The practice is performed following this tutorial: https://www.makeuseof.com/create-sentiment-analysis-model/, which used Trip Advisor Hotel Reviews dataset from Kaggle to build the sentiment analysis model

Dataset used to train the model in this practice: https://www.kaggle.com/datasets/kazanova/sentiment140/data, which is the dataset for ACM AI Tweetiment hackathon

```
! pip install tensorflow scikit-learn pandas numpy pickle5
```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.25.2)
Collecting pickle5
  Downloading pickle5-0.0.11.tar.gz (132 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 132.1/132.1 kB 1.4 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=23.5.26 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.5
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.9.0)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes~=0.2.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.3.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from tensorflow) (24.0)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in /usr/local/lib/pyt
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from tensorflow) (67.7.2)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.4.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (4.11.0)
Requirement already satisfied: wrapt<1.15,>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.14.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.3
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.63.0)
Requirement already satisfied: tensorboard<2.16,>=2.15 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.15.2)
Requirement already satisfied: tensorflow-estimator<2.16,>=2.15.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.15
Requirement already satisfied: keras<2.16,>=2.15.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.15.0)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.4)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow) (0
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15->tensor
Requirement already satisfied: google-auth-oauthlib<2,>=0.5 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15->tensorflow)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15->tensorfl
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.1
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15->tensorflow)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorb
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorbo
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.16
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from google-auth-oauthlib<2,>=0.5
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorb
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard<2.16,>=
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard<2
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard<2
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from werkzeug>=1.0.1->tensorboard<2.16,>
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in /usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1->google-au
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from requests-oauthlib>=0.7.0->google-auth
Building wheels for collected packages: pickle5
  Building wheel for pickle5 (setup.py) ... done
  Created wheel for pickle5: filename=pickle5-0.0.11-cp310-cp310-linux_x86_64.whl size=255313 sha256=ed069c789a1fb6151f1c0d75068019ba
  Stored in directory: /root/.cache/pip/wheels/7d/14/ef/4aab19d27fa8e58772be5c71c16add0426acf9e1f64353235c
Successfully built pickle5
Installing collected packages: pickle5
Successfully installed pickle5-0.0.11
```

```
import numpy as np
import pandas as pd
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
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 Embedding, Conv1D, GlobalMaxPooling1D, Dense, Dropout
import pickle5 as pickle
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Load data

```
df = pd.read_csv("/content/drive/MyDrive/Sentiment Analysis Dataset/tweetiment.csv", encoding = 'latin',header=None)
```

```
# Display the first few rows of the dataset
print(df.head())
```

```
   0           1                             2         3               4  \
0  0  1467810369  Mon Apr 06 22:19:45 PDT 2009  NO_QUERY  _TheSpecialOne_
1  0  1467810672  Mon Apr 06 22:19:49 PDT 2009  NO_QUERY    scotthamilton
2  0  1467810917  Mon Apr 06 22:19:53 PDT 2009  NO_QUERY         mattycus
3  0  1467811184  Mon Apr 06 22:19:57 PDT 2009  NO_QUERY          ElleCTF
4  0  1467811193  Mon Apr 06 22:19:57 PDT 2009  NO_QUERY           Karoli

                                                   5
0  @switchfoot http://twitpic.com/2y1zl - Awww, t...
1  is upset that he can't update his Facebook by ...
2  @Kenichan I dived many times for the ball. Man...
3    my whole body feels itchy and like its on fire
4  @nationwideclass no, it's not behaving at all....
```

```
# Select only the first and last columns
df = df[[5, 0]]
df.columns=['tweet', 'sentiment']
print(df.head())
```

```
                                               tweet  sentiment
0  @switchfoot http://twitpic.com/2y1zl - Awww, t...          0
1  is upset that he can't update his Facebook by ...          0
2  @Kenichan I dived many times for the ball. Man...          0
3    my whole body feels itchy and like its on fire           0
4  @nationwideclass no, it's not behaving at all....          0
```

```
sentiment_distribution = df['sentiment'].value_counts()
print(sentiment_distribution)
```

```
sentiment
0    800000
4    800000
Name: count, dtype: int64
```

```
# changed label to 0 for negative and 1 for positive
df['sentiment'] = df['sentiment'].replace(4,1)
df = df.sample(frac=1).reset_index(drop=True)
```

```
# Check for missing values in the 'text' column
missing_values= df['tweet'].isnull().sum()
print("Number of missing values in 'tweet' column of training dataset:", missing_values)

# Drop rows with missing values
df = df.dropna(subset=['tweet'])
```

```
Number of missing values in 'tweet' column of training dataset: 0
```

```
# Tokenization and Padding
tokenizer = Tokenizer(num_words=5000, oov_token='<OOV>')
tokenizer.fit_on_texts(df['tweet'])
word_index = tokenizer.word_index
sequences = tokenizer.texts_to_sequences(df['tweet'])
padded_sequences = pad_sequences(sequences, maxlen=100, truncating='post')
```

```
# Convert sentiment labels to one-hot encoded vectors
sentiment_labels = pd.get_dummies(df['sentiment']).values
```

```
# Splitting the dataset into training and testing Sets
x_train, x_test, y_train, y_test = train_test_split(padded_sequences, sentiment_labels, test_size=0.2)
```

Creating and Training the Neural Network

```
# Creating the Neural Network
model = Sequential()
model.add(Embedding(5000, 100, input_length=100))
model.add(Conv1D(64, 5, activation='relu'))
model.add(GlobalMaxPooling1D())
model.add(Dense(32, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(2, activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 100, 100)          500000

 conv1d_3 (Conv1D)           (None, 96, 64)            32064

 global_max_pooling1d_3 (Gl  (None, 64)                0
 obalMaxPooling1D)

 dense_6 (Dense)             (None, 32)                2080

 dropout_3 (Dropout)         (None, 32)                0

 dense_7 (Dense)             (None, 2)                 66

=================================================================
Total params: 534210 (2.04 MB)
Trainable params: 534210 (2.04 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
# Stop training early if not improving
from keras.callbacks import EarlyStopping

early_stopping = EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=True)
```

```
# Training the Neural Network
history = model.fit(x_train, y_train, epochs=20, batch_size=32, validation_data=(x_test, y_test), callbacks=[early_stopping])
```

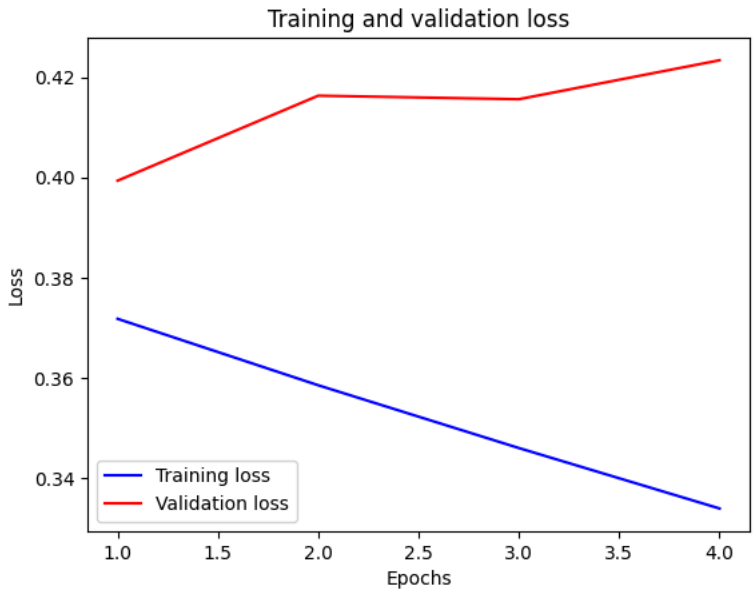
```
Epoch 1/20
40000/40000 [==============================] - 1260s 31ms/step - loss: 0.3718 - accuracy: 0.8365 - val_loss: 0.3994 - val_accuracy: 0.82
Epoch 2/20
40000/40000 [==============================] - 1207s 30ms/step - loss: 0.3585 - accuracy: 0.8430 - val_loss: 0.4163 - val_accuracy: 0.81
Epoch 3/20
40000/40000 [==============================] - 1215s 30ms/step - loss: 0.3460 - accuracy: 0.8485 - val_loss: 0.4156 - val_accuracy: 0.81
Epoch 4/20
40000/40000 [==============================] - 1205s 30ms/step - loss: 0.3339 - accuracy: 0.8540 - val_loss: 0.4234 - val_accuracy: 0.81
```

```
# Loss and accuracy values for training and validation sets
train_loss = history.history['loss']
val_loss = history.history['val_loss']
train_acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

# Print or visualize the loss and accuracy curves
import matplotlib.pyplot as plt

epochs = range(1, len(train_loss) + 1)

plt.plot(epochs, train_loss, 'b', label='Training loss')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```



Based on the plot, the training loss decreases sharply while the validation loss slowly increases, it indicates that the model is fitting the training data too closely, capturing noise or irrelevant patterns that don't generalize well to new data. Early Stopping stop training when the validation loss starts to increase.

```
# Evaluating the Performance of the Trained Model
from sklearn.metrics import f1_score

y_pred = np.argmax(model.predict(x_test), axis=-1)
y_true = np.argmax(y_test, axis=-1)
# Calculate accuracy score
# Calculate accuracy score
print("Accuracy:", accuracy_score(y_true, y_pred))
# Calculate F1-score
print("F1-score:", f1_score(y_true, y_pred, average='macro'))
```

```
10000/10000 [==============================] - 98s 10ms/step
Accuracy: 0.820946875
F1-score: 0.8209446776537175
```

```
# Saving the Model
model.save('my_tweetiment_analysis_model.h5')
with open('tokenizer.pickle', 'wb') as handle:
    pickle.dump(tokenizer, handle, protocol=pickle.HIGHEST_PROTOCOL)
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file via `m
  saving_api.save_model(
```

Using the Model to Classify the Sentiment of Given Text

```
# Load the saved model and tokenizer
import keras

model = keras.models.load_model('my_tweetiment_analysis_model.h5')
with open('tokenizer.pickle', 'rb') as handle:
    tokenizer = pickle.load(handle)


# Define a function to predict the sentiment of input text
def predict_sentiment(text, model, tokenizer):
    # Tokenize and pad the input text
    text_sequence = tokenizer.texts_to_sequences([text])
    text_sequence = pad_sequences(text_sequence, maxlen=100)

    # Make a prediction using the trained model
    predicted_rating = model.predict(text_sequence)[0]

    # Map the predicted sentiment index to its corresponding label
    sentiment_mapping = {0: 'negative', 1: 'positive'}
    predicted_index = np.argmax(predicted_rating)
    predicted_sentiment_label = sentiment_mapping[predicted_index]

    return predicted_index, predicted_sentiment_label


positive_texts = [
    "I loved the new book. It's amazing!",
    "The weather today is beautiful.",
    "I'm feeling great today.",
    "I had a fantastic time at the party last night!",
    "I'm really excited about the upcoming event.",
    "The customer service was excellent!",
    "I feel so happy right now.",
    "The meeting went well.",
    "The hotel room was clean and comfortable.",
    "I'm proud of my achievements.",
    "I'm grateful for all the support I've received.",
    "Today was a productive day!",
    "I received some wonderful news today.",
    "The food at the restaurant was delicious.",
    "I'm surrounded by amazing friends and family.",
    "I'm making progress towards my goals.",
    "I'm enjoying every moment of life.",
    "I'm filled with optimism about the future.",
    "I'm thankful for the opportunities I've been given.",
    "I'm blessed to have such a fulfilling life."
]
print(len(positive_texts))
```

```
20
```

```
for text in positive_texts:
    predicted_index, predicted_sentiment = predict_sentiment(text, model, tokenizer)
    print("Text:", text)
    print("Predicted Sentiment Index:", predicted_index)
    print("Predicted Sentiment Label:", predicted_sentiment)
    print()
```

```
Predicted Sentiment Label: positive

1/1 [==============================] - 0s 20ms/step
Text: I received some wonderful news today.
Predicted Sentiment Index: 1
Predicted Sentiment Label: positive

1/1 [==============================] - 0s 20ms/step
Text: The food at the restaurant was delicious.
Predicted Sentiment Index: 1
Predicted Sentiment Label: positive

1/1 [==============================] - 0s 26ms/step
Text: I'm surrounded by amazing friends and family.
Predicted Sentiment Index: 1
Predicted Sentiment Label: positive

1/1 [==============================] - 0s 26ms/step
Text: I'm making progress towards my goals.
Predicted Sentiment Index: 1
Predicted Sentiment Label: positive

1/1 [==============================] - 0s 23ms/step
Text: I'm enjoying every moment of life.
Predicted Sentiment Index: 1
Predicted Sentiment Label: positive

1/1 [==============================] - 0s 21ms/step
Text: I'm filled with optimism about the future.
Predicted Sentiment Index: 1
Predicted Sentiment Label: positive

1/1 [==============================] - 0s 21ms/step
Text: I'm thankful for the opportunities I've been given.
Predicted Sentiment Index: 1
Predicted Sentiment Label: positive

1/1 [==============================] - 0s 26ms/step
Text: I'm blessed to have such a fulfilling life.
Predicted Sentiment Index: 1
Predicted Sentiment Label: positive
```

```
negative_texts = [
    "The service at the restaurant was terrible.",
    "The traffic was awful this morning.",
    "The food tasted awful.",
    "I'm so disappointed with the product quality.",
    "The flight got delayed again.",
    "I can't stand this waiting.",
    "I'm tired of all this drama.",
    "I'm annoyed by all the noise outside.",
    "The internet connection is so slow.",
    "I'm so frustrated with this project.",
    "I'm feeling really down today.",
    "Everything seems to be going wrong.",
    "I'm overwhelmed with stress.",
    "I'm not looking forward to tomorrow.",
    "I'm exhausted from all the work.",
    "I'm upset about what happened.",
    "I'm not in the mood for anything.",
    "I'm disappointed in myself.",
    "I'm so lonely right now.",
    "I'm struggling to stay positive."
]
```

```
print(len(negative_texts))
```

```
20
```

```
for text in negative_texts:
    predicted_index, predicted_sentiment = predict_sentiment(text, model, tokenizer)
    print("Text:", text)
    print("Predicted Sentiment Index:", predicted_index)
    print("Predicted Sentiment Label:", predicted_sentiment)
    print()
```

```
1/1 [==============================] - 0s 22ms/step
Text: The service at the restaurant was terrible.
Predicted Sentiment Index: 0
Predicted Sentiment Label: negative
```

```
1/1 [==============================] - 0s 24ms/step
Text: The traffic was awful this morning.
Predicted Sentiment Index: 0
Predicted Sentiment Label: negative

1/1 [==============================] - 0s 24ms/step
Text: The food tasted awful.
Predicted Sentiment Index: 0
Predicted Sentiment Label: negative

1/1 [==============================] - 0s 21ms/step
Text: I'm so disappointed with the product quality.
Predicted Sentiment Index: 0
Predicted Sentiment Label: negative

1/1 [==============================] - 0s 24ms/step
Text: The flight got delayed again.
Predicted Sentiment Index: 0
Predicted Sentiment Label: negative

1/1 [==============================] - 0s 25ms/step
Text: I can't stand this waiting.
Predicted Sentiment Index: 0
Predicted Sentiment Label: negative

1/1 [==============================] - 0s 23ms/step
Text: I'm tired of all this drama.
Predicted Sentiment Index: 0
Predicted Sentiment Label: negative

1/1 [==============================] - 0s 26ms/step
Text: I'm annoyed by all the noise outside.
Predicted Sentiment Index: 0
Predicted Sentiment Label: negative

1/1 [==============================] - 0s 23ms/step
Text: The internet connection is so slow.
Predicted Sentiment Index: 0
Predicted Sentiment Label: negative

1/1 [==============================] - 0s 24ms/step
Text: I'm so frustrated with this project.
Predicted Sentiment Index: 0
Predicted Sentiment Label: negative

1/1 [==============================] - 0s 21ms/step
Text: I'm feeling really down today.
Predicted Sentiment Index: 0
Predicted Sentiment Label: negative

1/1 [==============================] - 0s 23ms/step
Text: Everything seems to be going wrong.
Predicted Sentiment Index: 0
```