Projeto NLP - Classificação de Vagas

Bruno Rodrigues Silva
Repositório Github do Projeto

June 2021

1 Objetivos

A motivação deste projeto é como um possível auxílio para construção de equipes de dados por funcionários de RH ou por pessoas que não entendam exatamente o papel de cada componente dentro de uma equipe de dados. O usuário coloca como entrada do modelo a descrição do funcionário que ele está procurando e o modelo retorna qual título este funcionário mais se adéqua em uma equipe de dados, classificando a vaga entre Cientista de Dados, Analista de Dados ou Engenheiro de Dados.

2 Métodos usados

O projeto foi desenvolvido em três etapas ordenadamente:

1. Data Scraping de vagas no Linkedin para a criação do dataset de descrições com marcação de título da vaga usando Selenium;

2. Criação de um modelo de Redes Neurais para classificação multi-classes usando Tensorflow Keras;

3. Criação de um Webapp para a interação mais fluida com o modelo usando Streamlit.

3 Corpus usado

O Dataset foi criado utilizando Selenium a partir do script de scraping disponível no arquivo *scraper/scraper.py*.

O processo consistiu em pesquisar por vagas com descrição em inglês de Cientista de Dados, Analista de Dados e Engenheiro de Dados nos Estados Unidos (mil de cada título, um título por vez), e buscar a descrição de cada vaga conforme fornecido pela empresa contratante.

O Dataset será disponibilizado somente para o avaliador do projeto, uma vez que não foram solicitadas as autorizações do Linkedin para a realização

do scraping e tampouco das empresas envolvidas, já que as mesmas não foram anonimizadas no processo.

4 Modelo

O modelo foi criado utilizando Tensorflow Keras e os métodos de processamento abordados durante o curso e principalmente nas aulas de classificação multi-classes utilizando redes neurais.

Para o tuning de hiper parâmetros foi desenvolvido um método personalizado de GridSearch para testar diferentes arquiteturas de redes, funções de ativação, batch_size, funções de custo e épocas de treinamento e para a criação de um relatório interativo em HTML buscando a obtenção de um modelo com a melhor arquitetura em tempo hábil. O arquivo com todos os testes realizados (mais de 400MB) pode ser solicitado caso necessário, mas um exemplo pequeno deste relatório está disponível no arquivo *tuning-process-report.html* e o melhor modelo encontrado por este processo estará evidenciado na próxima seção do relatório.

5 Resultados e Discussão

5.1 Tuning de hiper parâmetros

Conforme mencionado na sessão 4, foi criado um método personalizado para o tuning de hiperparâmetros do modelo que gera um relatório interativo do processo completo, após testes de centenas de combinações de hiperparâmetros e análises do relatório, a melhor arquitetura encontrada está esquematizada na Figura 1 e seu recorte no relatório na Figura 2 e teve seu desempenho com os seguintes hiper parâmetros:

- 8 camadas ocultas, com 8 neurônios cada;
- função de ativação tanh;
- batch size=64;
- 20 épocas de treinamento;
- função de custo SGD.

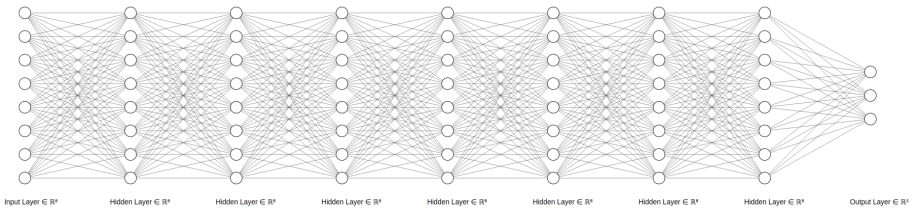

Figure 1: Diagrama do modelo escolhido, exatidão de 86.85% na base de teste.

Hiper parâmetros do modelo

units: [8, 8, 8, 8, 8, 8, 8, 8]
activation: tanh
batch: 64
optimizer: SGD
Acurácia na avaliação: 0.8684834241867065
Perda: 0.41729816794395447

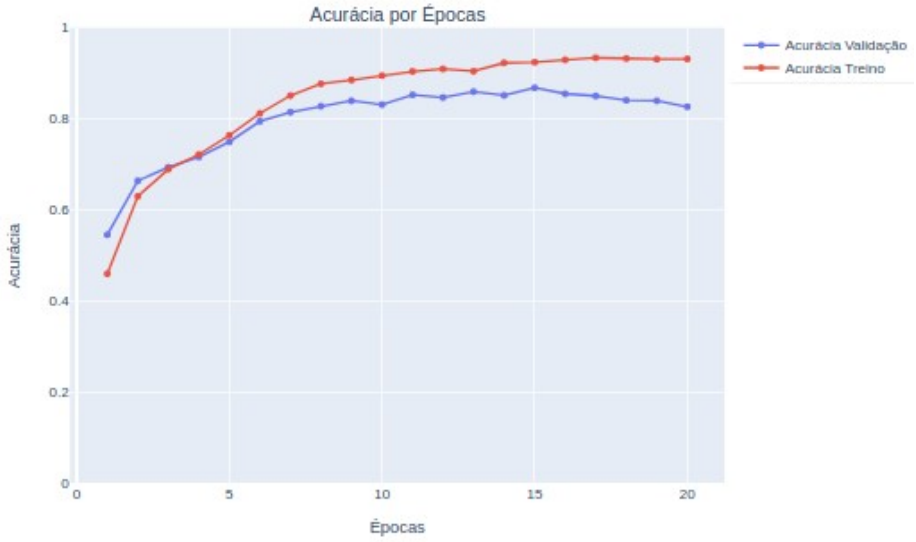

Figure 2: Recorte do relatório de tuning com o modelo vencedor.

5.2 Interação do usuário com o Modelo

Pensando no usuário final, também foi construído um WebApp com a biblioteca Streamlit onde o usuário pode escolher entre exemplos simples, exemplos tirados do Linkedin do autor (que não estão na base de treino nem de teste) ou um input

manual qualquer de interesse pessoal. O tutorial de utilização desta ferramenta de análise está no repositório do modelo, disponível no Github

O WebApp é extremamente simples e serve como prova de conceito, basta escolher entre as opções e o dashboard criará um gráfico de barras mostrando a probabilidade estimada pelo modelo entre os três títulos possíveis conforme demonstrado na Figura 3. Em testes manuais, não foi encontrada nenhuma descrição de vaga vinda do Linkedin do autor que o modelo classificou de maneira incorreta em relação ao título da vaga.

6 Conclusão

6.1 O corpus

Inicialmente o processo de scraping seria feito com vagas de Cientistas de Dados, Analistas de Dados e Engenheiros de Dados do Brasil, mas existiu muita dificuldade para a criação desse corpus por dois pontos principais:

1. Poucas vagas com uma descrição de qualidade para Analistas de dados. No Brasil - atualmente - o título de analista de dados é extremamente novo e ambíguo, isso impossibilita a criação de um corpus em português de qualidade para este público;

2. Diversas vagas no Linkedin de empresas brasileiras estão descritas em inglês, o que forçaria o uso de uma ferramenta de classificação de idioma antes de aceitação ao corpus;

Adição da classe Engenheiro de Machine Learning, que está sendo cada vez mais adotada por times de dados. Porém no momento não está com um volume suficiente para criação de um padrão bem estabelecido de perfil, variando muito entre empresas.

Além destes pontos, também seria interessante incluir uma classe externa às mencionadas sendo usada como nulo, visto que o modelo atual é forçado a escolher entre as três classes disponíveis sem a opção de não escolher nenhuma classe, o que pode ferir o desempenho.

6.2 Explicabilidade do modelo

Apesar do modelo lograr um resultado relativamente bom no corpus, essencialmente, o modelo está sendo tratado como uma caixa preta. De maneira geral, existem diversos métodos, como SHAP ou LIME, que poderiam ser utilizados para traçar uma visão mais precisa dos fatores que fazem o modelo conseguir uma separação entre as classes mencionadas. [1] [2]

6.3 Data Analyst vs Data Scientist

É interessante verificar que em vagas para Analistas de Dados o modelo não obteve uma boa separação com as vagas de Cientistas de Dados, conforme evidenciado na matriz de confusão da Figura 3, onde é observado que 19.26%

das vagas de Analistas são classificadas como Cientistas de Dados pelo modelo. Esta separação é algo complexo até para humanos e muitas vezes há uma sobreposição de papéis no setor com diferentes opiniões sobre como definir o título de cada profissional, como mencionado por Warsame em 2019 "But surprisingly, we've also come to appreciate how similar the two kindred professions actually are. In essence, they both seek to retrieve insights from datasets.". [3]

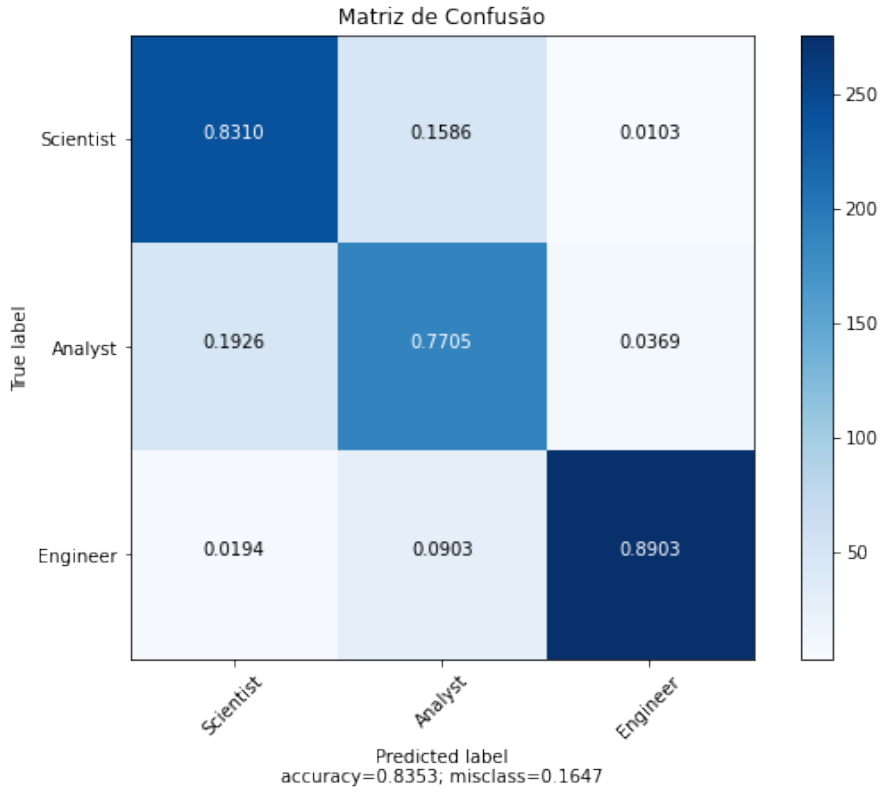

Figure 3: Matriz de confusão normalizada.

6.4 Possíveis caminhos para a modelagem

6.4.1 Mantendo a abordagem atual

Existem infinitos testes com diferentes hiper parâmetros que poderiam ser realizados para a criação de um modelo com melhor desempenho, sendo que a abordagem de utilização de uma rede neural somente com camadas totalmente conectadas obteve resultados satisfatórios para a profundidade desejada deste projeto. Porém esta abordagem está fadada a um limite do método utilizado,

portanto é possível despender um recurso alto de tempo na abordagem mencionada, porém o ganho seria marginal.

6.4.2 Redes neurais com atenção

Para uma próxima estratégia de modelagem, uma das ideias mais interessantes seria utilizar modelos recorrentes baseados em mecanismos de atenção - como redes LSTM Bi-Direcionais com atenção -, uma vez que já se mostraram superiores a modelos somente com camadas totalmente conectadas em estudos de classificação de textos (Huang, 2018 e Sun, 2020). [4] [5]

References

[1] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[2] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016.

[3] Mohamed A. Warsame. Data analyst vs. data scientist - a comparative analysis of the roles and responsibilities, points of overlap and expectations of the 'sexiest job of the 21st century'. 2019.

[4] Changshun Du and Lei Huang. Text classification research with attention-based recurrent neural networks. *International Journal of Computers Communications Control*, 13:50, 02 2018.

[5] Xiaobing Sun and Wei Lu. Understanding attention for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3418–3428, Online, July 2020. Association for Computational Linguistics.