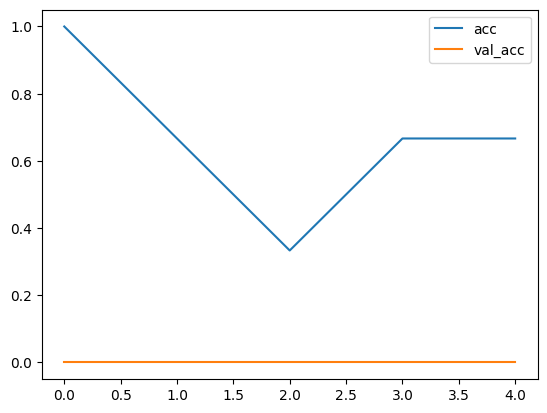
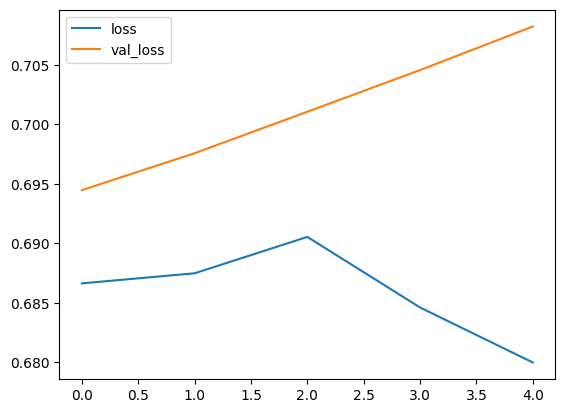
PROGRAM:

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
from tensorflow.keras.preprocessing.text import Tokenizer  
from tensorflow.keras.preprocessing.sequence import pad\_sequences  
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import LSTM, Dense, Dropout, SpatialDropout1D, Embedding  
  
# Create a dummy dataset  
data = {  
 'text': [  
 "I loved the flight, it was amazing!",  
 "The flight was okay, nothing special.",  
 "Worst experience ever, flight delayed and rude staff.",  
 "The flight attendants were friendly and helpful.",  
 "Terrible service, I'll never fly with this airline again."  
 ],  
 'airline\_sentiment': [  
 'positive',  
 'neutral',  
 'negative',  
 'positive',  
 'negative'  
 ]  
}  
  
# Create DataFrame  
df = pd.DataFrame(data)  
  
# Filter out neutral sentiment  
review\_df = df[df['airline\_sentiment'] != 'neutral']  
  
# Tokenize the text  
tweet = review\_df.text.values  
tokenizer = Tokenizer(num\_words=5000)  
tokenizer.fit\_on\_texts(tweet)  
  
# Determine vocabulary size  
vocab\_size = len(tokenizer.word\_index) + 1  
  
# Encode and pad sequences  
encoded\_docs = tokenizer.texts\_to\_sequences(tweet)  
padded\_sequence = pad\_sequences(encoded\_docs, maxlen=20)  
  
# Model architecture  
embedding\_vector\_length = 32  
model = Sequential()  
model.add(Embedding(vocab\_size, embedding\_vector\_length, input\_length=20))  
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'])  
  
# Model summary  
print(model.summary())  
  
# Model training  
sentiment\_label = review\_df.airline\_sentiment.factorize()  
history = model.fit(padded\_sequence, sentiment\_label[0], validation\_split=0.2, epochs=5, batch\_size=32)  
  
# Plot accuracy  
plt.plot(history.history['accuracy'], label='acc')  
plt.plot(history.history['val\_accuracy'], label='val\_acc')  
plt.legend()  
plt.savefig("Accuracy\_plot.jpg")  
plt.show()  
  
# Plot loss  
plt.plot(history.history['loss'], label='loss')  
plt.plot(history.history['val\_loss'], label='val\_loss')  
plt.legend()  
plt.savefig("Loss\_plot.jpg")  
plt.show()  
  
# Prediction function  
def predict\_sentiment(text):  
 tw = tokenizer.texts\_to\_sequences([text])  
 tw = pad\_sequences(tw, maxlen=20)  
 prediction = int(model.predict(tw).round().item())  
 print("Predicted label: ", sentiment\_label[1][prediction])  
  
# Test sentences  
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)

**OUTPUT:**

Model: "sequential\_1"  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 Layer (type) Output Shape Param #   
=================================================================  
 embedding\_1 (Embedding) (None, 20, 32) 896   
   
 spatial\_dropout1d\_1 (Spati (None, 20, 32) 0   
 alDropout1D)   
   
 lstm\_1 (LSTM) (None, 50) 16600   
   
 dropout\_1 (Dropout) (None, 50) 0   
   
 dense\_1 (Dense) (None, 1) 51   
   
=================================================================  
Total params: 17547 (68.54 KB)  
Trainable params: 17547 (68.54 KB)  
Non-trainable params: 0 (0.00 Byte)  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
None  
Epoch 1/5  
1/1 [==============================] - 4s 4s/step - loss: 0.6866 - accuracy: 1.0000 - val\_loss: 0.6945 - val\_accuracy: 0.0000e+00  
Epoch 2/5  
1/1 [==============================] - 0s 52ms/step - loss: 0.6875 - accuracy: 0.6667 - val\_loss: 0.6976 - val\_accuracy: 0.0000e+00  
Epoch 3/5  
1/1 [==============================] - 0s 53ms/step - loss: 0.6905 - accuracy: 0.3333 - val\_loss: 0.7011 - val\_accuracy: 0.0000e+00  
Epoch 4/5  
1/1 [==============================] - 0s 52ms/step - loss: 0.6846 - accuracy: 0.6667 - val\_loss: 0.7046 - val\_accuracy: 0.0000e+00  
Epoch 5/5  
1/1 [==============================] - 0s 62ms/step - loss: 0.6800 - accuracy: 0.6667 - val\_loss: 0.7082 - val\_accuracy: 0.0000e+00





1/1 [==============================] - 0s 240ms/step  
Predicted label: positive  
1/1 [==============================] - 0s 20ms/step  
Predicted label: positive