Sentiment Analysis

Import the libraries

```
# linear algebra
import numpy as np

# data processing, CSV file I/O
import pandas as pd
import os
```

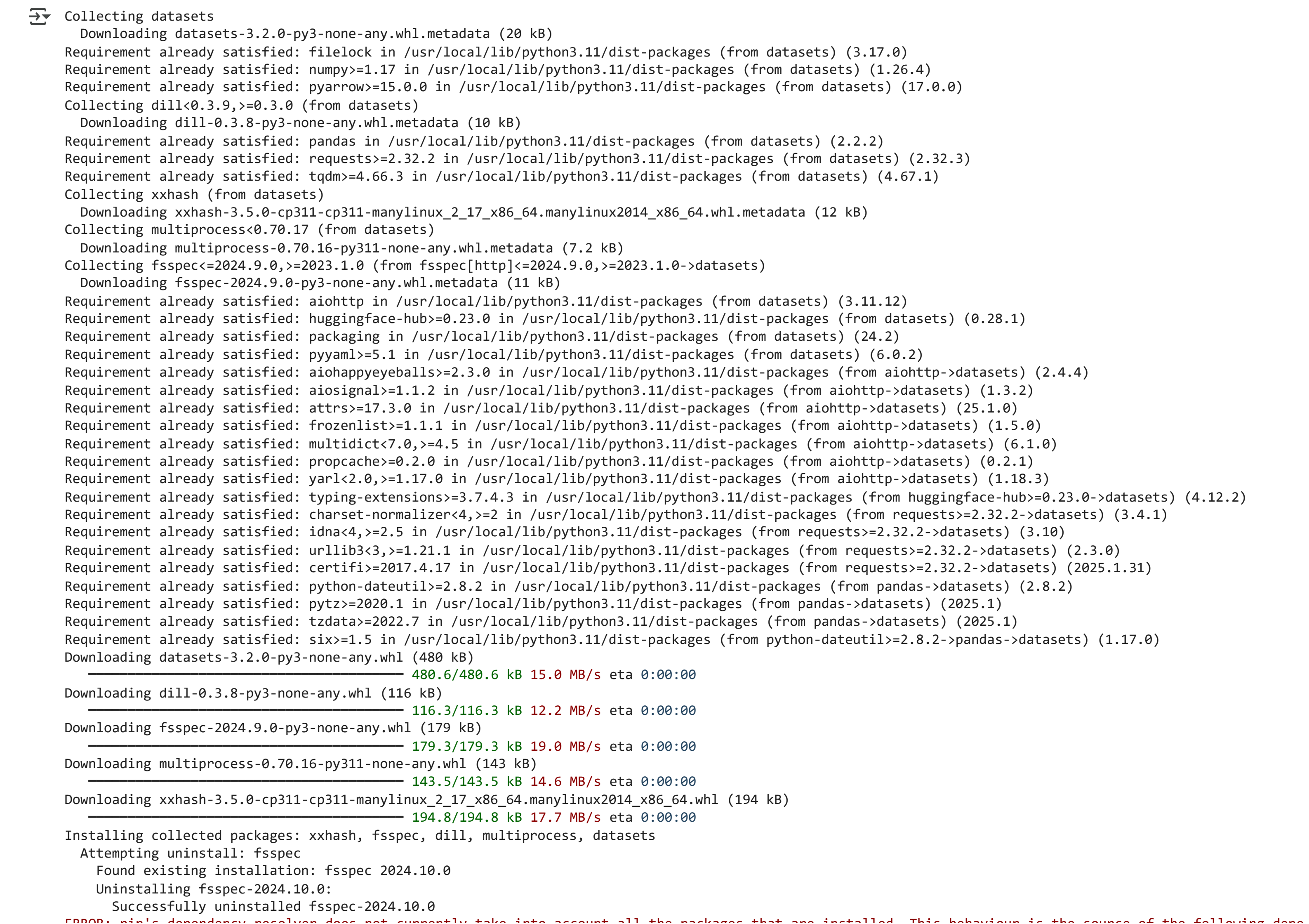
```
# !pip install -qU numpy pandas matplotlib seaborn scikit-learn tensorflow
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
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, LSTM, Dense, Dropout, SpatialDropout1D
from tensorflow.keras.callbacks import EarlyStopping
```

```
!pip install datasets
# load the data from imdb
from datasets import load_dataset

# Load IMDb dataset
dataset = load_dataset("imdb")

# Train DataFrame
df = dataset["train"].to_pandas()
```



```
Collecting datasets
  Downloading datasets-3.2.0-py3-none-any.whl.metadata (20 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from datasets) (3.17.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages (from datasets) (1.26.4)
Requirement already satisfied: pyarrow>=15.0.0 in /usr/local/lib/python3.11/dist-packages (from datasets) (17.0.0)
Collecting dill<0.3.9,>=0.3.0 (from datasets)
  Downloading dill-0.3.8-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (from datasets) (2.2.2)
Requirement already satisfied: requests>=2.32.2 in /usr/local/lib/python3.11/dist-packages (from datasets) (2.32.3)
Requirement already satisfied: tqdm>=4.66.3 in /usr/local/lib/python3.11/dist-packages (from datasets) (4.67.1)
Collecting xxhash (from datasets)
  Downloading xxhash-3.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (12 kB)
Collecting multiprocess<0.70.17 (from datasets)
  Downloading multiprocess-0.70.16-py311-none-any.whl.metadata (7.2 kB)
Collecting fsspec<=2024.9.0,>=2023.1.0 (from fsspec[http]<=2024.9.0,>=2023.1.0->datasets)
  Downloading fsspec-2024.9.0-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.11/dist-packages (from datasets) (3.11.12)
Requirement already satisfied: huggingface-hub>=0.23.0 in /usr/local/lib/python3.11/dist-packages (from datasets) (0.28.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from datasets) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from datasets) (6.0.2)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (2.4.4)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.3.2)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (25.1.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.5.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (6.1.0)
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (0.2.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.18.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.23.0->datasets) (4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests>=2.32.2->datasets) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests>=2.32.2->datasets) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests>=2.32.2->datasets) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests>=2.32.2->datasets) (2025.1.31)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas->datasets) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas->datasets) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas->datasets) (2025.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas->datasets) (1.17.0)
Downloading datasets-3.2.0-py3-none-any.whl (480 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 480.6/480.6 kB 15.0 MB/s eta 0:00:00
Downloading dill-0.3.8-py3-none-any.whl (116 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.3/116.3 kB 12.2 MB/s eta 0:00:00
Downloading fsspec-2024.9.0-py3-none-any.whl (179 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 179.3/179.3 kB 19.0 MB/s eta 0:00:00
Downloading multiprocess-0.70.16-py311-none-any.whl (143 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 143.5/143.5 kB 14.6 MB/s eta 0:00:00
Downloading xxhash-3.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.8/194.8 kB 17.7 MB/s eta 0:00:00
Installing collected packages: xxhash, fsspec, dill, multiprocess, datasets
  Attempting uninstall: fsspec
    Found existing installation: fsspec 2024.10.0
    Uninstalling fsspec-2024.10.0:
      Successfully uninstalled fsspec-2024.10.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following depe
```

```
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following depe
gcsfs 2024.10.0 requires fsspec==2024.10.0, but you have fsspec 2024.9.0 which is incompatible.
torch 2.5.1+cu124 requires nvidia-cublas-cu12==12.4.5.8; platform_system == "Linux" and platform_machine == "x86_64", but you have nvidia-cublas-cu12 12.5.3.
torch 2.5.1+cu124 requires nvidia-cuda-cupti-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you have nvidia-cuda-cupti-cu12
torch 2.5.1+cu124 requires nvidia-cuda-nvrtc-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you have nvidia-cuda-nvrtc-cu12
torch 2.5.1+cu124 requires nvidia-cuda-runtime-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you have nvidia-cuda-runtime-
torch 2.5.1+cu124 requires nvidia-cudnn-cu12==9.1.0.70; platform_system == "Linux" and platform_machine == "x86_64", but you have nvidia-cudnn-cu12 9.3.0.75
torch 2.5.1+cu124 requires nvidia-cufft-cu12==11.2.1.3; platform_system == "Linux" and platform_machine == "x86_64", but you have nvidia-cufft-cu12 11.2.3.61
torch 2.5.1+cu124 requires nvidia-curand-cu12==10.3.5.147; platform_system == "Linux" and platform_machine == "x86_64", but you have nvidia-curand-cu12 10.3.
torch 2.5.1+cu124 requires nvidia-cusolver-cu12==11.6.1.9; platform_system == "Linux" and platform_machine == "x86_64", but you have nvidia-cusolver-cu12 11.
torch 2.5.1+cu124 requires nvidia-cusparse-cu12==12.3.1.170; platform_system == "Linux" and platform_machine == "x86_64", but you have nvidia-cusparse-cu12 1
torch 2.5.1+cu124 requires nvidia-nvjitlink-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you have nvidia-nvjitlink-cu12 1
Successfully installed datasets-3.2.0 dill-0.3.8 fsspec-2024.9.0 multiprocess-0.70.16 xxhash-3.5.0
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning: 
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Cola
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
README.md: 100%                                    7.81k/7.81k [00:00<00:00, 440kB/s]
train-00000-of-00001.parquet: 100%                 21.0M/21.0M [00:00<00:00, 86.9MB/s]
test-00000-of-00001.parquet: 100%                  20.5M/20.5M [00:00<00:00, 113MB/s]
unsupervised-00000-of-00001.parquet: 100%          42.0M/42.0M [00:00<00:00, 71.8MB/s]
Generating train split: 100%                       25000/25000 [00:00<00:00, 93268.76 examples/s]
Generating test split: 100%                        25000/25000 [00:00<00:00, 114060.37 examples/s]
Generating unsupervised split: 100%                50000/50000 [00:00<00:00, 128668.18 examples/s]
```

```
# print the first 5 data values
df.head()
```

```
                                                 text  label
0   I rented I AM CURIOUS-YELLOW from my video sto...      0
1             "I Am Curious: Yellow" is a risible and preten...      0
2                If only to avoid making this type of film in t...      0
3          This film was probably inspired by Godard's Ma...      0
4                Oh, brother...after hearing about this ridicul...      0
```

Next steps: Generate code with df View recommended plots New interactive sheet

```
# count the label
df['label'].value_counts()
```

```
       count
label
    0  12500
    1  12500

dtype: int64
```

```
# print the column names in the dataframe
print(df.columns)
```

```
Index(['text', 'label'], dtype='object')
```

Exploratory Data Analysis (EDA)

```
df.isnull().sum()
```

```
           0
text       0
label      0

dtype: int64
```

Check the distribution of sentiments

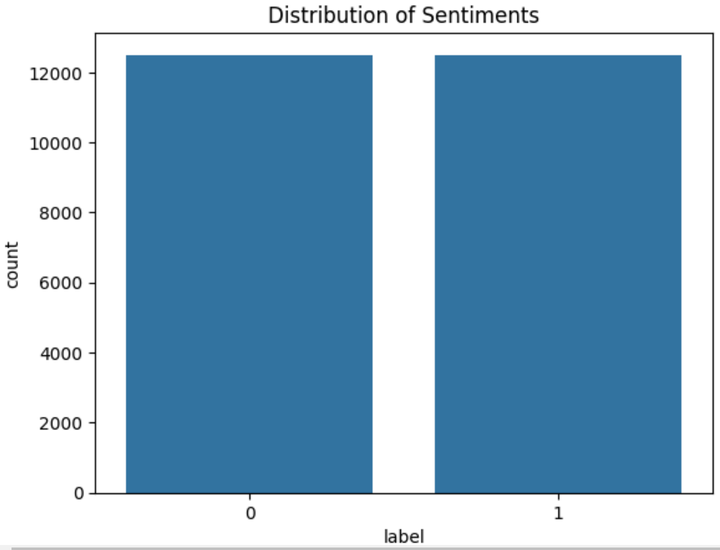
```
import seaborn as sns

# Making a distribution plot of the data

print("The distribution of Sentiments is : ")
print()


sns.countplot(x='label', data=df)
plt.title('Distribution of Sentiments')
plt.show()
```

```
The distribution of Sentiments is :
```



Splitting the data into trian and test set

```
# Train , test split 80:20
x_train, x_test, y_train, y_test = train_test_split(df['text'], df['label'], test_size=0.2, random_state=42)
```

```
x_train
```

```
                                                    text
23311       I borrowed this movie despite its extremely lo...
23623        After the unexpected accident that killed an i...
 1020   On the 1998 summer blockbuster hit BASEketball...
12645    Can Scarcely Imagine a Better Movie Than This<...
 1533        A still famous but decadent actor (Morgan Free...
  ...                                                    ...
21575     My discovery of the cinema of Jan Svankmajer o...
 5390          The story is similar to ET: an extraterrestria...
  860        I have read the novel Reaper of Ben Mezrich a ...
15795          Went to see this finnish film and I've got to ...
23654       I first saw "Breaking Glass" in 1980, and thou...

20000 rows × 1 columns

dtype: object
```

```
y_train
```

	label
23311	1
23623	1
1020	0
12645	1
1533	0
...	...
21575	1
5390	0
860	0
15795	1
23654	1

20000 rows × 1 columns

dtype: int64

```
# count the zero and one
print(y_train.value_counts())
```

```
label
1    10015
0     9985
Name: count, dtype: int64
```

```
# count the zero and one
print(y_test.value_counts())
```

```
label
0    2515
1    2485
Name: count, dtype: int64
```

Tokenization and Padding

```
# oov_token set the words as OOV (Out of Vocab )which are not present in Vocab
# keep the most fequent 5000 words in the vocab

tokenizer = Tokenizer(num_words=5000, oov_token='<OOV>')
tokenizer.fit_on_texts(x_train)
```

Tokenize the dataset

```
x_train_seq = tokenizer.texts_to_sequences(x_train)
x_test_seq = tokenizer.texts_to_sequences(x_test)
```

Apply Padding

```
# Since all the inputs are of different length apply padding

max_length = 200
X_train_pad = pad_sequences(x_train_seq, maxlen=max_length, padding='post', truncating='post')
X_test_pad = pad_sequences(x_test_seq, maxlen=max_length, padding='post', truncating='post')
```

Build the LSTM MODEL

```
# define the model architecture
model = Sequential([
    Embedding(input_dim=5000, output_dim=128),
    SpatialDropout1D(0.2),
    LSTM(64, dropout=0.2, recurrent_dropout=0.2),
    Dense(64, activation='relu'),
    Dropout(0.2),
    Dense(1, activation='sigmoid')
])


# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])


# Early Stopping to prevent overfitting
early_stop = EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=True)
```

Train the model

```
history = model.fit(
    X_train_pad, y_train,
    epochs=10,
    batch_size=64,
    validation_data=(X_test_pad, y_test),
    callbacks=[early_stop]
)
```

```
Epoch 1/10
313/313 ━━━━━━━━━━━━━━━━━━━━ 257s 759ms/step - accuracy: 0.5085 - loss: 0.6930 - val_accuracy: 0.5508 - val_loss: 0.6825
Epoch 2/10
313/313 ━━━━━━━━━━━━━━━━━━━━ 218s 650ms/step - accuracy: 0.5461 - loss: 0.6823 - val_accuracy: 0.6100 - val_loss: 0.6501
Epoch 3/10
313/313 ━━━━━━━━━━━━━━━━━━━━ 262s 649ms/step - accuracy: 0.6340 - loss: 0.6226 - val_accuracy: 0.5512 - val_loss: 0.6792
Epoch 4/10
313/313 ━━━━━━━━━━━━━━━━━━━━ 258s 636ms/step - accuracy: 0.6330 - loss: 0.6138 - val_accuracy: 0.5970 - val_loss: 0.6508
Epoch 5/10
313/313 ━━━━━━━━━━━━━━━━━━━━ 204s 642ms/step - accuracy: 0.7354 - loss: 0.5275 - val_accuracy: 0.8452 - val_loss: 0.3618
Epoch 6/10
313/313 ━━━━━━━━━━━━━━━━━━━━ 197s 630ms/step - accuracy: 0.8676 - loss: 0.3237 - val_accuracy: 0.8560 - val_loss: 0.3344
Epoch 7/10
313/313 ━━━━━━━━━━━━━━━━━━━━ 206s 643ms/step - accuracy: 0.9030 - loss: 0.2516 - val_accuracy: 0.8586 - val_loss: 0.3467
Epoch 8/10
313/313 ━━━━━━━━━━━━━━━━━━━━ 200s 638ms/step - accuracy: 0.9229 - loss: 0.2090 - val_accuracy: 0.8564 - val_loss: 0.3685
Epoch 9/10
313/313 ━━━━━━━━━━━━━━━━━━━━ 201s 636ms/step - accuracy: 0.9377 - loss: 0.1718 - val_accuracy: 0.8516 - val_loss: 0.3846
```

```
# Model Summary
model.summary()
```

```
Model: "sequential"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━┓
┃ Layer (type)                         ┃ Output Shape                ┃         Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━┩
│ embedding (Embedding)                │ (None, 200, 128)            │         640,000 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ spatial_dropout1d (SpatialDropout1D) │ (None, 200, 128)            │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ lstm (LSTM)                          │ (None, 64)                  │          49,408 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense (Dense)                        │ (None, 64)                  │           4,160 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dropout (Dropout)                    │ (None, 64)                  │               0 │
├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense_1 (Dense)                      │ (None, 1)                   │              65 │
└──────────────────────────────────────┴─────────────────────────────┴─────────────────┘
 Total params: 2,080,901 (7.94 MB)
 Trainable params: 693,633 (2.65 MB)
 Non-trainable params: 0 (0.00 B)
```

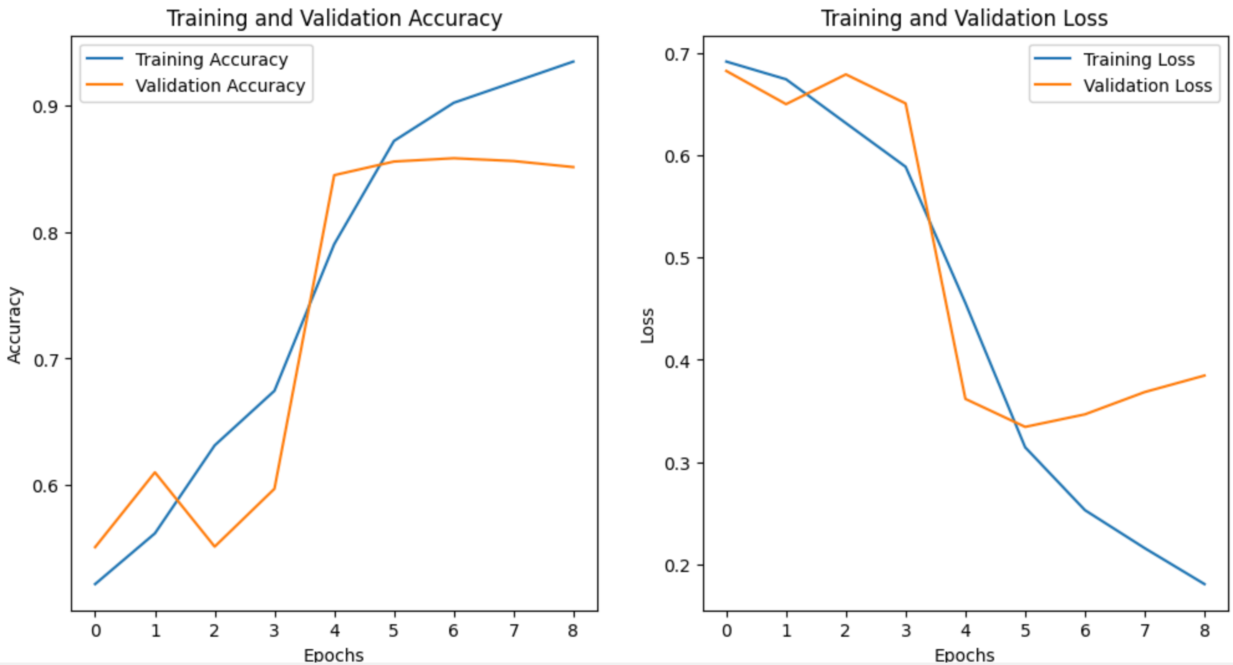
Evaluate the model

```
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.show()
```



```
y_pred = model.predict(X_test_pad)
y_pred = (y_pred > 0.5).astype(int)
```

```
157/157 ━━━━━━━━━━━━━━━━━━━━ 17s 109ms/step
```

```
print(classification_report(y_test, y_pred))
```

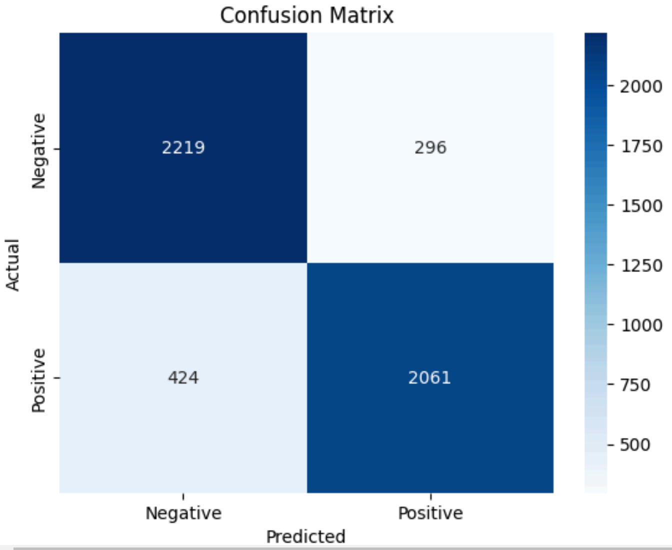
```
              precision    recall  f1-score   support

           0       0.84      0.88      0.86      2515
           1       0.87      0.83      0.85      2485

    accuracy                           0.86      5000
   macro avg       0.86      0.86      0.86      5000
weighted avg       0.86      0.86      0.86      5000
```

Confusion Matrix

```
conf_matrix = confusion_matrix(y_test, y_pred)
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Negative', 'Positive'], yticklabels=['Negative', 'Positive'])
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```


Confusion Matrix
Actual
Negative
Positive
Negative
Positive
Predicted