INSTITUTO UNIVERSITÁRIO DE LISBOA

Mestrado em Ciência de Dados – 2023/2024

Processamento e Modelação de Big Data

Deteção de Reviews Spam em Produtos de Moda e Acessórios na Amazon

Docente: João Pedro Oliveira

Alunos: Filipa Rodrigues nº 99865, Mariana Borralho nº 120417 e Nuno Ferreira nº 120557

Índice

1. Introdução 3
2. Formulação do Problema 3
3. Base de Dados 4
3.1 Valores Omissos 4
3.2 Transformação e criação de variáveis 4
3.3 Balanceamento dos dados 5
3.4 Relações entre variáveis 6
3.5 Variáveis irrelevantes 7
4. Modelação 7
5. Resultados 8
6. Conclusão 10
7. Bibliografia 11
8. Anexos 12

1. Introdução

No âmbito da unidade curricular de Processamento e Modelação de Big Data, integrada no mestrado em Ciência de Dados, foi proposto a realização do presente projeto. Para a elaboração do mesmo foi selecionada uma base de dados da Amazon, que contém reviews de produtos na categoria "Clothing, Shoes, and Jewelry".
O objetivo deste projeto é desenvolver e implementar uma solução computacional para responder a um problema de análise de dados em larga escala, especificamente focado na distinção entre reviews autênticas e tentativas de manipulação (spam). A análise preliminar dos dados foi essencial para identificar as variáveis mais relevantes, com base nas quais foram desenvolvidos e testados vários modelos em amostras reduzidas.
O projeto incluiu a modelação de diferentes algoritmos de aprendizagem supervisionada, tais como Regressão Logística, Support Vector Machine, Árvores de Decisão, Florestas Aleatórias e Gradient-Boosted Trees. Cada modelo foi treinado e avaliado utilizando métricas de desempenho como a accuracy, precision e recall, a fim de determinar qual a abordagem que oferece a melhor capacidade de distinguir entre reviews autênticas e de spam.

2. Formulação do Problema

O problema central deste projeto é desenvolver e implementar um modelo computacional eficaz na identificação de reviews de produtos da Amazon classificadas como spam. A distinção entre reviews genuínas e spam é desafiadora devido à subjetividade e variabilidade textual das avaliações, bem como às diversas motivações por trás das reviews manipuladas. O objetivo é criar um modelo que possa analisar automaticamente as características das reviews, identificando padrões que diferenciem avaliações legítimas de tentativas de manipulação.

Para abordar este problema, foram exploradas duas abordagens principais:

- **Processamento de Texto:** Utilizando uma pipeline personalizada que incluiu etapas de tokenização, remoção de stop words, transformação com HashingTF e cálculo do IDF (Inverse Document Frequency). Esta abordagem visa representar as reviews textuais de forma numérica adequada para a modelação.
- **Feature Engineering:** Selecionando e transformando colunas relevantes para a modelação, como ‘helpfulTotalRatio’, ‘reviewUpVotes’, ‘reviewLength’, ‘isWeekend’, entre outras, facilitando a criação de recursos que melhoram a precisão dos modelos preditivos.

Estas abordagens são cruciais para garantir que os consumidores recebam informações confiáveis sobre os produtos que estão a considerar comprar, bem como manter a integridade das avaliações dos produtos da plataforma. Isto não apenas protege os consumidores como também preserva a reputação da Amazon como uma plataforma de compras confiável.

3. Base de Dados

A base de dados escolhida para este projeto, extraída do Kaggle e intitulada "Clothing_Shoes_and_Jewelry", compreende aproximadamente 5.504.331 reviews de produtos. Cada registo na base de dados é composto por múltiplos campos que são fundamentais para a análise e modelação dos dados. Estes incluem um identificador único para cada review ('_id'), o número de identificação standard da Amazon para produtos ('asin'), e a categoria do produto que indica o segmento de mercado a que pertence, como roupas, sapatos ou joias. Adicionalmente, cada review classificada como spam ou não spam ('class'), contém informações sobre quantas pessoas consideraram a review útil ('helpful'), a avaliação geral atribuída ao produto pelo utilizador ('overall'), o texto completo da review ('reviewText'), a data da publicação ('reviewTime' e 'unixReviewTime'), e o identificador e nome ou pseudónimo do utilizador ('reviewerID', 'reviewerName'). A análise destes campos permite uma avaliação detalhada das características das reviews, fundamental para o desenvolvimento de modelos preditivos eficazes.
Dada a magnitude dos dados, optou-se por realizar análises preliminares numa amostra reduzida de 5%.

3.1 Valores Omissos

Foram eliminadas 13.180 observações devido à ausência de informação no campo 'reviewerName'. Esta exclusão foi motivada pela necessidade de manter a consistência dos dados, evitando assim análises comprometidas pela falta de identificação clara do utilizador. Desta forma, o conjunto de dados final para análise passou a ter 5.491.151 observações.

3.2 Transformação e criação de variáveis

Transformações e criações de variáveis fundamentais enriquecem a análise e a performance do modelo preditivo. Desta forma, diversas alterações foram aplicadas ao dataframe original, priorizando detalhes subtis nas reviews que podem revelar padrões importantes.

Comprimento das reviews: Inicialmente, foram criadas variáveis para quantificar o comprimento do texto das reviews e dos resumos. Estas variáveis têm o potencial de influenciar a classificação das reviews.

Média das avaliações e popularidade dos produtos: Procedeu-se, então, à avaliação da média das avaliações dos produtos e da popularidade dos mesmos, agregadas pelo identificador do produto (asin). Estas medidas fornecem uma perspetiva sobre como o produto é recebido pelo mercado e a sua relevância no conjunto de dados.

Variável helpful: A coluna helpful foi transformada em helpfulTotalRatio, representando a percentagem de utilidade, e introduziu-se reviewUpvotes para quantificar os votos positivos de cada review. Adicionalmente, a variável helpfulTotalRatio foi categorizada em três grupos distintos usando o método Bucketizer. Esta transformação permite uma análise mais detalhada sobre como as reviews são percebidas em termos de utilidade pelos outros utilizadores.

Variável containsQuestion: Foi implementada a funcionalidade de detetar perguntas no texto das reviews, adicionando uma coluna containsQuestion. Esta variável ajuda a identificar se o texto da review solicita mais informações ou promove interação, indicando uma possível procura por diálogo entre utilizadores.

Variável hasLink: Também se introduziu a capacidade de identificar links nos textos das reviews, com a criação da variável binária hasLink. Esta característica é crucial para filtrar conteúdos que possam ser classificados como spam.

Variáveis de tempo: A coluna reviewTime foi convertida para formato de data, facilitando análises baseadas em períodos específicos. Derivando desta variável, foi criada a variável isWeekend que indica se a review ocorreu durante o fim de semana.

Nº de reviews por user: Por fim, agregou-se o número de reviews feitas por cada utilizador, associando esta contagem a cada linha de review. Esta abordagem permite identificar utilizadores mais ativos e avaliar a sua influência potencial na credibilidade das avaliações.

Estas transformações no conjunto de dados são essenciais para a construção de um modelo preditivo robusto, facilitando uma abordagem analítica mais rica e detalhada, que contribui significativamente para a precisão e interpretabilidade dos resultados do modelo.

3.3 Balanceamento dos dados

Para ajustar o desequilíbrio entre as classes da variável target no dataset, aplicou-se a técnica de undersampling. A distribuição inicial apresentava 1.166.879 observações para a classe 0 (não spam) e 4.324.271 observações para a classe 1 (spam), revelando um desequilíbrio acentuado (Anexo 1). Calculou-se uma taxa de balanceamento, correspondendo à razão entre o número de observações da classe minoritária e a classe maioritária. Seguidamente, realizou-se o undersampling na classe maioritária, ajustando a sua presença no conjunto de dados para igualar a da classe minoritária. O conjunto final ficou com 1.166.879 observações na classe 0 e 1.168.698 na classe 1, assegurando um balanceamento adequado para evitar o enviesamento nos modelos.

3.4 Relações entre variáveis

Para otimizar os modelos preditivos e compreender melhor as inter-relações entre as variáveis, foi construída uma matriz de correlação que inclui variáveis quantitativas e binárias, como class, productRating, entre outras. Este processo permite identificar variáveis com comportamentos semelhantes no conjunto de dados. Em casos de elevada correlação, opta-se por incluir apenas uma das variáveis no modelo. Esta abordagem ajuda a reduzir a complexidade do modelo, a evitar multicolinearidade e a manter a integridade das informações.

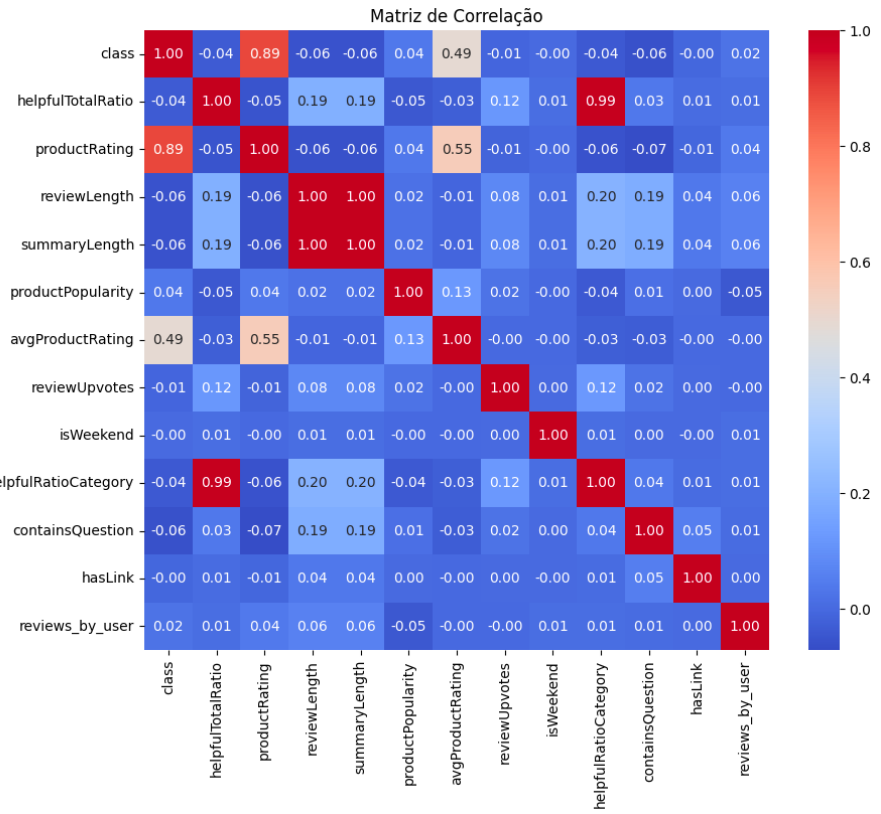


Fig. 1 - Matriz de correlações das variáveis quantitativas e binárias no dataset

Com base na análise da matriz de correlação apresentada, algumas observações e ajustes podem ser realizados para melhorar a compreensão das relações entre as variáveis e a adequação do modelo preditivo:

- Observa-se uma forte correlação entre class e productRating (0.89), o que indica que a classificação do produto pode influenciar diretamente a perceção das avaliações como spam ou não spam.

- Existe uma correlação elevada entre productRating e avgProductRating (0.55), o que é esperado dado que ambas as variáveis representam avaliações de produtos. Por uma questão de ambas as variáveis serem redundantes, optou-se por considerar apenas a

variável avgProductRating em detrimento de productRating. Apesar de a variável productRating aparentemente apresentar uma relação mais forte com a variável target, resultava em overfit dos modelos implementados.

- A variável helpfulTotalRatio mostra uma forte correlação com helpfulRatioCategory (0.99). De forma a evitar o problema de multicolienaridade eliminou-se a variável criada.

- As variáveis reviewLength e summaryLength apresentam uma correlação perfeita (1.00), indicando redundância, pois ambas medem o comprimento dos textos das reviews e resumos. Uma vez mais, de forma a evitar o problema anteriormente mencionado, removeu-se uma das variáveis, summaryLength.

- Muitas variáveis apresentam correlações muito baixas ou quase nulas com outras variáveis, como isWeekend, que mostra pouca ou nenhuma correlação com outros características das reviews, indicando uma influência limitada do fim de semana como variável explicativa.

O coeficiente de Pearson, ideal para relações lineares, pode ser menos eficaz com variáveis binárias, como class e isWeekend. Assim, é preciso ter alguma prudência e o uso de métodos alternativos, como regressão logística e árvores de decisão para captar interações complexas ou não lineares, garantindo assim uma modelagem mais robusta e interpretações mais precisas.

3.5 Variáveis irrelevantes

Neste processo, optou-se por remover várias colunas do dataframe consideradas irrelevantes para o problema em análise. Foram excluídas as variáveis category, que não oferece informações distintivas cruciais (apenas existe uma categoria); asin, o identificador padrão de produtos da Amazon que é irrelevante para a determinação da autenticidade das reviews; reviewerID e reviewerName, porque a identidade dos utilizadores não afeta a análise do conteúdo das reviews; id e unixReviewTime, que são específicos e não acrescentam valor analítico significativo; e reviewTime e summary, dados considerados não essenciais para o foco da análise. A eliminação destas colunas simplifica os modelos, focando nas características das reviews que realmente podem indicar a sua autenticidade, além de garantir eficiência computacional e clareza nos resultados.

4. Modelação

Para a modelação, após a preparação e transformação dos dados, foram implementados modelos de aprendizagem supervisionada utilizando PySpark. O processo incluiu a preparação dos dados textuais e numéricos.

Primeira Abordagem (Text Processing)

Na parte de tratamento de texto, utilizou-se uma pipeline de pré-processamento que incluiu etapas de tokenização, remoção de palavras irrelevantes (stop words), transformação com HashingTF e cálculo do IDF (Inverse Document Frequency). Esta pipeline foi ajustada aos dados e aplicada para transformar as reviews textuais em uma representação numérica adequada para a modelação.

Segunda Abordagem (Feature Engineering)

Inicialmente, selecionaram-se as colunas relevantes para a modelação, como 'helpfulTotalRatio', 'reviewUpVotes', 'reviewLength', 'isWeekend', 'productPopularity', 'avgProductRating', 'containsQuestion', 'hasLink' e 'reviews_by_user'. Utilizou-se o VectorAssembler para combinar essas colunas em uma única coluna de recursos denominada features, transformando o DataFrame para incluir a nova coluna features e a coluna class, que serve como variável alvo.

Para facilitar a análise e a modelação em escala reduzida, realizou-se a amostragem de 1% e 5% do DataFrame, salvando os DataFrames como arquivos Parquet para utilização posterior nos modelos de aprendizagem supervisionada.

Os modelos implementados incluem Regressão Logística, Support Vector Machine (SVM), Árvores de Decisão, Florestas Aleatórias e Gradient-Boosted Trees. A Regressão Logística foi utilizada para prever a probabilidade de uma instância pertencer a uma determinada classe (spam ou não spam). O SVM foi usado para encontrar um hiperplano que separa as classes de dados com a maior margem possível. Árvores de Decisão foram utilizadas para tomar decisões baseadas em condições sobre as características dos dados, representadas em forma de árvore. Florestas Aleatórias combinam múltiplas árvores de decisão para formar um único modelo mais robusto (ensemble learning). Finalmente, Gradient-Boosted Trees combinam várias árvores de decisão ajustadas aos erros residuais das árvores anteriores.

5. Resultados

Para avaliar o desempenho dos modelos implementados recorreu-se a métricas de desempenho como a accuracy, precision, recall e F1-score para as duas: classe 0 (não spam) e classe 1 (spam) (Anexo 2).

Primeira Abordagem (Text Processing)

O modelo de **Regressão Logística** obteve uma accuracy de 74%. Para a classe 0, a precision foi de 74% e o recall de 75%, com um F1-score de 74%. Na classe 1, a precision foi de 74% e o recall de 72%, resultando num F1-score de 73%. O modelo apresenta um desempenho equilibrado entre as duas classes.

O **Support Vector Machine** destacou-se como o modelo mais eficaz, considerando as diferentes métricas, com uma accuracy, de 79%. Para a classe 0, a precision foi de 79% e o recall de 80%, o

que resulta um F1-score de 80%. Para a classe 1, a precision foi de 80% e o recall de 78%, com um F1-score de 79%.

As **Árvores de Decisão** apresentaram uma accuracy de 67%. Para a classe 0, a precision foi de 63% e o recall de 85%, resultando num F1-score de 73%. Para a classe 1, a precision foi de 77% e o recall de 49%, comum F1-score de 60%. O desempenho é inconsistente entre as classes, limitando a utilidade do modelo.

O modelo **Florestas Aleatórias** registou menor accuracy, de 61%, comparando com os restantes. Para a classe 0, a precision foi de 63% e o recall de 56%, resultando num F1-score de 59%. Para a classe 1, a precision foi de 60% e o recall de 67%, com um F1-score de 63%. O modelo apresenta desempenho fraco e inconsistente.

O **Gradient-Boosted Trees** obteve uma accuracy de 72%. Para a classe 0, a precision foi de 68% e o recall de 84%, com um F1-score de 75%. Para a classe 1, a precision foi de 78% e o recall de 59%, com um F1-score de 67%. O modelo é robusto, mas não é o melhor para todos os cenários.

Segunda Abordagem (Feature Engineering)

O modelo de **Regressão Logística** mostrou uma performance equilibrada entre as duas classes, com uma accuracy de 71%, com boa precision e recall para a classe 0, de 72% e 67%, respetivamente. Para a classe 1, a precision foi de 70% e o recall de 75%. Este modelo é capaz de identificar os exemplos para ambas as classes.

O **Support Vector Machine** obteve uma accuracy semelhante de 71%, porém com uma distribuição diferente entre classes. Para a classe 0, a precision foi alta de 75%, mas o recall foi menor de 62%. Para a classe 1, o recall foi bastante alto, de 80%, enquanto a precision obteve um valor mais baixo de 68%. Os resultados mostram que o SVM é eficaz na identificação de exemplos da classe 1, com um desempenho um pouco inferior na classe 0 em comparação com a Regressão Logística.

O modelo **Árvores de Decisão** alcançou uma accuracy ligeiramente superior, de 72%. Para classe 0, o modelo apresenta um recall de 66% e uma precision razoável de 74%. Para a classe 1, a precision foi alta, de cerca de 70%, mas o recall foi de apenas 77%. Os resultados sugerem que o modelo é melhor em identificar exemplos da classe 0 do que da classe 1.

As **Florestas Aleatórias** também atingiram uma accuracy de 73%. Para classe 0, o modelo apresentou uma alta precision de 74% e um recall mais baixo de 69%. Por outro lado, na classe 1, o recall foi bastante alto de cerca de 76% e a precision foi moderada de 72%. Este modelo destaca-se na identificação da classe 1, com uma capacidade menos consistente na identicação de exemplos da classe 0.

O modelo **Gradient-Boosted Trees** apresentou uma accuracy, de 73%. Para classe 0, tanto a precision quanto o recall são equilibrados, tendo registando os 72% e 75%, respetivamente. Para classe 1, o modelo também apresenta uma boa precision de 75% e um recall de 71%. Este modelo demostra um desempenho consistente e equilibrado em ambas as classes.

6. Conclusão

Os modelos desenvolvidos possuem uma capacidade específica em distinguir entre reviews genuínas e spam, embora ainda existam margens para melhoria. A resolução precisa da classificação desempenha um papel fundamental na confiança das reviews, influenciando diretamente a tomada de decisão dos consumidores no momento da compra.

No processo de tratamento de texto, a utilização de uma pipeline personalizada revelou-se a abordagem mais eficaz para a classificação das reviews textuais. Esta pipeline foi adaptada aos dados e incluiu etapas de tokenização, remoção de stop words, transformação com HashingTF e cálculo do IDF (Inverse Document Frequency), permitindo assim uma representação numérica adequada para a modelação. Destaca-se que a abordagem de processamento de texto foi a mais eficaz comparada com o feature engineering tradicional.

Entre os modelos avaliados, a melhor técnica para classificação de spam foi utilizando o SVM. Este modelo obteve os melhores resultados, demonstrando uma grande capacidade de distinguir entre reviews genuínas e spam.

Além da abordagem de processamento de texto, também exploramos uma abordagem de feature engineering. Esta abordagem envolveu a seleção e transformação de várias colunas relevantes para a modelação, facilitando a criação de recursos que melhoram a precisão dos modelos preditivos. Embora eficaz, esta abordagem não superou o desempenho da pipeline de processamento de texto.

Este projeto destaca a importância de uma abordagem iterativa na construção e melhoria contínua dos modelos de análise de dados em larga escala. Tal abordagem é essencial para garantir uma maior confiabilidade para os consumidores online, promovendo assim a integridade e transparência das plataformas de e-commerce como a Amazon.

Melhorias Futuras

Para aprimorar ainda mais o projeto, várias abordagens podem ser consideradas:

- **Técnicas Avançadas de Processamento de Linguagem Natural (NLP):** Implementar técnicas mais avançadas de NLP, como embeddings de palavras (Word2Vec ou GloVe) ou modelos baseados em transformadores (BERT, GPT), poderia melhorar significativamente a representação de texto e a precisão dos modelos.

- **Ajuste de Hiperparâmetros:** Realizar uma busca mais extensa e sistemática de hiperparâmetros para cada modelo pode aumentar o desempenho. Técnicas como Grid Search ou Random Search, bem como métodos mais avançados como o Bayesian Optimization, poderiam ser utilizadas.

- **Modelos Ensemble:** Combinar diferentes modelos preditivos através de técnicas de ensemble learning, como Stacking, pode melhorar a robustez e a precisão do sistema de detecção de spam.

7. Bibliografia

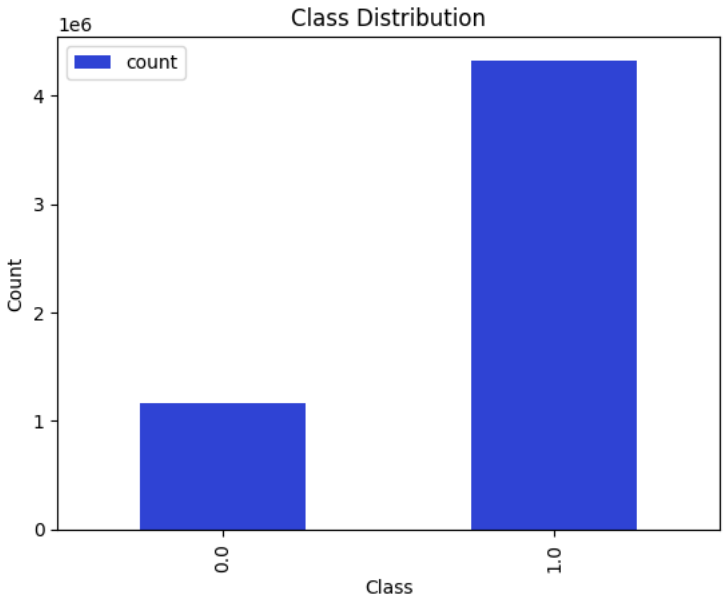
Naveed, H. N. (n.d.). Amazon Product Review Spam and Non-Spam. Kaggle. Disponível em: https:/www.kaggle.com/datasets/naveedhn/amazon-product-review-spam-and-non-spam?select=Clothing_Shoes_and_Jewelry

Hussain, N., Mirza, H. T., Hussain, I., Iqbal, F., & Memon, I. (2020). Review Detection Using the Linguistic and Spammer Behavioral Methods. IEEE. Disponível em: https://www.researchgate.net/publication/339821179_Spam_Review_Detection_Using_the_Linguistic_and_Spammer_Behavioral_Methods

Sampath, A. (n.d.). Amazon Review Spam Detection. Kaggle. Disponível em: https://www.kaggle.com/code/abhilashsampath/amazon-review-spam-detection

8. Anexos

Anexo 1 - Desequilíbrios entre as classes da variável target



Anexo 2 - Resultados das duas abordagens

	Accuracy	Classe0_Precison	Classe0_Recall	Classe0_F1-Score	Classe1_Precision	Classe1_Recall	Classe1_F1-Score
Regressão Logística	0,74	0,74	0,75	0,74	0,74	0,72	0,73
Support Vector Machine	0,79	0,79	0,80	0,80	0,80	0,78	0,79
Árvores de Decisão	0,67	0,63	0,85	0,73	0,77	0,49	0,60
Florestas Aleatórias	0,61	0,63	0,56	0,59	0,60	0,67	0,63
Gradient-Boosted Trees	0,72	0,68	0,84	0,75	0,78	0,59	0,67

	Accuracy	Classe0_Precison	Classe0_Recall	Classe0_F1-Score	Classe1_Precision	Classe1_Recall	Classe1_F1-Score
Regressão Logística	0,71	0,72	0,67	0,70	0,70	0,75	0,73
Support Vector Machine	0,71	0,75	0,62	0,68	0,68	0,80	0,74
Árvores de Decisão	0,72	0,74	0,66	0,70	0,70	0,77	0,74
Florestas Aleatórias	0,73	0,74	0,69	0,71	0,72	0,76	0,74
Gradient-Boosted Trees	0,73	0,72	0,75	0,73	0,75	0,71	0,73