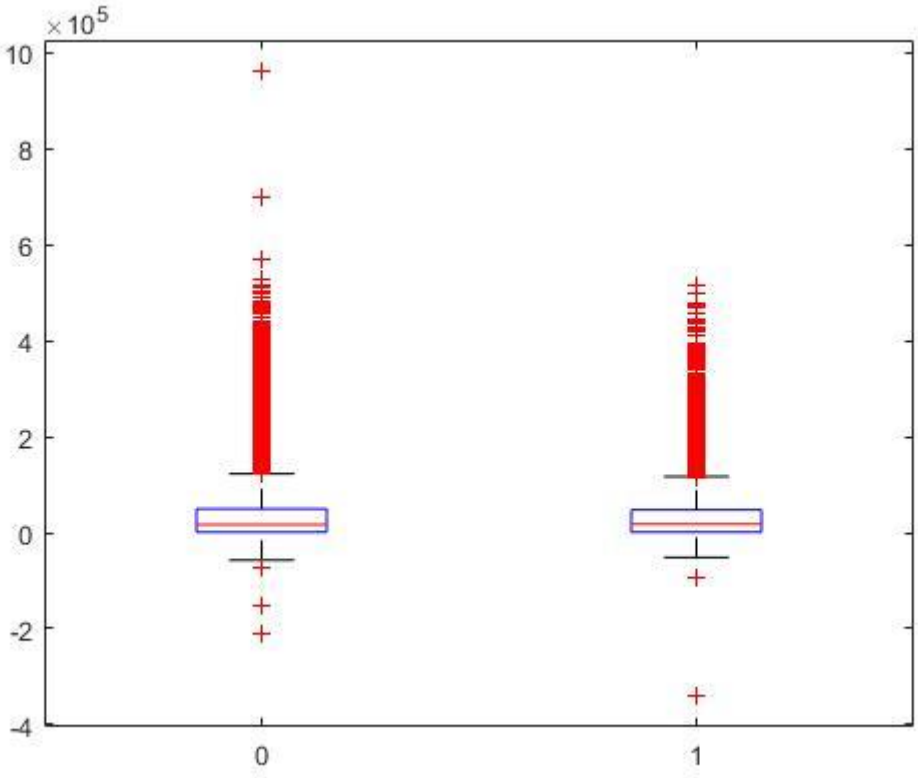

Figura 17 – Boxplot of the variable X17, April 2005 Bill State, for each class;

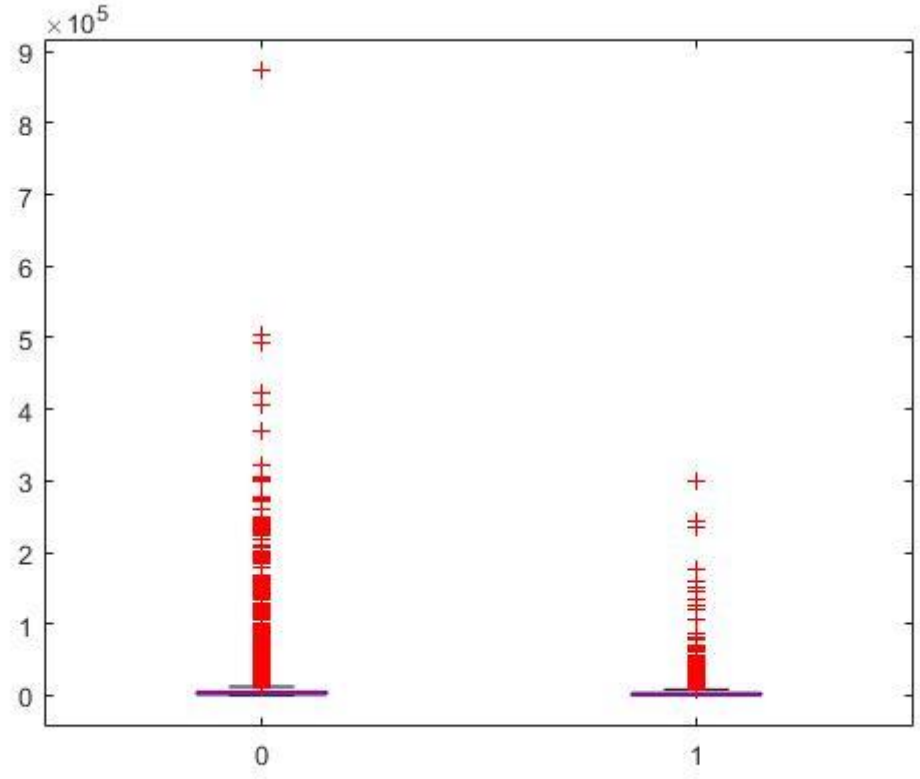

Figura 18 – Boxplot of the variable X18, September 2005 Previous State, for each class;

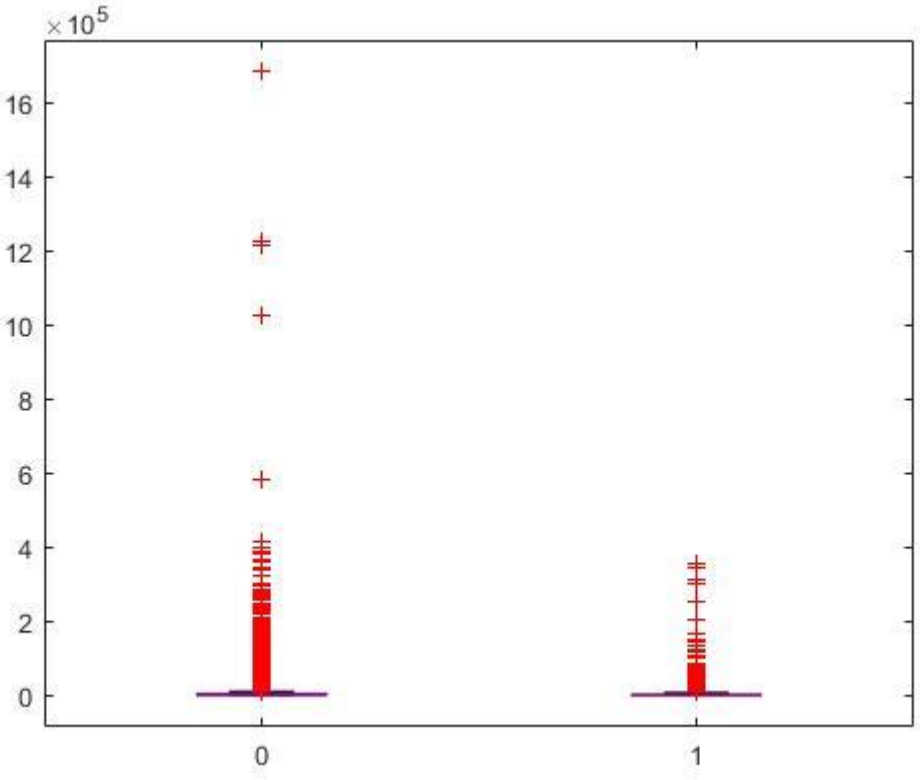

Figura 19 – Boxplot of the variable X19, August 2005 Previous State, for each class;

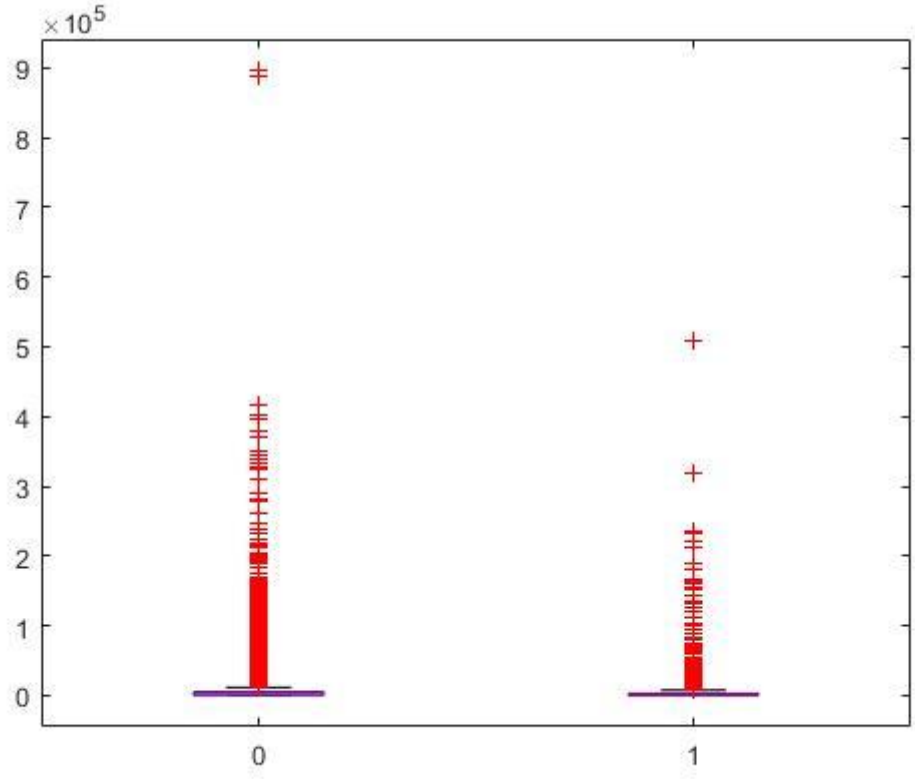

Figura 20 – Boxplot of the variable X20, July 2005 Previous State, for each class;

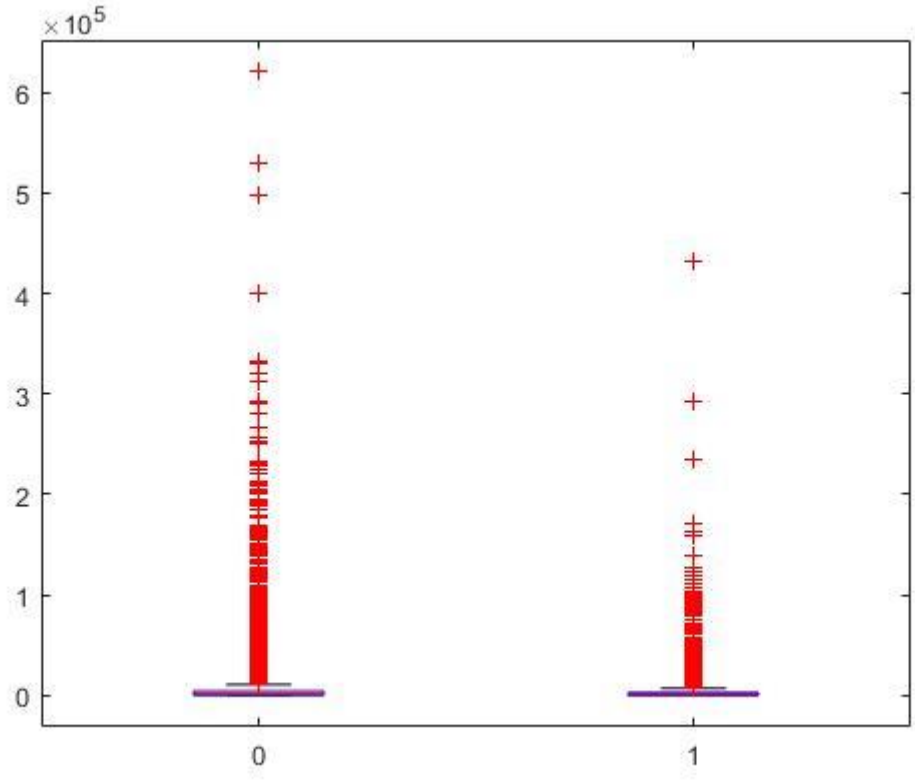

Figura 21 – Boxplot of the variable X21, June 2005 Previous State, for each class;

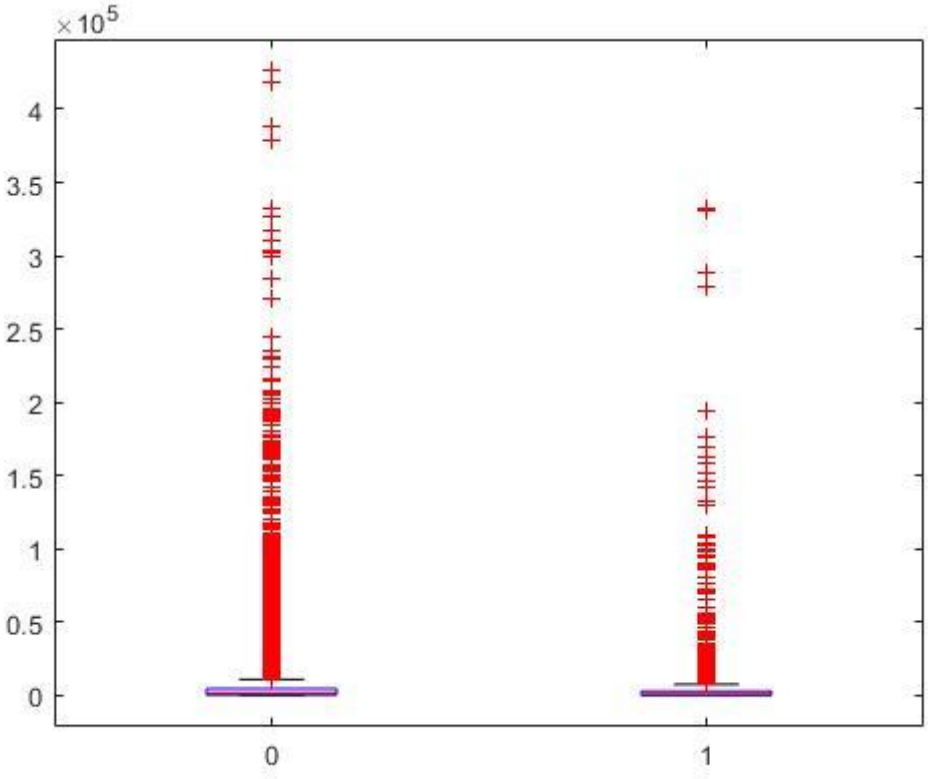

Figura 22 – Boxplot of the variable X22, May 2005 Previous State, for each class;

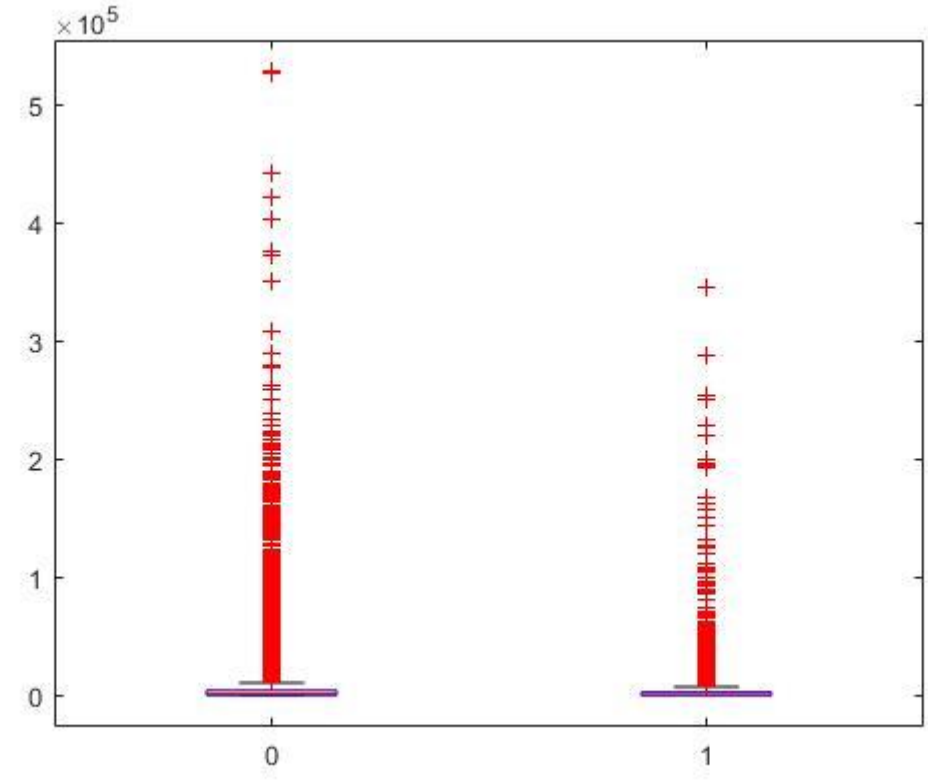

Figura 23 – Boxplot of the variable X23, April 2005 Previous State, for each class;

Seleção e Redução de *Features*

De forma simplificada, com vista a seleccionar e remover *features* cuja contribuição para o problema de classificação é mínima, procede-se à aplicação de técnicas de **redução de *features***. Inicia-se esta tarefa com **ordenação das *features* mais discriminantes** através da implementação do algoritmo de **Kruskal-Wallis** (e posterior comparação com resultados obtidos na matriz correlação, de modo a evitar ambiguidades). Simultaneamente analisa-se a **AUC** de tais variáveis separadamente, auxiliados dos **boxplots** e **histogramas** respectivos. Combinando as interpretações é possível determinar quais as ***features*** que irão ser **decisivas** na distinção a realizar.

Por fim, aplica-se o ***Principal Component Analysis*** (PCA) para redução da dimensão do *dataset*.

- Teste de ***Kruskal-Wallis***;

Esta prática é seguida da aplicação de **correlação**, no qual as *features* com χ^2 inferior à da mediana e que tenham uma correlação superior a 10% com uma das *features* com χ^2 superiores ou iguais à mediana são removidas.

Tabela 1 – Variáveis em estudo ordenadas de acordo com χ^2 (após realização do teste de Kruskal Wallis);

	1
PAY_0	2.5616e+03
PAY_2	1.4116e+03
PAY_3	1.1380e+03
PAY_4	905.0114
LIMIT_BAL	862.7564
PAY_AMT1	772.7156
PAY_5	758.8177
PAY_AMT2	683.8024
PAY_6	609.3657
PAY_AMT3	582.8512
PAY_AMT4	491.3393
PAY_AMT6	442.4420
PAY_AMT5	407.7629
EDUCATION	59.0562
SEX	47.9038
MARRIAGE	21.0506
BILL_AMT1	19.2428
BILL_AMT2	7.2573

Matriz Correlação

Tabela 2 - Matriz correlação (após remoção das variáveis ambíguas)

	PAY_0	PAY_2	PAY_3	PAY_4	LIMIT_BAL	PAY_AMT1	PAY_5	PAY_AMT2	PAY_6	SEX	MARRIAGE
PAY_0	100	67.2164	57.4245	53.8841	27.1214	7.9269	50.9426	7.0101	47.4553	5.7643	1.9917
PAY_2	67.2164	100	76.6552	66.2067	29.6382	8.0701	62.2780	5.8990	57.5501	7.0771	2.4199
PAY_3	57.4245	76.6552	100	77.7359	28.6123	0.1295	68.6775	6.6793	63.2684	6.6096	3.2688
PAY_4	53.8841	66.2067	77.7359	100	26.7460	0.9362	81.9835	0.1944	71.6449	6.0173	3.3122
LIMIT_BAL	27.1214	29.6382	28.6123	26.7460	100	19.5236	24.9411	17.8408	23.5195	2.4755	10.8139
PAY_AMT1	7.9269	8.0701	0.1295	0.9362	19.5236	100	0.6089	28.5576	0.1496	0.0242	0.5979
PAY_5	50.9426	62.2780	68.6775	81.9835	24.9411	0.6089	100	0.3191	81.6900	5.5064	3.5629
PAY_AMT2	7.0101	5.8990	6.6793	0.1944	17.8408	28.5576	0.3191	100	0.5223	0.1391	0.8093
PAY_6	47.4553	57.5501	63.2684	71.6449	23.5195	0.1496	81.6900	0.5223	100	4.4008	3.4345
SEX	5.7643	7.0771	6.6096	6.0173	2.4755	0.0242	5.5064	0.1391	4.4008	100	3.1389
MARRIAGE	1.9917	2.4199	3.2688	3.3122	10.8139	0.5979	3.5629	0.8093	3.4345	3.1389	100

- ***Area Under Curve (AUC)***

A partir da seleção realizada no ponto anterior, de onde resultam as **11 *features*** presentes na tabela 2, inspeciona-se a **AUC** para cada *feature* com o objectivo de auxiliar a avaliação realizada a partir do Kruskal-Wallis complementada pela correlação.

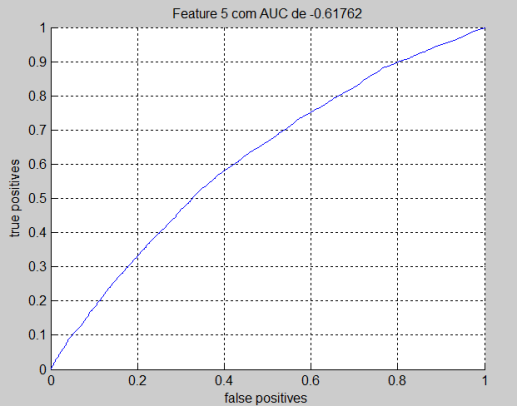

Figura 24: Gráfico tipo da *ROC curve* para a variável com melhor taxa de AUC

- **Principal Component Analysis (PCA)**

Ambiguidade entre variáveis com grande correlação entre si, mas com elevado χ^2, é descomplexada nesta etapa com o uso do PCA. Sua eliminação da base de dados não seria de todo vantajoso, tendo em vista a **redução de dimensões** a que será sujeita nesta fase, que poderia causar perdas de informação importantes, daí estas não terem sido menosprezadas na fase anterior.

De forma a evitar uma redução de dimensões incauta realizou-se primeiramente um PCA sobre a bases de dados, sem quaisquer restrições acerca do número de dimensões desejado para este novo espaço representativo vetorial. Para tal, recorreu-se ao "***Scree test*", como** demonstrado de seguida.

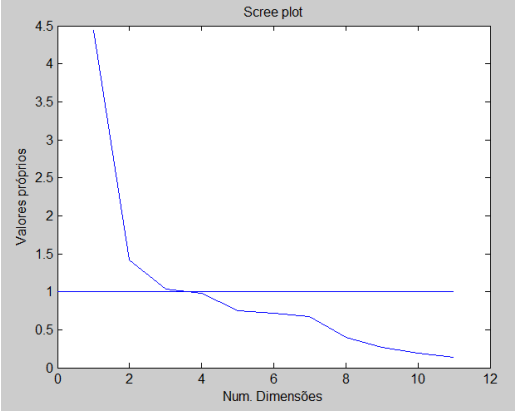

Figura 25 – Scree plot.

A partir do ***Scree plot*** podemos realizar o *Kaiser test* ou o *scree test* de forma a otimizar a escolha do número de dimensões que será considerada na representação dos dados em estudo, tentando que tal representação seja o mais fiel possível à informação inicial disponível.

Neste caso, como não se verifica qualquer zona minimamente estável no *Scree plot* obtido, de maneira a aplicar o *Kaiser test*, decidiu-se escolher o número de componentes principais a utilizar considerando o ***Scree test*** (baseado nos valores próprios), complementado pela **percentagem de variância** que cada componente principal consegue explicar. O número total de dimensões a considerar será igual ao número de componentes principais aqui utilizadas.

Interpretando o *scree plot* aqui obtido, verifica-se que a curva resultante da ligação entre os vários valores próprios intercepta a recta y=1 em x=3 (aproximadamente **3 dimensões**), que corresponde a cerca de 62,7% da variância explicada a partir das 3 principais componentes, que não é considerada suficiente para o trabalho em causa. Como tal, de forma a ter em conta este critério de forma mais preponderante, para uma futura apresentação de resultados mais fiéis, resolveu-se testar o PCA para **5 dimensões**, que tem cerca de 80% da variância explicada.

No entanto, na prática (a partir da 1ª parte deste projecto) verificou-se que não existiam grandes diferenças, em termos de assertividade no que diz respeito à classificação dos dados, entre a escolha de 3 ou 5 dimensões. Como tal, de modo a facilitar **interpretações** e diminuir a **complexidade computacional**, utilizam-se **3 componentes** na restante análise.

Classificação

Terminada a seleção de *features* inicia-se a etapa de classificação de dados. Aqui, com o intuito de explorar os vários algoritmos existentes e respetiva performance testar-se-á classificação **supervisionada** e **não supervisionada**.

Para tal, criaram-se várias combinações de grupos de treino e teste, cada um com igual número de dados pertencentes às diferentes classes.

Começamos com **4 tipos de classificação supervisionada**, onde se destaca:

- Classificação supervisionada:
  - Funções de distância: **Mahalanobis** e **Euclideana**;
  - **Análise discriminante linear** (LDA);
  - ***K-nearest neighbor*** (K-NN);
  - ***SVM;***
  - Classificador Bayesiano.

Nota para o facto de se ter realizado um ***downsample*** dos dados, para alguns dos algoritmos mais "pesados" a nível computacional, devido às suas grandes dimensões (como se verificou necessário para o K-NN). Isto está associado a uma perda de informação que será tida em conta nas conclusões finais.

- LDA

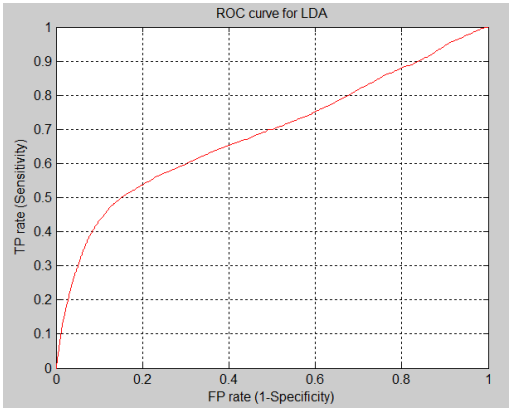

Figura 26 – Curva ROC para classificador LDA (a 3 dimensões)

Para este primeiro caso, a partir da curva ROC é possível verificar que para valores altos de sensibilidade (como pretendido) existem também valores bastante elevados de "FP rate" (aproximadamente uma proporcionalidade direta), o que está associado a uma **incerteza** bastante elevada nos resultados obtidos, o que é indesejável.

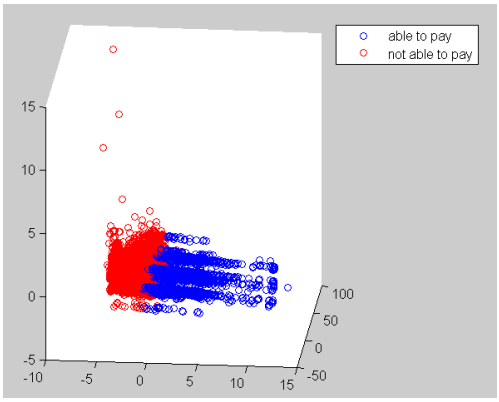

Figura 27- Distribuição da classificação dos vários pontos, segundo as 3 dimensões determinadas a partir do PCA, resultantes do LDA.

Tabela 3 – Matriz confusão para o LDA a 3 dimensões. A horizontal estão as labels previstas e na vertical as labels verdadeiras.

LDA3	Pos	Neg
'Pos'	4174	2462
'Neg'	8327	15037

Tabela 4 – Valores obtidas para a sensibilidade e especificidade no LDA para 3 dimensões.

sensibilidade	especificidade	accuracy
0.62899	0.6436	0.64037

- Mahalanobis

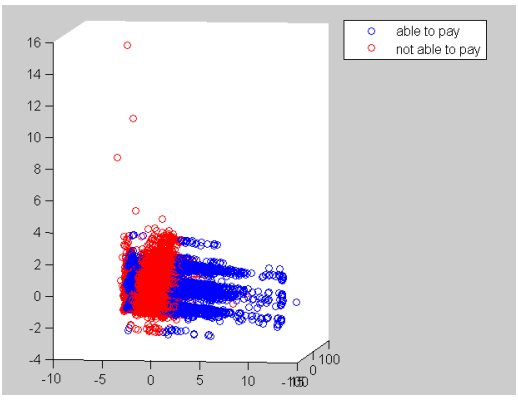

Figura 28 – Distribuição das classificações do vários pontos, segundo as 3 dimensões obtidas com o PCA, de acordo com a função distância Mahalanobis;

Tabela 5 – Matriz confusão dos resultados da classificação com a função distância Mahalanobis;

MAH3	Pos	Neg
'Pos'	4245	2391
'Neg'	9109	14255

Tabela 6 – Valores obtidos para a sensibilidade, especificidade e accuracy respeitantes à função Mahalanobis;

sensibilidade	especificidade	accuracy
0.63969	0.61013	0.61667

- <u>Euclidean</u>

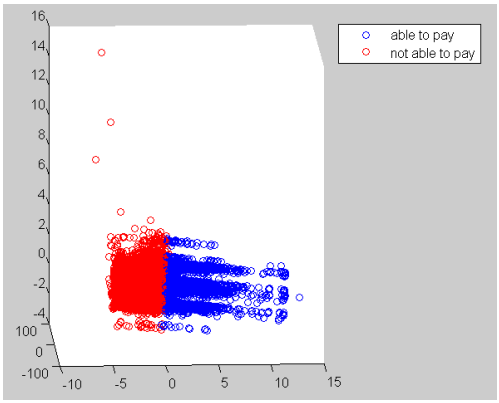

Figura 29 – Distribuição da classificação dos vários pontos, para as 3 componentes principais obtidas com o PCA, resultante da função distância Euclideana;

Tabela 7 – Matriz confusão para a função de distância Euclideana.

EUC3	Pos	Neg
'Pos'	4211	2425
'Neg'	8312	15052

Tabela 8 - Valores obtidos para a sensibilidade, especificidade e accuracy respeitantes à função Euclideana.

sensibilidade	especificidade	accuracy
0.63457	0.64424	0.6421

- *K-Nearest Neighbour (leave-one-out cross validation* com k=5)

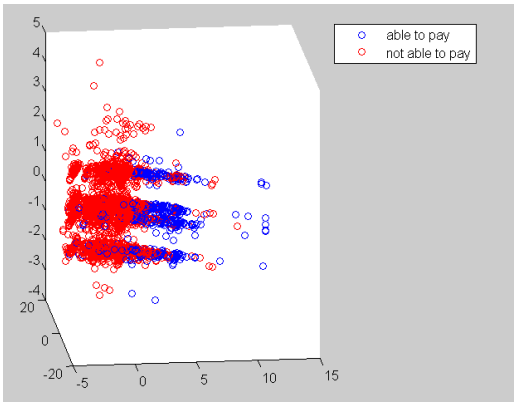

Figura 30 – Distribuição da classificação dos vários pontos, para as 3 componentes principais obtidas com o PCA, resultante da função K-Nearest Neighbour;

Tabela 9: Valores obtidos para a sensibilidade, especificidade e accuracy respeitantes à função K-NN.

Sensitivity	Specificity	Accuracy
0.37195	0.9152	0.79048

- SVM

Tabela 10: Valores obtidos para a sensibilidade respeitantes ao SVM para a Kernel Function 'linear'. Gamma corresponde ao Kernel Parameter e C é a constante de regularização.

	Gamma				
C	0.1	1	10	100	1000
0.1	0,6284	0,6273	0,5903	0,9526	0,9526
1	0,6282	0,6276	0,6205	0,5076	0,9526
10	0,0039	0,6284	0,6273	0,5903	0,9526
100	0,0921	0,6282	0,6276	0,6202	0,5076
1000	0,4284	0,4280	0,6286	0,6273	0,5903

Tabela 11: Valores obtidos para a especificidade respeitantes ao SVM para a Kernel Function 'linear'. Gamma corresponde ao Kernel Parameter e C é a constante de regularização.

	Gamma				
C	0.1	1	10	100	1000
0.1	0,6591	0,6602	0,7314	0,0874	0,0874
1	0,6596	0,6602	0,6690	0,8558	0,0874
10	0,9992	0,6591	0,6602	0,7314	0,0874
100	0,8771	0,6594	0,6601	0,6689	0,8558
1000	0,8706	0,8508	0,6591	0,6602	0,7314

Tabela 12: Valores obtidos para a sensibilidade respeitantes ao SVM para a Kernel Function 'rbf'. Gamma corresponde ao Kernel Parameter e C é a constante de regularização.

	Gamma				
C	0.1	1	10	100	1000
0.1	0,8840	0,5033	0,4564	0,6915	0,9524
1	0,6484	0,6006	0,4590	0,5860	0,9524
10	0,6392	0,6256	0,4728	0,6026	0,9524
100	0,6510	0,6239	0,4753	0,6073	0,5877
1000	0,6687	0,6047	0,5003	0,4945	0,6047

Tabela 13: Valores obtidos para a especificidade respeitantes ao SVM para a Kernel Function 'rbf'. Gamma corresponde ao Kernel Parameter e C é a constante de regularização.

	Gamma				
C	0.1	1	10	100	1000
0.1	0,2674	0,8558	0,8889	0,4952	0,0885
1	0,5941	0,7692	0,8905	0,7548	0,0885
10	0,5633	0,7381	0,8815	0,7036	0,0885
100	0,5390	0,7220	0,8788	0,6936	0,7514
1000	0,5138	0,7244	0,8545	0,8540	0,6972

○ Classificador Bayesiano

De seguida, com base no *Maximum Likelihood estimation of parameters of Gaussian mixture model* (calculado pela função MATLAB mlcgmm) que é um modelo probabilístico que assume que toda a DATA aqui presente é gerada a partir de uma mistura de distribuições Gaussianas finitas, testa-se este classificador bayesiano:

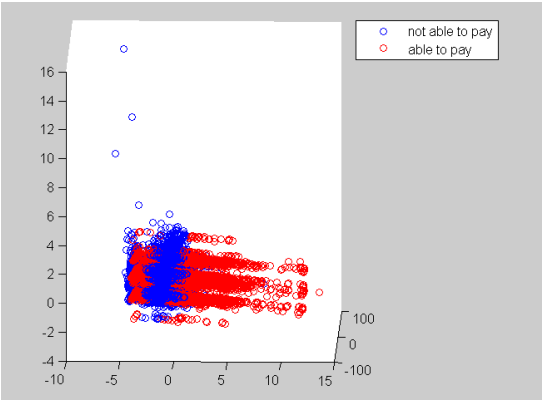

Figura 31: Distribuição da classificação dos vários pontos, para as 3 componentes principais obtidas com o PCA, resultante da Classificação *Bayesiana;*

Tabela 14: Matriz confusão resultante da Classificação *Bayesiana*

STAT3	Pos	Neg
'Pos'	4932	1704
'Neg'	12830	10534

Tabela 15: Valores obtidos para a sensibilidade, especificidade e accuracy respeitantes à Classificação *Bayesiana*

sensibilidade	especificidade	accuracy
0.74322	0.45086	0.51553

De forma a contrastar com esta primeira parte de classificação supervisionada, resolveu-se experimentar um método não supervisionado.

- Classificação não supervisionada:
  - **Hierarchical Clustering**

No Hierarchical Clustering foram realizadas algumas parametrizações das quais se destaca aquela com função distância *correlation* e *linkage weighted* (figura 29).

- Hierarchical Clustering

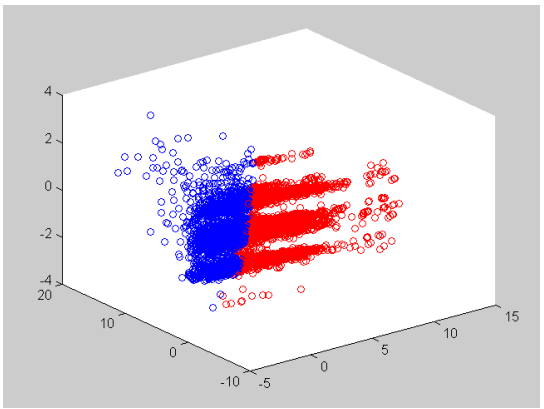

Figura 32: Distribuição da classificação dos vários pontos para as 3 componentes principais obtidas com o PCA, de acordo com Hierarchical Clustering. **Parametrização:** função distância *correlation* e *linkage* weighted

Tabela 16: Matriz confusão para Hierarchical Clustering.

HIER3	Pos	Neg
'Pos'	1776	4860
'Neg'	2979	3656

Tabela 17: Valores obtidos para a sensibilidade, especificidade e accuracy respeitantes a hierarchical Clustering

sensibilidade	especificidade	accuracy
0.26763	0.55102	0.40931

Discussão

No caso em estudo é pretendida, da forma mais fiel possível, a distinção entre indivíduos que **conseguem repor o dinheiro** que lhes foi emprestado pelo banco dos que **não conseguem**, pelo que se deseja a obtenção de uma alta taxa de Verdadeiros Positivos ("TP rate") e Verdadeiros Negativos em conjunto com uma baixa taxa de Falsos Positivos ("FP rate") e Falsos Negativos, distinguindo-se a ***accuracy*** como a métrica mais apropriada para tal necessidade, como é apresentado na tabela seguinte:

Tabela 18: Accuracy obtida em todos os algoritmos de classificação

Algoritmo	LDA	Mahalanobis	Euclidean	K-NN	Bayes	SVM (linear/rbf)	Hierarchical
Accuracy(%)	64,03	61,66	64,21	79,05	51,55	65/73	40,93

Passando à interpretação dos dados em estudo, inicia-se por analisar a forma como se deu a **redução de features**. Verificou-se que a situação inicialmente experimentada de visualização dos **boxplots** para cada variável não nos permitia analisar a preponderância que cada uma poderia ter na separação das diferentes classes. Para todas as variáveis categóricas verifica-se semelhante posicionamento relativo entre as "caixas" (gama entre o quartil de 25% e o de 75%). Já em relação às variáveis contínuas existem certas diferenças nos histogramas para cada classe (barras maiores para classe 0 por exemplo) que poderão influenciar positivamente a nossa análise futura.

Considerando uma taxa de **AUC** mínima para aceitação da variável de 80%, verifica-se que nenhuma chega mesmo a atingir este valor. Na melhor das hipóteses, obtém-se uma taxa de 61%, o que está muito longe do esperado. Dados estes resultados inconclusivos, resolveu-se continuar a análise apenas com as *features* discriminadas pelo método de Kruskal-Wallis, ignorando a AUC.

Prosseguindo a nossa análise com o arranjo da informação disponível em si, a **divisão equitativa dos dados** em termos de classes foi feita no sentido de obter um treino e teste dos classificadores completo (obtenção do melhor modelo de classificação possível) e evitar erros típicos associados a uma má distribuição dos dados pelos grupos mencionados (criando tendências indesejáveis que favoreça a classe com maior número de amostras por exemplo). Isto verificou-se inicialmente com o *Hierarchical Clustering* que favorecia o agrupamento de dados com *labelling* negativo (especificidade próxima dos 100%) em detrimento do reconhecimento de Verdadeiros positivos (sensibilidade próxima dos 0%).

Para alguns casos foi também necessário realizar um **downsample** dos dados como é exemplo o K-NN, de forma a evitar problemas de computação associados à validação cruzada *leave-one-out* que aqui é realizada e que envolve múltiplas combinações entre grupo treino e teste. De forma semelhante, o *Hierarchical Clustering* foi também testado sobre um grupo de dados de menores dimensões devidos às repetidas notificações de escassez de memória no que diz respeito ao processamento das funções distância.

Comentando os resultados finais, face à primeira mudança realizada em relação à primeira parte do projecto, em termos da construção do grupo treino e teste **não se verificam melhorias significativas** ao nível da eficácia classificativa (exemplo do algoritmo **LDA, *Euclidean distance*)** tendo até havido uma **diminuição de *accuracy*** por exemplo para **o *Mahalanobis***. No entanto, pode-se destacar, mais uma vez, pela positiva o *Hierarchical Clustering,* cuja sensibilidade passou de uma taxa nula para 26%, devido ao aumento do número de dados pertencentes à classe positiva, demonstrativa da desigualdade existente anteriormente em termos de constituição do grupo sujeito a classificação.

Ao nível das **parametrizações** experimentadas, o *Hierarchical Clustering* foi mais uma vez o algoritmo que demonstrou uma diferença em termos de performance mais notória (algo que como já mencionado estava também fortemente ligado à má construção do grupo a testar), tendo havido uma mudança decisiva ao nível do método como é "desenhada" a árvore hierárquica para este tipo de cluster (parametrização mencionada na figura 32).

Ainda a este nível, referência para a forma como no K-NN foram escolhidos os k *neighbours,* feita com base em conclusões retiradas a partir de artigos lidos, por experiências anteriores no que diz respeito ao uso deste algoritmo e por experimentação de algumas parametrizações neste trabalho (tendo-se apresentado apenas resultados associados a uma parametrização para a qual k=5).

Considera-se que aqui a **distribuição dos dados** (demasiado juntos) não contribui para uma correta classificação, que por ser realizada com base em distâncias complica todo o processo. Alguns tipos de parametrização experimentados apresentavam melhores valores para a *accuracy,* no entanto, tal devia-se a uma sensibilidade quase nula e a uma especificidade muito grande provenientes do facto de todos os dados serem classificados com uma só classe (dada a proximidade entre dados).

Para além disto, existe sempre um erro associado à existência de ***outliers*** (que também aqui se verificam), cujas distâncias ao ponto de referência influenciam os resultados finais obtidos. Numa tentativa de reparar tal defeito excluiu-se qualquer dado (*outlier*) demasiado

afastado do resto da DATA (ação realizada apenas para o Hierarchical Clustering, como está presente no código MATLAB), tendo-se verificado na mesma uma incapacidade deste algoritmo em se adaptar à distribuição dos dados.

Realizando uma análise quantitativa comparativa dos algoritmos, em termos gerais é possível desde logo verificar uma uniformidade existente nos resultados obtidos para os variados métodos. Mais especificamente, e considerando que os resultados associados a todas as métricas variam entre **0** e **1**, verificou-se que o valor de **especificidade** obtido para o K-NN **(91,5%)** terá sido o mais elevado, dentre os métodos considerados. Em termos de **sensibilidade**, todos os métodos rondaram valores entre os 26**%** (*Hierarchical Clustering*) e os **74%** (Bayesiano), o que também não é de todo positivo. Por fim, a ***accuracy*** atinge um valor máximo de **79%** para o K-NN**,** destacando-se dos restantes algoritmos cuja performance fica muito distante do desejável.

Apesar do valor elevado de *accuracy* obtido para o K-NN, o SVM (que seria o algoritmo no qual se depositou mais expectativas) apresenta também uma boa performance. Dentre as parametrizações realizadas, pode-se destacar aquela proveniente do **Kernel Function 'rbf'** (com um **gamma=1 e C=1**) com uma **especificidade de 76%,** uma **sensibilidade de 60% e uma *accuracy* de 73%.** Verificam-se em variadas ocasiões especificidades e sensibilidades elevados, no entanto, esta aqui escolhida é a que apresenta melhor combinação das três métricas mencionadas.

No cômputo geral, o ***Hierarchical Clustering*** apresenta-se como o **pior** classificador, não chegando a *accuracy* a atingir os 50%.

Para quase todos os métodos de classificação utilizados verificou-se uma **boa distinção de 2 grupos** (sem grandes misturas na disposição dos pontos de diferentes cores, classes, no *plot* a 3D). No entanto, mais uma vez se salienta o facto de os dados pertencentes a estas 2 classes se encontrarem tão próximos o que poderá explicar os **maus resultados** obtidos em termos de **sensibilidade, especificidade e *accuracy***. Ou seja, apesar dos dados terem sido bem agrupados, tal agrupamento foi feito segundo um "*labelling*" errado.

Como possíveis **fontes de erro** poderemos apresentar o ***Downsampling*** realizado para alguns dos métodos (por exemplo *Hierarchical Clustering*, K-NN) que poderá ter levado a uma diminuição dos dados demasiado elevada, comprometendo a obtenção de uma correcta classificação.

Já a *accuracy* de **51%** associada ao classificador **Bayesiano** poderá ter a sua origem na "**não-gaussianidade**" verificada para qualquer uma das 3 *features* em uso (teste realizado a partir do algoritmo de **Kolmogorov-Smirnov**, critério complementar do classificador **Bayesiano**), o que por si só é já uma premonição para os resultados menos positivos que se obtêm no final.

Em relação à classificação não supervisionada, dada a distribuição espacial dos dados já mencionada (onde não é possível distinguir de imediato dois grupos de dados), desde logo se poderia ter uma noção de que o *Hierarchical clustering* poderia levar a conclusões erradas. Para além disso, a ausência de um **grupo treino** (característica deste tipo de classificação) impede a obtenção de resultados com melhor qualidade (com a criação de um modelo de classificação).

Num cenário bancário, onde uma boa gestão do dinheiro aí depositado é prioridade máxima, de maneira a que seja construída uma boa imagem à volta das políticas praticadas pelo mesmo (permitindo progressão), é necessário que a eficácia com que se realiza o *labelling* dos clientes (aqui estudado) se aproxime o mais possível do perfeito. Só assim se conseguirá verdadeiramente fidelizar os clientes actuais e atrair novos clientes.

Neste sentido, e tendo em conta os maus resultados já denunciados, poder-se-ia escolher o método **K-NN** ou o **SVM** como os **mais aconselháveis** a ser usados por parte dos informáticos no banco de Taiwan. Uma vez mais salienta-se que mesmo estes métodos se apresentam como **insuficientes** para uso em situações reais, hipotecando as intenções de se realizar uma eficaz discriminação.

Conclusão

Como primeira constatação, um breve reparo para as modificações realizadas com sucesso nesta 2ª parte do projecto:

- Utilização de técnicas de classificação mais versáteis (como por exemplo SVM);
- Utilização de grupos de treino e de teste com igual tamanho e tratamento (que por interpretação dos resultados não contribuíram para qualquer tipo de optimização significativo);
- Redução da influência de outliers (como foi tentado no *Hierarchical Clustering,* sem grande sucesso no entanto).

Em relação aos resultados obtidos admite-se que houve **insucesso** em relação ao pretendido. Apesar de ter sido feita uma separação razoável dos vários pontos obtidos a partir da conjugação das componentes do PCA (algo visível nos vários gráficos obtidos a partir do *clustering* realizado para 3 dimensões), ou seja, sem misturas de pontos pertencentes a diferentes classes, o respetivo "*labelling*" realizado não correspondeu em termos de eficácia, como se verifica a partir dos baixos valores de sensibilidade, especificidade e *accuracy* obtidos em termos gerais.

Interpretando estes resultados no que diz respeito ao problema em causa, a sensibilidade, especificidade e *accuracy* apresentam-se respectivamente como a melhor métrica para:

1. **Identificação dos clientes cumpridores**, que poderá em casos extremos levar à perda do cliente (que procurará outros bancos com vista a obter tal empréstimo);
2. **Identificação dos clientes incumpridores**;
3. **Reconhecimento geral** da avaliação realizada pelo banco de Taiwan em relação às condições apresentadas por cada um dos seus clientes;

A obtenção de baixos valores para qualquer uma destas métricas é indesejável para uma saudável manutenção do banco, principalmente a longo prazo:

1. A percentagem de clientes capazes de pagar o seu empréstimo, mas que não foram reconhecidos como tal, o que por sua vez impossibilitou a realização desse mesmo empréstimo, **perdendo-se uma oportunidade de negócio**.
2. **Mau investimento do dinheiro do banco**, que dificilmente será reavido, tendo os restantes clientes que suportar tais consequências;

Denunciados e explicados os maus resultados, e consequentes implicações, é necessário explorar possíveis soluções que numa próxima fase deste projecto nos possam garantir melhores *outcomes*, entre as quais se destacam:

1. Redução da influência dos *outliers* mais eficaz (poderá levar a perdas de informação ou de casos exclusivos de possível interesse) seguindo exemplo do que foi feito no *Hierarchical Clustering*;
2. Experimentação mais abrangente de diferentes critérios de eliminação de variáveis (com o objectivo de obter *features* mais discriminativas);
3. Modificação das variáveis contínuas para categóricas (nominais ou ordinais), o que poderá ser mais difícil dada a perda de informação que lhe é subjacente (e que não será aqui realizado dada a nossa inexperiência em tal, o que poderia sabotar possíveis resultados);
4. Utilização de uma classificação baseada numa validação cruzada (apenas realizada para o K-NN que por sinal apresentou os melhores resultados) onde numa primeira fase se criaria um modelo de classificação para o grupo de dados em estudo (com base num grupo de treino e teste), e posteriormente a inclusão de um grupo de validação, cuja avaliação de *accuracy* seria realmente o alvo de interesse (algo que aqui na maior parte das vezes não é possível devido às especificações das funções MATLAB usadas, que impedem classificação de um grupo validação de acordo com probabilidades a priori);
5. Realização de outro tratamento inicial dos dados (normalização);
6. Realização de um maior número de parametrizações (como por exemplo na *linkage* realizada no Hierarchical Clustering ou a escolha do k no K-NN, que aqui por ser tão elevado poderá ter levado a um *overfitting* dos resultados).

Como nota final, dado o caracter deste projeto, onde se procura a implementação correta de todos os algoritmos e sua interpretação, considera-se que os objetivos foram cumpridos.