Team AK

Name and NetID of Team Members

Member 1 - Kartik Milind Ghate
NetID - kghate2

Member 2 - Anushka Amitsingh Mohane
NetID. - amohane2

Team Captain – Kartik Milind Ghate

Free Topic – IMDB Movie Review Sentiment Analysis

IMDB Movie Review Sentiment Analysis project is a Machine Learning (ML) model that will classify a given review as positive or negative based on the words present in it. It will use Natural Language Processing (NLP) for this purpose. It will be trained and performed on the Kaggle IMDB dataset.

The dataset is a CSV file from Kaggle containing IMDB reviews. The packages and modules proposed to be used for this project are Python, NumPy, Pandas. Tensorflow was used to train the model and it was on NLP. The data was be tokenized and converted to small packets called sequences. This data was padded and then input in the model for training. Hence, the trained model produced quite accurate results when subjected to test runs.

The task was divided between team members to create a tokenizer and predict.py file. The team members worked together to train and create the model. The model was created on jupyter and the code scripting was distributed among the members equally.

Kartik – Finding the dataset, predict.py file, worked on creating and training the model. Also worked on creating a simple front end to display the output.

Anushka – Finding the dataset, created the tokenizer, worked on creating and training the model. Also worked on creating a simple front end to display the output.

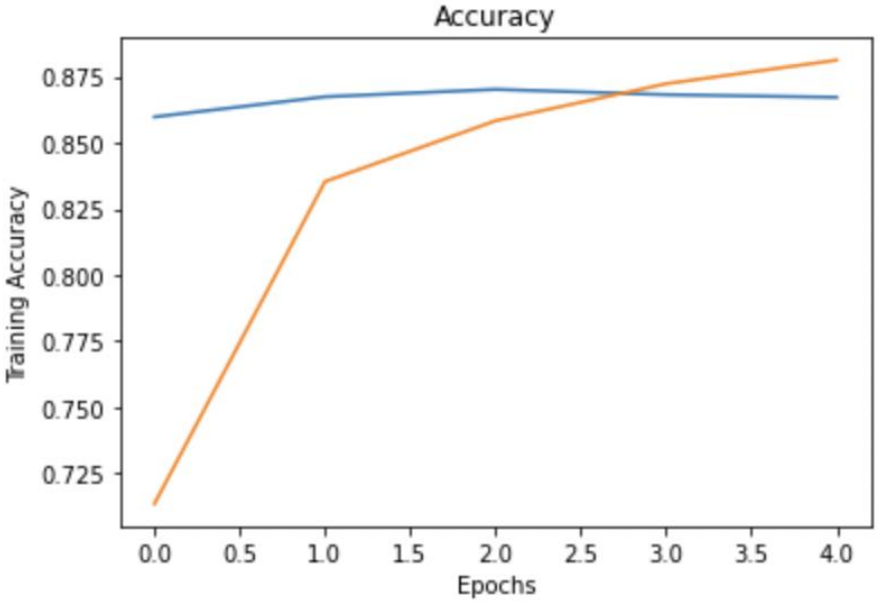
The work done by Anushka took almost 20 hours and the work done by Kartik took almost 19.5 hours.

The model, created in python using NumPy, pandas and TensorFlow was trained using the dataset found on Kaggle. The dataset is a csv file containing movie reviews which contained a list of positive and negative reviews. We first extract the files from the folder, load them into the model and then train the model to achieve greater accuracy.

The dataset was in CSV format. There were 40000 reviews in the train file and 5000 reviews in the test file.

The data was initially tokenized, then converted to sequences, padded and then inserted into the model for training.

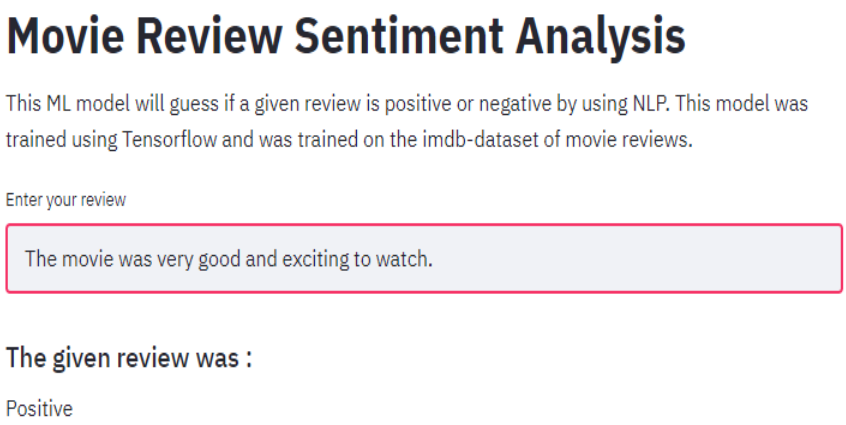
From the code, we achieve almost 86% accuracy and the graph plotted shows the same as well. Various parameters were used to train the model to achieve greater accuracy.



```
Train on 40000 samples, validate on 5000 samples
Epoch 1/5
40000/40000 [==============================] - 11s 274us/sample - loss: 0.5549 - accuracy: 0.7135 - val_loss: 0.3613
- val_accuracy: 0.859
Epoch 2/5
40000/40000 [==============================] - 7s 174us/sample - loss: 0.3796 - accuracy: 0.8353 - val_loss: 0.3140 -
val_accuracy: 0.86
Epoch 3/5
40000/40000 [==============================] - 6s 159us/sample - loss: 0.3339 - accuracy: 0.8583 - val_loss: 0.3088 -
val_accuracy: 0.87
Epoch 4/5
40000/40000 [==============================] - 7s 172us/sample - loss: 0.3112 - accuracy: 0.8723 - val_loss: 0.3121 -
val_accuracy: 0.86
Epoch 5/5
40000/40000 [==============================] - 7s 167us/sample - loss: 0.2948 - accuracy: 0.8813 - val_loss: 0.3186 -
val_accuracy: 0.86
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 120, 16)           160000
_________________________________________________________________
global_average_pooling1d (Gl (None, 16)                0
_________________________________________________________________
dropout (Dropout)            (None, 16)                0
_________________________________________________________________
dense (Dense)                (None, 6)                 102
_________________________________________________________________
dropout_1 (Dropout)          (None, 6)                 0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 7
=================================================================
Total params: 160,109
Trainable params: 160,109
Non-trainable params: 0
_________________________________________________________________
```

A very simple to use online UI was deployed to display the output and provide appropriate results. It classified the reviews as either positive or negative.



The project user just needs python to be installed on their system with libraries to be imported which are NumPy, Pandas and TensorFlow. The modules can be installed into the system using simple commands in the terminal namely:

pip install numpy

pip install pandas

pip install tensorflow

The team members faced some problems where the tokenizer was not working, there were problems with the dataset not loading into the model, and variable errors. The team members worked together to solve these issues and helped each other in each way possible to code.