Relatório: Processo de Desenvolvimento de Modelo de Classificação de Texto

Patrick Canto de Carvalho

1

1. Introdução

Este relatório visa apresentar o processo de desenvolvimento de um modelo de aprendizado de máquina capaz de classificar textos de acordo com o nível de dificuldade. As categorias utilizadas são: ensino fundamental I e II, ensino médio e ensino superior.

2. Metodologia

2.1. Ferramentas

- Linguagem de programação Python;
- Biblioteca Scikit-Learn para criação dos modelos de aprendizagem de máquina;
- Jupyter notebooks para documentar processo de desenvolvimento;
- Framework Flask para criação da aplicação web;
- Biblioteca Spacy para análise de texto.

2.2. Separação e tratamento dos dados

A partir dos arquivos presentes no dataset, foi criado um único arquivo csv com os dados categorizados. Foram acrescentados a esses dados outras estatísticas obtidas a partir da biblioteca spacy que podem vir a ser úteis.

O conjunto de dados foi dividido entre treino e teste, sendo 80% para treino e 20% para teste. O conjunto de dados se encontra desbalanceado, sendo assim, foi usada a opção class_weights='balanced' nos modelos, quando aplicável.

Para codificação dos textos, foi utilizada a abordagem *bag of words*. Este método representa o texto como um vetor de frequências de palavras, considerando todo o vocabulário do corpus.

2.3. Experimentos e análise dos resultados

Foram selecionados para uma investigação inicial as seguintes técnincas de classificação discutidas por [Kowsari et al. 2019] e [Sen et al. 2020]:

- Random Forest Classifier (SVM);
- Stochastic Gradient Descent (SGD);
- Decision Tree Classifier;
- K Nearest Neighbors Classifier.

Para cada técnica, foram realizadas 30 execuções com sementes de randomização diferentes e foram computadas as métricas de acurácia, precisão, recall e f1-score, bem como as matrizes de confusão médias normalizadas. O resumo dos resultados é apresentado na 1 e o gráfico comparativo detalhado é apresentado na figura 1.

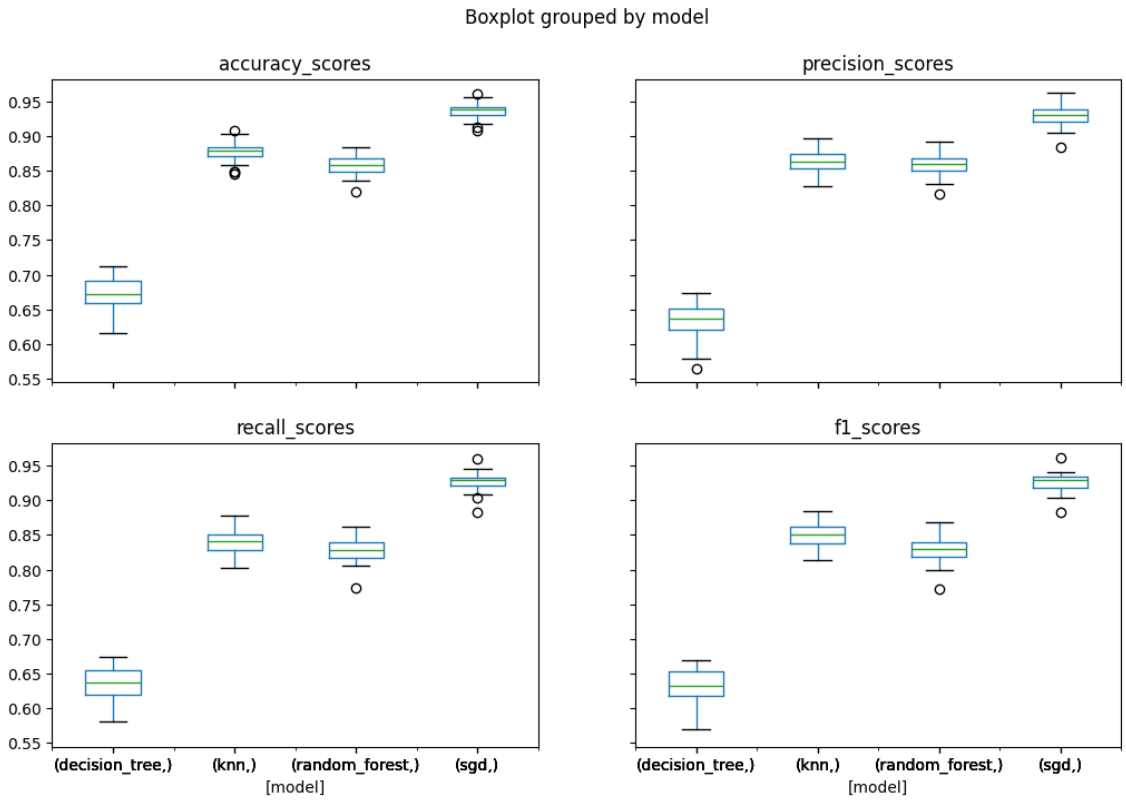

Figure 1. Comparativo entre os algoritmos Decision Tree, KNN, Random Forest e SGD segundo as métricas de acurácia, precisão, *recall* e *f1-score*

Algoritmo	Acurácia	Precisão	Recall	F1-score
Random Forest	0.857612	0.859203	0.829544	0.830805
SGD	**0.937260**	**0.928300**	**0.926063**	**0.925859**
Decision Tree	0.672676	0.633877	0.636016	0.633681
KNN	0.877244	0.864790	0.841041	0.850206

Table 1. Comparação entre os valores médios das métricas: acurácia, precisão, recall e f1-score para as técnicas estudadas.

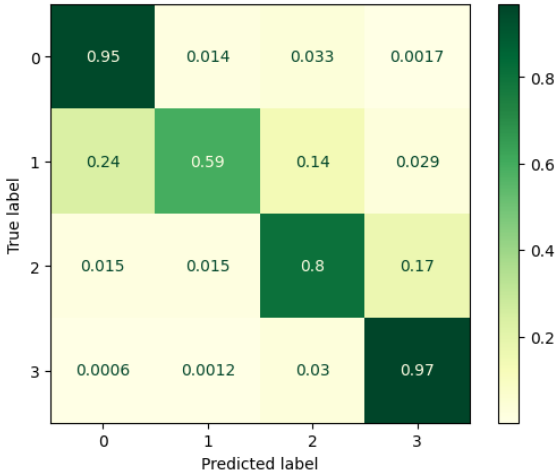

Figure 2. Matriz de confusão para o algoritmo Random Forest.

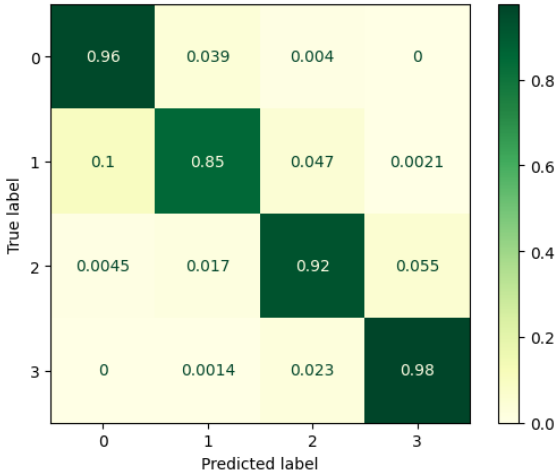

Figure 3. Matriz de confusão para o algoritmo SGD.

E, para visualizar o comportamento dos algoritmos de forma mais detalhada, foram geradas as matrizes de confusão médias normalizadas para cada técnica, apresentadas nas figuras 2, 3, 4 e 5.

É possível perceber que, de acordo com essas métricas, o modelo que apresentou melhor desempenho foi o SGD. Sendo assim, para melhor visualizar as categorias que apresentaram maior dificuldade de classificação, é apresentada na figura 6 a matriz de confusão do SGD com a diagonal principal zerada. De acordo com o gráfico, o principal erro é classificar um texto do ensino fundamental II como sendo do ensino fundamental I. No entanto, nota-se que é uma tendência do geral modelo errar para categorias adjacentes.

Tendo sido identificada a melhor dentre as técnicas estudadas, foi selecionado desta técnica o melhor modelo produzido durante o experimento (acurácia média de 0.96) para que pudesse ser feito um refinamento em seus hiper-parâmetros utilizando a função GridSearchCV da biblioteca Scikit-Learn. No entanto, não foi notada melhoria. A matriz de confusão do melhor modelo treinado é mostrada na figura 7.

Este modelo foi salvo para ser usado na aplicação web, juntamente com os as classes auxiliares necessárias para codificar o texto. Foi criado também um gerador de inversões de frases. O sistema é acessível pelo navegador e as instruções para executar estão no arquivo readme.md. Mais detalhes sobre o processo de desenvolvimento podem ser encontrados no arquivo text-complexity.ipynb.

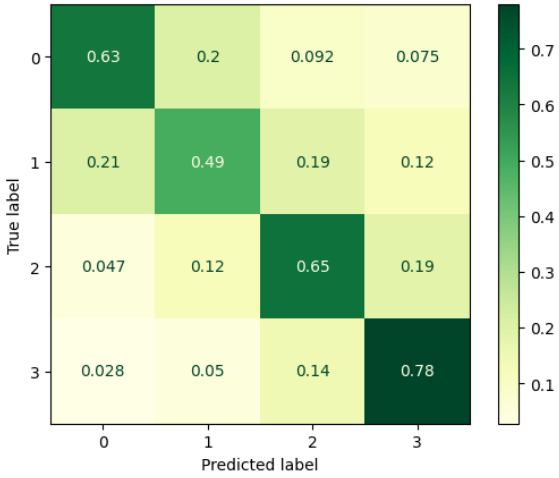

Figure 4. Matriz de confusão para o algoritmo Decision Tree.

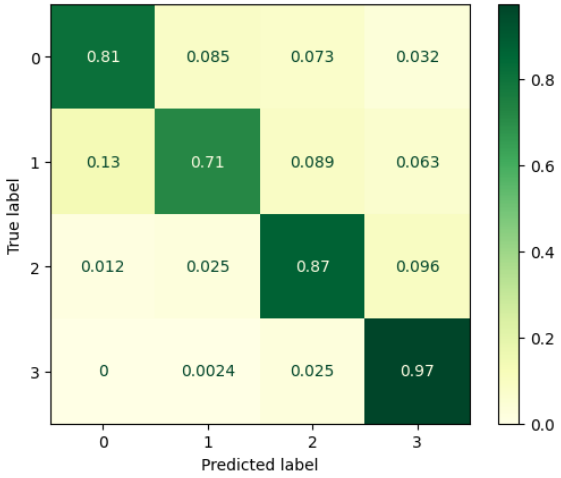

Figure 5. Matriz de confusão para o algoritmo KNN.

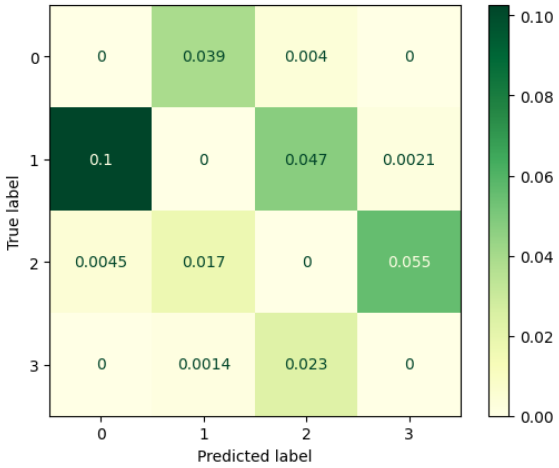

Figure 6. Matriz de confusão para o algoritmo SGD com a diagonal principal zerada.

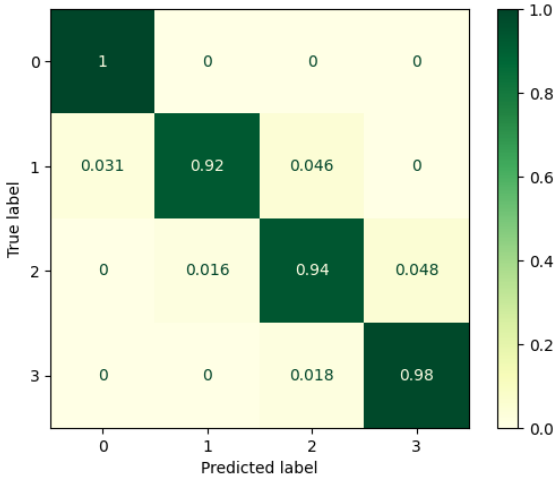

Figure 7. Matriz de confusão o melhor modelo treinado (SGD).

3. Gerador de inversões em frases

Foi criada a estrutura para a implementação do inversor de frases. Contudo, não foi possível implementá-lo por não ter tido tempo hábil para entender o funcionamento das bibliotecas disponíveis para este fim. De qualquer forma, é possível assumir que a classificação das frases geradas se manteria igual à da frase original, tendo em vista que a técnica de codificação utilizada considera apenas o vocabulário presente no texto.

4. Conclusão

A melhor técnica identificada foi a Stochastic Gradient Descent (SGD), obtendo o melhor desempenho em todas as métricas utilizadas. O melhor modelo obteve acurácia de aproximadamente 0.96. No entanto, este realiza classificações equivocadas, especialmente entre classes consideradas adjacentes. Devido à limitação no tempo, não foi possível implementar algumas etapas que poderiam tornar mais robusto o sistema desenvolvido. Sendo assim, podem ser listadas algumas melhorias a serem feitas futuramente:

- Considerar mais características do texto, tendo em vista que o método de codificação utilizado analisa apenas a frequência de palavras para caracterizar o texto, desconsiderando a estrutura sintática e semântica.
- É possível analisar de maneira mais aprofundada os textos do conjunto de dados fornecido, de modo a possibilitar um tratamento melhor dos dados antes do treinamento do modelo, incluindo a remoção de outliers.

References

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., and Brown, D. (2019). Text classification algorithms: A survey. *Information*, 10(4):150.

Sen, P. C., Hajra, M., and Ghosh, M. (2020). Supervised classification algorithms in machine learning: A survey and review. In *Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018*, pages 99–111. Springer.