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 1

Belo Horizonte
2022

Base de dados

Escolhemos a base de dados *Sentiment Analysis in Python* (Análise de sentimentos em Python). Nesta base, o autor Rob Mulla mostra como aplicar duas técnicas diferentes para analisar os sentimentos dos comentários de compra de comida na Amazon, que são VADER (*Valence Aware Dictionary and sentiment Reasoner*) e RoBERTa.

O motivo de escolhermos esta base de dados é porque depois queremos avaliar se o comentário do cliente condiz com a nota de avaliação que ele forneceu pelo seu pedido de comida na Amazon.

Separamos as 500 primeiras linhas para aplicar a técnica RoBERTa, que é baseada no modelo BERT, para classificar as frases como positivas, negativas ou neutras e criar uma nova coluna com esta classificação que é o rótulo.

É esta classificação que vamos utilizar como base para avaliar o desempenho do algoritmo de aprendizado *Naive Bayes Multinomial*.

Antes de utilizar o *Naive Bayes* não foi aplicado nenhum pré-processamento nestas 500 linhas. A base de dados está desbalanceada, com 399 instâncias com classificação positiva, 64 como negativa e 37 como neutra. Não há dados ausentes nestas 500 instâncias.

Aplicamos o *Naive Bayes Multinomial* para classificar de novo todas as instâncias, para informar se o comentário é positivo, negativo ou neutro. Nele utilizamos o atributo de entrada Text para classificar, onde primeiro retiramos as *stopwords* e substituímos a coluna Text pela new_Phrase com a frase tratada. Depois todas as frases se transformam em arrays, compondo cada posição com uma palavra. Com os arrays formados é treinado 80% dos dados, ou seja, 400 instâncias. A previsão é feita com toda a base, para compararmos o desempenho do *Naive Bayes Multinomia*l com a classificação feita pela técnica RoBERTa.

É gerado a matriz de confusão conforme os resultados.

Resultados obtidos

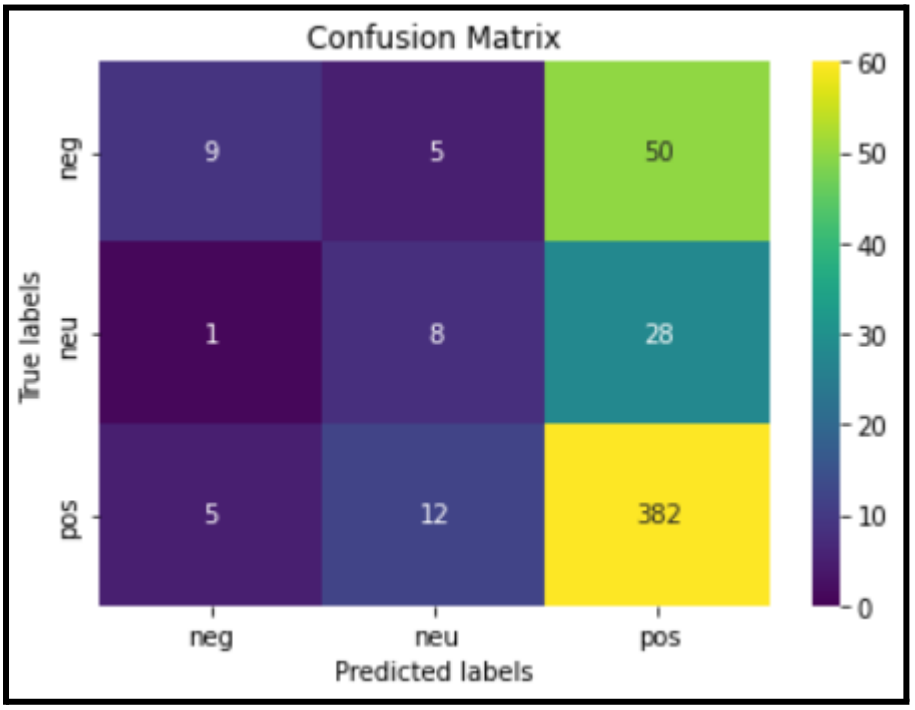

Figura 1: *Matriz de Confusão*

```
VP neg:9        VP neu:8        VP pos:382
FN neg:55       FN neu:29       FN pos:17
FP neg:6        FP neu:17       FP pos:78
VN neg:430      VN neu:446      VN pos:23
Accuracy: 0.798
F-Measure: 0.45843963459145537
Recall: 0.4380782333084965
Precision: 0.5834782608695651

VP: Verdadeiro Positivo
FN: Falso Negativo
FP: Falso Positivo
VN: Verdadeiro Negativo
```

Figura 2: *Métricas geradas conforme matriz de confusão*

Como a base de dados está desbalanceada podemos ver que nas classificações de negativo e neutro o erro é maior, pois a quantidade de instâncias classificadas em RoBERTa nestas duas classes é muito inferior à classificação positiva, que é 399 de 500.

Pelo valor da acurácia percebemos que as classificações feitas pelo *Naive Bayes* estão bem próximas da classificação feita pela RoBERTa.

As métricas *F-Measure, Recall* e *Precision* foram feitas sobre as 3 classificações e poderão ser utilizadas para comparar com o próximo algoritmo de aprendizado que será aplicado na próxima etapa deste trabalho.

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

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.