Classificação de Spam utilizando o Classificador Naive Bayes

Caio da Silva Pinheiro[1]

[1]Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET-RJ)
Petrópolis- RJ- Brasil

caio.pinheiro.1@aluno.cefet-rj.br

Abstract. *Spam detection in emails remains a critical challenge in digital communication. This article explores the application of Multinomial Naive Bayes for spam classification. The model achieved 98.65% accuracy on the test set, and the results indicate that Naive Bayes, despite its simplicity, is highly effective for spam filtering when combined with proper preprocessing and imbalance mitigation techniques.*

Resumo. *A detecção de spam em e-mails permanece um desafio crítico na comunicação digital. Este artigo explora a aplicação do Multinomial Naive Bayes para classificação de spam, o modelo alcançou 98.65% de acurácia no teste, os resultados indicam que o Naive Bayes, apesar de sua simplicidade, é altamente eficaz para filtragem de spam quando combinado com pré-processamento adequado e técnicas de mitigação de desbalanceamento.*

1. Introdução

O e-mail se popularizou com o crescimento da internet e, atualmente, é praticamente obrigatório que qualquer pessoa possua um endereço de e-mail, especialmente com o uso crescente de smartphones. Com esse aumento, o volume de mensagens eletrônicas também cresceu exponencialmente, trazendo consigo um grande desafio: a identificação e filtragem de spam.

Spam (Sending and Posting Advertisement in Mass, ou ”Enviar e Postar Publicidade em Massa”) é o termo usado para classificar e-mails indesejados, frequentemente contendo propagandas, golpes, vírus e ataques de phishing. Esses e-mails, muitas vezes, são prejudiciais tanto para a segurança quanto para a produtividade dos usuários.

A detecção eficaz de spam tornou-se cada vez mais desafiadora, uma vez que os criadores desses e-mails têm refinado suas táticas, fazendo com que pareçam cada vez mais legítimos. Isso fez com que filtros tradicionais, baseados apenas em regras, se mostrassem insuficientes para distinguir entre spam e e-mails legítimos.

Este artigo tem como objetivo explorar a aplicação de algoritmos de aprendizado de máquina com ênfase no Naive Bayes, na classificação de mensagens como spam ou não, buscando aprimorar a precisão na identificação de e-mails indesejados.

2. Descrição do Problema

O conjunto de dados utilizado neste trabalho é o *Spam Emails Dataset* [Wagih 2023], disponível em: Kaggle. O *dataset* possui 5572 entradas e apenas duas colunas:

- **Category**: Esta coluna possui dois valores possíveis, ”ham”(não spam) e ”spam”, indicando a classificação de cada e-mail.
- **Message**: Contém o conteúdo textual de cada e-mail, que será analisado para determinar sua categoria.

Vale destacar que o conjunto de dados apresenta um desequilíbrio de classes, com uma predominância de e-mails classificados como ”ham”, o que pode impactar o desempenho do modelo e requerer técnicas específicas para lidar com esse desbalanceamento.

3. Descrição da Solução

Para a resolução deste problema de classificação de spam, foi adotada uma abordagem de aprendizado supervisionado com o uso do *Multinomial Naive Bayes*. A solução foi estruturada em várias etapas, com destaque para a transformação de dados, o pré-processamento, a escolha do modelo e o tratamento do desbalanceamento entre as classes. Para a implementação do modelo, foram utilizadas as bibliotecas NumPy [Walt et al. 2011], Pandas [McKinney 2010] e scikit-learn [Pedregosa et al. 2011].

3.1. Transformação da Coluna `Category` em Valores Numéricos (Binários)

A coluna `Category` original, que contém os valores *ham* (não spam) e *spam*, foi transformada em uma coluna binária, onde *ham* foi mapeado para o valor 0 e *spam* para o valor 1. Essa transformação é necessária, pois o modelo de aprendizado de máquina não pode lidar diretamente com variáveis categóricas, sendo necessário representar as classes numericamente para que o algoritmo consiga processá-las corretamente.

3.2. Pré-processamento e Vetorização com TF-IDF

Antes de treinar o modelo, foi realizada a etapa de pré-processamento nos dados de texto. A principal técnica utilizada para converter os textos dos e-mails em uma forma que pudesse ser processada pelo modelo foi a *TF-IDF* (Term Frequency-Inverse Document Frequency). Essa técnica atribui um peso a cada palavra com base em sua frequência no documento e em sua raridade no conjunto de dados, o que permite destacar palavras mais significativas e reduzir o impacto de palavras muito comuns, mas irrelevantes (como *stopwords*). A transformação *TF-IDF* foi aplicada sobre a coluna `Message`, gerando uma representação vetorial densa que foi utilizada como entrada para o modelo.

3.3. Uso do Modelo Multinomial Naive Bayes

Para a classificação, optou-se pelo *Multinomial Naive Bayes* (`MultinomialNB`), que é particularmente adequado para tarefas de classificação de texto, como no caso da detecção de spam. O `MultinomialNB` é mais eficaz em problemas onde os dados podem ser representados por frequências de palavras (como é o caso aqui). Ele assume que as características (palavras) seguem uma distribuição multinomial, o que é uma suposição razoável para dados de texto.

Embora existam outras variantes do Naive Bayes, como o *Gaussian Naive Bayes* (`GaussianNB`) e o *Categorical Naive Bayes* (`CategoricalNB`), esses não foram escolhidos para este problema. O `GaussianNB` assume que as características seguem uma distribuição normal, o que não é o caso para os dados de texto, onde as características (palavras) não têm uma distribuição contínua. Já o `CategoricalNB` é mais

adequado para variáveis categóricas, mas os dados de entrada, após a vetorização, são representados por valores contínuos (resultantes da transformação *TF-IDF*), o que torna o `MultinomialNB` a escolha mais apropriada.

Além disso, foi realizado um ajuste no parâmetro α do modelo *Multinomial Naive Bayes*, que controla a suavização das probabilidades. O valor de α foi ajustado para 0.08, o que ajudou a melhorar a generalização do modelo, evitando tanto o sobreajuste quanto o subajuste. A escolha desse valor foi feita com base em uma avaliação empírica do desempenho do modelo durante o treinamento e validação cruzada, buscando um equilíbrio entre a precisão e a capacidade de adaptação às características do conjunto de dados.

3.4. Tratamento do Desbalanceamento com o Parâmetro `class_prior`

O conjunto de dados apresenta um desbalanceamento significativo entre as classes, com uma predominância de e-mails classificados como *ham* (não spam). Para lidar com esse desbalanceamento, foi utilizado o parâmetro `class_prior` do `MultinomialNB`. Esse parâmetro permite ajustar as probabilidades a priori das classes, de modo que o modelo penalize menos os erros na classe minoritária (spam), melhorando sua capacidade de detecção de e-mails indesejados. Dessa forma, o modelo consegue gerar previsões mais equilibradas e evitar um viés em favor da classe majoritária.

4. Resultados

Nesta seção, são apresentados os resultados obtidos com a aplicação do modelo *Multinomial Naive Bayes* para a tarefa de classificação de e-mails como *ham* ou *spam*. Diversas métricas foram utilizadas para avaliar a performance do modelo, incluindo a acurácia, validação cruzada, matriz de confusão, e o *log-loss*.

4.1. Relatório de Classificação

O relatório de classificação, apresentado na Tabela 1, fornece as métricas de precisão, recall e F1-score para as classes *ham* e *spam*. Essas métricas ajudam a entender o desempenho do modelo em termos de acurácia e de como ele lida com o desbalanceamento das classes.

Tabela 1. Relatório de Classificação

Classe	Precisão	Recall	F1-score	Suporte
Ham (0)	0.99	0.99	0.99	966
Spam (1)	0.94	0.96	0.95	149
Acurácia	98.65%			
Macro Avg	0.97	0.98	0.97	1115
Weighted Avg	0.99	0.99	0.99	1115

4.2. Acurácia

A acurácia do modelo foi avaliada tanto no conjunto de treinamento quanto no conjunto de teste. Os resultados são os seguintes:

- Acurácia no Treinamento: 99.84%
- Acurácia no Teste: 98.65%

Observa-se que o modelo apresenta uma boa performance tanto no conjunto de treinamento quanto no conjunto de teste.

4.3. Validação Cruzada

A validação cruzada foi realizada para avaliar a robustez do modelo. As acurácias médias obtidas durante a validação cruzada são:

- Acurácia média da validação cruzada no Treinamento: 98.04%
- Acurácia média da validação cruzada no Teste: 97.93%

Esses valores indicam a consistência do modelo ao longo de diferentes divisões dos dados, refletindo a capacidade de generalização.

4.4. Matriz de Confusão

A matriz de confusão, apresentada na Figura 1, fornece uma visão detalhada do desempenho do modelo nas diferentes classes. Ela mostra o número de falsos positivos, falsos negativos, verdadeiros positivos e verdadeiros negativos.

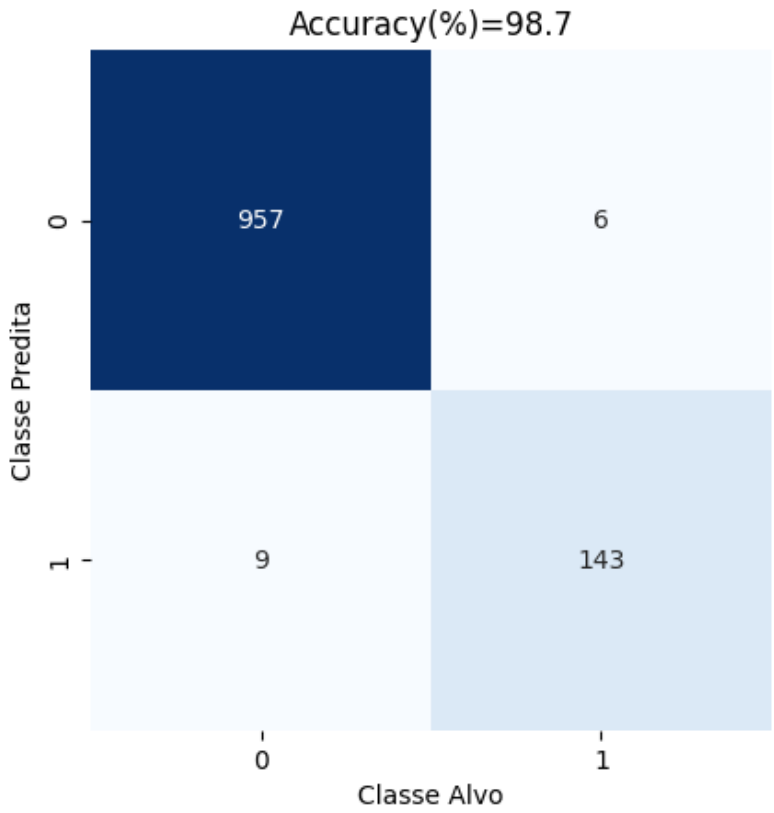

Figura 1. Matriz de Confusão

4.5. Log-Loss

O *log-loss* do modelo foi de 0.06. Esse valor indica o quanto o modelo penaliza as previsões incorretas. Um *log-loss* mais baixo indica uma boa capacidade do modelo de fornecer previsões próximas aos valores reais.

4.6. Discussão dos Resultados

Os resultados mostram que o modelo de *Multinomial Naive Bayes* obteve uma boa performance na classificação de e-mails, com acurácias elevadas no treinamento e no teste, além de um *log-loss* baixo. A matriz de confusão evidencia que o modelo consegue distinguir de forma eficaz entre as classes *ham* e *spam*. A validação cruzada demonstra que o modelo é robusto e generaliza bem para novos dados, mesmo com o desbalanceamento das classes.

5. Conclusão

Este trabalho demonstrou a eficácia do algoritmo Multinomial Naive Bayes na detecção de spam, mesmo em cenários de desbalanceamento de classes. A combinação de TF-IDF para representação textual e ajuste de priors resultou em um modelo com alta acurácia (98.65%), recall para spam (96%) e baixa log-loss (0.06), indicando confiança nas previsões. A validação cruzada confirmou a robustez do modelo, com variação mínima entre treino e teste.

Como limitações, destaca-se a dependência da qualidade do pré-processamento textual e a possível dificuldade em generalizar para novos padrões de spam não presentes no dataset. Para trabalhos futuros, sugere-se:

- Testar modelos como XGBoost ou redes neurais com embeddings contextualizados.
- Coletar mais dados da classe spam para reduzir o desbalanceamento.

Os resultados reforçam que técnicas clássicas de aprendizado de máquina, quando bem aplicadas, permanecem competitivas em problemas de classificação textual.

Referências

[McKinney 2010] McKinney, W. (2010). Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, pages 56–61.

[Pedregosa et al. 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., and Thirion, B. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

[Wagih 2023] Wagih, A. (2023). Spam emails dataset. Disponível em: Kaggle. Acessado em: 10 out. 2023.

[Walt et al. 2011] Walt, S. v. d., Colbert, S. C., and Varoquaux, G. (2011). The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30.