ENGENHARIA DE COMPUTAÇÃO

INTELIGÊNCIA ARTIFICIAL

Avaliação Continuada 2 (AC2)

Turma: CP404TIN2

Vinícius Lourenço Claro Cardoso – 180618

Professor: Renato Moraes Silva

Sorocaba/SP
17/11/2021

SUMÁRIO

1. INTRODUÇÃO

A inteligência artificial pode ser usada para resolver diversos tipos de problemas e facilitar a vida dos humanos ser usada para resolver tarefas que exigem capacidade humana ligada a inteligência, como raciocínio, percepção de ambiente e a habilidade para a tomada de decisão.

Nesse trabalho, será explorado três métodos de classificação: Árvore de Decisão, Perceptron e Multi-Layer Perceptron. Cada um desses métodos trabalha de formas distintas, mas são capazes de resolver problemas parecidos, variando o seu grau de acuracidade.

Abaixo, podemos ver como é a estrutura de alguns desses métodos de classificação:

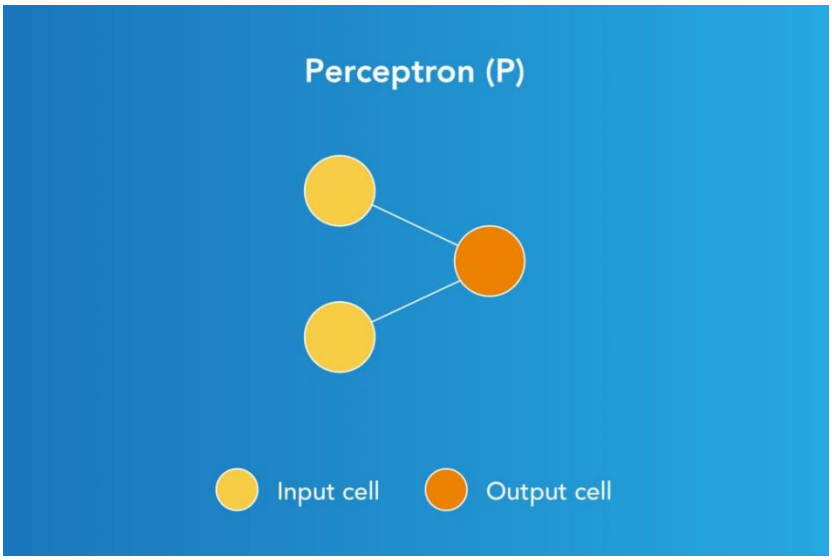

Figura 1 O método de classificação usando Perceptron.

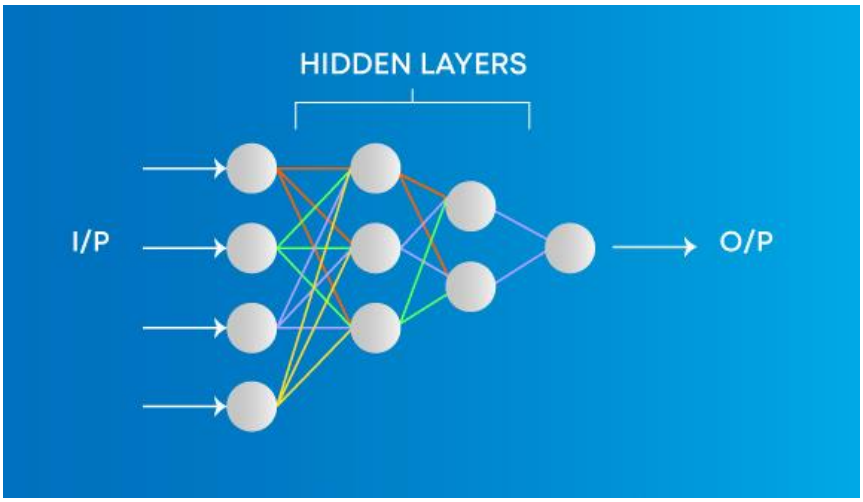

Figura 2 O método de classificação Multi-Layer Perceptron.

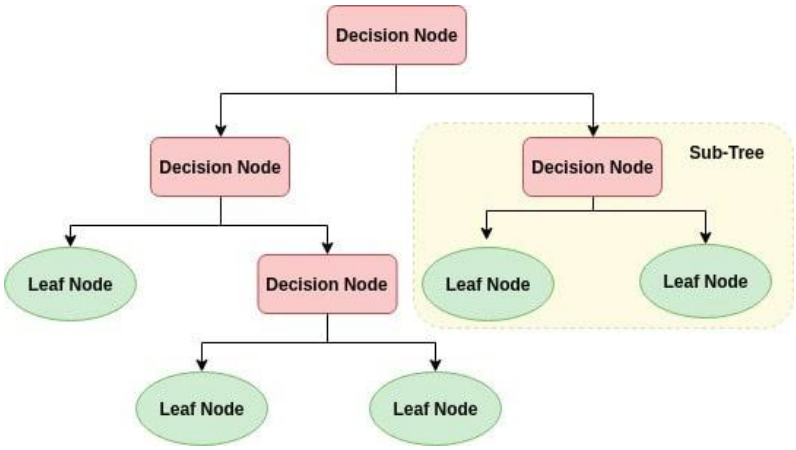

Figura 3 O método de classificação Árvore de Decisão.

Com esses métodos, o problema que será resolvido é o de classificação de mensagens de Spam. Através do Kraggle, uma plataforma onde é possível encontrar diversas bases de dados, foi escolhido uma base chamada “Spam Text Message Classification”, que possui um total de 5157 mensagens únicas em inglês, entre eles, 87% de mensagens reais (ham) e mensagens de ruins (spam).

Dessa forma, o objetivo desse trabalho será criar um modelo que possa predizer, a partir de um texto, se o texto é uma mensagem de spam ou é uma mensagem real.

2. MÉTODOLOGIA

Para começar a realizar o objetivo de classificar o texto para dizer se é spam ou não, antes, é preciso realizar algumas etapas de processamento do texto para torná-lo mais compreensível para a máquina.

Assim, para os três métodos, foi realizado as seguintes etapas de pré-processamento:

- Deixar todas as palavras com letras minúsculas
- Substituir os números pela tag <*NUMBER*>
- Substituir todas as URLS pela tag <*URL*>
- Substituir todos os emails pela tag <*EMAIL*>
- Substituir o símbolo de moeda pela tag <*MONEY*>
- Substituir todos os caracteres não-alfanuméricos por um espaço em branco.
- Remover as entidades do HTML como: “>”, “>”.

Além disso, foi realizado um processo chamado de estemização, no qual diminui uma palavra para os eu radical, tornando uma palavra como “flies” para “fli”.

Ademais, também foi aplicado um outro processo chamado de remoção de stopwords, que remove palavras muito comuns de uma língua. Essa etapa em especial, foi usada uma biblioteca chamada nltk que contém as stopwords de diversas línguas, e como o trabalho será feito sob um texto em inglês, palavras como: “i”, “me”, “my”, “myself”, “we” e outras foram removidas.

E por fim, foi feito uma análise para remover valores duplicados que, de um total de 5572, foi notado que apenas 5157 são valores únicos. E com a remoção de valores duplicados, foi verificado se existia algum valor faltando que poderia comprometer o treinamento, e dada a qualidade do dataset, não foi encontrado nenhum.

Após o processamento, foi realizado a etapa de Word Embedding, que consiste em transformar cada palavra do nosso dataset em um número, de forma que, seja possível usar esses valores durante o treinamento. Durante sua criação, palavras que apareciam menos de 2 vezes foram removidas e também foi usado uma seed igual a zero com apenas 1 worker para manter uma consistência entre os resultados da criação do Word Embedding.

E por fim, com os valores de treinamento devidamente processados, foi realizado uma divisão 70-30 para separar em treinamento e teste, respectivamente. Essa é uma divisão que visa garantir um bom treinamento que visa evitar problemas com overfitting e underfitting.

Sobre os métodos escolhidos, falando primeiro da Árvore de Decisão, foi escolhido para o parâmetro de critério (criterion) o valor de entropy (entropia), assim

como, para a profundidade máximo, máximo de folhas e mínimo de folhas foram deixados no seu estado padrão, None (nenhum).

Sobre o Perceptron, foi escolhido um total de 100 interações, assim como, foi setado uma taxa de aprendizado de 0,1 e parâmetros como tol (tolerância), penalty (penalidade) e class_weight (peso de cada classe) foram mantidos no seu valor padrão, None (nenhum).

E por fim, o Multi-Layer Perceptron, foi escolhido uma organização com 2 camadas, nas quais possuem, respectivamente, 5 e 2 neurônios. Além disso, o número de interações máximo foi setado para 2000, a tolerância para 10^{-6}, o algoritmo de ativação para as camadas escondidas foi selecionado o relu (função linear retificada), e para o algoritmo de optimização dos pesos foi escolhido o lgfgs (um otimizador da família de métodos quasi-newton) que tem uma convergência melhor e performa melhor em datasets menores.

E é interessante notar também, que para o último método, foi realizado uma etapa a mais para processar, tantos os valores quantos as classes, que consistiu em usar LabelEncoder e StandardScaler da biblioteca Sklearn, que foram responsáveis em transformar as classes de texto para um número e escalar os valores de treino e teste para uma escala próxima de zero.

3. RESULTADOS

A abaixo, será exibido uma tabela que contém a acurácia, a precisão e a f-medida de cada um dos métodos usados:

Medidas	Árvore de Decisão	Perceptron	Multi-Layer Perceptron
Acurácia	0,93	0,92	0,96
Precisão (Ham)	0,96	0,93	0,98
Precisão (Spam)	0,72	0,77	0,81
F-Medida (Ham)	0,96	0,95	0,98
F-Medida (Spam)	0,73	0,60	0,84

Com a tabela acima, podemos tirar algumas conclusões, a primeira é que de todas as acurácias, a melhor foi a do método Multi-Layer Perceptron em 96%, com Árvore de Decisão e Perceptron com 93% e 92%, respectivamente.

Além da medida de acurácia, é importante analisar também a Precisão, que assim como a acurácia, o Multi-Layer teve o melhor valor, 98% em Ham e 81% em Spam. Isso indica que ele praticamente está classificando mensagens verdadeiras como verdadeiras, e errando certa de 19% das mensagens de Spam e marcando elas como verdadeira.

De certa forma, apesar do nosso modelo classificar algumas mensagens de Spam como verdadeiras, o mais importante, nesse cenário, é não classificar mensagens verdadeiras como Spam porque elas seriam mais difíceis de corrigir. E com os dados, vemos que a porcentagem de acerto de mensagens verdadeiras é muito alta, o que é um bom resultado.

E sobre a F-Medida, o que mais se diferencia é com relação a mensagens de Spam, a Multi-Layer Perceptron teve uma acurácia de 3%~4% maior que os outros métodos, contudo, sua F-Medida foi superior de 11%-24%, o que é uma ótima melhoria que nos indica que ele está classificando bem não só as mensagens verdadeiras como as mensagens de spam.

E por fim, foi realizado um experimento usando validação cruzada para identificar o quanto afeta o nosso modelo a distribuição dos dados. A validação cruzada consiste em particionar os dados, e ir realizando a predição e treinamento com algumas das partições e treinando com outras, de forma, a treinar toda o modelo com uma mistura muito uniforme de dados.

Para essa validação cruzada, foi usado o método de ShuffleSplit em conjunto com a função cross_val_score, com 5 partições e uma divisão de 70-30 para dados de treino e teste, e os dados obtidos são os seguintes:

Medidas	Árvore de Decisão	Perceptron	Multi-Layer Perceptron
Acurácia	94% ± 0%	91% ± 2%	96% ± 0%

Com essas acurácias, vemos que a acurácia do método de Árvore de Decisão deu uma ligeira melhorada, de 93% para 94%, a Perceptron piorou em 1% mas com um desvio de 2% e o Multi-Layer Perceptron se manteve o mesmo.

Portanto, podemos ver que no nosso caso, acredito que pela quantidade suficiente de amostras e uma divisão de 70-30, foi possível mitigar o problema da distribuição de dados.

Para um futuro trabalho, seria possível se aprofundar melhor ao utilizar o LabelEncoder e StandardScaler não só para o Multi-Layer Perceptron mas também para os outros métodos. Além disso, para o Perceptron, aumentar também a quantidade de interações e para o Multi-Layer Perceptron aumentar o número de camadas e neuronios para testar a possibilidade de uma melhoria devido a um tempo maior de treinamento e uma abstração melhor dos neurônios e camadas.

Todo o código desse projeto, assim como os notebooks, podem ser encontrados no link: https://github.com/H4ad/h4ad.facens.artificial-inteligence.text-classification/.

REFERÊNCIAS

TEXT Preprocessing in Python: Steps, Tools, and Examples. [S. l.], 17 nov. 2021. Disponível em: https://medium.com/@datamonsters/text-preprocessing-in-python-steps-tools-and-examples-bf025f872908. Acesso em: 17 nov. 2021.

PRECISÃO e revocação. [*S. l.*], 17 nov. 2021. Disponível em: https://pt.wikipedia.org/wiki/Precis%C3%A3o_e_revoca%C3%A7%C3%A3o. Acesso em: 17 nov. 2021.

O QUE é Inteligência artificial? Como funciona, exemplos e aplicações. [S. l.], 17 nov. 2021. Disponível em: https://www.totvs.com/blog/inovacoes/o-que-e-inteligencia-artificial/. Acesso em: 17 nov. 2021.

HTML Entities - HTML symbols - HTML characters. [*S. l.*], 17 nov. 2021. Disponível em: https://www.devmedia.com.br/html-entities-html-symbols-html-characters/1011. Acesso em: 17 nov. 2021.

IPYNB import another ipynb file. [*S. l.*], 17 nov. 2021. Disponível em: https://stackoverflow.com/questions/20186344/ipynb-import-another-ipynb-file. Acesso em: 17 nov. 2021.

SPAM Text Message Classification. [*S. l.*], 17 nov. 2021. Disponível em: https://www.kaggle.com/team-ai/spam-text-message-classification. Acesso em: 17 nov. 2021.

PYTHON Machine Learning - Part 1 : Scikit-Learn Perceptron | packtpub.com. [*S. l.*], 17 nov. 2021. Disponível em: https://www.youtube.com/watch?v=oLane_Vh3CU. Acesso em: 17 nov. 2021.

HOW can I import one Jupyter notebook into another. [*S. l.*], 17 nov. 2021. Disponível em: https://stackoverflow.com/questions/50382248/how-can-i-import-one-jupyter-notebook-into-another. Acesso em: 17 nov. 2021.

SKLEARN.LINEAR_MODEL.PERCEPTRON. [*S. l.*], 17 nov. 2021. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Perceptron.html. Acesso em: 17 nov. 2021.

INTRODUÇÃO ao Scikit-learn - Parte 3: avaliando a qualidade do modelo via cross-validation. [*S. l.*], 17 nov. 2021. Disponível em: http://computacaointeligente.com.br/outros/intro-sklearn-part-3/. Acesso em: 17 nov. 2021.

AVALIAÇÃO de modelos, cross-validation e data leakage. [*S. l.*], 17 nov. 2021. Disponível em: http://computacaointeligente.com.br/conceitos/avaliando-performance-cross-validation/. Acesso em: 17 nov. 2021.

SKLEARN.NEURAL_NETWORK.MLPCLASSIFIER. [*S. l.*], 17 nov. 2021. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#. Acesso em: 17 nov. 2021.

DEEP Neural Multilayer Perceptron (MLP) with Scikit-learn. [*S. l.*], 17 nov. 2021. Disponível em: https://towardsdatascience.com/deep-neural-multilayer-perceptron-mlp-with-scikit-learn-2698e77155e. Acesso em: 17 nov. 2021.

NUMPY.CONCATENATE. [*S. l.*], 17 nov. 2021. Disponível em: https://numpy.org/doc/stable/reference/generated/numpy.concatenate.html. Acesso em: 17 nov. 2021.

SKLEARN.NEURAL_NETWORK.MLPCLASSIFIER. [S. l.], 17 nov. 2021. Disponível em: https://docs.w3cub.com/scikit_learn/modules/generated/sklearn.neural_network.mlpclassifier#sklearn.neural_network.MLPClassifier.fit. Acesso em: 17 nov. 2021.

QUASI-NEWTON method. [*S. l.*], 17 nov. 2021. Disponível em: https://en.wikipedia.org/wiki/Quasi-Newton_method. Acesso em: 17 nov. 2021.

A GENTLE Introduction to the Rectified Linear Unit (ReLU). [*S. l.*], 17 nov. 2021. Disponível em: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/. Acesso em: 17 nov. 2021.

SKLEARN.PREPROCESSING.ORDINALENCODER. [*S. l.*], 17 nov. 2021. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OrdinalEncoder.html. Acesso em: 17 nov. 2021.

3 Ways to Encode Categorical Variables for Deep Learning. [*S. l.*], 17 nov. 2021. Disponível em: https://machinelearningmastery.com/how-to-prepare-categorical-data-for-deep-learning-in-python/. Acesso em: 17 nov. 2021.