In [1]:
```
# Sentiment Analysis
```

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM,Dense, Dropout, SpatialDropout1D
from tensorflow.keras.layers import Embedding
```

```
WARNING:tensorflow:From C:\Users\KIIT\anaconda3\lib\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.
```

In [6]:
```
df = pd.read_csv('C:/Users/KIIT/Downloads/Tweets.csv')
```

In [7]:
```
review_df = df[['text','airline_sentiment']]

print(review_df.shape)
review_df.head(5)
```

```
(14640, 2)
```

Out[7]:

	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

In [12]:
```
df.columns
```

Out[12]:
```
Index(['tweet_id', 'airline_sentiment', 'airline_sentiment_confidence',
       'negativereason', 'negativereason_confidence', 'airline',
       'airline_sentiment_gold', 'name', 'negativereason_gold',
       'retweet_count', 'text', 'tweet_coord', 'tweet_created',
       'tweet_location', 'user_timezone'],
      dtype='object')
```

In [11]:
```
review_df = review_df[review_df['airline_sentiment'] != 'neutral']

print(review_df.shape)
review_df.head(5)
```

```
(11541, 2)
```

Out[11]:

	text	airline_sentiment
1	@VirginAmerica plus you've added commercials t...	positive
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

In [13]:
```
review_df["airline_sentiment"].value_counts()
```

Out[13]:
```
negative    9178
positive    2363
Name: airline_sentiment, dtype: int64
```

In [14]:
```
sentiment_label = review_df.airline_sentiment.factorize()
sentiment_label
```

Out[14]:
```
(array([0, 1, 1, ..., 0, 1, 1], dtype=int64),
 Index(['positive', 'negative'], dtype='object'))
```

In [15]:
```
tweet = review_df.text.values
```

In [17]:
```
from tensorflow.keras.preprocessing.text import Tokenizer
```

In [18]:
```
tokenizer = Tokenizer(num_words=5000)

tokenizer.fit_on_texts(tweet)
```

In [30]:
```
vocab_size = len(tokenizer.word_index) + 1
```

In [19]:
```
encoded_docs = tokenizer.texts_to_sequences(tweet)
```

In [22]:
```
from tensorflow.keras.preprocessing.sequence import pad_sequences

padded_sequence = pad_sequences(encoded_docs, maxlen=200)
```

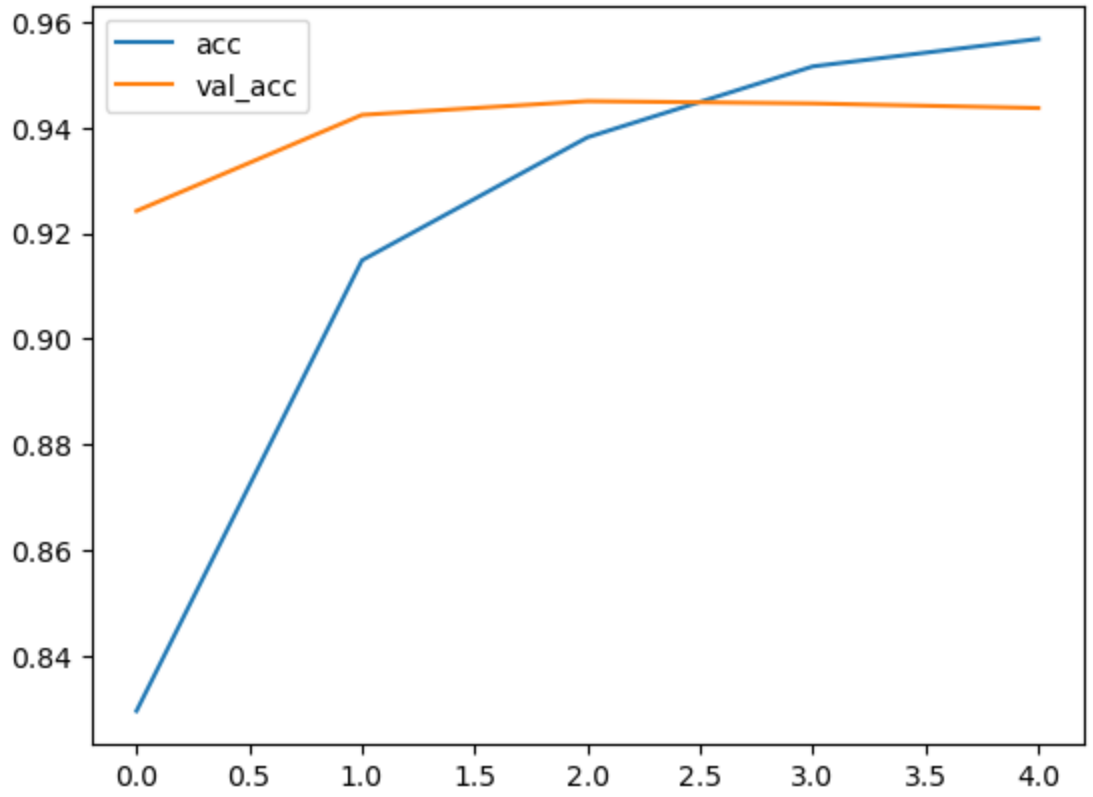
In [31]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM,Dense, Dropout, SpatialDropout1D
from tensorflow.keras.layers import Embedding


embedding_vector_length = 32
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())
```

```
WARNING:tensorflow:From C:\Users\KIIT\anaconda3\lib\site-packages\keras\src\optimizers\__init__.
py:309: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer inste
ad.

Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 200, 32)           423488

 spatial_dropout1d (Spatial  (None, 200, 32)           0
 Dropout1D)

 lstm (LSTM)                 (None, 50)                16600

 dropout (Dropout)           (None, 50)                0

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 440139 (1.68 MB)
Trainable params: 440139 (1.68 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

In [32]:
```
history = model.fit(padded_sequence,sentiment_label[0],validation_split=0.2, epochs=5, batch_siz
```

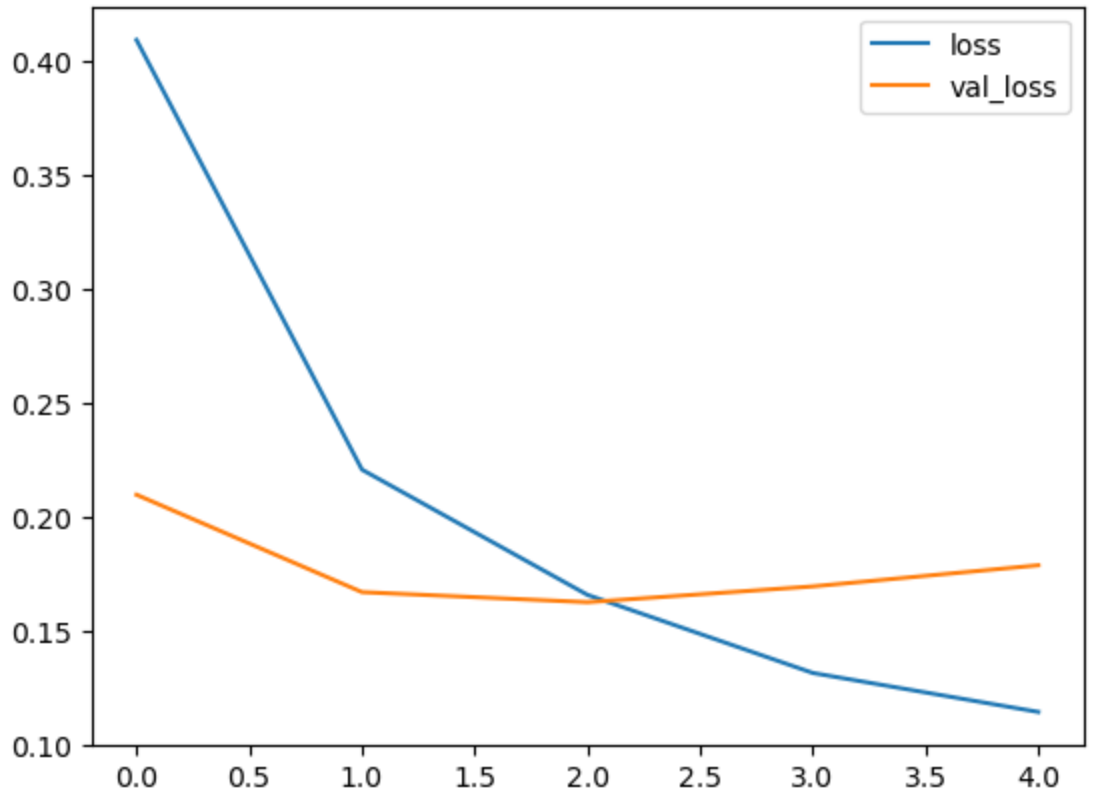
```
Epoch 1/5
WARNING:tensorflow:From C:\Users\KIIT\anaconda3\lib\site-packages\keras\src\utils\tf_utils.py:49
2: The name tf.ragged.RaggedTensorValue is deprecated. Please use tf.compat.v1.ragged.RaggedTens
orValue instead.

WARNING:tensorflow:From C:\Users\KIIT\anaconda3\lib\site-packages\keras\src\engine\base_layer_ut
ils.py:384: The name tf.executing_eagerly_outside_functions is deprecated. Please use tf.compat.
v1.executing_eagerly_outside_functions instead.

289/289 [==============================] - 42s 136ms/step - loss: 0.4093 - accuracy: 0.8295 - va
l_loss: 0.2099 - val_accuracy: 0.9242
Epoch 2/5
289/289 [==============================] - 40s 140ms/step - loss: 0.2209 - accuracy: 0.9149 - va
l_loss: 0.1671 - val_accuracy: 0.9424
Epoch 3/5
289/289 [==============================] - 42s 146ms/step - loss: 0.1660 - accuracy: 0.9381 - va
l_loss: 0.1627 - val_accuracy: 0.9450
Epoch 4/5
289/289 [==============================] - 42s 147ms/step - loss: 0.1317 - accuracy: 0.9516 - va
l_loss: 0.1697 - val_accuracy: 0.9446
Epoch 5/5
289/289 [==============================] - 42s 145ms/step - loss: 0.1146 - accuracy: 0.9568 - va
l_loss: 0.1789 - val_accuracy: 0.9437
```

In [33]:
```
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")
```



<Figure size 640x480 with 0 Axes>

In [35]:
```
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")
```



<Figure size 640x480 with 0 Axes>

In [36]:
```
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 = "I enjoyed my journey on this flight."
predict_sentiment(test_sentence1)

test_sentence2 = "This is the worst flight experience of my life!"
predict_sentiment(test_sentence2)
```

```
1/1 [==============================] - 0s 264ms/step
Predicted label:  positive
1/1 [==============================] - 0s 34ms/step
Predicted label:  negative
```

In [ ]: