Pontifícia Universidade Católica de Minas Gerais

Inteligência Artificial
Professora Cristiane Neri Nobre

Curso: Ciência da Computação, 4º período, turno manhã

Cecília Capurucho Bouchardet
Danielle Dias Vieira
Felipe Vilas Boas Marprates
João Augusto dos Santos Silva
Thiago de Campos Ribeiro Nolasco

Trabalho prático - Etapa 3

Belo Horizonte
2022

Etapa 3

Para a etapa 3 do Trabalho Prático, nos foi pedido para implementar mais um algoritmo de aprendizado de máquina para a nossa base de dados. Como estamos trabalhando com a classificação de frases, escolhemos o *Random Forest* para realizar essa classificação, tendo em vista que já utilizamos os algoritmos RoBERTa e *Naive Bayes Multinomial* anteriormente, e agora estamos comparando os 3 algoritmos para verificar qual é o mais viável para o nosso caso de uso.

Primeiramente executamos o algoritmo com a base desbalanceada, o que verificamos na etapa anterior do Trabalho Prático da disciplina, é que apresenta um impacto significativo nos resultados da execução do aprendizado de máquina. Os resultados com a base desbalanceada não foram muito otimistas, já que a classe majoritária (positivo) apresentou recall igual a 1.0, resultado perfeito, com todas as instâncias corretamente classificadas, o que nos levou a desconfiança dos resultados, tendo em vista que nenhum algoritmo de *Machine Learning* é capaz de alcançar um resultado de 100%. Para as outras classes notamos que a grande maioria das instâncias tendem para a classe majoritária, mostrando a necessidade de realizarmos a etapa de pré-processamento para balanceamento da base de dados.

Após o balanceamento da base de dados e análise dos resultados obtidos, verificamos a melhora das métricas para as classes negativo e neutro, onde conseguimos elevar o número de instâncias que foram corretamente classificadas. O recall, f-score e precision melhoraram e se tornaram mais realistas, nos mostrando que o algoritmo *Random Forest* obteve um resultado superior aos resultados obtidos pelo *Naive Bayes Multinomial* para as classes minoritárias e um pouco inferior para a classe majoritária.

Aprofundando um pouco mais na comparação entre os algoritmos, na média das métricas analisadas, o *Random Forest* balanceado (Figura 8) foram superiores em 0.01 quando comparado com o *Naive Bayes Multinomial* também balanceado (Figura 4) e na matriz de confusão conseguimos também perceber um maior acerto no *Random Forest* balanceado em relação ao *Naive*, o que nos leva a concluir que ambos algoritmos possuem um alto nível de acerto e que são eficientes para o PLN (processamento de linguagem natural).

Resultados obtidos com algoritmo *Naive Bayes Multinomial*

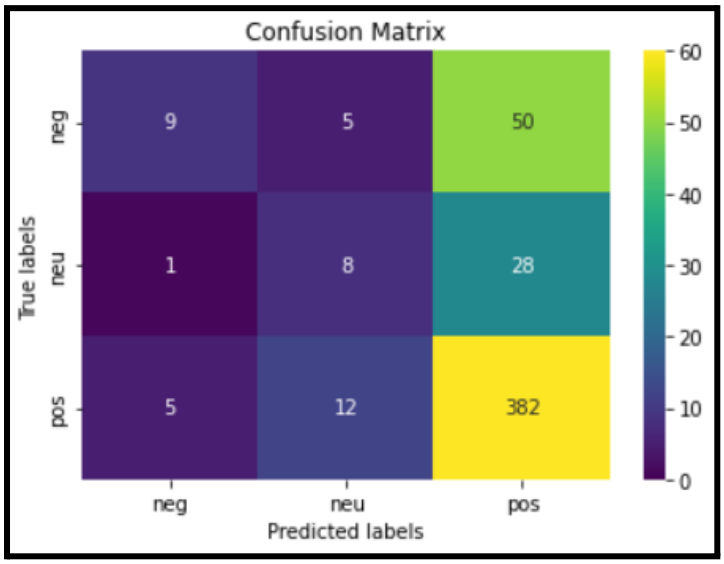

Figura 1: *Matriz de Confusão não-balanceada*

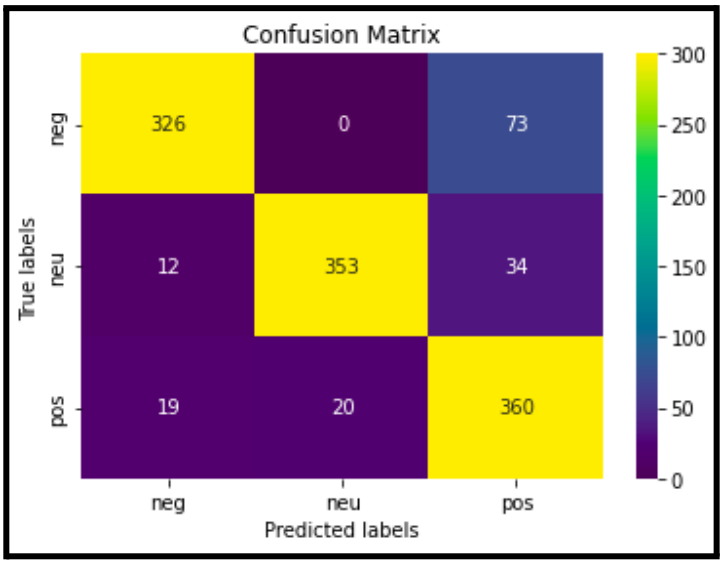

Figura 2: *Matriz de Confusão balanceada*

	precision	recall	f1-score	support
negative	0.60	0.14	0.23	64
neutral	0.32	0.22	0.26	37
positive	0.83	0.96	0.89	399
accuracy			0.80	500
macro avg	0.58	0.44	0.46	500
weighted avg	0.76	0.80	0.76	500

Figura 3: *Métricas geradas conforme matriz de confusão não-balanceada*

	precision	recall	f1-score	support
negative	0.91	0.82	0.86	399
neutral	0.95	0.88	0.91	399
positive	0.77	0.90	0.83	399
accuracy			0.87	1197
macro avg	0.88	0.87	0.87	1197
weighted avg	0.88	0.87	0.87	1197

Figura 4: *Métricas geradas conforme matriz de confusão balanceada*

Resultados obtidos com algoritmo *Random Forest*

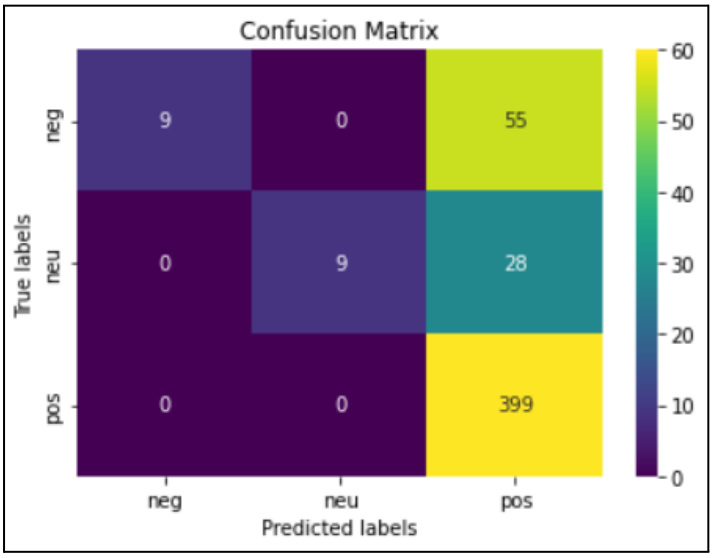

Figura 5: *Matriz de Confusão não-balanceada*

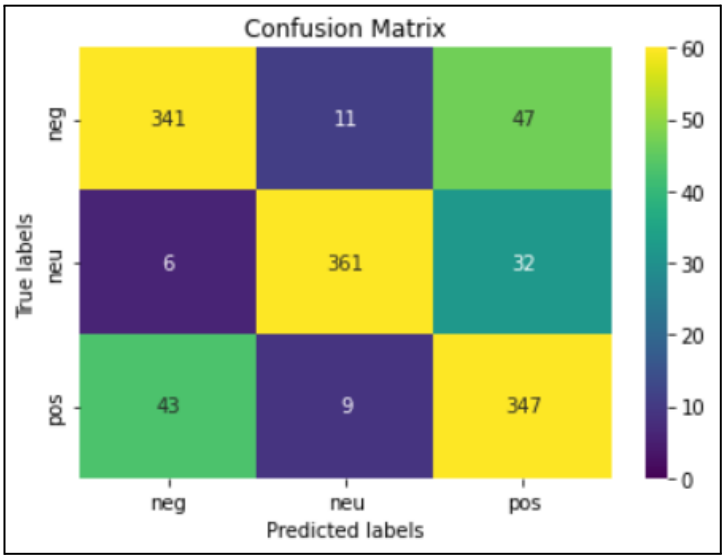

Figura 6: *Matriz de Confusão balanceada*

	precision	recall	f1-score	support
negative	1.00	0.14	0.25	64
neutral	1.00	0.24	0.39	37
positive	0.83	1.00	0.91	399
accuracy			0.83	500
macro avg	0.94	0.46	0.51	500
weighted avg	0.86	0.83	0.78	500

Figura 7: *Métricas geradas conforme matriz de confusão não-balanceada*

	precision	recall	f1-score	support
negative	0.87	0.85	0.86	399
neutral	0.95	0.90	0.93	399
positive	0.81	0.87	0.84	399
accuracy			0.88	1197
macro avg	0.88	0.88	0.88	1197
weighted avg	0.88	0.88	0.88	1197

Figura 8: *Métricas geradas conforme matriz de confusão balanceada*

Links

Base de dados no Kaggle:
https://www.kaggle.com/code/robikscube/sentiment-analysis-python-youtube-tutorial/notebook

Notebook com o código de classificação da base de dados em positivo, negativo e neutro utilizando a técnica RoBERTa:
https://colab.research.google.com/drive/1GmB-1Qdap5Dz4b1ISdgAozFVqAJiDsa0?usp=sharing

Notebook com o código de aprendizado de máquina com o algoritmo *Naive Bayes Multinomial*:
https://colab.research.google.com/drive/1cDcBNP-tW4c1-IXklZQ0DLp9y_yaMPRU?usp=sharing

Notebook com o código de aprendizado de máquina com o algoritmo *Naive Bayes Multinomial para base balanceada*:
https://colab.research.google.com/drive/1FMsau3yPi8lWvMvRJeHoQgmXgjwETJ0K?usp=sharing

Notebook com o código de aprendizado de máquina com o algoritmo *Random Forest*:
https://colab.research.google.com/drive/1UrpTlPmdhYLQyRX9-RKpR6nheqgVPfWA?usp=sharing

Link do GitHub com todos os arquivos utilizados no trabalho:
https://github.com/joaoaugustoss/TrabalhoPratico-IA

Referências

scikit-learn. sklearn.naive_bayes.MultinomialNB. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html. Acesso em 16 de setembro de 2022.

imbalanced-learn.imblearn.over_sampling.RandomOverSampler.Disponível em: https://imbalanced-learn.org/stable/over_sampling.html?highlight=random%20over%20sampler. Acesso em 29 de setembro de 2022.

sklearn.ensemble.RandomForestClassifier. Disponível em: https://scikit-learn.org/stable/modules/ensemble.html. Acesso em 14 de outubro de 2022.