Sentiment-Analysis

This project implements sentiment analysis using LSTM (Long Short-Term Memory) networks to classify the sentiment (positive, negative, neutral) of text data.

LONGSHORT-TERMMEMORYNETWORKS: LSTM stands for Long Short-Term Memory Networks. It is a variant of Recurrent Neural Networks. RNNs are usually used with sequential data such as text and audio. Usually, while computing an embedding matrix, the meaning of every word and its calculations (which are called hidden states) are stored. If the reference of a word, let's say a word is used after 100 words in a text, then all these calculations RNNs cannot be stored in its memory. That's why RNNs are not capable of learning these long-term dependencies.

DATASET:

	text	airline_sentiment
0	@VirginAmerica What @dhepburn said.	neutral
1	@VirginAmerica plus you've added commercials t...	positive
2	@VirginAmerica I didn't today... Must mean I n...	neutral
3	@VirginAmerica it's really aggressive to blast...	negative
4	@VirginAmerica and it's a really big bad thing...	negative
5	@VirginAmerica seriously would pay $30 a fligh...	negative
6	@VirginAmerica yes, nearly every time I fly VX...	positive
7	@VirginAmerica Really missed a prime opportuni...	neutral
8	@virginamerica Well, I didn't...but NOW I DO! :-D	positive
9	@VirginAmerica it was amazing, and arrived an ...	positive

Twitter US Airline Sentiment | Kaggle

CODE:

```
pip install pandas matplotlib tensorflow import
pandas as pd
data = pd.read_csv("Tweets.csv")
data.columns
review_data = data[['text','airline_sentiment']]
print(review_data.shape) review_data.head(10)

review_data = review_data[review_data['airline_sentiment'] != 'neutral']
print(review_data.shape) review_data.head(5)
review_data["airline_sentiment"].value_counts()

sentiment_label = review_data.airline_sentiment.factorize() #0 represents
positive sentiment and the 1 represents negative sentiment.
sentiment_label
tweet = review_data.text.values
```

```
from tensorflow.keras.preprocessing.text import Tokenizer

tokenizer = Tokenizer(num_words=50000)

tokenizer.fit_on_texts(tweet)

encoded_docs = tokenizer.texts_to_sequences(tweet)
```

```
from tensorflow.keras.preprocessing.sequence import pad_sequences
padsequence = pad_sequences(encoded_docs, maxlen=200)

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import LSTM,Dense, Dropout, SpatialDropout1D

from tensorflow.keras.layers import Embedding embedding_vector_length
= 32

vocab_size=13250

model = Sequential()

model.add(Embedding(vocab_size,embedding_vector_length, input_length=200))

model.add(SpatialDropout1D(0.25))

model.add(LSTM(50, dropout=0.5, recurrent_dropout=0.5))

model.add(Dropout(0.2))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy',optimizer='adam',
metrics=['accuracy']) print(model.summary())


history = model.fit(padsequence,sentiment_label[0],validation_split=0.2,

epochs=5, batch_size=32)

import matplotlib.pyplot as plt

plt.plot(history.history['accuracy'], label='acc')

plt.plot(history.history['val_accuracy'], label='val_acc')

plt.legend() plt.show()

plt.savefig("Accuracy plot.jpg")


plt.plot(history.history['loss'], label='loss')

plt.plot(history.history['val_loss'], label='val_loss')

plt.legend()

plt.show()

plt.savefig("Loss plt.jpg") def

predict_sentiment(text):

    tw = tokenizer.texts_to_sequences([text])
```

```
    tw = pad_sequences(tw,maxlen=200)

    prediction = int(model.predict(tw).round().item()) print("Predicted
    label: ", sentiment_label[1][prediction])

test_sentence1 = input("Enter the FIRST sentence to TEST: ")
predict_sentiment(test_sentence1)
```

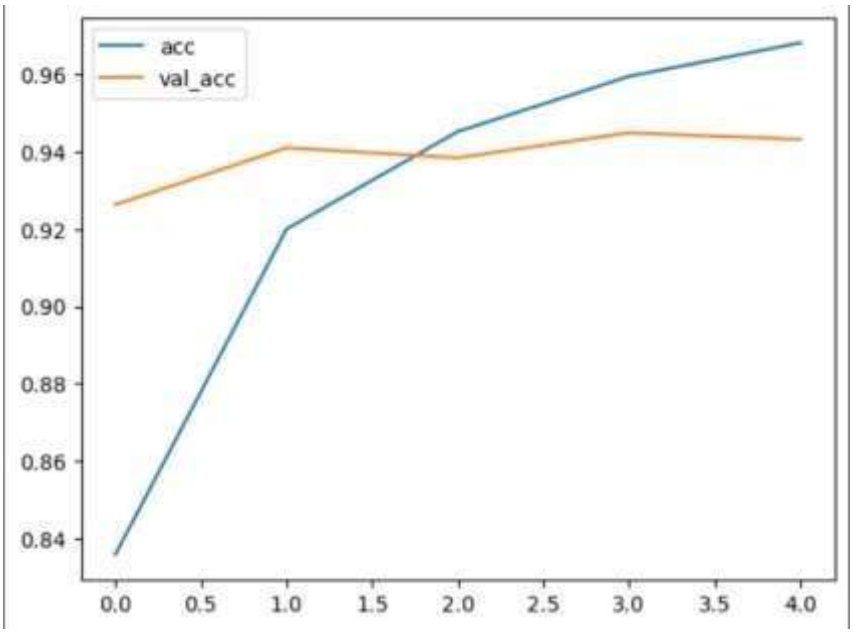
test_sentence2 =input("Enter the SECOND sentence to TEST:") predict_sentiment(test_sentence2)

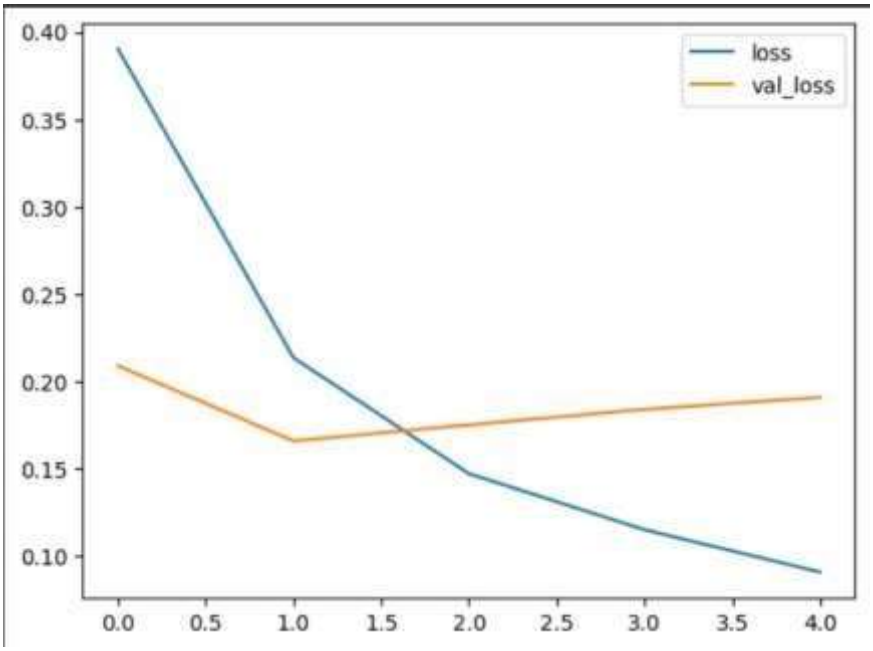
Output:

```
[15] history = model.fit(padsequence,sentiment_label[0],validation_split=0.2, epochs=5, batch_size=32)

Epoch 1/5
289/289 [==============================] - 91s 303ms/step - loss: 0.3903 - accuracy: 0.8358 - val_loss: 0.2089 - val_accuracy: 0.9264
Epoch 2/5
289/289 [==============================] - 86s 299ms/step - loss: 0.2135 - accuracy: 0.9201 - val_loss: 0.1658 - val_accuracy: 0.9411
Epoch 3/5
289/289 [==============================] - 90s 312ms/step - loss: 0.1471 - accuracy: 0.9453 - val_loss: 0.1750 - val_accuracy: 0.9385
Epoch 4/5
289/289 [==============================] - 91s 315ms/step - loss: 0.1150 - accuracy: 0.9596 - val_loss: 0.1839 - val_accuracy: 0.9450
Epoch 5/5
289/289 [==============================] - 89s 308ms/step - loss: 0.0907 - accuracy: 0.9683 - val_loss: 0.1988 - val_accuracy: 0.9433
```

Here X-Axis is no.of Epochs






```
test_sentence1 = input("Enter the FIRST sentence to TEST: ")
predict_sentiment(test_sentence1)

Enter the FIRST sentence to TEST: I felt very comfortable in the journey
1/1 [==============================] - 0s 46ms/step
Predicted label:  positive

test_sentence2 =input("Enter the SECOND sentence to TEST:")
predict_sentiment(test_sentence2)

Enter the SECOND sentence to TEST:I feel suffocating and fragrance is not good in the cabin
1/1 [==============================] - 0s 44ms/step
Predicted label:  negative
```

LIBRARIES USED :

PANDAS Pandas allows us to analyze big data and make conclusions based on statistical theories.Pandas can clean messy data sets, and make them readable and relevant. MATPLOTLIB Matplotlib is a Python library focused on data visualization and primarily used for creating graphs, plots, histograms, and bar charts. It is compatible for plotting data from SciPy, NumPy, and Pandas. TENSORFLOW Tensorflow automatically computes a function's derivatives within a high-level language. TensorFlow can be used to visualize machine learning models KERAS Keras is a module inside the Tensorflow library that is designed specifically for developing the neural networks for ML models. It can run on top of Theano and TensorFlow to train neural networks.

Summary:

In this machine learning project, we will build a binary text classifier that classifies the sentiment of the tweets into positive and negative. We will obtain more than 94% accuracy on validation. This interesting project helps businesses across the domains understand customers' sentiments/feelings towards their brands.