Summary

1. Import libraries and dataset
2. Delete all columns and keep just the reviews with their labels
   1. Before that, you can do further investigating for data quality such as looking if there are negative review with high classes or vice versa since this suggests that there is something is wrong with these reviews and we need to pay more attention to them
3. Delete the observations missing review (just labels)
4. Drop duplicates observations (same review with the same label)
5. Check the distribution of the class labels (+ve reviews by far more than the -ve reviews 82:17)
6. Create a cleaning function that will be used for WordCloud & ML models. This function removes:
   1. URL links
   2. Tags
   3. Mentions (start with @)
   4. Remove ‘ from the stepwords so words like “don’t” don’t get deleted since we are doing sentiment analysis not classification
   5. Do lowering so all words will be full if lowercase letters
   6. Do tokenization
   7. Remove punctuation & stepwords (stepwords used from a library, I did not create my own)
   8. Lemmatization normalization so each token is return to its root in the dictionary
7. By using this cleaning function we do WordCloud for each class (+ve & -ve) to see the most important for each class. I found that the 1st two frequent tokens are repeated for both such as (dress & top) so I considered them as stepwords
8. I divided the dataset into three sets (training/validation/testing) I also create a set that has both the training & validation together and called it X\_train\_full
9. For ML models: two types of vectorizers has been used: count vectorizer & TF-IDF vectorizer
10. For each vectorizer, 5 algorithms have been used which are: Logistic Regression, Naïve Bayes, Support Vector Machine, Random Forest, Ada boost
11. Two more models have been built, deep learning GRU & LSTM models. For these, Keras tokenizer has been used, with filtering of punctual & numbers along with lowering the letters. However, no normalization has been used. For the deep learning models, fixed number of tokens for each review was needed, so the choosing process was based on that at least 95% of the reviews are with full information without truncations. Also, GRU & LSTM needed embedding size which represents the size of the vector that represent each token 50 was used. We noticed in these two models that the overfit directly which indicates that maybe batch normalization is needed since the dropout is used already.
12. BERT model also was used. For BERT, TPU from Google Colab was used