Project Assignment - Default of credit card clients

Diogo A. Rosário[1] - uc2023185395, João E.M. Raposo[2] - uc2023147060

14 de maio de 2024

UNIVERSIDADE D
COIMBRA

Conteúdo

1 Introdução

No cenário financeiro global, as instituições bancárias enfrentam desafios constantes na gestão de riscos associados ao crédito, especialmente quando se trata da incapacidade dos clientes em relação aos pagamentos de cartões de crédito. A capacidade de prever com precisão se um cliente será capaz de cumprir as suas obrigações financeiras futuras é uma necessidade crucial para garantir a estabilidade e a sustentabilidade dos serviços financeiros. Neste contexto, a análise preditiva torna-se uma ferramenta indispensável, empregando técnicas avançadas de aprendizado de máquina e extração de dados para identificar padrões e tendências nos dados dos clientes. Em particular, a classificação binária emerge como uma abordagem eficaz para prever o risco de incapacidade, onde o objetivo é categorizar os clientes em duas classes distintas: capazes ou incapazes de cumprir com o pagamento do crédito.

2 Objetivo

Neste trabalho propõe-se explorar e desenvolver classificadores para prever se um determinado cliente será capaz de pagar (ou não) o crédito que arrecadou no próximo mês, com base num conjunto de dados coletados em Taiwan. Fazendo uso de técnicas de aprendizagem / máquina, o estudo visa investigar a relação entre as características individuais dos clientes e a probabilidade de incapacidade, fornecendo informações valiosas para as instituições financeiras na tomada de decisões estratégicas e na mitigação de riscos.

Ao longo deste trabalho, serão abordados aspectos fundamentais da construção e avaliação de modelos preditivos, incluindo a seleção e engenharia de características relevantes, a aplicação de algoritmos de aprendizagem / máquina, a validação e otimização do desempenho do modelo, bem como a interpretação dos resultados obtidos.

3 Dataset

O dataset utilizado neste estudo foi compilado em outubro de 2005 e contém informações de 30.000 clientes de um banco significativo em Taiwan. Os dados estão disponíveis publicamente no seguinte link: [Yeh16]. Dos 30.000 clientes analisados, 5.529 deles (22,12%) foram identificados como portadores de cartões de crédito com histórico de pagamento atrasados. Esses dados oferecem uma perspectiva abrangente do comportamento financeiro dos clientes em relação aos seus pagamentos através do uso cartão de crédito.

O dataset inclui 23 características (*features*) que foram cuidadosamente selecionadas para distinguir entre clientes que são capazes de pagar suas faturas de cartão de crédito no próximo mês e aqueles que podem enfrentar dificuldades de pagamento. Essas características estão resumidas na Tabela 1 abaixo.

ID	Name	Possible Values
X1	Amount of the given credit	Includes both the individual consumer credit and his/her family (supplementary) credit in dollars.
X2	Gender	1 = male; 2 = female
X3	Education	1 = graduate school; 2 = university; 3 = high school; 4 = others.
X4	Marital status	1 = married; 2 = single; 3 = others.
X5	Age	Age in years.
X6 - X11	History of past payment from April to September, 2005: X6 = the repayment status in September, 2005 X7 = the repayment status in August, 2005 ... X11 = the repayment status in April, 2005.	-1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; 8 = payment delay for eight months; ... 9 = payment delay for nine months and above.
X12-X17	Amount of bill statement: X12 = in September, 2005; X13 = in August, 2005; . . . X17 = in April, 2005.	Amount in dollars.
X18-X23	Amount of previous payment. X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; ... X23 = amount paid in April, 2005	Amount in dollars.

Figura 1: Tabela das *Features*

4 Experiência

Neste capítulo, pretendemos descrever como toda a experiência foi preparada e organizada.

A experiência começou com a preparação do ambiente de trabalho. Optamos por uma abordagem colaborativa e eficiente, utilizando o Google Drive como plataforma principal. Criamos uma pasta dedicada e organizamos os diretórios de forma a facilitar o acesso aos dados garantindo que todos os membros da grupo pudessem trabalhar de forma colaborativa e acessar os dados de maneira transparente.

Com o ambiente devidamente configurado, mergulhámos na análise inicial dos dados. Utilizamos o Google Colab como ambiente de desenvolvimento, aproveitando a sua perfeita sincronização e integração com o Google Drive. Importamos todas as dependências necessárias e começamos a explorar e analisar os dados.

5 Ferramentas de Análise

No âmbito do nosso projeto de *machine learning*, foram utilizados duas ferramentas de análise que desempenham um papel fundamental na compreensão e modelagem dos dados. Entre estas ferramentas destacam-se o *scikit-learn* (*sklearn*) e o *Matplotlib*.

O *scikit-learn* é uma biblioteca em *Python* que oferece uma vasta gama de algoritmos de machine learning e ferramentas para pré-processamento de dados e avaliação de modelos. Com o *scikit-learn*, é possível implementar algoritmos de classificação, regressão, *clustering* e outras técnicas de análise de dados. Além disso, esta biblioteca proporciona uma interface simples e consistente, o que facilita bastante o desenvolvimento e a experimentação com diferentes modelos.

Juntamente com o *scikit-learn*, foi utilizado o *Matplotlib* para visualização de dados. O *Matplotlib* é uma biblioteca de visualização em *Python* que permite criar gráficos de alta qualidade para representar os dados de forma clara e compreensível. Com o *Matplotlib*, consegue-se gerar gráficos de dispersão, histogramas, gráficos de barras e muitos outros tipos de visualizações, que são essenciais para analisar padrões nos dados, identificar relações entre variáveis e comunicar os resultados do meu projeto.

5.1 Análise de *Features*

5.1.1 *PCA*

Análise de componentes principais (*Principal Component Analysis*) é uma metodologia com base na projeção da informação em direções onde os dados variam mais, ou seja, onde existe uma maior variância. Nestas projeções, são escolhidas as melhores em ordem de reter a maioria da informação e a preservação da informação é medida em termos da variabilidade dos dados.

5.1.2 *LDA*

Análise linear discriminante (*Linear Discriminant Analysis*) é também um técnica de redução de *features* usada com sucesso em muitos problemas estatistícos de reconhecimento de padrões. Tem como objetivo primário separar amostras de grupos distintos ao colocá-los num espaço maximizando a separabilidade entre classe enquanto minimizando a variabilidade dentro deles. Este resultado pode ser usado em redução da dimensionalidade do classificador ou numa classificação linear.

5.1.3 *Kruskal-Wallis*

Para esta experiência utilizamos o teste de **Kruskal-Wallis** [Man22a] [bhu22], que é uma técnica estatística não paramétrica utilizada para determinar a existência de diferenças estatisticamente significativas entre os medianas de duas ou mais amostras independentes. Neste caso específico, o teste de **Kruskal** foi calculado para cada feature do nosso dataset.

Com este teste pretendemos verificar quais das features obtêm valores de **kruskal** mais elevados pois isto indica que existe uma maior probabilidade de as medianas entre os grupos sejam diferentes. Para além disso, valores elevados podem sugerir que uma determinada feature pode ser um bom indicador entre as restantes.

5.1.4 *Kolmogorov-Smirnov*

Ao aplicarmos o teste de Kolmogorov-Smirnov [Man22b] [Ast24] aos resultados obtidos pelos classificadores desenvolvidos, é possível avaliar se as previsões geradas pelos modelos estão de acordo com a distribuição real dos dados. Esta análise é fundamental para verificar se os modelos são capazes de capturar as características essenciais do fenômeno em estudo.

5.1.5 *Gaussian Distribution - Univariate*

A Distribuição Gaussiana, também conhecida como distribuição normal, é uma distribuição de probabilidade contínua que descreve a probabilidade de uma única variável aleatória assumir certos valores. No caso univariado, a distribuição é caracterizada por uma curva em forma de sino simétrica quando plotada, com a média (μ) representando o centro da distribuição e a variância (σ^2) controlando a dispersão dos pontos de dados em torno da média. A função de densidade de probabilidade (PDF) de uma distribuição gaussiana univariada é dada por:

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

5.1.6 *Gaussian Distribuition - Bivariate*

A Distribuição Gaussiana no caso bivariado refere-se a uma distribuição normal multivariada envolvendo duas variáveis aleatórias. Descreve a probabilidade conjunta de observar combinações específicas de valores para as duas variáveis. Semelhante ao caso univariado, é simétrica em torno do seu vetor médio, com contornos elípticos no espaço bidimensional. A função de densidade de probabilidade (PDF) de uma distribuição gaussiana bivariada é dada por:

$$f(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$

5.2 Análise de Classificadores

5.2.1 Matriz de Confusão

A matriz de confusão ajuda a visualizar o desempenho de um modelo de classificação. Ela permite identificar quais os tipos de erros que o o modelo está a cometer e em que quantidades. A matriz é composta por 4 campos principais: verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos.

5.2.2 Exatidão (Accuracy)

Indica a proporção total de classificações corretas:

- (VP + VN) / (Total de amostras)

5.2.3 Precisão (Precision)

Indica a proporção de classificações positivas que estavam corretas:

- VP / (VP + FP)

5.2.4 Sensibilidade (Recall)

Indica a proporção de amostras positivas que foram identificadas corretamente:

- VP / (VP + FN)

5.2.5 *F1-Score*

É uma métrica que tem em conta a precisão e a sensibilidade:

- 2 * ((Precisão * Sensibilidade)/(Precisão + Sensibilidade))

5.2.6 *ROC Curves*

A curva ROC é um gráfico que mostra a relação entre a taxa de positivos verdadeiros (TPR) e a taxa de falsos positivos (FPR) para diferentes valores do limiar de classificação. O limiar de classificação é um valor que é usado para determinar se um evento é positivo ou negativo. Um valor alto significa que um evento precisa ter uma alta probabilidade de ser positivo para ser classificado como positivo, enquanto um valor baixo significa que um evento precisa ter uma baixa probabilidade de ser positivo para ser classificado como negativo.

As curvas ROC podem ser usadas para comparar o desempenho de diferentes modelos de classificação. Elas também podem ser usadas para selecionar o limiar de classificação ideal para um modelo de classificação específico.

Além disso, o teste de *Kolmogorov-Smirnov* será utilizado para comparar diferentes modelos entre si, permitindo identificar qual deles apresenta um desempenho estatísticamente superior em termos de adequação à distribuição dos dados observados. Essa comparação é crucial para a seleção do modelo mais robusto e confiável.

5.2.7 Erro Quadrático Médio *(Mean Squared Error, MSE)*

O Erro Quadrático Médio representa a média da diferença ao quadrado entre os valores originais e previstos no conjunto de dados. Ele mede a variância dos resíduos:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

5.2.8 Raiz do Erro Quadrático Médio *(Root Mean Squared Error, RMSE)*

O Erro Quadrático Médio da Raiz é a raiz quadrada do Erro Quadrático Médio. Ele mede o desvio padrão dos resíduos:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

5.2.9 Erro Absoluto Médio *(Mean Absolute Error, MAE)*

O erro absoluto médio representa a média da diferença absoluta entre os valores reais e previstos no conjunto de dados. Ele mede a média dos resíduos no conjunto de dados:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

6 Classificadores

Para alguns dos classificadores usados neste projeto, é importante referir que foi usado a seguinte distribuição dos dados:

- Para o conjunto de treino foi usado 80% do dataset
- Para o conjunto de teste foi usado 20% do dataset

Neste projeto, é importante referir quais os classificadores que foram usados e que tiveram o seu *output* estudado:

6.1 Classificador - *Minimum Distance*

O classificador de distância mínima é usado para classificar dados desconhecidos em classes que minimizam a distância entre os dados e a classe no espaço multi-característica. A distância é definida como um índice de similaridade de forma que a distância mínima seja idêntica à máxima similaridade. As seguintes distâncias são frequentemente utilizadas neste procedimento:

- **6.1.1 Euclidiana**

  No contexto da classificação de padrões, a distância Euclidiana é comumente empregada para medir a semelhança ou dissimilaridade entre dois elementos de um conjunto de dados, representados por vetores de características. Ela é definida como a raiz quadrada da soma dos quadrados das diferenças entre os elementos correspondentes dos vetores.

- **6.1.2 Mahalanobis**

  Por outro lado, a distância de Mahalanobis é uma generalização da distância Euclidiana que leva em consideração a correlação entre as variáveis e que ao contrário da distância Euclidiana, que assume independência entre as variáveis, a distância de Mahalanobis leva em conta a covariância entre elas. Essa distância é calculada pela normalização das diferenças entre as observações pelas respectivas variâncias e covariâncias.

6.2 Classificador - *Naive Bayes*

O Classificador Naive Bayes é uma técnica popular de aprendizagem / máquina que se baseia no teorema de Bayes para realizar a classificação de dados. Existem várias variantes deste classificador, entre as quais se destacam:

- **6.2.1 Classificador - *Gaussian Naive Bayes***

  Este classificador é especialmente útil quando os dados seguem uma distribuição gaussiana, também conhecida como distribuição normal. Neste método, presume-se que os valores das características são independentes e seguem uma distribuição gaussiana em cada classe.

- **6.2.2 Classificador - *Bernoulli Naive Bayes***

  Este classificador é adequado para dados binários, onde cada característica pode ter apenas dois valores possíveis, como verdadeiro ou falso, 0 ou 1. Este classificador assume que as características são distribuídas de acordo com a distribuição de Bernoulli, que é uma distribuição de probabilidade discreta.

6.3 Classificador - *KNN*

O algoritmo *k-Nearest Neighbors (k-NN)* é uma técnica de aprendizagem / máquina usada para classificação e regressão. Ele classifica pontos novos do conjunto de dados com base na maioria dos seus k vizinhos mais próximos no espaço de características. O valor de k, o número de vizinhos considerados, é um hiperparâmetro que influencia a sensibilidade ao ruído e a suavidade das fronteiras de decisão. Embora simples de entender e implementar, o k-NN pode ser computacionalmente caro em grandes conjuntos de dados e requer a escolha adequada da métrica de distância e de k para um desempenho eficaz.

6.4 Classificador - *Random Forest*

O algoritmo *Random Forest* é uma extensão do método de *bagging*, pois utiliza tanto o *bagging* quanto a aleatoriedade de características para criar uma floresta de árvores de decisão não correlacionadas. A aleatoriedade de características, também conhecida como bagging de características ou "método do subespaço aleatório", gera um subconjunto aleatório de características, o que garante baixa correlação entre as árvores de decisão. [IBM]

6.5 Classificador - *SVM*

As Máquinas de Vetores de Suporte (SVM) são um modelo de aprendizagem supervisionada usado para classificação e regressão. Elas encontram o hiperplano que melhor separa os dados em diferentes classes, maximizando a margem entre elas. As SVM são eficazes em espaços de características de alta dimensão e podem lidar com dados não lineares usando funções de kernel. No entanto, elas podem ser sensíveis à escolha de parâmetros e exigem tempo de treinamento computacional.

6.6 Classificador - *AdaBoost*

O AdaBoost é um algoritmo de aprendizagem de máquina que combina múltiplos classificadores fracos para formar um classificador forte. Ele funciona dando mais peso às instâncias de dados classificadas incorretamente em cada iteração do treinamento, resultando em um classificador final robusto e preciso. O AdaBoost é eficaz em lidar com conjuntos de dados desequilibrados e é menos propenso ao overfitting. No entanto, pode ser sensível a outliers e ruído nos dados, e o tempo de treinamento pode ser mais longo do que outros métodos.

7 Resultados

7.1 Análise de Dados

7.1.1 *PCA*

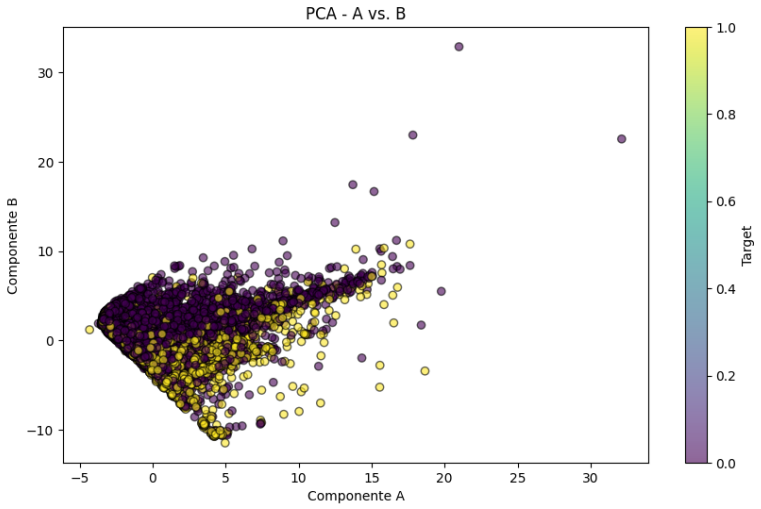

Figura 2: Representação das features 2D com as duas primeiras componentes (X1 e X2)

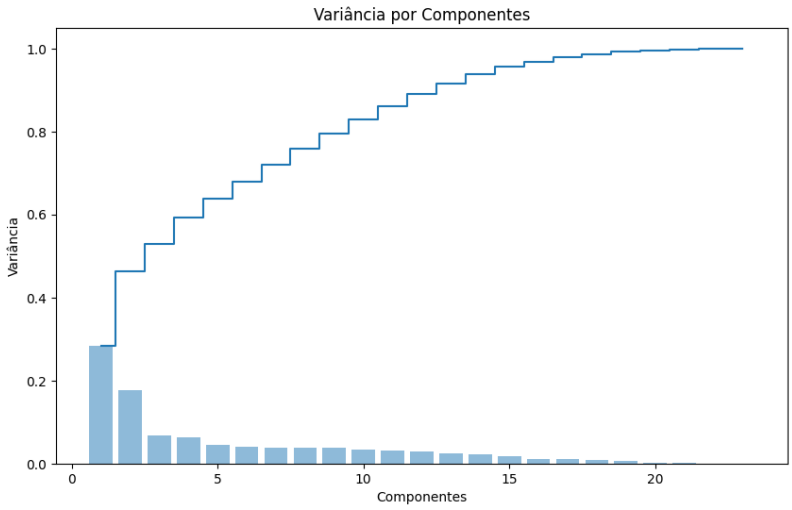

Figura 3: Representação variancia para cada feature

Com os resultados do PCA, decidiu-se usar e diminuir a dimensão do nosso dataset para 9 componentes (mais de metade do número inicial) acabando por se diminuir a complexidade do problema e obtendo uma variância de 79,6 %.

Esta escolha foi feita com base nos resultados observados da figura 3 deste relatório, onde se é possível observar o rácio da variância ao longo do número de componentes.

Na figura 2, observamos o uso das duas primeiras componentes com maior variância para a construção do gráfico bidimensional. As cores indicam a classe de cada cliente, onde o amarelo representam os clientes que são bons pagadores e o azul representa os que não conseguiram pagar. É também possível observar uma certa separação entre as classes, com os bons pagadores concentrados um pouco abaixo das 0 unidades da componente B e os devedores um pouco acima. Esta separação indica que com estas duas primeiras componentes principais já é possível capturar e observar informações discriminativas.

Para concluir, por causa da "maldição"de dimensionalidade, a representação do dataset no PCA não é a melhor, pois apesar de se conseguir ver uma pequena diferenciação, a distribuição dos dados dentro de cada classe parece ser homogênea, sem grandes concentrações em áreas específicas.

7.1.2 *LDA*

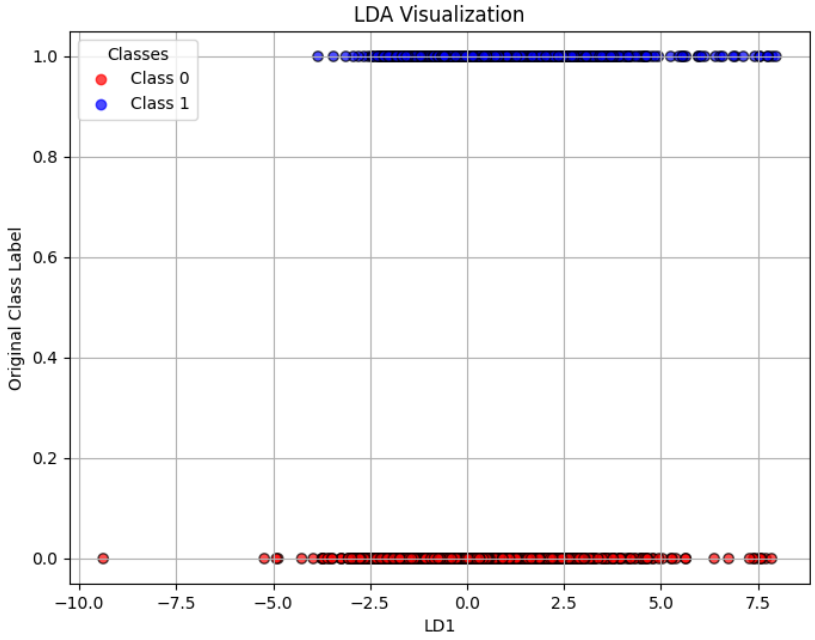

Figura 4: Representação LDA

É possível observar uma boa separação entre as classes no gráfico, com os bons pagadores (class 1) concentrados no lado direito do gráfico e os o que não conseguiram pagar (class 0) no lado esquerdo. Esta separação indica que o LDA é capaz de discriminar entre as classes com base nas variáveis do dataset. A quantidade de pontos sobrepostos é pequena, o que indica que a separação entre as classes é bastante precisa.

7.1.3 *Kruskal-Wallis*

Feature	Valor de Kruskal	Valor p.
X6	2561.57	0.0
X7	1411.56	6.44e-309
X8	1138.03	1.78e-249
X9	905.01	7.99e-199
X1	862.76	1.22e-189
X18	772.71	4.61e-170
X10	758.82	4.85e-167
X19	683.80	9.95e-151
X11	609.36	1.54e-134
X20	582.85	8.99e-129
X21	491.34	7.28e-109
X23	442.44	3.18e-98
X22	407.76	1.12e-90
X3	59.05	1.53e-14
X2	47.90	4.47e-12
X4	21.05	4.47e-6
X12	19.24	1.15e-5
X13	7.24	0.00706
X14	4.81	0.028
X15	2.095	0.15
X16	1.41	0.24
X5	0.79	0.37
X17	0.00017	0.989

Tabela 1: Tabela que contêm os valores calculados a partir do teste de Kruskal-Wallis

Ao examinarmos os valores de ***Kruskal*** na tabela acima para as diferentes características (*features*), observamos uma variação considerável nas suas magnitudes. As características **X6, X7 e X8** (status de reembolso nos últimos 3 meses) apresentam os maiores valores de *Kruskal*, seguidas por **X9, X1** (X9: o status do reembolso em abril, 2005; X1: Valor do crédito concedido) e assim por diante. Isto sugere que estas características possuem variações substanciais entre os grupos que estão a ser comparados.

Por outro lado, características como **X17, X5, X16 e X15** (X5: idade; restantes são valor do extrato da fatura em diferentes meses) têm valores de *Kruskal* relativamente baixos em comparação com as outras. Isto indica que estas características têm variações menos pronunciadas entre os grupos.

No entanto, é importante notar que os valores de *Kruskal* por si só não fornecem informações sobre a direção ou a natureza das diferenças entre os grupos.

Durante o desenvolvimento do projeto, o grupo decidiu então, quando usado este tipo de teste, usar as primeiras 9 caracteristicas mais discriminativas para construir os classificadores.

7.1.4 Kolmogorov-Smirnov

Feature1	Feature2	Estatistica KS	P-value	Significância
X1	X2	1.0	0.0	Significante
X1	X3	1.0	0.0	Significante
X1	X4	1.0	0.0	Significante
X1	X5	1.0	0.0	Significante
X1	X6	1.0	0.0	Significante
X1	X7	1.0	0.0	Significante
X1	X8	1.0	0.0	Significante
X1	X9	1.0	0.0	Significante
X2	X3	0.179033	0.0	Significante
X2	X4	0.060833	1.07×10^{-48}	Significante
X2	X5	1.0	0.0	Significante
X2	X6	0.772733	0.0	Significante
X2	X7	0.852067	0.0	Significante
X2	X8	0.859567	0.0	Significante
X2	X9	0.883	0.0	Significante
X3	X4	0.168267	0.0	Significante
X3	X5	1.0	0.0	Significante
X3	X6	0.772267	0.0	Significante
X3	X7	0.8516	0.0	Significante
X3	X8	0.8591	0.0	Significante
X3	X9	0.882533	0.0	Significante
X4	X5	1.0	0.0	Significante
X4	X6	0.770933	0.0	Significante
X4	X7	0.850267	0.0	Significante
X4	X8	0.857767	0.0	Significante
X4	X9	0.8812	0.0	Significante
X5	X6	1.0	0.0	Significante
X5	X7	1.0	0.0	Significante
X5	X8	1.0	0.0	Significante
X5	X9	1.0	0.0	Significante
X6	X7	0.079333	1.46×10^{-82}	Significante
X6	X8	0.086833	7.49×10^{-99}	Significante
X6	X9	0.110267	2.67×10^{-159}	Significante
X7	X8	0.0101	0.0931	Não Significante
X7	X9	0.030933	6.65×10^{-13}	Significante
X8	X9	0.023433	1.38×10^{-7}	Significante

Tabela 2: Resultados do teste KS

Os resultados do teste ***KS (Kolmogorov-Smirnov)*** indicam a comparação da distribuição empírica de pares de características (*features*) em relação à hipótese nula de que elas são amostradas da mesma distribuição.

A maioria dos pares de características (*Feature 1* e *Feature 2*) possui um valor de estatística KS igual a 1.0, indicando que há uma diferença completa entre as distribuições dessas características. Isto sugere que essas características têm comportamentos distintos ou padrões de valores que são facilmente distinguíveis.

7.1.5 *Gaussian Distribution - Univariate*

Para a distribuição gaussiana univariada, utilizamos as 9 características mais discriminantes obtidas pelo teste de Kruskal. Para cada característica, elaboramos o gráfico da distribuição gaussiana univariada

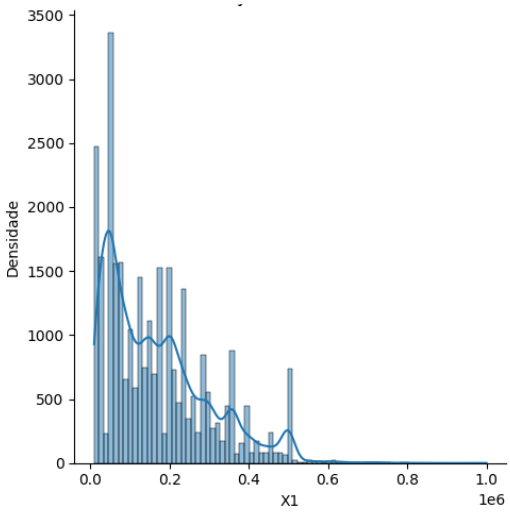

Figura 5: Distribuição Gaussiana Univariada para X1

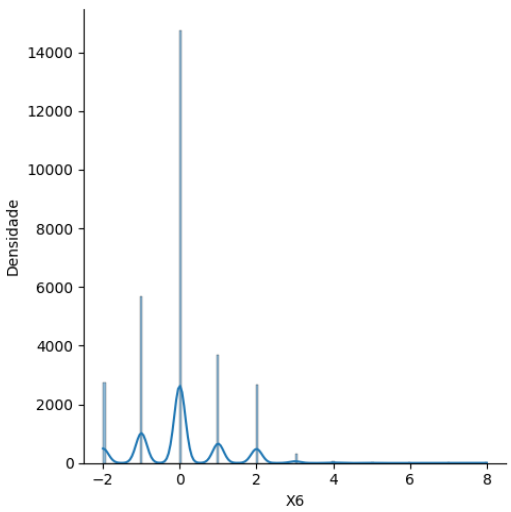

Figura 6: Distribuição Gaussiana Univariada para X6

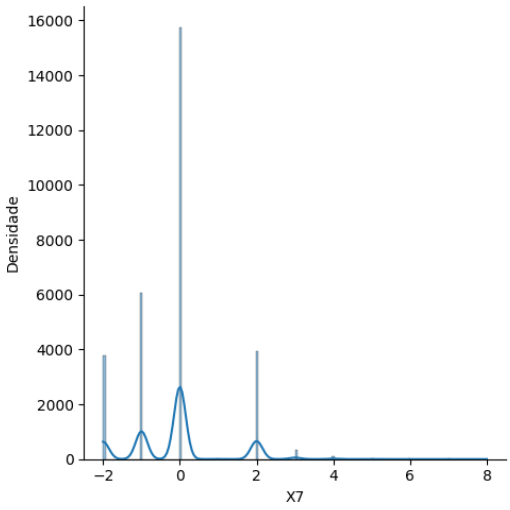

Figura 7: Distribuição Gaussiana Univariada para X7

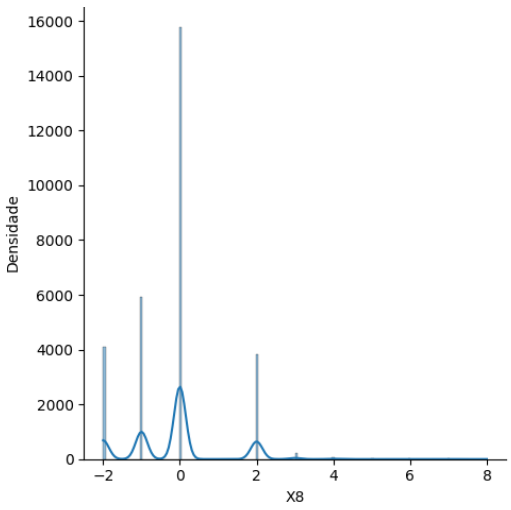

Figura 8: Distribuição Gaussiana Univariada para X8

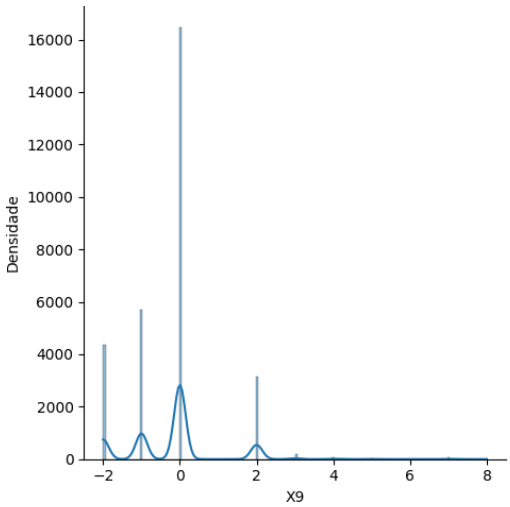

Figura 9: Distribuição Gaussiana Univariada para X9

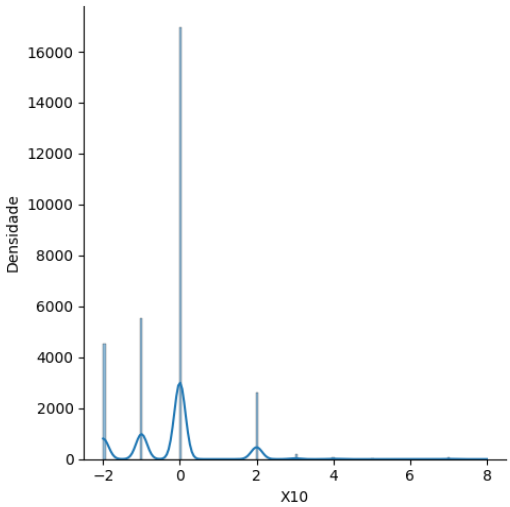

Figura 10: Distribuição Gaussiana Univariada para X10

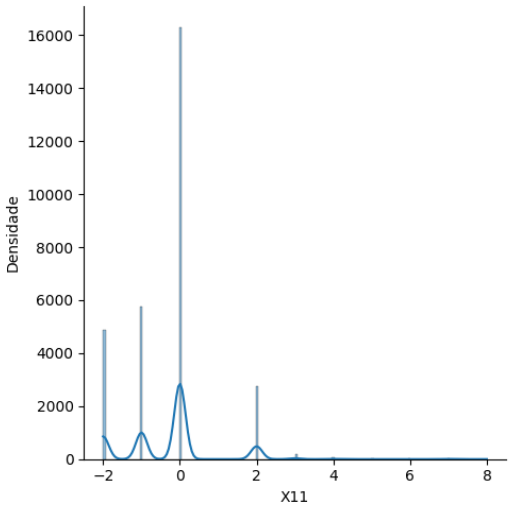

Figura 11: Distribuição Gaussiana Univariada para X11

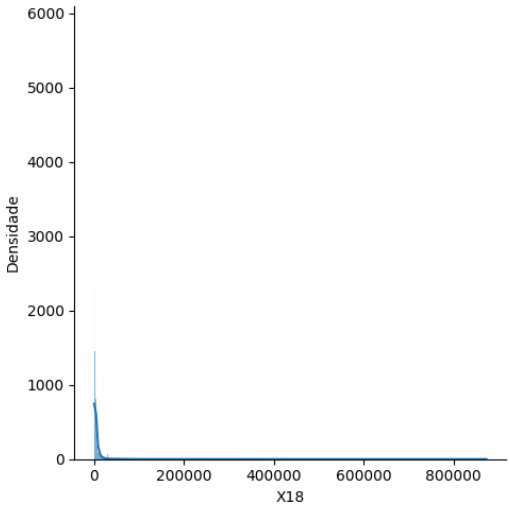

Figura 12: Distribuição Gaussiana Univariada para X18

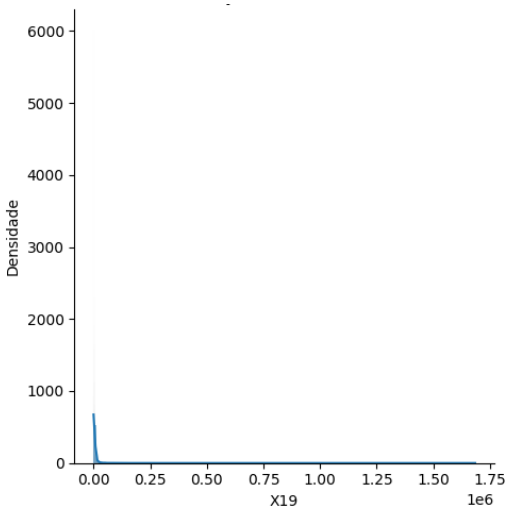

Figura 13: Distribuição Gaussiana Univariada para X19

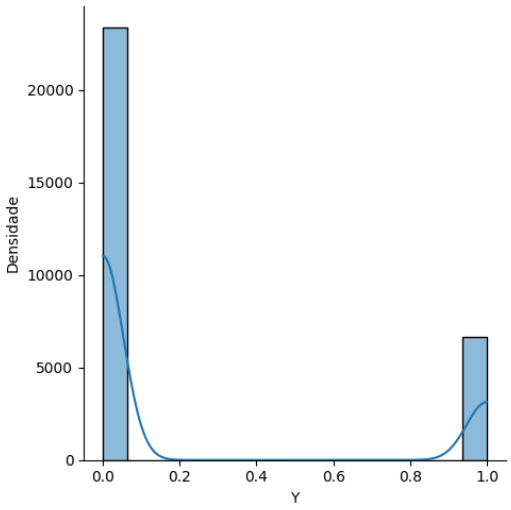

Figura 14: Distribuição Gaussiana Univariada para o Target

7.1.6 *Gaussian Distribuition - Bivariate*

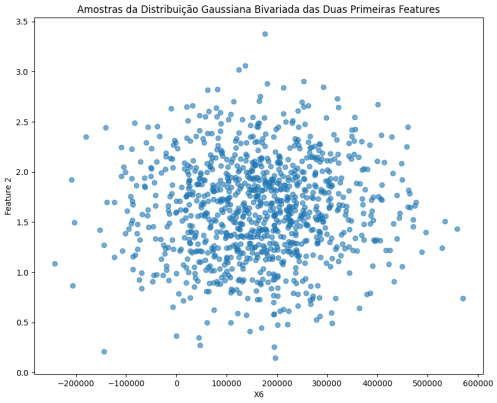

Figura 15: Distribuição Gaussiana Bivariada para X1 e X2

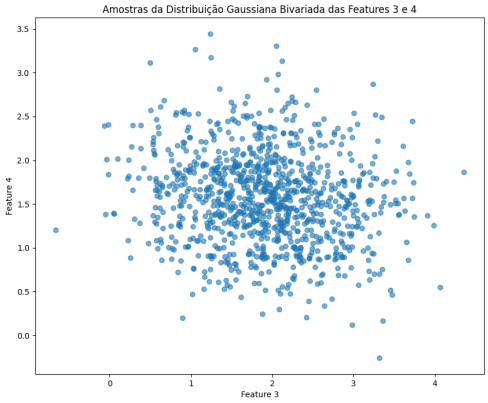

Figura 16: Distribuição Gaussiana Bivariada para X3 e X4

7.2 Classificadores

7.2.1 Classificador - *Minimum Distance with Fisher LDA*

Para criarmos o classificador *Minimum Distance with Fisher LDA*, utilizamos os dados de treino, incluindo as 9 características mais discriminantes resultantes do teste de *Kruskal*. Para além disso, é importante notar que os resultados vão ser os mesmos para o uso da distância *euclidian*, tanto para a de *mahalanobis*. Apresentamos agora os resultados da análise do classificador:

Accuracy	81.05%
Specificity	97.54%
Precision	71.67%
Recall	22.16%
F1 Score	31.85%
Mean Squared Error	18.95%
Root Mean Squared Error	43.53%
Mean Absolute Error	18.95%
ROC (AUC)	72.0%

Tabela 3: Tabela de resultados do Classificador *Minimum Distance with Fisher LDA*

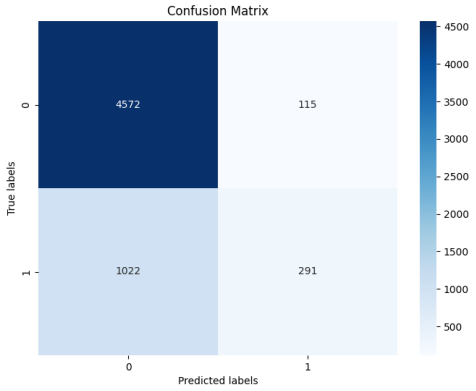

Tabela 4: Matrix Confusão do Classificador *Minimum Distance with Fisher LDA*

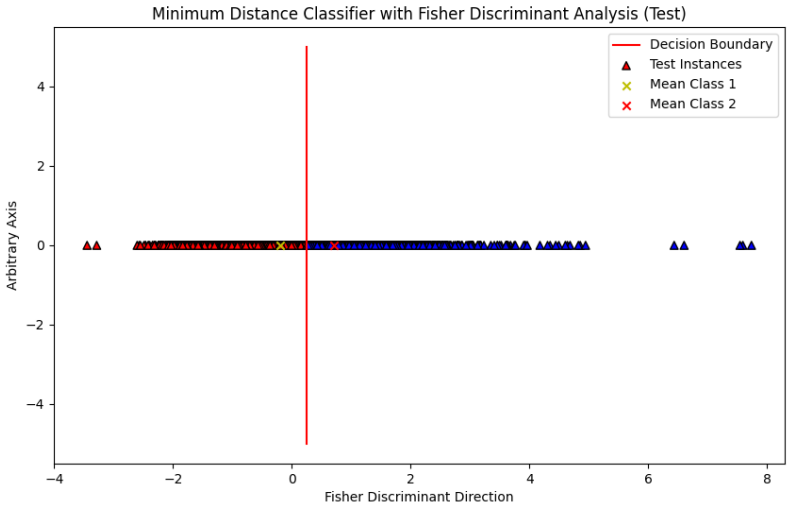

Figura 17: *Minimum Distance with Fisher LDA*

Os resultados indicam que o modelo tem uma precisão global de 81,05%, o que é bastante positivo. No entanto, ele parece ter dificuldades em identificar corretamente os casos positivos, como indicado pelo baixo recall de 22,16% e pelo *F1 Score* de 31,85%. A especificidade alta (97,54%) sugere que o modelo é eficaz em identificar casos negativos. Os erros médios absoluto e quadrático, juntamente com o erro quadrático médio raiz, estão todos em torno de 18,95% a 43,53%, o que indica que o modelo pode não estar tão preciso nas suas previsões. A área sob a curva ROC é de 72,0%, sugerindo um desempenho moderado na capacidade de discriminação do modelo.

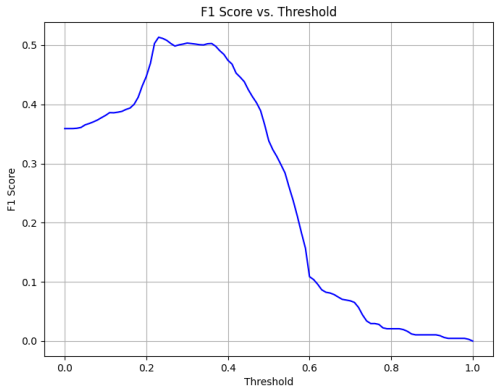

Figura 18: *F1 score* para o classificador *Minimum Distance with Fisher LDA*

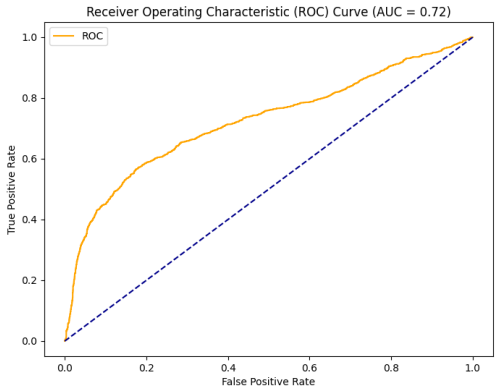

Figura 19: Curvas de ROC para o classificador *Minimum Distance with Fisher LDA*

Matriz Confusão: Após a realização do teste, obtivemos os seguintes resultados na matriz de confusão:

- Verdadeiros Positivos (TP): 291
- Falsos Negativos (FN): 1022
- Falsos Positivos (FP): 115
- Verdadeiros Negativos (TN): 4572

F1 score: Inicialmente, o modelo apresentou um *F1 Score* moderado, em torno de 0.35, indicando um equilíbrio razoável entre precisão e *recall*. No entanto, ao longo do tempo, observou-se uma melhoria gradual no *F1 Score*, aumentando para cerca de 0.5, o que sugere uma progressão positiva no desempenho do modelo. Entretanto, ao chegar ao valor de threshold de 0.4, houve uma queda acentuada no *F1 Score*, destacando uma sensibilidade significativa do modelo à alteração desse parâmetro específico.

7.2.2 Classificador - *Gaussian Naive Bayes*

Para criarmos o classificador *Gaussian Naive Bayes*, utilizamos os dados de treino, incluindo as 9 características mais discriminantes resultantes do teste de *Kruskal*. Apresentamos agora os resultados da análise do classificador:

Accuracy	56.7%
Specificity	52.67%
Precision	29.61%
Recall	71.1%
F1 Score	41.8%
Mean Squared Error	43.3%
Root Mean Squared Error	65.8%
Mean Absolute Error	43.3%
ROC (AUC)	69.0%

Tabela 5: Tabela de resultados do Classificador *Gaussian Naive Bayes*

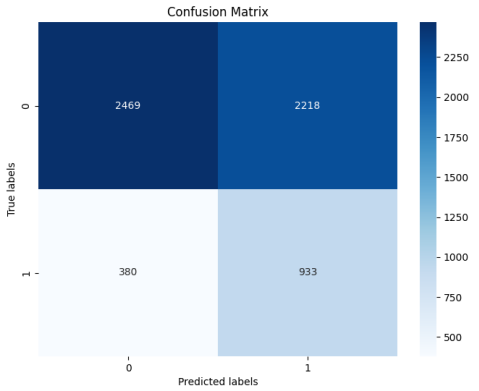

Tabela 6: Matrix Confusão do Classificador *Gaussian Naive Bayes*

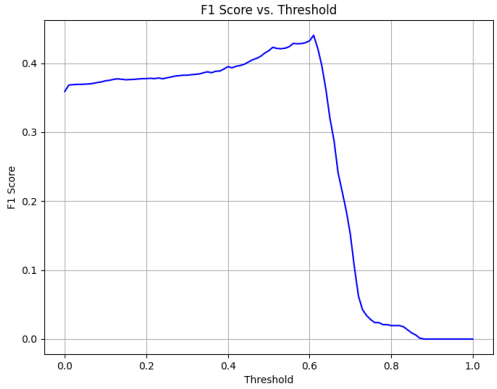

Figura 20: *F1 score* para o classificador *Gaussian Naive Bayes*

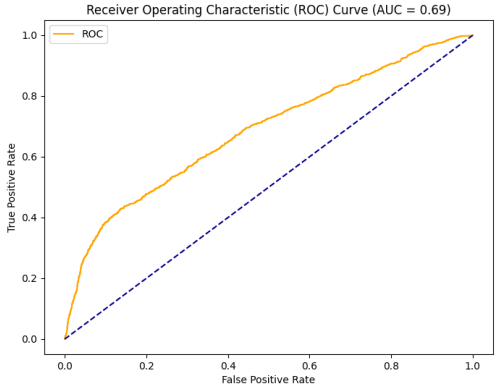

Figura 21: Curvas de ROC para o classificador *Gaussian Naive Bayes*

Os resultados sugerem que o modelo tem uma precisão global de 56,7%, o que é apenas moderado. Ele tem uma especificidade de 52,67%, o que indica uma capacidade moderada de identificar casos negativos. No entanto, a precisão (29,61%) indica que o modelo está a ter dificuldades em evitar falsos positivos. Por outro lado, o recall (71,1%) é relativamente alto, o que sugere que o modelo está a identificar a maioria dos casos positivos. O F1 Score (41,8%) é uma média harmónica entre precisão e recall, e é também moderado.

Os erros médios absoluto e quadrático, juntamente com o erro quadrático médio raiz, estão todos em torno de 43,3% a 65,8%, o que indica que o modelo pode não estar tão preciso em suas previsões. A área sob a curva ROC é de 69,0%, sugere um desempenho moderado na capacidade de discriminação do modelo, embora não seja tão alto quanto o esperado para um bom modelo de classificação.

Matriz Confusão: Após a realização do teste, obtivemos os seguintes resultados na matriz de confusão:

- Verdadeiros Positivos (TP): 933
- Falsos Negativos (FN): 380
- Falsos Positivos (FP): 2218
- Verdadeiros Negativos (TN): 2469

F1 score: Entre um *threshold* de 0,35 e 0,45, o *F1 score* permanece relativamente estável e aumenta ligeiramente. Isso sugere que neste intervalo de *threshold*, o modelo está equilibrar bem a *precision* e o *recall*, resultando num *F1 score* consistente e incremental.

Quando o *threshold* atinge 0,6, observa-se uma queda acentuada no *F1 score*. Isso indica que, ao aumentar o threshold para 0,6, o modelo está a tornar-se mais conservador nas suas previsões, resultando numa diminuição significativa na capacidade de equilibrar a *precision* e o *recall*.

7.2.3 Classificador - *Bernoulli Naive Bayes*

Para criarmos o classificador *Bernoulli Naive Bayes*, utilizamos os dados de treino, incluindo as 9 características mais discriminantes resultantes do teste de *Kruskal*. Apresentamos agora os resultados da análise do classificador:

Accuracy	79.1%
Specificity	89.52%
Precision	52.56%
Recall	41.43%
F1 Score	46.33%
Mean Squared Error	21.0%
Root Mean Squared Error	45.82%
Mean Absolute Error	21.0%
ROC (AUC)	73.0%

Tabela 7: Tabela de resultados do Classificador *Bernoulli Naive Bayes*

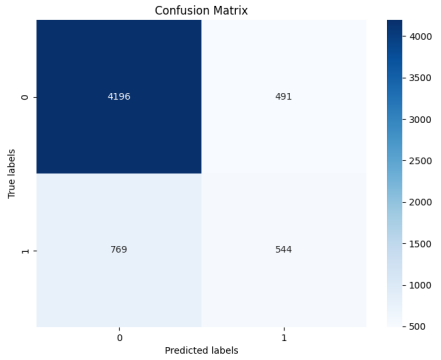

Tabela 8: Matrix Confusão do Classificador *Bernoulli Naive Bayes*

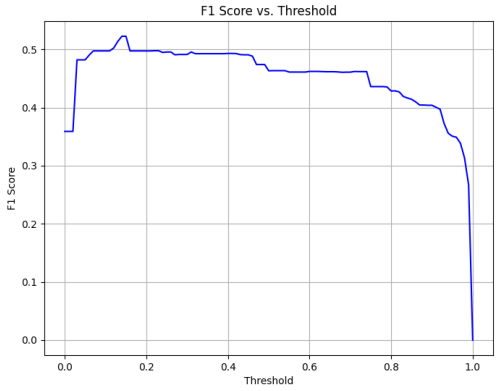

Figura 22: *F1 score* para o classificador *Bernoulli Naive Bayes*

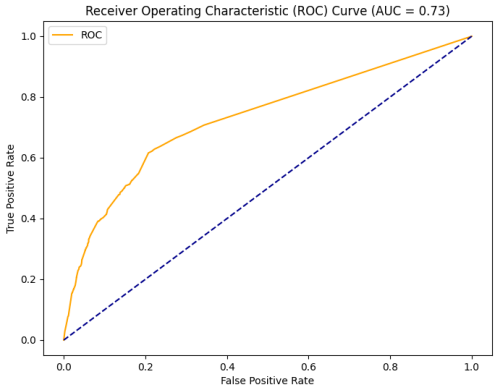

Figura 23: Curvas de ROC para o classificador *Bernoulli Naive Bayes*

Estes resultados indicam que o modelo de classificação *Bernoulli* tem uma precisão global de 79,1%, o que é positivo. A especificidade de 89,52% sugere uma boa capacidade de identificar casos negativos. No entanto, a precisão de 52,56% indica que o modelo pode estar a ter dificuldades em evitar falsos positivos. O *recall* (41,43%) é relativamente baixo, o que sugere que o modelo está a perder alguns casos positivos. O *F1 Score* (46,33%) é moderado, indicando um equilíbrio entre precisão e *recall*.

Os erros médios absoluto e quadrático, juntamente com o erro quadrático médio raiz, estão todos em torno de 21,0% a 45,82%, o que sugere uma precisão razoável nas previsões do modelo. A área sob a curva *ROC* é de 73,0%, indica um desempenho moderado na capacidade de discriminação do modelo. Em suma, este modelo parece ter um desempenho sólido, mas pode haver margem para melhorias, especialmente em termos de *recall* e precisão.

Matriz Confusão: Após a realização do teste, obtivemos os seguintes resultados na matriz de confusão:

- Verdadeiros Positivos (TP): 544
- Falsos Negativos (FN): 769
- Falsos Positivos (FP): 491
- Verdadeiros Negativos (TN): 4196

F1 score: Inicialmente, o F1 score começa num intervalo de valores em torno de 0.4, o que sugere um equilíbrio moderado, pois indica que o modelo está a conseguir obter um bom balanço entre *precision* e *recall*. À medida que o threshold aumenta para 0.5, o *F1 score* também aumenta, indicando uma melhoria na capacidade do modelo de equilibrar *precision* e o *recall*. No entanto, à medida que o *threshold* continua a aumentar até atingir 0.8, o *F1 score* começa a cair drásticamente. Ao aumentarmos o threshold para 0.8, o modelo torna-se muito conservador nas suas previsões, resultando em uma diminuição significativa na capacidade de equilibrar a *precision* e o*recall*.

7.2.4 Classificador - *KNN*

Para criarmos o classificador *KNN*, utilizamos os dados de treino, incluindo as 9 características mais discriminantes resultantes do teste de *Kruskal*. Apresentamos agora os resultados da análise do classificador:

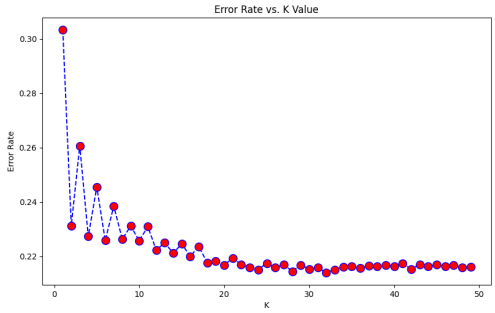

Figura 24: Relação entre o valor de K e o erro

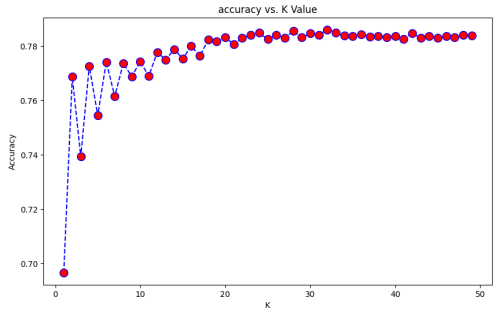

Figura 25: Relação entre o valor de K e exatidão

As duas últimas figuras são representativas do cálculo do valor de K. No exercício de calcular o valor de K para este classificador, obtivemos um valor de K igual a 31.

Accuracy	78.42%
Specificity	97.69%
Precision	54.62%
Recall	9.87%
F1 Score	16.72%
Mean Squared Error	21.58%
Root Mean Squared Error	46.46%
Mean Absolute Error	21.58%
ROC (AUC)	64.0%
Minimum Error (K = 31)	21.41%
Minimum Accuracy (K = 31)	78.42%

Tabela 9: Tabela de resultados do Classificador *KNN*

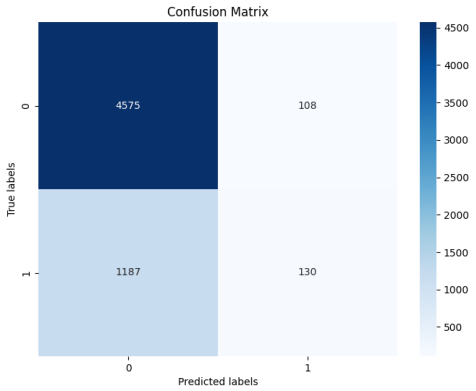

Tabela 10: Matrix Confusão do Classificador *KNN*

Estes resultados revelam que o modelo de classificação *KNN* apresenta uma precisão global de 78,42%, o que é bastante razoável. A especificidade de 97,69% sugere uma excelente capacidade de identificar casos negativos. No entanto, a precisão de 54,62% indica que o modelo pode estar a ter dificuldades em evitar falsos positivos. O recall (9,87%) é muito baixo, sugerindo que o modelo está a perder a maioria dos casos positivos. Consequentemente, o F1 Score (16,72%) é bastante baixo, indicando um desempenho insatisfatório na harmonização entre precisão e *recall*.

Os erros médios absoluto e quadrático, juntamente com o erro quadrático médio raiz, estão todos em torno de 21,58% a 46,46%, o que sugere uma precisão razoável nas previsões do modelo. A área sob a curva *ROC* é de 64,0%, indicando um desempenho moderado na capacidade de discriminação do modelo. O modelo alcança a menor taxa de erro (21,41%) e a maior precisão (78,42%) quando o parâmetro K é igual a 31.

Em suma, embora o modelo apresente uma boa especificidade, a baixa *recall* e o baixo *F1 Score* sugerem que ele pode não ser adequado para identificar casos positivos. Isso pode ser uma área de melhoria para o modelo.

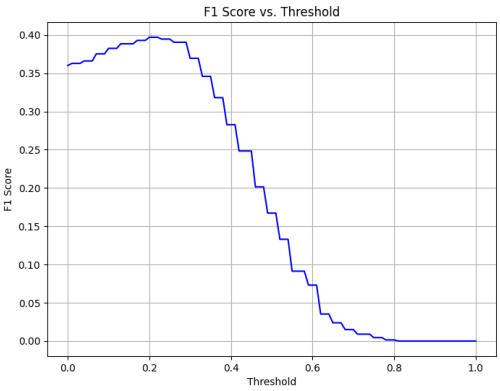

Figura 26: Gráfico do *F1 Score* do Classificador *KNN*

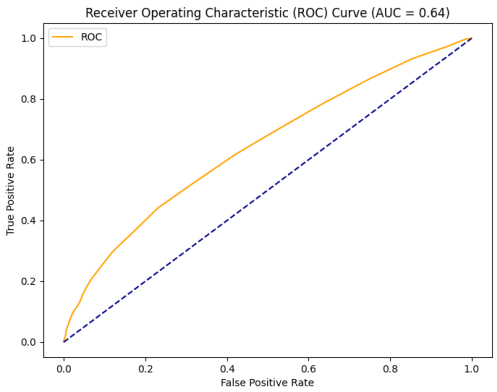

Figura 27: *ROC curves* do Classificador *KNN*

Matriz Confusão: Após a realização do teste, obtivemos os seguintes resultados na matriz de confusão:

- Verdadeiros Positivos (TP): 130
- Falsos Negativos (FN): 1187
- Falsos Positivos (FP): 108
- Verdadeiros Negativos (TN): 4575

F1 score: Inicialmente, o F1 score começa num intervalo de valores em torno de 0.35, o que sugere um equilíbrio moderado, pois indica que o modelo está a conseguir obter um bom balanço entre *precision* e *recall*. À medida que o threshold aumenta para 0.4, o *F1 score* também aumenta, indicando uma melhoria na capacidade do modelo de equilibrar *precision* e o *recall*. No entanto, à medida que o *threshold* continua a aumentar até atingir 0.3, o *F1 score* começa a cair drásticamente. Ao aumentarmos o threshold para 0.8, o modelo torna-se muito conservador nas suas previsões, resultando em uma diminuição significativa na capacidade de equilibrar a *precision* e o *recall*.

7.2.5 Classificador - *Random Forest*

Para criarmos o classificador *Random Forest*, utilizamos os dados de treino, incluindo as 9 características mais discriminantes resultantes do teste de *Kruskal*. Apresentamos agora os resultados da análise do classificador:

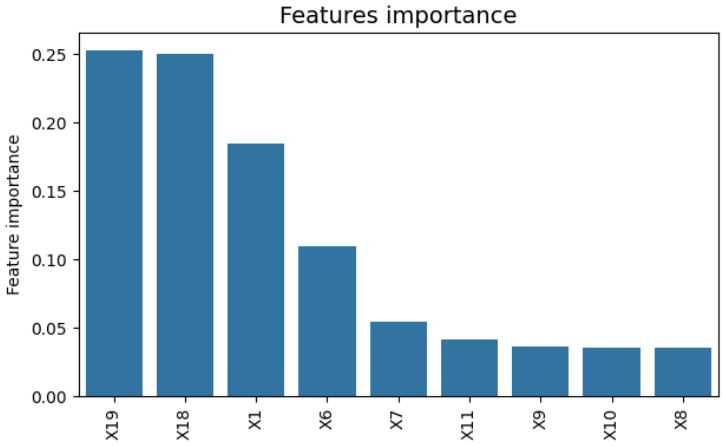

Figura 28: Tabela do cálculo da importância das *Features* atribuída pelo classificador.

A tabela anterior indica a importância que o classificador dá às determinadas *features*. Conseguimos observar que as *features* que o classificador dá mais importância são a: **X19** e **X18**.

Accuracy	80.12%
Specificity	92.38%
Precision	57.19%
Recall	36.32%
F1 Score	44.43%
Mean Squared Error	19.88%
Root Mean Squared Error	44.59%
Mean Absolute Error	19.88%
ROC (AUC)	74.0%

Tabela 11: Tabela de resultados do Classificador *Random Forest*

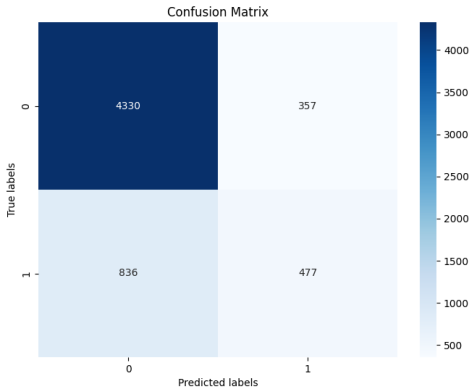

Tabela 12: Matrix Confusão do Classificador *Random Forest*

Estes resultados indicam que o modelo de classificação *Random Forest* apresenta uma precisão global de 80,12%, o que é positivo. A especificidade de 92,38% sugere uma boa capacidade de identificar casos negativos. A precisão de 57,19% indica que o modelo está razoavelmente bom em evitar falsos positivos. No entanto, o *recall* (36,32%) é moderado, sugerindo que o modelo está a identificar apenas uma parte dos casos positivos. O *F1 Score* (44,43%) é uma média harmónica entre precisão e *recall*, e é também moderado.

Os erros médios absoluto e quadrático, juntamente com o erro quadrático médio raiz, estão todos em torno de 19,88% a 44,59%, o que sugere uma precisão razoável nas previsões do modelo. A área sob a curva *ROC* é de 74,0%, indicando um desempenho moderado na capacidade de discriminação do modelo.

Em suma, este modelo parece ter um desempenho sólido, com uma boa precisão global e capacidade de identificar casos negativos. No entanto, pode haver margem para melhorias na identificação de casos positivos, como indicado pelo *recall* moderado.

Matriz Confusão: Após a realização do teste, obtivemos os seguintes resultados na matriz de confusão:

- Verdadeiros Positivos (TP): 477
- Falsos Negativos (FN): 836
- Falsos Positivos (FP): 357
- Verdadeiros Negativos (TN): 4330

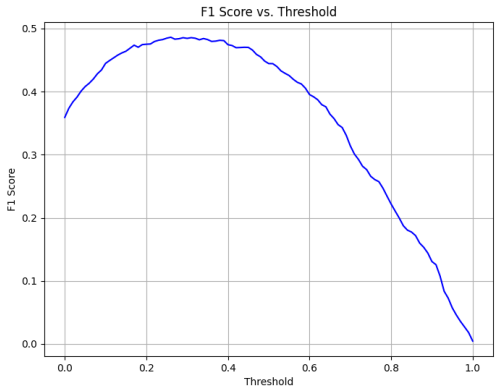

Figura 29: *F1 score* para o classificador *Random Forest*

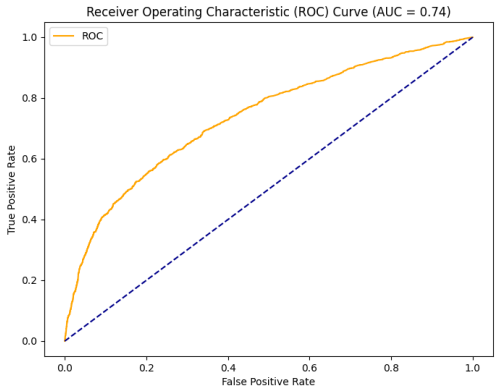

Figura 30: Curvas de *ROC* para o classificador *Random Forest*

F1 score: Inicialmente, o F1 score começa num intervalo de valores em torno de 0.35, o que sugere um equilíbrio moderado, indicando que o modelo está a conseguir obter um bom balanço entre *precision* e *recall*. À medida que o threshold aumenta para um valor muito perto de 0.3, o *F1 score* também aumenta, indicando uma melhoria na capacidade do modelo de equilibrar *precision* e o *recall*. No entanto, quando o treshold aumenta até ao valor 0.4, o valor do *F1 score* descresce até chegar a 0, fazendo com o modelo torne-se muito conservador nas suas previsões, resultando em uma diminuição significativa na capacidade de equilibrar a *precision* e o *recall*.

7.2.6 Classificador - *SVM Default*

Para este classificador, usamos o dataset inteiro, sem qualquer pré-processamento e usou-se um valor de *C = 1, Kernel='rbf', gamma='scale'*. Os próximos gráficos e a tabela, são respetivos à aptidão obtida do classificador.

Accuracy	82.32%
Specificity	95.93%
Precision	70.67%
Recall	34.46%
F1 Score	46.32%
Mean Squared Error	17.68%
Root Mean Squared Error	42.05%
Mean Absolute Error	17.68%
ROC (AUC)	65.20%

Tabela 13: Tabela de resultados do Classificador *SVM Default*

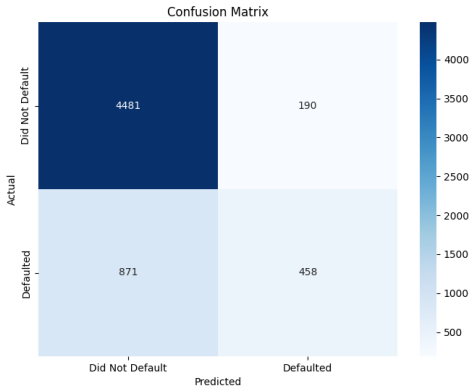

Tabela 14: Matrix Confusão do Classificador *SVM Default*

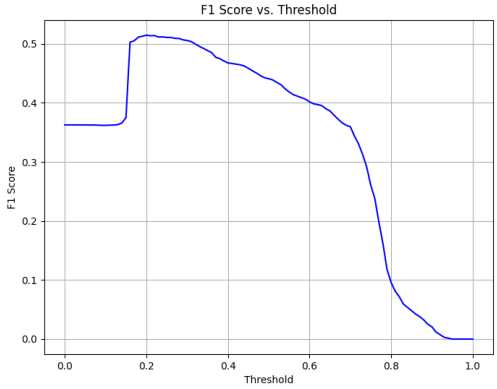

Figura 31: *F1 score* para o classificador *SVM Default*

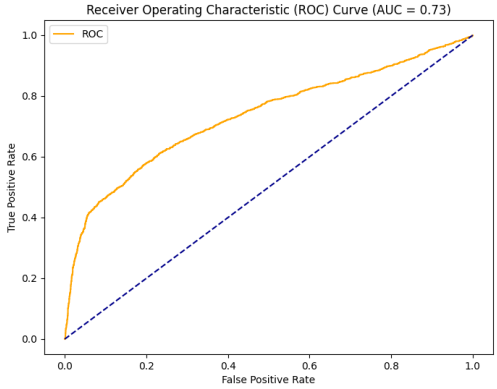

Figura 32: Curvas de *ROC* para o classificador *SVM Default*

Estes resultados indicam que o classificador *SVM Default* apresenta uma precisão global de 82,32%, o que é bastante positivo. A especificidade de 95,93% sugere uma excelente capacidade de identificar casos negativos. A precisão de 70,67% indica que o modelo está relativamente bom em evitar falsos positivos. No entanto, o recall (34,46%) é pouco moderado, o que sugere que o modelo está a identificar apenas uma parte dos casos positivos. O *F1 Score* (46,32%) é também moderado.

Os erros médios absoluto e quadrático, juntamente com o erro quadrático médio da raiz, estão todos em torno dos valores 17,68% a 42,05%, o que sugerem uma precisão razoável nas previsões do modelo. A área sob a curva *ROC* é de 65,20%, indicando um desempenho moderado na capacidade de discriminação do modelo.

Em resumo, este modelo parece ter um desempenho sólido, com uma boa precisão global e capacidade de identificar casos negativos. No entanto, assim como alguns dos modelos anteriores, há espaço para melhorias na identificação de casos positivos, como indicado pelo *recall* moderado.

Matriz Confusão: Após a realização do teste, obtivemos os seguintes resultados na matriz de confusão:

- Verdadeiros Positivos (TP): 458
- Falsos Negativos (FN): 871
- Falsos Positivos (FP): 190
- Verdadeiros Negativos (TN): 4481

F1 score: Inicialmente, o F1 score começa num intervalo de valores em torno de 0.35, o que sugere um equilíbrio moderado, indicando que o modelo está a conseguir obter um bom balanço entre *precision* e *recall*. À medida que o threshold aumenta, passando um pouco do valor de 0.5, o *F1 score* também mantém o seu valor estável sofrendo de uma subida repentina, indicando uma melhoria na capacidade do modelo de equilibrar *precision* e o *recall*. No entanto, há medida que o treshold vai aumentando, o valor do *F1 score* vai decrescendo gradualmente até ao valor do threshold de 0.7, onde após este marco sofre uma descida abruta, fazendo com o modelo torne-se muito conservador nas suas previsões, resultando em uma diminuição significativa na capacidade de equilibrar a *precision* e o *recall*.

7.2.7 Classificador - *SVM com GridSearch*

Para este classificador, usamos o dataset inteiro, sem qualquer pré-processamento e usou-se um valor de *C = 1000, Kernel='rbf', gamma=0.001*. Estes valores foram já estudados e calculados, fazendo uso da técnica *Grid Search Cross-Validation* para fazer *tuning* dos seguintes hiperparâmetros tendo em conta a exatidão como métrica de avaliação:

- C :[0.5,0.1,1,10,100,1000];
- gamma :['scale', 1,0.1, 0.01,0.001,0.0001];
- kernel :['rbf', 'sigmoid']

Os próximos gráficos e a tabela, são respetivos à aptidão obtida do classificador.

Accuracy	81.92%
Specificity	95,66%
Precision	67.98%
Recall	32.83%
F1 Score	44.27%
Mean Squared Error	18.08%
Root Mean Squared Error	42.52%
Mean Absolute Error	18.08%
ROC (AUC)	73%

Tabela 15: Tabela de resultados do Classificador *SVM with GridSearch*

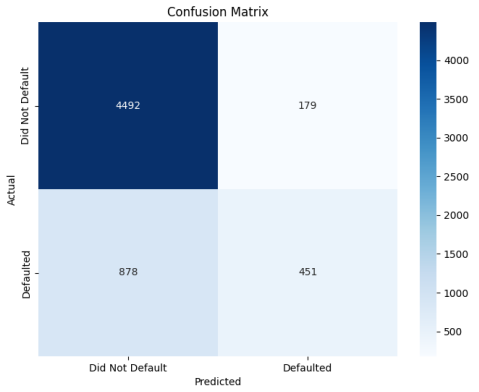

Tabela 16: Matrix Confusão do Classificador *SVM with GridSearch*

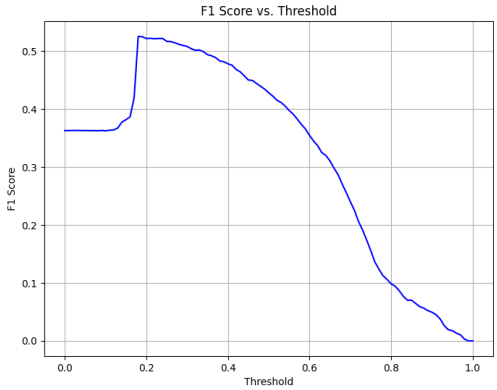

Figura 33: *F1 score* para o classificador *SVM with GridSearch*

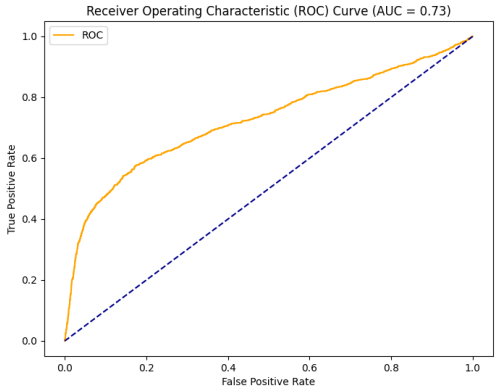

Figura 34: Curvas de *ROC* para o classificador *SVM with GridSearch*

Os resultados indicam que o classificador *SVM with GridSearch* apresenta uma precisão global de 82,38%, o que é bastante positivo. A especificidade de 96,16% sugere uma excelente capacidade de identificar casos negativos. A precisão de 71,58% indica que o modelo está relativamente bom em evitar falsos positivos. No entanto, o recall (33,93%) é moderado, sugerindo que o modelo está a identificar apenas uma parte dos casos positivos. O F1 Score (46,04%) é também moderado.

Os erros médios absoluto e quadrático, juntamente com o erro quadrático médio raiz, estão todos em torno de 17,61% a 41,97%, o que sugere uma precisão razoável nas previsões do modelo. A área

sob a curva ROC é de 65,05%, indicando um desempenho moderado na capacidade de discriminação do modelo.

Em suma, este modelo também parece ter um desempenho sólido, com uma boa precisão global e capacidade de identificar casos negativos. No entanto, assim como alguns dos modelos anteriores, há espaço para melhorias na identificação de casos positivos, como indicado pelo recall moderado. **Matriz Confusão:** Após a realização do teste, obtivemos os seguintes resultados na matriz de confusão:

- Verdadeiros Positivos (TP): 451
- Falsos Negativos (FN): 876
- Falsos Positivos (FP): 179
- Verdadeiros Negativos (TN): 4492

F1 score: Observa-se que o *F1 score* é semelahnte ao do classificador anterior. Inicialmente, o F1 score começa num intervalo de valores em torno de 0.35, o que sugere um equilíbrio moderado, indicando que o modelo está a conseguir obter um bom balanço entre *precision* e *recall*. À medida que o threshold aumenta, passando um pouco do valor de 0.5, o *F1 score* também mantém o seu valor estável sofrendo de uma subida repentina, indicando uma melhoria na capacidade do modelo de equilibrar *precision* e o *recall*. No entanto, há medida que o treshold vai aumentando, o valor do *F1 score* vai decrescendo gradualmente fazendo com o modelo torne-se muito conservador nas suas previsões, resultando em uma diminuição significativa na capacidade de equilibrar a *precision* e o *recall*.

7.2.8 Classificador - *SVM com GridSearch e Kruskal*

Para este classificador, já foi usado as melhores 9 features com base no Kruskall e usou-se um valor de *C = 1000, Kernel='rbf', gamma=0.001*. Estes valores foram estudados e calculados da mesma maneira que o classificador anterior, fazendo uso da técnica *Grid Search Cross-Validation* para fazer *tuning* dos seguintes hiperparâmetros:

- C :[0.5,0.1,1,10,100,1000];
- gamma :['scale', 1,0.1, 0.01,0.001,0.0001];
- kernel :['rbf', 'sigmoid']

Accuracy	82.36%
Specificity	95.61%
Precision	69.89%
Recall	35.81%
F1 Score	47.36%
Mean Squared Error	17.63%
Root Mean Squared Error	41.99%
Mean Absolute Error	17.63%
ROC (AUC)	73%

Tabela 17: Tabela de resultados do Classificador *SVM with GridSearch and Kruskal*

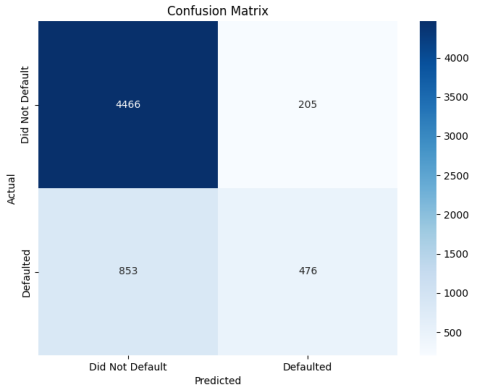

Tabela 18: Matrix Confusão do Classificador *SVM with GridSearch and Kruskal*

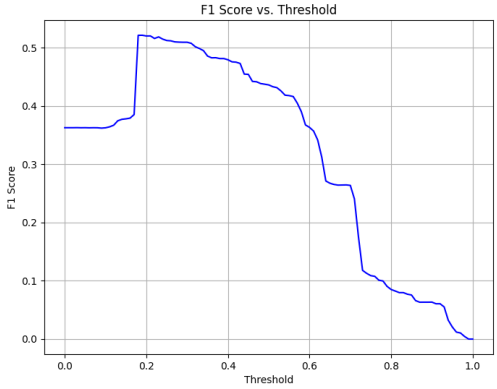

Figura 35: *F1 score* para o classificador *SVM with GridSearch and Kruskal*

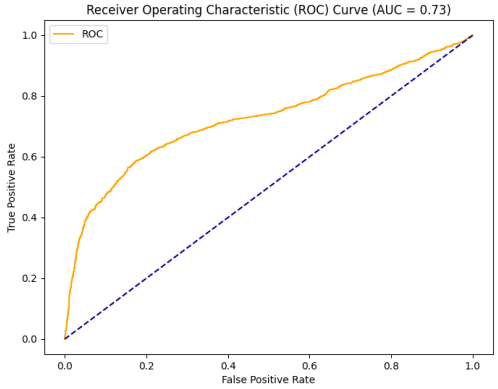

Figura 36: Curvas de *ROC* para o classificador *SVM with GridSearch and Kruskal*

Os resultados obtidos com o classificador *SVM with GridSearch and Kruskal* revelam uma performance positiva. Com uma precisão global de 82.36%, o modelo demonstra uma habilidade considerável para fazer previsões corretas. A especificidade de 95.61% indica uma notável capacidade de distinguir eficazmente os casos negativos e a precisão de 69.89% sugere uma relativa habilidade em evitar falsos positivos.

Por outro lado, o *recall* de 35.81% mostra que o modelo está a deixar escapar uma proporção significativa dos casos positivos. O *F1 Score* atinge 47.36%, o que é um valor moderado, mas aponta para uma certa necessidade de melhoria na capacidade de identificar corretamente os casos positivos.

No que diz respeito à precisão das previsões, os erros médios absoluto e quadrático, assim como o erro quadrático médio da raiz, estão em torno de 17.63%, o que indica uma precisão aceitável. A área sob a curva *ROC* é de 73%, o que sugere um desempenho razoável na capacidade de discriminação do modelo.

Em suma, apesar de apresentar uma boa precisão global e uma capacidade notável de identificar casos negativos, o modelo ainda carece de aprimoramento na identificação dos casos positivos, conforme o observado no *recall*.

Matriz Confusão: Após a realização do teste, obtivemos os seguintes resultados na matriz de confusão:

- Verdadeiros Positivos (TP): 476
- Falsos Negativos (FN): 853
- Falsos Positivos (FP): 205
- Verdadeiros Negativos (TN): 4466

F1 score: Observa-se que o *F1 score* é semelahnte ao do classificador anterior. Inicialmente, o F1 score começa num intervalo de valores em torno de 0.35, o que sugere um equilíbrio moderado, indicando que o modelo está a conseguir obter um bom balanço entre *precision* e *recall*. À medida que o threshold aumenta, passando um pouco do valor de 0.5, o *F1 score* também mantém o seu valor estável sofrendo de uma subida repentina, indicando uma melhoria na capacidade do modelo de equilibrar *precision* e o *recall*. No entanto, há medida que o treshold vai aumentando, o valor do *F1 score* vai decrescendo gradualmente fazendo com o modelo torne-se muito conservador nas suas previsões, resultando em uma diminuição significativa na capacidade de equilibrar a *precision* e o *recall*.

7.2.9 Classificador - *AdaBoost*

Para este classificador, foram usadas as melhores 9 features com base no Kruskall e usou-se um valor de *random state=42, algorithm='SAMME.R', learning rate=0.8, número de estimadores=100*. Para escolher estes valores, os mesmos foram estudados e calculados para descobrir o melhor conjunto de hiperparâmetros tendo com base a exatidão e o gasto computacional.

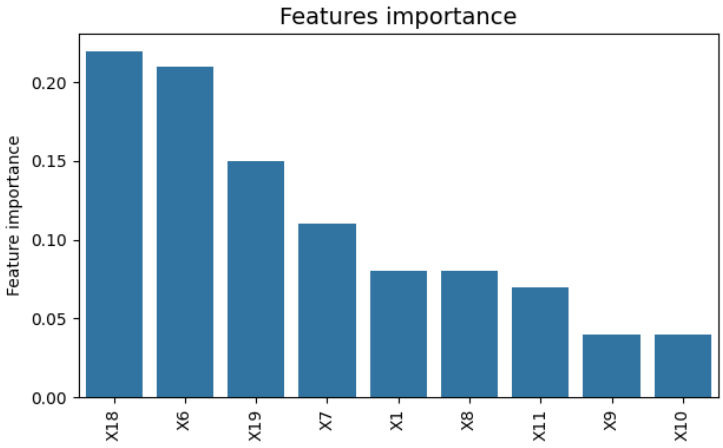

Figura 37: Tabela do cálculo da importância das *Features* atribuída pelo classificador.

A tabela anterior indica a importância que o classificador dá às determinadas *features*. Conseguimos observar que as *features* que o classificador dá mais importância são a: **X18** e **X6**. Os resultados

Accuracy	82.15%
Specificity	95.80%
Precision	69.84%
Recall	34.16%
F1 Score	45.88%
Mean Squared Error	17.84%
Root Mean Squared Error	42.24%
Mean Absolute Error	17.84%
ROC (AUC)	77%

Tabela 19: Tabela de resultados do Classificador *AdaBoost*

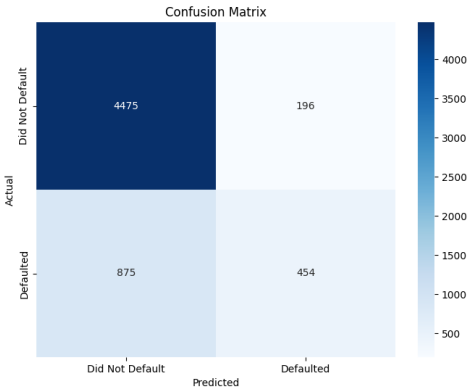

Tabela 20: Matrix Confusão do Classificador *AdaBoost*

obtidos com o classificador *AdaBoost* indicam uma performance sólida, mas com margem para melhorias. Com uma precisão global de 82.15%, o modelo demonstra uma capacidade considerável em fazer previsões corretas. A especificidade de 95.80% reflete uma excelente habilidade do modelo em identificar corretamente os casos negativos, enquanto a precisão de 69.84% sugere uma relativa habilidade em evitar falsos positivos.

Entretanto, o *recall* de 34.16% mostra que o modelo está a deixar escapar uma proporção significativa dos casos positivos. O *F1 Score* atinge os 45.88%, o que é um valor moderado, mas aponta para a necessidade de melhoria na capacidade de identificar corretamente os casos positivos.

No que diz respeito à precisão das previsões, os erros médios absoluto e quadrático, assim como o erro quadrático médio da raiz, estão em torno de 17.84%, o que indica uma precisão aceitável. A área

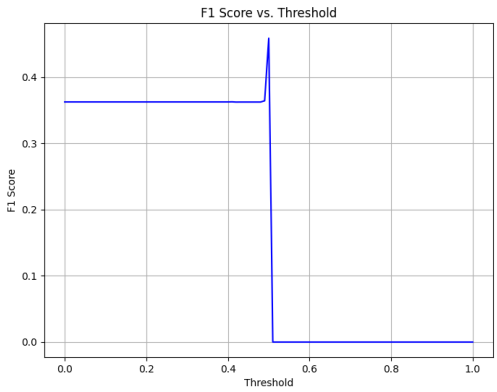

Figura 38: *F1 score* para o classificador *AdaBoost*

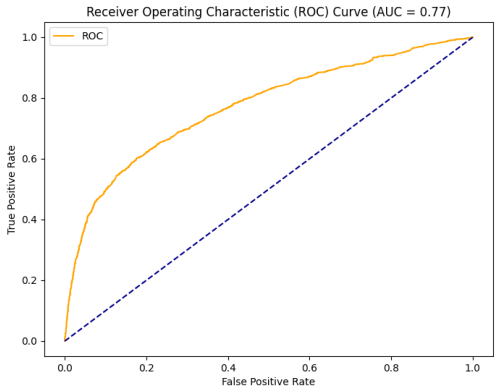

Figura 39: Curvas de *ROC* para o classificador *AdaBoost*

sob a curva ROC é de 77%, o que sugere um desempenho moderado na capacidade de discriminação do modelo.

Em suma, embora o modelo *AdaBoost* apresente uma boa precisão global e uma capacidade notável de identificar casos negativos, há espaço para melhorias na identificação dos casos positivos, conforme o observador pelo o valor do *recall*.

Matriz Confusão: Após a realização do teste, obtivemos os seguintes resultados na matriz de confusão:

- Verdadeiros Positivos (TP): 454
- Falsos Negativos (FN): 875
- Falsos Positivos (FP): 196
- Verdadeiros Negativos (TN): 4475

F1 score: Inicialmente, o F1 score começa num intervalo de valores em torno de 0.37, o que sugere um equilíbrio moderado, indicando que o modelo está a conseguir obter um bom balanço entre *precision* e *recall*. À medida que o threshold aumenta, chegando ao valor de 0.5, o *F1 score* sofre de uma subida repentina e logo após esta subida, desce até 0, indicando que em logo após de 0.5, o treshold é um ponto de equilíbrio muito delicado entre precisão e recall, começando a classificar incorretamente muitos verdadeiros positivos como negativos.

8 Discussão

Tendo em conta o nosso objetivo incial de auxiliar o banco na identificação de casos em que os indivíduos não serão capazes de pagar o empréstimo no próximo mês (classe 0), é crucial atribuir maior importância aos classificadores que demonstrem uma especificidade mais elevada pois o banco quererá precaver-se de casos onde a pessoa nao consiga pagar mesmo o empréstimo mas é assinalada como consegue, embora não descartando as outras métricas associadas a cada classificador.

No que diz respeito à **exatidão**, o classificador *SVM Default* destaca-se como o melhor, com uma taxa de 82,32%, enquanto o *Gaussian Naive Bayes* apresenta um desempenho significativamente inferior, com apenas 56,7%. Esta discrepância pode ser atribuída à natureza simplista do modelo naive Bayes, que assume independência entre as variáveis, o que pode não ser o caso nos dados do nosso conjunto.

Analisando a **especificidade**, o *KNN* com K=31 revela-se como o melhor classificador, atingindo uma taxa impressionante de 97,69%. Por outro lado, o *Gaussian Naive Bayes* novamente apresenta um desempenho não muito ótimo, com uma especificidade de apenas 52,67%.

Quanto à **precisão**, o *Fisher LDA* destaca-se como o melhor classificador, com uma taxa de 71,67%, enquanto o *Gaussian Naive Bayes* continua a ter um desempenho insatisfatório, com apenas 29,61%. Esta disparidade pode ser atribuída à tendência do modelo *Naive Bayes* para subestimar a probabilidade de certas classes, resultando em uma baixa precisão.

Por fim, em relação à **sensibilidade**, o *Gaussian Naive Bayes* apresenta-se como o melhor classificador, com uma taxa de 71,1%, enquanto o *KNN*com K=31 mostra uma sensibilidade de apenas 9,87%. Uma possível explicação para este resultado é a capacidade do *Naive Bayes* de lidar melhor com conjuntos de dados não equilibrados, como é o caso aqui, onde a classe 0 é predominante.

Além dos resultados dos classificadores mencionados anteriormente, é essencial destacar os desempenhos dos outros classificadores desenvolvidos. Os diferentes modelos SVM, incluindo *SVM* com *Gridsearch* e *Gridsearch* com *Kruskal*, demonstraram resultados bastante semelhantes ao da *SVM Default* em todas as métricas avaliadas. A razão poderá estar inserida na ótica de que os ajustes nos parâmetros do SVM não tiveram um impacto significativo no desempenho do modelo.

Por outro lado, o classificador *Adaboost* destacou-se com uma especificidade de 95,8%, o que o torna particularmente eficaz na identificação de casos em que os indivíduos não serão capazes de pagar o empréstimo no próximo mês, minimizando assim os falsos positivos. No entanto, é importante notar que o *Adaboost* demonstrou uma sensibilidade relativamente baixa de 34%, o que significa que pode não ser tão eficaz na captura de todos os casos positivos, resultando em uma taxa mais alta de falsos negativos.

É de grande relevância ainda ter em atenção, que durante todo o projeto, faltou ao grupo fazer um balaceamento do dataset, pois existe mais casos onde o cliente consegue pagar o empréstimo no próximo mês.

Para este balanceamento podia-se fazer um *Under-sampling*, diminuindo de forma aleatória o tamanho da classe com maior quantidade, testar com vários datasets balanceados para descobrir um melhor que durante as experiências demonstrassem melhores resultados. O mesmo podia ser feito para um *Over-sampling*, que é usado quando a quantidade de dados de uma das classes é insuficiente. Aqui o método tenta equilibrar o conjunto de dados aumentando o tamanho das amostras. Em vez de se livrar das amostras que estão em demasia, novas amostras são geradas usando por exemplo repetição, bootstrapping ou *SMOTE* (*Synthetic Minority Over-Sampling Technique*).

9 Conclusão

Em suma, a nossa investigação permite afirmar que o classificador mais eficaz para identificar os casos em que os indivíduos não conseguem pagar no próximo mês é o KNN com um valor de k igual a 31. A elevada especificidade de 97% e sensibilidade tornam-no uma opção sólida para esta tarefa específica, garantindo uma deteção fiável dos casos negativos.

Contudo, se o intuito for identificar os casos em que os indivíduos conseguem pagar no próximo mês, o classificador *Gaussian Naive Bayes* pode ser preferível do que o inverso. Apesar dos resultados inferiores em comparação com o KNN para a classe 0, o Naive Bayes demonstrou uma capacidade superior para lidar com conjuntos de dados desequilibrados, o que pode ser crucial ao tentar identificar os casos positivos.

Assim, a escolha do classificador mais apropriado dependerá das necessidades específicas do projeto e das prioridades em termos de identificação de casos positivos ou negativos. Ambos os classificadores têm as suas vantagens e limitações.

Bibliografia

[Yeh16] I-Cheng Yeh. *Default of Credit Card Clients*. 2016. URL: https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients.

[bhu22] bhuwanesh. *How to Perform a Kruskal-Wallis Test in Python*. 2022. URL: https://www.geeksforgeeks.org/how-to-perform-a-kruskal-wallis-test-in-python/.

[Man22a] SciPy v1.12.0 Manual. *scipy.stats.kruskal*. 2022. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kruskal.html.

[Man22b] SciPy v1.12.0 Manual. *scipy.stats.kstest*. 2022. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kstest.html.

[Ast24] AsthaMehta. *ML — Kolmogorov-Smirnov Test*. 2024. URL: https://www.geeksforgeeks.org/ml-kolmogorov-smirnov-test/.

[IBM] IBM. *What is random forest?* URL: https://www.ibm.com/topics/random-forest. (accessed: 09.03.2024).