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 2

Belo Horizonte
2022

# Etapa 2

Para a Etapa 2 do Trabalho Prático, nos foi pedido para executar os processos de pré-processamento na base de dados escolhida e após a análise para verificar qual etapa de pré-processamento deveria ser realizada, constatamos que a nossa base não possui nenhum tipo de atributo ausente, mas a base está desbalanceada com as três possíveis classes de classificação, o que está nos gerando problemas com a classificação das classes minoritárias, pois estão com as métricas baixas em relação a classe majoritária.

A fim de mantermos a base com uma boa quantidade de instâncias a serem classificadas, optamos em utilizar o algoritmo *Random Over Sampler* da biblioteca *imbalanced-learn* que gera novas instâncias de forma aleatória, deixando todas as classes da base com a mesma quantidade de instâncias.

A nossa base inicial possui 500 elementos, sendo eles 399 classificados como *positive*, 64 como *negative* e 37 como *neutral*, o que nos gera boas métricas para a classe *positive*, mas métricas ruins para as outras classes devido a grande diferença de tamanho entre elas. Após o balanceamento da base, aumentamos as instâncias classificadas como *negative* e *neutral*, deixando-as com 399 instância cada e com isso a nossa base deixou de ter 500 elementos e passou para 1197.

Para entender a diferença causada pela otimização da base, veja as comparações abaixo:

## Resultados obtidos

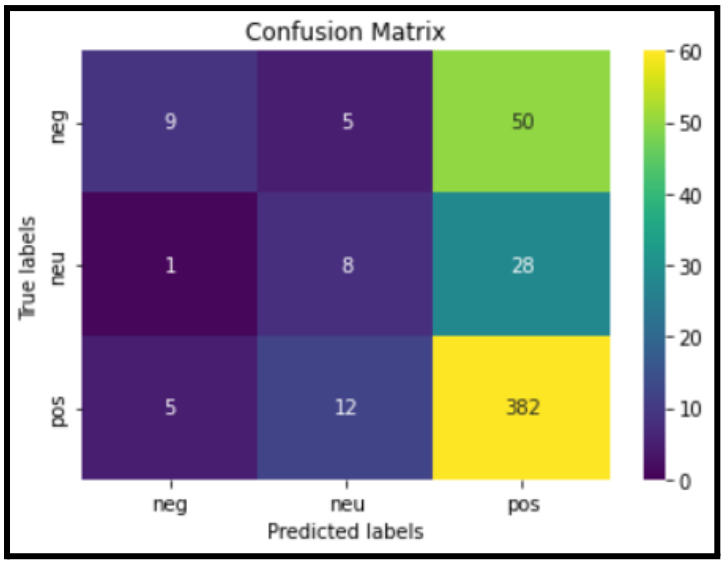


***Figura 1:*** *Matriz de Confusão não-balanceada*

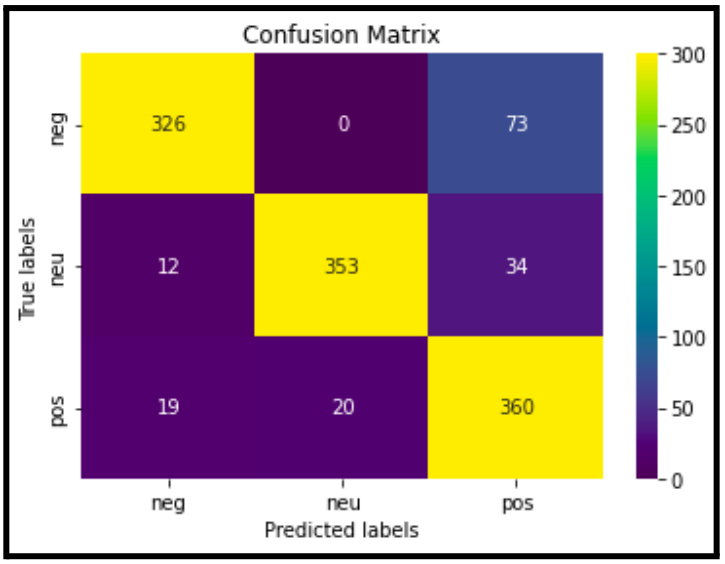


***Figura 2:*** *Matriz de Confusão balanceada*

| | 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*

Com a análise dos resultados obtidos e comparando-os, é facilmente perceptível a diferença causada pelo balanceamento da base de dados para todas as classes de nossa tabela. As métricas das classes negative e neutral apresentaram um

aumento significativo, já que no primeiro teste elas eram as classes minoritárias do modelo e por esse motivo, assim como é visível na matriz de confusão (Figura 1) para as classes *negative* e *neutral* existiram muitos erros de classificação que foram corrigidos e mostrados na segunda matriz de confusão (Figura 2).

# 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#scrollTo=UIKWxOs01US9

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

# 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.