ML Engineer Assignment – Wysa
Submitted by: Sarthak Srivastava

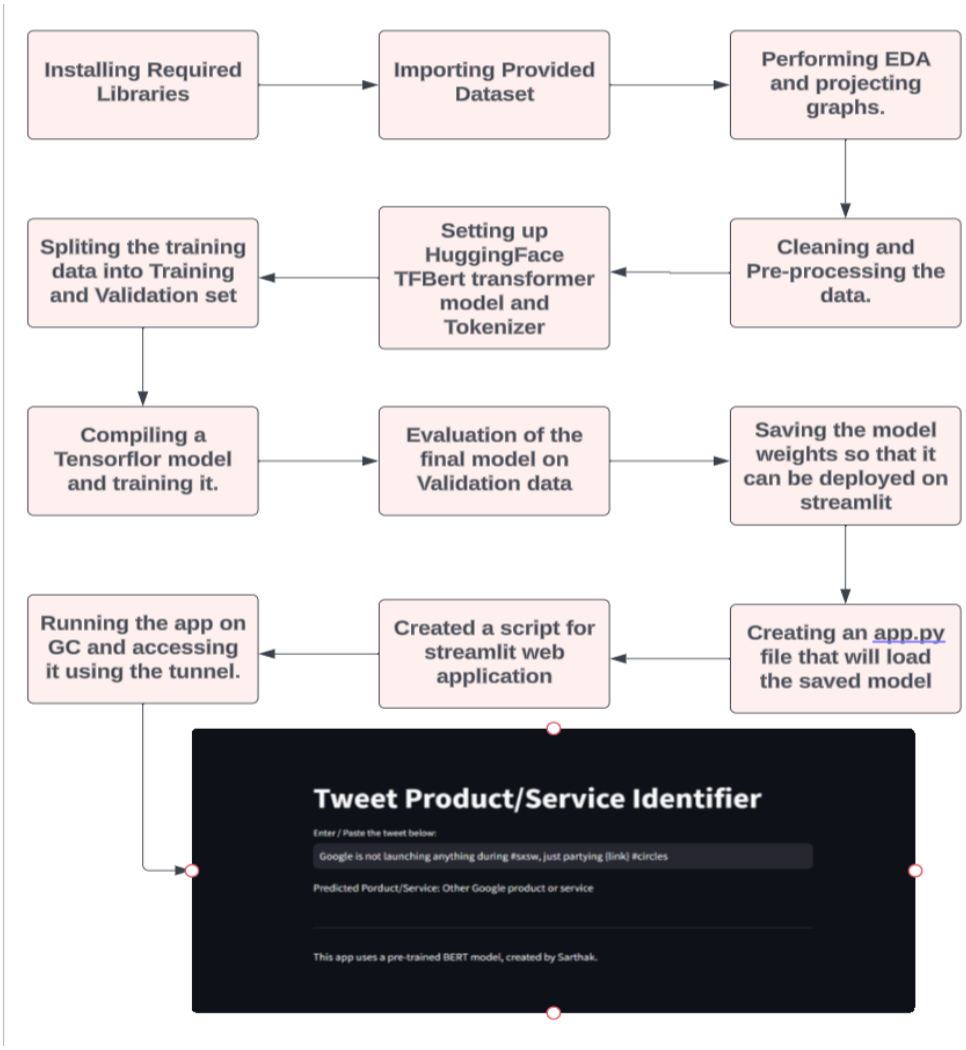
Problem Statement:
What emotion is directed towards which product?

Proposed Solution Outline:

1. I have fine-tuned a pre-trained BERT base (uncased) transformer model sourced from HuggingFace library to predict towards which product a particular tweet is directed at. (example: iPad, Android, Google,etc)
2. I already have a similar project wherein I have used LoRA Adpater for PEFT of a BERT model for sentiment analysis. Link: sarthak-sr/PEFT_on_BERT_using_LoRA (github.com)
3. For this project I have focused on the text classification part. I have delivered the following:

   3.1 **An Open-ended EDA of the data** (In the form of GC notebook) with various graph.
   3.2 A source code for the model setup and training.
   3.3 A .csv file containing result of 'test' data provided alongside the assignment.
   3.4 A streamlit web application that can be deployed on Google collab itself.
   3.5 A demo video for better understanding.

Code Walkthrough:



Evaluation Metrics and Results:

Train Validation Split: 80 – 20

Loss function used on model training: Sparse Categorical Cross Entropy
Optimizer: Adam
LR:0.00001
Evaluation Metrics: Accuracy

Training loss and accuracy:
loss: 0.2608 - accuracy: 0.9487

Validation loss and accuracy:
loss: 0.4569 - accuracy: 0.8741

Model Deployment and Maintenance:

To replicate the model, upload the provided .pynb file into any GC Notebook, change the runtime from CPU to GPU, and run all. At the bottom a link will be displayed which will be a tunnel link for the streamlit application. The password would be the Public IP of the GC Notebook, which would be mentioned in the code block itself.

I have deployed the model in Google Collab itself due to computational boundaries. However the same model can be build into a Docker Image with Streamlit app + saved model baked in with, something like this:

$ mlem build docker --image.name myimage --model mymodel --server streamlit

Monitoring:

Monitoring an ML model deployed as a Streamlit web app is crucial to ensure its performance, reliability, and effectiveness over time. Here are some key parameters to monitor:

1. Prediction Accuracy:
   Track the model's accuracy over time to identify any degradation in performance.

2. Latency:
   Monitor the time taken for the model to make predictions. Sudden increases in latency may indicate performance issues.

3. Throughput:
   Measure the number of predictions the model makes within a given time frame. Ensure that the system can handle the expected load.

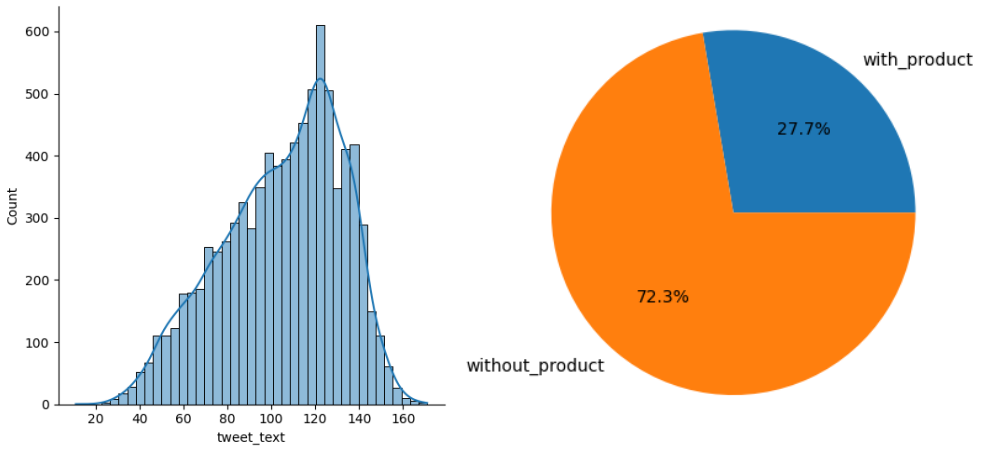
4. Resource Utilization:
   Monitor CPU, memory, and GPU usage to identify potential resource bottlenecks. This is crucial for maintaining a responsive and efficient application.

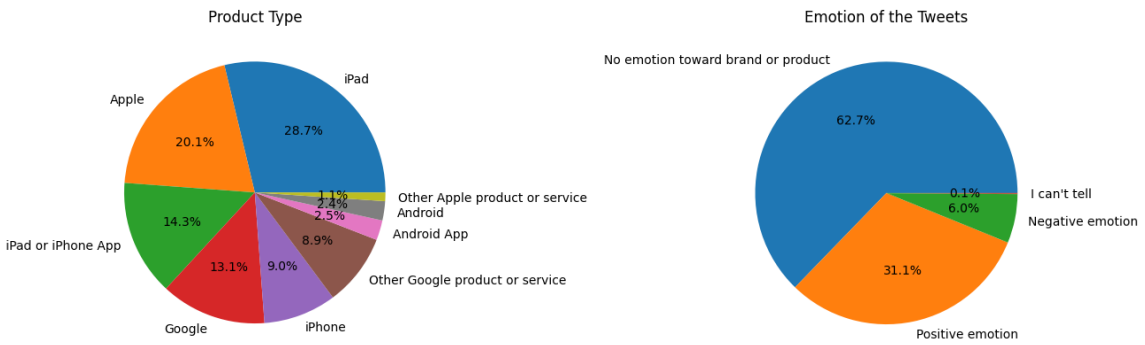
5. Error Rates:
   Keep an eye on the rate of errors or misclassifications. This can help identify issues with the model's generalization or issues with the data.

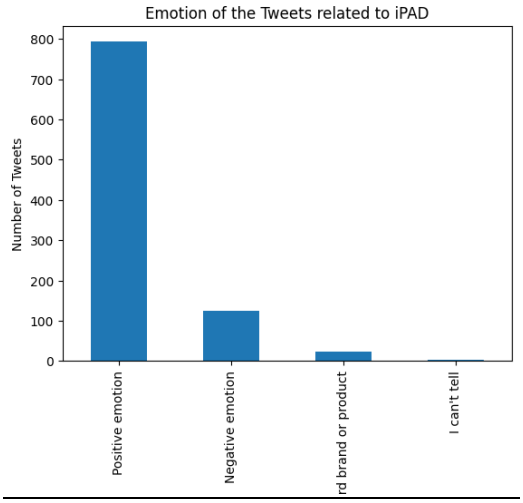
Conclusion and Data/Result Imporvement:

1. More number of epochs can be run for better accuracy and results.
2. Manually data can be labelled for ‘No targeted product/service’ as many of tweets are not pointed towards any service/product, hence they confuse the model as model is forced to predict any one of the 9 categories.
3. Better data cleaning and feature engineering can be done.
4. A detailed EDA can also result in extracting some new features or insight which might be useful in the model building.

Some graphs from EDA:







Sarthak Srivastava
Email: sarthak.s.1603@gmail.com