Aprendizado Supervisionado Utilizando Máquina de Vetores de Suporte Para Detecção de Spam Em Mensagens De E-mail

1st Ícaro Peretti Baseggio
Instituto Federal Catarinense
Videira, Brasil
icaroperetti50@gmail.com

***Resumo*—Com a utilização de técnicas do Machine Learning (aprendizado de máquina) e a linguagem de programação python, foi possível desenvolver um algoritmo capaz de identificar se uma mensagem recebida por e-mail trata-se ou não de um spam, mensagens indesejadas que são propagas em massa via mensagens de e-mail. Utilizando-se de recursos fornecidos pela biblioteca sklearn, desenvolvida com foco em aprendizagem de máquina, para realizar o treinamento de modelos de classificação, a fim de obter o resultado da verificação para tal identificação.**

***Index Terms*—machine learning, detecção de spam, aprendizado supervisionado, máquinas de vetores de suporte**

I. Introdução

Desde o advento da tecnologia e o surgimento da internet uma das mais significativas evoluções foi a capacidade de se comunicar de forma rápida e por meio de mensagens texto. Uma das pioneiras delas é o e-mail, criado pelo programador Ray Tomlinson com o objetivo de comunicar-se com seus colegas de trabalho[3].No ano de 2021 cerca de 319.6 bilhões de e-mails foram enviados e recebidos ao redor do mundo[2].

Todavia, nem todas as mensagens que são recebidas por e-mail todos os dias são mensagens desejadas, existe uma grande quantidade de spams (Stupid Pointless Annoying Messages), que se tratam de mensagens indesejadas que acabam sendo entregue de maneira online pelo serviço de e-mail,assim como acontecia com os correios antigamente[5], durante o período de outubro de 2020 a setembro de 2021, durante o mês de julho de 2021 houve uma grande concentração de e-mails contendo spam com cerca 283 bilhões de e-mails[4]. Tendo em vista a vasta quantidade de e-mails contendo spams sendo enviado globalmente técnicas para detecção destes e-emails houveram de ser desenvolvidas. Uma delas é a utilização de Máquinas de Suporte de vetor, mais conhecidas como Support Vector Machines[5][6].

II. Referencial Teórico

A. Aprendizado de Máquina

O aprendizado de máquina é uma subárea da inteligência artificial que tem por objetivo o desenvolvimento de sistemas computacionais capaz de obter conhecimento de maneira autônoma [1]. São algoritmos capazes de identificar padrões em um conjunto de dados tornando-os capazes de alterar seu comportamento de forma autônoma baseando-se em sua própria experiência e uma menor intervenção humana. O aprendizado de máquina pode ser aplicado de duas formas, de maneira supervisionada e de maneira não supervisionada.

B. Aprendizado Supervisionado

Os algoritmos de aprendizado de máquina que utilizam aprendizado supervisionado, de acordo com buscam obter um bom classificador a partir de uma base de dados rotulada, ou seja, se há uma base de dados com múltiplas imagens de gatos e cachorros cada uma adequadamente classificada (rotulada) o objetivo será desenvolver um modelo capaz de identificar a partir de novas imagens não rotuladas fornecidas ao algoritmo se nelas há um gato ou um cachorro. Outra forma é realizar a extração de características de uma imagem e converte-las para um vetor de valores de atributos. Ou seja, este tipo de aprendizagem depende de um professor externo[1].

- Atributos descrevem características ou aspectos de um exemplo podendo estes serem nominais (cores) ou contínuos (peso, números reais ou números inteiros)

A figura abaixo demonstra a geração de um modelo de aprendizado supervisionado. Os dados representam o conjunto de treinamento Xi....Xn, cada instância Xi = {Xi1...Xim} possui m atributos sendo estes categóricos ou continuos e estão associados a classe y. Com base nos exemplos o algoritmo de aprendizado de máquina é aplicado para gerar o classificador representado pelo f(x)[9]

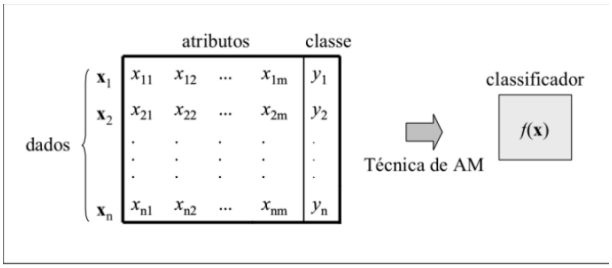

Figura 1. Indução de classificação em aprendizado supervisionado

C. Aprendizado Não Supervisionado

Em algoritmos de aprendizado de máquina que utilizam o aprendizado não supervisionado,o modelo receberá apenas os atributos de cada dado, sem a informação de que classe aquele dado pertence. Por meio do reconhecimento de algum padrão identificado pelo modelo no conjunto dos dados, ele irá tentar agrupar estes dados, conforme novos dados são adicionados é realizado um cálculo para tentar prever para qual grupo o dado pertence.

D. Máquina de Vetores de Suporte (SVM)

As máquinas de vetores de suporte foram desenvolvidas por Vapnik e possuem aplicação na resolução de problemas de classificação e regressão e trabalham de forma supervisionada.Neste artigo será aplicado para classificação visto que o problema a ser solucionado será classificar uma mensagem de e-mail como é spam ou não é spam [9].

Fundamentalmente, as máquinas de vetores de suporte buscam obter hiperplanos com a melhor separação entre as classes existentes para o conjunto de dados. Considerando um conjunto de dados binários 1 - Spam e 0 - Não Spam, a SVM tem por fim o intuito de construir um classificador capaz de separar novos exemplos desconhecidos em duas instâncias de forma adequada, a partir de dados que foram recebidos de forma rotulada durante a fase de treinamento do modelo. Basicamente, existindo duas classes e um conjunto de pontos pertencentes a cada classe, a SVM irá determinar o hiperplano que separa os pontos visando posicionar o maior numero de pontos de uma classe do mesmo lado e distanciando cada classe deste hiperplano [9].

Observando a figura a baixo, tendo como embasamento as características do formato das orelhas e o tamanho do focinho de gatos e cachorros, existe uma linha divisória para separar os gatos dos cachorros porém pode-se perceber que ao lado direito onde seriam a classe dos cachorros há um gato que de maneira incorreta foi classificado devido ao formato de seu focinho e falta de orelhas pontiagudas mas que se encontra na fronteira de divisão, estes dados que se encontram nesta fronteira são determinados vetores de suporte, fundamentais para determinar as margens, a partir destas margens um hiperplano ótimo será aplicado pelo SVM[10].

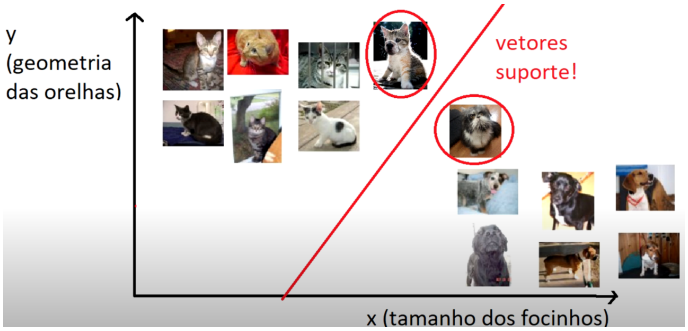

Figura 2. Hiperplano gatos e cachorros

E. Funções de Kernel

1) Kernel Linear: O Kernel linear é utilizado quando o conjunto de dados pode ser separado linearmente, ou seja, dados que podem ser separados em uma só linha. Normalmente utilizado quando existe um grande numero de recursos em um conjunto de dados, por exemplo na classificação de texto, um problema onde cada alfabeto é um novo recurso. A figura abaixo demonstra a comparação entre dados linearmente separaveis e não linearmente separáveis[11]

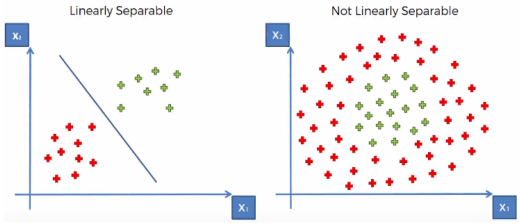

Figura 3. Separação de Classes de Conjuntos Lineares e Não Lineares [12]

2) Kernel Gaussiano (RBF): Este tipo de Kernel é amplamente utilizado para resolução de problemas de aprendizado de máquina, inclusive vem embutido em diversas bibliotecas de linguagem de programação que usam o algoritmo SVM. Na máquina utilizando o kernel RBF é possível resolver problemas que não sejam originalmente lineares através do mapeamento para um espaço maior de dimensões. Para obter o melhor resultado existem dois parâmetros que podem ser variados sendo eles y (gamma) e C (custo). Neste tipo de kernel os centros são definidos a partir dos vetores de suporte obtidos. Na figura abaixo o aglomerado de pontos indicam instâncias das classes e a tonalidade de fundo a qual classe pertence aquele conjunto[9].

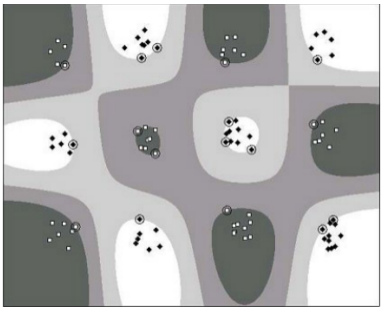

Figura 4. Separação de Classes Rbf

III. METODOLOGIA

Para realizar o desenvolvimento do presente trabalho a linguagem de programação escolhida foi o Python, o ambiente de desenvolvimento foi construído por meio da plataforma Colaboratory fornecida de forma gratuita pelo Google devido a existência de diversas ferramentas já instaladas visando o trabalho com Python e aprendizado de máquina. Para realizar o carregamento e pré-processamento do conjunto de dados foi utilizada a biblioteca pandas, biblioteca de código aberto voltada para a manipulação e análise de dados. Para realização do treinamento dos modelos a biblioteca escolhida foi a sklearn que já se encontra no ambiente do colab e devido a sua vasta aplicação para o machine learning torna o processo de desenvolvimento menos complexo, bibliotecas para impressão gráficos como a pyplot e para balanceamento de dados como a imblearn também foram utilizadas. Além disto foi definido o conjunto de dados, encontrado na plataforma kaggle através de múltiplas pesquisas, a plataforma trata-se de uma comunidade para pesquisadores e estudantes de aprendizado de máquina onde existem diversos conjuntos de dados para diferentes aplicações. O conjunto de dados escolhido possui cerca de 5171 mensagens e sua classificação sendo 0 para não spam e 1 para spam de diversas mensagens de e-mail na língua inglesa.

Previamente foi realizada uma análise do conjunto, inicialmente o conjunto de dados é carregado para um dataframe do pandas e então a remoção de todos os dados nulos é realizada, duas colunas contendo um identificador e a descrição textual da classe também foram removidas por serem desnecessárias, visto que a mensagem e a classe representada de forma binária seriam suficientes. Em seguida um método de vetorização de textos foi aplicado, chamado de CountVectorizer e também disponibilizado pela biblioteca sklearn, as mensagens de e-mail serão convertidas para um vetor onde cada vetor será uma palavra contida na mensagem,como mostra a imagem:

```
[enron, methanol, meter, 988291, follow, note,...
[hpl, nom, january, 9, 2001, see, attached, fi...
[neon, retreat, ho, ho, ho, around, wonderful,...
[photoshop, windows, office, cheap, main, tren...
[indian, springs, deal, book, teco, pvr, reven...
```

Figura 5. Mensagens vetorizadas

e então transformadas para um valor inteiro retornando assim um novo vetor contendo a quantidade de vezes que determinada palavra apareceu em uma mensagem.Logo após uma checagem da quantidade de classes é realizada, o que possibilitou a identificação de que as classes estavam desbalanceadas, como é possível observar na imagem a quantidade de mensagens spam é menor do que a quantidade de mensagens legítimas.

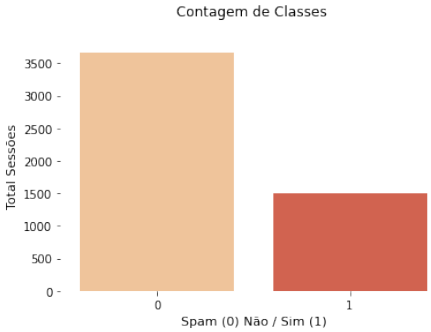

Figura 6. Classes Desbalanceadas

Contagem de classes após aplicação da técnica de smote, para realizar o balanceamento destas classes foi utilizado o smote combinado com o oversampling que tem por objetivo aumentar a quantidade de dados na classe faltante[8].

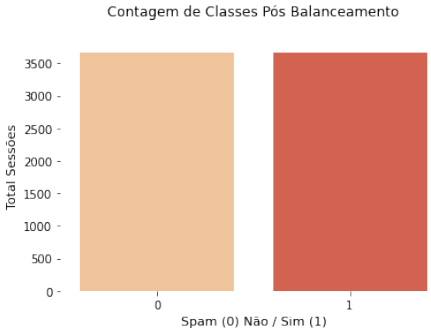

Figura 7. Classes Balanceadas

Após as etapas de pré-processamento e também de balanceamento dos dados o novo conjunto de dados foi separado entre conjuntos de teste e treinamento utilizando o método train_test_split também fornecido pela biblioteca sklearn, o conjunto foi dividido, sendo $\frac{2}{3}$ para treinamento e o restante será utilizado para teste e em sequência os modelos são treinados.

IV. RESULTADOS

Após a execução dos testes com máquinas de vetores de suporte utilizando os kernels linear e rbf foi possível obter bons resultados para ambos. A SVM combinada com o kernel linear obteve uma acurácia de 97% com uma taxa de verdadeiros positivos de 98% e uma taxa de verdadeiros negativos de 96%.

```
Acc: 0.9748349834983498
TPR: 0.9857142857142858
TNR: 0.9643435980551054
```

Figura 8. Resultados SVM Kernel Linear

Na matriz de confusão gerada é possível observar que o modelo foi capaz de prever que 1190 e-mails que não eram spam foram classifidos como não sendo spam, 44 e-mails que eram spam e foram classificados como não sendo spam, 17 e-mails que eram spam e foram classificados como não spam e 1173 e-mails que eram spam e realmente foram classificados como spam.

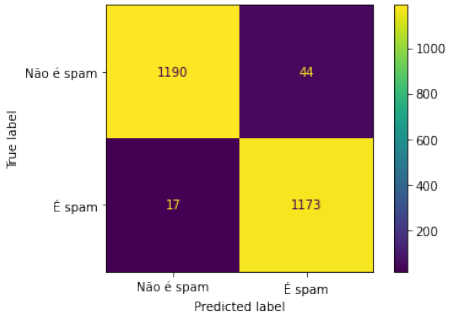

Figura 9. Matriz de Confusão SVM Kernel Linear

Já a curva ROC que é uma importante métrica a ser analisada para problemas de classificação, a curva ROC é capaz de mostrar quão capaz o modelo é de separar classes. A curva ROC recebe dois parâmetros, a taxa de verdadeiro positivo e a taxa de falsos positivos. O principal objetivo da curva ROC é permitir a melhor analize da AUC (Area under the curve) que é uma métrica que varia entre 0.0 até 1.0 onde um modelo que erra 100% das previsões possui uma AUC igual a 0 e um modelo que acerta 100% possui uma AUC de 1.0, portanto quanto mais próximo de 1 a AUC o algoritmo será superior na separação de classes[7]. A Figura 10 mostra a curva ROC do modelo utilizando o kernel linear com uma AUC de 0.98 indicando que o modelo é capaz de realizar uma boa separação de classes.

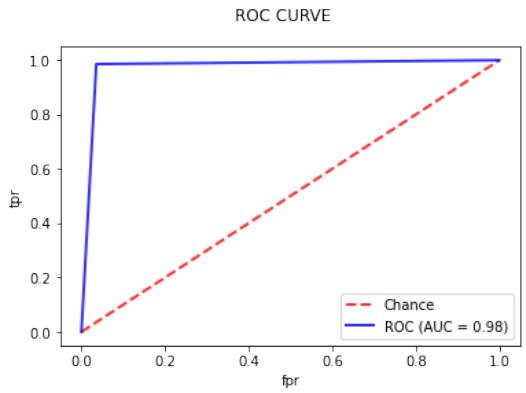

Figura 10. Curva ROC SVM Kernel Linear

Para o modelo SVM combinado ao kernel rbf foi apresentada uma acurácia de 94% com uma taxa de verdadeiros positivos de 99% e uma taxa de verdadeiros negativos de 89%.

```
Acc: 0.9434818481848185
TPR: 0.9966386554621849
TNR: 0.8922204213938412
```

Figura 11. Resultados SVM Kernel Rbf

Na matriz de confusão gerada é possível observar que o modelo foi capaz de prever que 1101 e-mails que não eram spam foram classifidos como não sendo spam, 133 e-mails que eram spam foram classificados como não sendo spam, 4 e-mails que eram spam foram classificados como não spam e 1186 e-mails que eram spam foram classificados como spam.

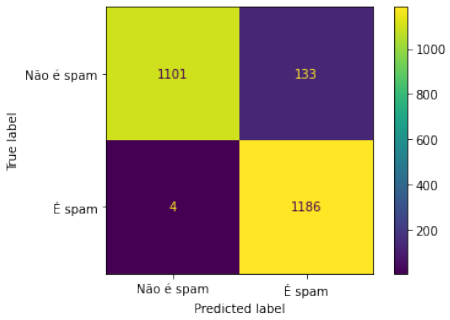

Figura 12. Matriz de Confusão SVM Kernel Linear

Já a métrica AUC apresentada pelo modelo que utilizou o kernel rbf apresentou um valor de 0.94 indicando novamente que o modelo também é capaz de realizar uma separação de classes adequada.

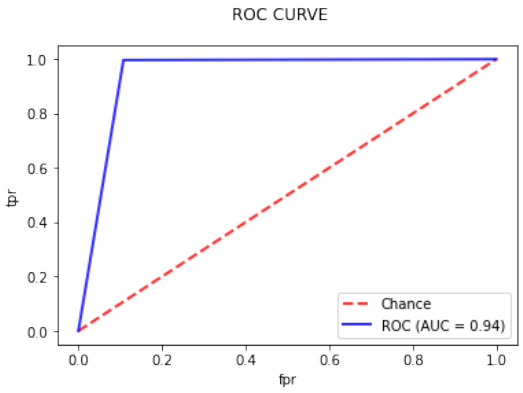

Figura 13. Curva ROC Kernel Rbf

V. Conclusão

O presente artigo teve como fundamento de seu estudo a aplicação do Machine Learning (aprendizado de maquina) para detecção de spam em mensagens de e-mail utilizando máquinas de suporte de vetores, e por meio da separação do

conjunto entre treinamento e teste, a aplicação do conjunto de testes resultou em boas taxas tanto de acurácia quanto de quantidade de verdadeiros positivos e verdadeiros negativos para ambos os modelos, além da métrica AUC que ficou muito próxima de 1 indicando que os modelos são capazes de realizar uma boa separação de classes. Sendo uma taxa de 97% de acurácia com 98% de verdadeiros positivos e 96% de verdadeiros negativos para a Svm com kernel linear, e então para o segundo modelo com uma acurácia de 94% com uma taxa de verdadeiros positivos de 99% e uma taxa de verdadeiros negativos de 89%. Portanto o objetivo proposto pelo artigo foi atigindo com a construção de dois modelos de aprendizado de máquinas supervisionados que é capaz de realizar a classificação de mensagens de e-mail que potencialmente viriam a conter spam.

REFERÊNCIAS

[1] MONARD, M.; BARANAUSKAS, J. Capítulo 4 Conceitos sobre Aprendizado de Máquina. [s.l: s.n.]. Disponível em: https://dcm.ffclrp.usp.br/ augusto/publications/2003-sistemas-inteligentes-cap4.pdf Acesso em: 28 nov. 2022.

[2] EMAILS sent per day 2025 — Statista. Disponível em: https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide/#:˜ text=In%202021,%20there%20were%20an,daily%20e-mails%20by%202025. Acesso em: 11 dez. 2022.

[3] FUSSIGER, Francis. Infográfico - Evolução do email. Disponível em: https://www.dinamize.com.br/blog/infografico-evolucao-email/#:˜text=O%20primeiro%20email%20surgiu%20em,troca%20de%20mensagens%20de%20texto.. Acesso em: 11 dez. 2022.

[4] DIXON, S. Global average daily spam volume 2021 — Statista. 28 abr. 2022. Disponível em:https://www.statista.com/statistics/1270424/daily-spam-volume-global/. Acesso em: 11 dez. 2022.

[5] CARLOS, A.; SANTOS. Máquinas de suporte vetorial e sua aplicação na detecção de spam. [s.l: s.n.]. Disponível em: https://bityli.com/zMBcs. Acesso em: 11 dez. 2022.

[6] WU, D.; VAPNIK, V. Support Vector Machines for Spam Categorization. IEEE TRANSACTIONS ON NEURAL NETWORKS, v. 10, n. 5, 1999. Disponível em: https://www.site.uottawa.ca/ nat/Courses/NLP-Course/itnn$_1$999$_0$9$_1$048.pdf.$Acessoem$10dez.2022.

[7] RODRIGUES, V. Entenda o que é AUC e ROC nos modelos de Machine Learning. Disponível em: https://medium.com/bio-data-blog/entenda-o-que-%C3%A9-auc-e-roc-nos-modelos-de-machine-learning-8191fb4df772. Acesso em: 09 dez. 2022.

[8] Over-sampling methods Version 0.10.0. Disponível em: https://imbalanced-learn.org/stable/references/over_sampling.html. Acesso em: 08 dez. 2022.

[9] OLIVEIRA, G. Máquina de Vetores Suporte: estudo e análise de parâmetros para otimização de resultado. [s.l: s.n.]. Disponível em: https://www.cin.ufpe.br/ tg/2010-2/gmoj.pdf. Acesso em: 9 dez. 2022.

[10] DISCRETO, U. Cachorros, Gatos e Máquinas de Vetores de Suporte (Sem Código). Disponível em: https://www.youtube.com/watch?v=FAKlbZD5Vugt=816s. Acesso em: 8 dez. 2022.

[11] BAJAJ, P. CRIANDO SVM DE KERNEL LINEAR EM PYTHON. Disponível em: https://acervolima.com/criando-svm-de-kernel-linear-em-python/. Acesso em: 12 dez. 2022.

[12] MLMATH.IO. Math Behind SVM(Kernel Trick). Disponível em: https://ankitnitjsr13.medium.com/math-behind-svm-kernel-trick-5a82aa04ab04. Acesso em: 13 dez. 2022.