Dataset: The sentiment analysis dataset contained three columns containing indices, airline_sentiment, and text. As a preprocessing step, I removed the column index as a data frame containing a column for index by default. The text data had the tags and some other Miserables for which I performed text cleaning using the library re.

Preprocessing for ML models: Then performed a train-test split. I tried some machine learning models as well as some deep learning models. For the ML models, I tried Logistic, SVC, RandomForest, and XG boost. For these, I used tfidf vectorizers as a preprocessing step to further feed the models with.
ML models results:

Models	Accuracies
Logistic	0.91
SVC	0.914
XgBoost	0.89
Random Forest	0.89

Preprocessing for DL models: After this, I tried some deep learning networks when it comes to learning meanings from the text NN usually provides an edge. For implementing deep networks certain steps needed to be performed. After these steps, I implemented the Bert base model which has 12 layers and 768 hidden units to generate the embeddings from the cleaned text. After this, I developed some models using CuDNNLSTM layers connecting further with some feed-forward layers. As Bert had proved to be providing an edge in many tasks of NLP so using that made the model get more contextualized embeddings than one with encoders such as OneHot encoders.
Some simple NN were also implemented with different seq lengths which are present in the deep learning model file in Github.

Models	Accuracies
NN(with LSTM node=128)	0.922
NN(with LSTM node=64)	0.915
NN(simple)	0.9233

Since we needed to implement the API as well. So, I decided to use my best ML model as creating API with these large Deep Learning Networks was a time-consuming and challenging task as well. Similarly for the train, loading of the dataset with the OOPs I developed with the SVC classifier.