Análise Comparativa de Modelos de Classificação para Detecção de Spam em Emails

Marcus Toledo

Universidade Tecnológica Federal do Paraná (UTFPR)
Dois Vizinhos – PR – Brazil

marcustoledo@alunos.utfpr.edu.br

Resumo. *Este artigo apresenta uma análise comparativa de diversos modelos de classificação utilizados em sistemas inteligentes, aplicando-os a um dataset de detecção de spam. Foram avaliados 7 algoritmos: Naive Bayes, Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbors e Rede Neural. O dataset utilizado foi obtido da UCI Machine Learning Repository, contendo características de emails com classificações de spam ou não-spam. O processo envolveu a normalização dos dados e a divisão em conjuntos de treinamento e teste. Cada modelo foi treinado e avaliado com base na acurácia, relatório de classificação e matriz de confusão.*

Abstract. *This paper presents a comparative analysis of various classification models used in intelligent systems, applying them to a spam detection dataset. Seven algorithms were evaluated: Naive Bayes, Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbors, and Neural Network. The dataset, sourced from the UCI Machine Learning Repository, contains features of emails labeled as spam or non-spam. The process involved normalizing the data and splitting it into training and testing sets. Each model was trained and evaluated based on accuracy, classification report, and confusion matrix..*

1. Introdução

A classificação de emails como spam ou não spam é um desafio importante no campo da filtragem de conteúdo digital, devido ao grande volume de emails indesejados que circulam diariamente. Esses emails não só causam incômodo aos usuários, mas também podem representar riscos à segurança, como phishing e disseminação de malware. Este trabalho utiliza o dataset "Spambase" do UCI Machine Learning Repository para treinar e avaliar diversos modelos de classificação com o objetivo de identificar emails de spam.

2. Fundamentação Teórica

Nesta parte, apresentarei os conceitos e teorias fundamentais por trás dos modelos de classificação utilizados:

- **Naive Bayes**: Um classificador probabilístico baseado no teorema de Bayes com a suposição da independência entre as características.
- **Logistic Regression**: Um modelo estatístico que utiliza uma função logística para modelar a probabilidade de um evento binário.
- **Decision Tree**: Um modelo de aprendizado que utiliza uma estrutura de árvore para tomar decisões baseadas em características dos dados.

- **Random Forest**: Um conjunto de árvores de decisão que melhora a precisão de classificação através da combinação de múltiplas árvores.
- **Support Vector Machine**: Um modelo de aprendizado que encontra o hiperplano ótimo para separar classes em um espaço de características.
- **K-Nearest Neighbors**: Um método de classificação baseado na proximidade dos dados em relação aos vizinhos mais próximos.
- **Rede Neural Artificial:** Um modelo inspirado no funcionamento do cérebro humano, composto por camadas de neurônios artificiais que processam dados de entrada.

2.1. Dataset

A base de dados utilizada neste estudo é a "Spambase", disponível no UCI Machine Learning Repository. Esta base de dados foi desenvolvida pela Hewlett-Packard Labs e é amplamente utilizada para estudos de classificação de emails como spam ou não spam. Essa base contém um total de 4601 amostras, cada uma representando um email que pode ser classificado como spam ou não spam, e possui 57 características numéricas extraídas de cada email, além de uma coluna adicional que indica a classe do email (spam ou não spam).

As características podem ser divididas em 3 categorias principais:

- **Frequência de Palavras**: Representa a porcentagem de palavras específicas encontradas no email em relação ao total de palavras.
- **Frequência de caracteres especiais**: Representa a frequência de caracteres especiais no email.
- **Comprimento de sequência de caracteres em caixa alta**: Representa a estatística sobre a sequência de caracteres em caixa alta no email.

Para esse estudo, foram selecionadas todas as características do dataset para melhorar a eficiência do treinamento dos modelos. Os dados foram divididos em conjuntos de treinamento (80%) e teste (20%) para permitir a avaliação do desempenho dos modelos. Além disso, os dados foram normalizados para garantir que todas as características estejam na mesma escala, o que é importante para algoritmos que são sensíveis à escala de dados, como SVM e redes neurais.

3. Metodologia

Para esse experimento utilizei todas as 57 características disponíveis no dataset, para garantir uma análise completa e abrangente, também realizei a divisão dos dados em conjuntos de treinamento e teste com uma semente aleatória para garantir a reprodutibilidade dos resultados. Além disso foi realizada a normalização das características com a biblioteca StandardScaler para padronizar os dados.

3.1. Métricas de avaliação

Para avaliar o desempenho dos modelos de classificação, utilizamos as seguintes métricas:

- **Acurácia**: Medida que indica a proporção de previsões corretas em relação ao total de previsões. Foi utilizada tanto a acurácia percentual quanto a quantidade de amostras corretamente classificadas.
- **Matriz de confusão**: Permite visualizar o desempenho do modelo, mostrando a distribuição de previsões verdadeiras e falsas para cada classe (spam e não spam). A matriz de confusão ajuda a identificar erros específicos do modelo.
- **Taxa de Verdadeiro Positivo (TPR) e Taxa de Verdadeiro Negativo (TNR)**: Representa respectivamente a proporção de verdadeiros positivos (spam identificado corretamente) em relação ao total de casos positivos e a proporção de verdadeiros negativos (não spam identificado corretamente) em relação ao total de casos negativos.
- **Relatório de Classificação**: Inclui precisão, recall e F1-score para cada classe. Essas métricas são importantes para entender como o modelo se comporta em termos de classificação.
- **Pontuação Composta**: Uma métrica geral combinando todas as métricas mencionadas acima. A pontuação composta é calculada como a média de acurácia, precisão média, recall médio, F1-score médio, TPR e TNR, oferecendo uma visão holística do desempenho do modelo.

3.2. Treinamento

O treinamento foi realizado utilizando métodos implementados pela biblioteca Scikit-learn. A seguir uma pequena lista contendo todos os parâmetros e métodos utilizados durante o treinamento dos modelos:

- **Naive Bayes:** Utilizamos o método GaussianNB que implementa o algoritmo de Naive Bayes Gaussiano.
- **Logistic Regression**: Utilizamos o método LogisticRegression com o parâmetro *max_iter=1000*, que implementa o algoritmo de Regressão Logística com um máximo de 1000 iterações.
- **Decision Tree**: Utilizamos o método DecisionTreeClassifier, que implementa o algoritmo de Árvore de Decisão sem parâmetros adicionais especificados.
- **Random Tree**: Utilizamos o método RandomForestClassifier, que implementa o algoritmo de Floresta Aleatória sem parâmetros adicionais especificados.
- **Support Vector Machine**: Utilizamos o método SVC, que implementa o algoritmo de Máquina de Vetores de Suporte com o kernel padrão (RBF - Radial Basis Function)
- **K-Nearest Neighbors**: Utilizamos o KneighborsClassifier, que implementa o algoritmo de K-Nearest Neighbors com o número padrão de vizinhos (k=5)
- **Rede Neural Artificial**: Utilizamos o método MLPClassifier com os parâmetros *hidden_layer_sizes=(12,8)* e *max_iter=1000,* que implementa uma Rede Neural

Artificial com duas camadas ocultas contendo 12 e 8 neurônios, respectivamente, e um máximo de 1000 iterações.

4. Resultados

Nesta seção, apresentamos uma análise comparativa do desempenho dos modelos de classificação utilizados neste estudo. Os gráficos a seguir ilustram as principais métricas avaliadas, incluindo acurácia, a taxa de verdadeiros positivos (TPR), a taxa de verdadeiros negativos (TNR), precisão, recall e f1-score. Essas métricas fornecem uma visão abrangente de eficácia de cada modelo em diferentes aspectos, permitindo uma comparação detalhada do desempenho.

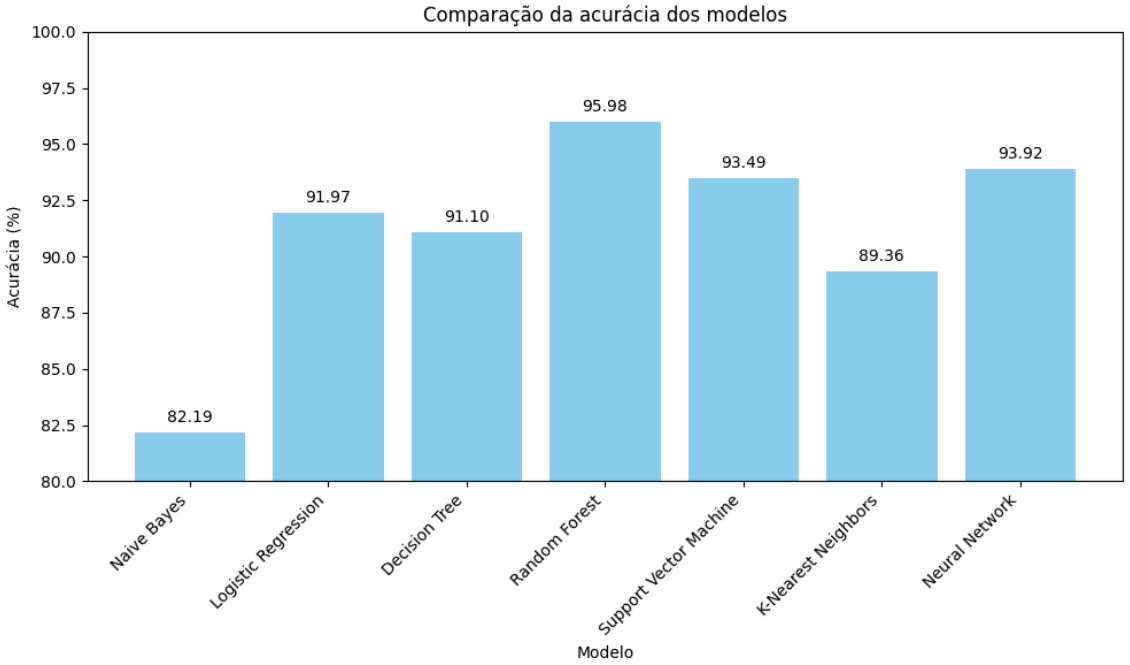

Figure 1. Gráfico de comparação da acurácia entre os modelos.

O primeiro gráfico destaca a acurácia dos diferentes modelos de classificação avaliados. A acurácia é uma métrica fundamental que representa a proporção de previsões corretas feitas pelo modelo em relação ao total de previsões. É uma medida geral da eficiência do modelo, indicando o quão bem ele está performando ao considerar todas as classes. O modelo Random Forest destacou-se com a maior acurácia, sugerindo um desempenho robusto na identificação correta das classes.

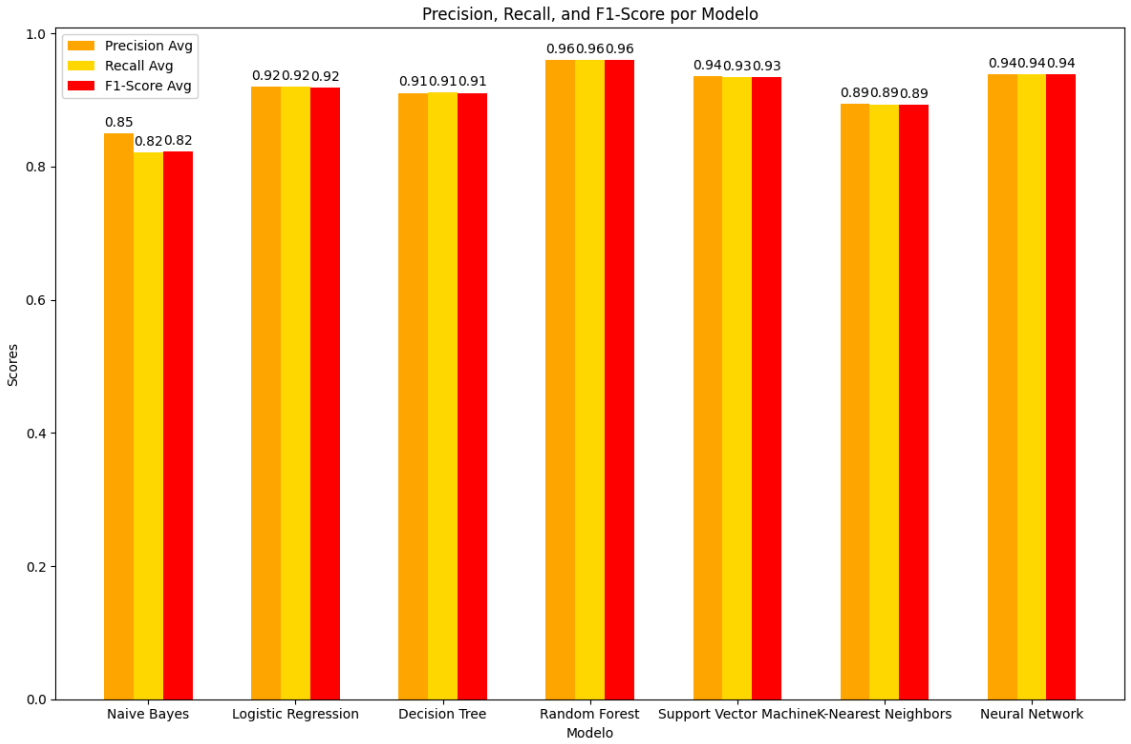

Figure 2. Gráfico de comparação entre as médias de precisão, recall e f1-score por modelo.

No segundo gráfico é apresentado uma comparação entre as médias de precisão, recall e f1-score para cada modelo. A precisão indica a proporção de verdadeiros positivos entre todas as previsões positivas feitas pelo modelo. O recall, também conhecido como sensibilidade, mede a capacidade do modelo de identificar corretamente os verdadeiros positivos em relação ao total de positivos reais. O f1-score é a média harmônica da precisão e do recall, proporcionando uma única métrica que equilibra ambas. Essas métricas são essenciais para avaliar a qualidade das previsões dos modelos, especialmente em cenários onde a distribuição de classes é desequilibrada.

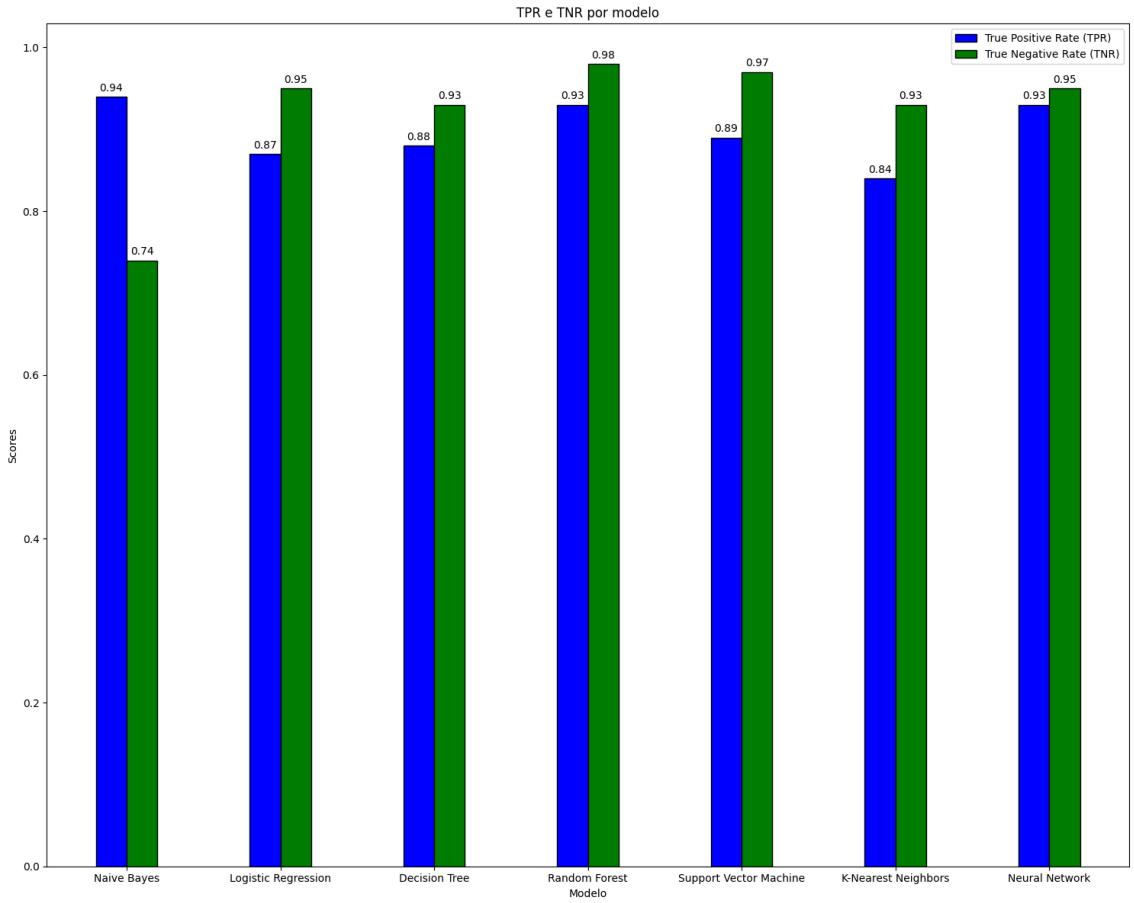

Figure 3. Gráfico de comparação entre as taxas de Verdadeiros Positivos e Verdadeiros Negativos por modelo.

Em seguida, os gráficos de TPR e TNR apresentam a comparação entre a Taxa de Verdadeiros Positivos e a Taxa de Verdadeiros Negativos para os modelos avaliados. A TPR, também conhecida como sensibilidade, mede a proporção de verdadeiros positivos corretamente identificados pelo modelo em relação ao total de positivos reais. Já a TNR, ou especificidade, indica a proporção de verdadeiros negativos corretamente identificados pelo modelo em relação ao total de negativos reais. Essas métricas são cruciais para entender a performance dos modelos em identificar corretamente ambas as classes.

Além disso, a TPR está diretamente relacionada ao recall, uma vez que ambas as métricas medem a capacidade do modelo de recuperar todos os verdadeiros positivos. A precisão, por outro lado, indica quantos dos positivos previstos são realmente corretos, e o f1-score combina estas duas métricas para proporcionar uma visão balanceada da performance do modelo. Dessa forma, TPR e recall fornecem insights complementares, enquanto precisão e f1-score ajudam a entender a eficácia geral do modelo na classificação correta das amostras.

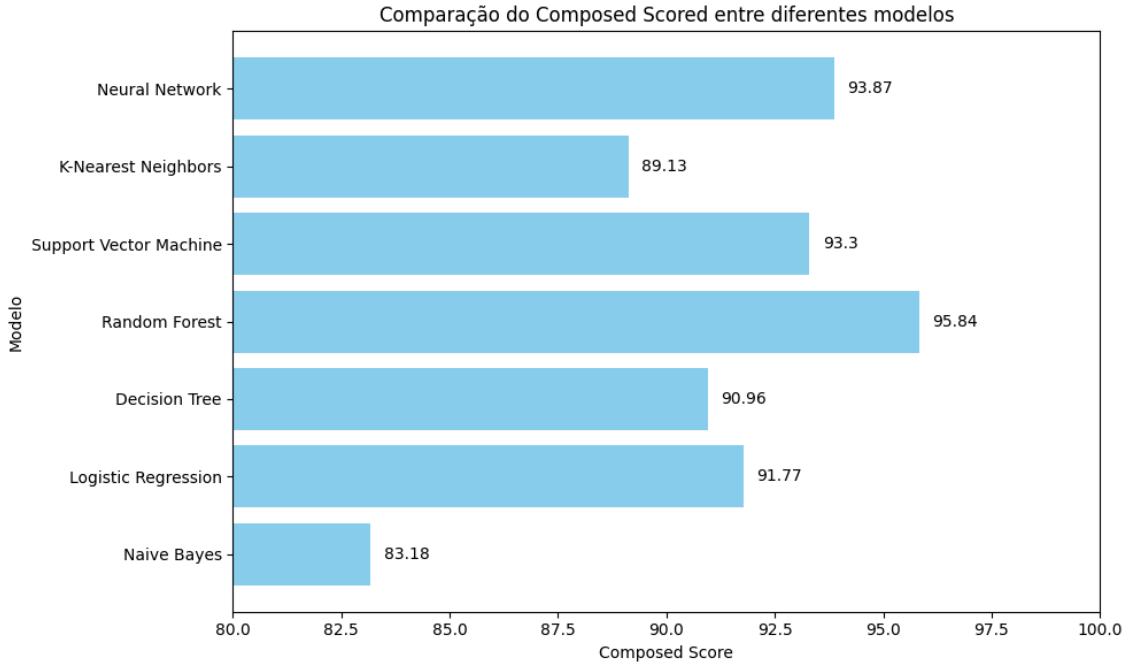

Figure 4. Gráfico de comparação entre a pontuação composta por modelo.

O gráfico final compara a pontuação composta de cada modelo, uma métrica agregada que considera várias dimensões de desempenho. A pontuação composta é calculada a partir de diferentes métricas de avaliação, proporcionando uma visão consolidada do desempenho geral de cada modelo. Este gráfico ajuda a identificar qual modelo oferece o melhor equilíbrio entre todas as métricas consideradas, facilitando a escolha do modelo mais adequado para o problema em questão. O Random Forest novamente se destaca com a maior pontuação composta, indicando sua superioridade no equilíbrio das diversas métricas de performance.

5. Conclusão

Neste estudo, realizamos uma análise comparativa de diversos modelos de classificação para avaliar seu desempenho em várias métricas essenciais. Os modelos foram avaliados com base na acurácia, precisão, recall, f1-score, taxa de verdadeiros positivos (TPR) e taxa de verdadeiros negativos (TNR). Além disso, calculamos uma pontuação composta para fornecer uma visão consolidada do desempenho geral de cada modelo.

Os resultados mostraram que o modelo Random Forest se destacou em várias métricas, apresentando a maior acurácia e uma pontuação composta superior. Este modelo também teve altos índices de TPR e TNR, indicando sua eficácia em prever ambas as classes com alta precisão. Outros modelos, como o Support Vector Machine e a Neural Network, também demonstraram desempenho promissor, embora um pouco inferior ao Random Forest.

A análise detalhada das métricas individuais, como precisão, recall e f1-score, revelou a capacidade dos modelos em diferentes aspectos de classificação. Por exemplo, enquanto alguns modelos apresentaram alta precisão, outros destacaram-se no recall, o que é crucial em cenários com classes desbalanceadas.

Em conclusão, o modelo Random Forest mostrou-se como a escolha mais equilibrada e eficiente para a tarefa proposta, considerando todas as métricas avaliadas.

References

H. Zhang (2004). The optimality of Naive Bayes. Proc. FLAIRS.

Scikit-learn. "Naive Bayes", https://scikit-learn.org/stable/modules/naive_bayes.html.

Amazon Web Services. "O que é Regressão Logística?". https://aws.amazon.com/pt/what-is/logistic-regression/.

Scikit-learn. "Decision Trees", https://scikit-learn.org/stable/modules/tree.html#decision-trees.

Wikipedia. Random forest, https://en.wikipedia.org/wiki/Random_forest

Scikit-learn. Support Vector Machines, https://scikit-learn.org/stable/modules/svm.html.

Scikit-learn, K-nearest neighbors classifier, https://scikit-learn.org/stable/modules/neighbors.html.

Scikit-learn, Neural network models, https://scikit-learn.org/stable/modules/neural_networks_supervised.html.

Dhande, Saurabh. Confusion Matrix Explained: Calculating Accuracy, TPR, FPR, TNR, Precision and Prevalence, https://medium.com/@saurabhdhandeblog/confusion-matrix-explained-calculating-accuracy-tpr-fpr-tnr-precision-and-prevalence-87557fe8714d