Project Report: Twitter Sentiment Analysis of University of Southern California.

Sentiment analysis of tweets is a growing field in natural language processing that aims to automatically classify the emotions conveyed by tweets. It involves using machine learning algorithms to analyze large volumes of tweets and categorize them based on the emotions they convey, such as positive, negative, or neutral. The accuracy of these algorithms is critical, as they can provide valuable insights into consumer behavior, political trends, and public opinion. Therefore, developing accurate models that can classify tweet sentiment in real-time has become a crucial task for many researchers and businesses.

1. Objectives:

1. Web Scraping the tweets from twitter for the University of southern California.
2. Preprocessing the tweet data by removing unnecessary characters such as punctuation and stop words.
3. Extracting the features from the preprocessed text using the bag of words approach and TF-IDF
4. Implementing a sequential neural network model using embedding layer LSTM, CNN, GRU, and dense output layers to classify the sentiment of the tweet.
5. Training the model using binary cross-entropy loss and the Adam-optimizer to get a better fit for the model.
6. Evaluating the performance of the model on the testing set using metrics and accuracy, precision, recall and F1 Score and visualize the prediction metrics using a confusion matrix.
7. Testing the model by predicting the sentiment of the sentence for the given samples.

2. Data Collection Process:

Steps involved in creating the Twitter Developer account:

Step 1: Creating a Twitter Developer Account.I have created a twitter developer account to pull the tweets for my selected university.

Step 2: After signing up to the account we should create new twitter app to get access to the api secrets and keys.

Step 3: Generate the twitter Api Key, Api Secret, Access token and access token secret.

These keys and secrets are used to connect to the twitter and fetch the tweets.

The data collection process involved using the Twitter API and the Tweepy library to scrape tweets from Twitter. I started by authenticating my credentials using the OAuthHandler provided by Tweepy. After successful authentication, I proceeded to search for tweets using specific hashtags related to the universities we chose from the given link. I collected tweets for the hashtags as per the university (["#USC", '#fighton', '#uscfootball', '#trojan', '#uscgrad', '#usctrojans',

'#universityofsoutherncalifornia', '#uscalumni']) and collected a minimum of 1500 tweets for all the hashtags.

To ensure that I have collected unique and distinct tweets, I have used a set to store the tweets and checked if a tweet was already in the set before adding it. I also specified the language of the tweets I wanted to collect, which in our case was English.

After collecting the tweets, I wrote them to a CSV file(usc_tweets.csv) with columns for the username, date, time, and tweet text. I used the CSV module to write the data to the file.

In [263]:
```
import tweepy
auth = tweepy.OAuthHandler("6R6MJUxLEpLIrkSJGiBguKMRv",
    "eyIukOMw4h9zkUJfMlcrDtJcdUXT9l9WOtgIc0c5dDOf9twdBY")
auth.set_access_token("1033566666305589248-r3tbQuBFTyv6NoMomhfYprOkSypSYD",
    "ewvQW8I3zBMlHwTpg1iDArytjcJ24RwdXRuGJwbJ69cIH")

api = tweepy.API(auth)

try:
    api.verify_credentials()
    print("Authentication is verified")
except:
    print("Authentication Failed")
```

```
Authentication is verified
```

In [115]:
```
import tweepy
import csv

hashtags = ['#southerncalifornia']

#I have used the following tags for craetion of my dataset.
#["#USC", '#fighton', '#uscfootball', '#trojan', '#uscgrad', '#usctrojans', '#universityofsoutherncalifornia', '#uscalumni']

unique_tweets = set()

for hashtag in hashtags:
    for tweet in tweepy.Cursor(api.search_tweets, q=hashtag, lang="en", tweet_mode="extended").items(400):
        if tweet not in unique_tweets:
            unique_tweets.add(tweet)

with open(f"usc_tweets.csv", "a", newline="", encoding="utf-8") as file:
    writer = csv.writer(file)
    writer.writerow(["Username", "Date", "Time", "Tweet"])
    for tweet in unique_tweets:
        writer.writerow([tweet.user.screen_name, tweet.created_at.date(), tweet.created_at.time(), tweet.full_text])
```

Figure 2.1 API Call and Extracting tweets from the twitter.

Usc_tweets.csv:

	A	B	C	D
1	Username	Date	Time	Tweet
2	gustavokralj	5/3/2023	1:10:40	For the North American Synod Report the Synod on Synodality is a â€˜Generational Projectâ€™
3	InsideUSC	5/3/2023	1:01:01	#USC playing Long Beach State at Angel Stadium tonight https://t.co/eWw3TGRHCt
4	jacki_essays	5/3/2023	0:41:13	Hmu for essays,assignments...
5	hankwasiak	5/3/2023	0:20:23	Part 3 of Sorority Dinner! Had a great time!
6	hankwasiak	5/3/2023	0:10:16	Part 2 of Sorority Dinner Invite!
7	hankwasiak	5/3/2023	0:01:08	Invite to Sorority dinner! Thank you to Annie for asking me!
8	will_camardella	5/2/2023	23:56:06	My view from the Legends Suites at Angel Stadium, where #USC will battle Long Beach State in about an hour. https://t.co/Sec7D7BvV3
9	InsideUSC	5/2/2023	23:49:48	It was a big deal when #USC reclaimed the Victory Bell in 1952 https://t.co/Y0O6IUo1Wu
10	GGreensss	5/2/2023	23:00:51	- Why SC MUST go after Keon Coleman
11	USC_Rivals	5/2/2023	22:57:55	Our Jeff McCulloch (@Rivals_Jeff) checked in with four-star Rivals100 priority #USC #FightOn OL target DeAndre Carter for the latest on his Trojans recruitment.
12	MicrotelGarners	5/2/2023	22:32:08	Is there really a city lost in the sea's depths? Catch The Lost City: Between Art and Science and other exhibitions daily at the #USC McKissick Museum!

Figure 2.2 Screenshot of the extracted tweets.

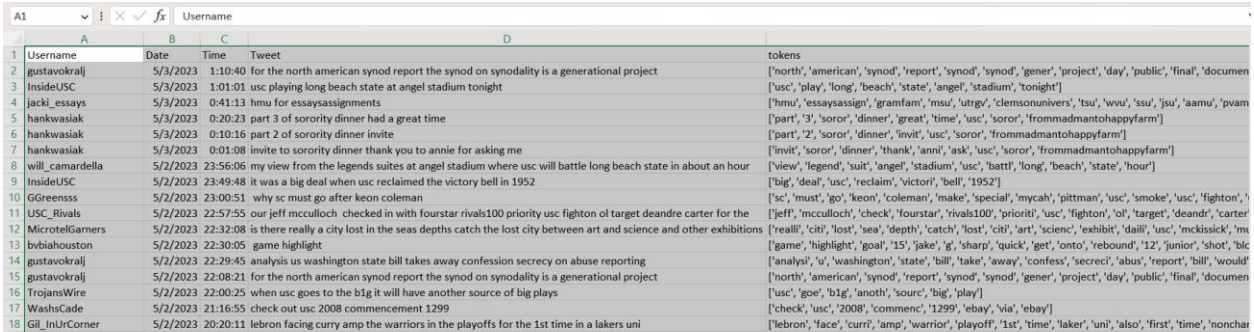
3. Preprocessing:

Preprocessing is important as it involves cleaning, transforming raw text data into a format that can be easily understood by a machine learning algorithm.

I have followed the below data preprocessing steps:

1. I have removed all the urls present in the tweets using the regular expression. This is helpful because URLs often do not provide any useful information. Additionally, URLs can sometimes be a source of noise in the data and make it more difficult to analyze and understand.
2. I have removed all the mentions from the tweet as they are used to reference other twitter users and it will not be useful for text analysis.
3. Next, I am converting all the text to lowercase to make the words uniform and consistent in their representation. By converting all the text to lowercase, we can reduce the number of unique words in the dataset, which can improve the efficiency of machine learning model.
4. I have removed the punctuation from the tweet as they donot carry specific meaning to the text analysis.
5. In the next step I am removing the stop words and performed stemming and lemmatization. Removing the stop words helps to reduce the dimensionality of the data and improve the accuracy of the analysis. Stemming and lemmatization are techniques used to reduce words to their base or root form. This is done to reduce the number of unique words in the data, which helps to improve the performance of machine learning algorithms.

Preprocessed Tweets:



	A	B	C	D	tokens
1	Username	Date	Time	Tweet	
2	gustavokralj	5/3/2023	1:10:40	for the north american synod report the synod on synodality is a generational project	['north', 'american', 'synod', 'report', 'synod', 'synod', 'gener', 'project', 'day', 'public', 'final', 'documen
3	InsideUSC	5/3/2023	1:01:01	usc playing long beach state at angel stadium tonight	['usc', 'play', 'long', 'beach', 'state', 'angel', 'stadium', 'tonight']
4	jacki_essays	5/3/2023	0:41:13	hmu for essaysassignments	['hmu', 'essaysassign', 'gramfam', 'msu', 'utrgv', 'clemsonunivers', 'tsu', 'wvu', 'ssu', 'jsu', 'aamu', 'pvam
5	hankwasiak	5/3/2023	0:20:23	part 3 of sorority dinner had a great time	['part', '3', 'soror', 'dinner', 'great', 'time', 'usc', 'soror', 'frommadmantohappyfarm']
6	hankwasiak	5/3/2023	0:10:16	part 2 of sorority dinner invite	['part', '2', 'soror', 'dinner', 'invit', 'usc', 'soror', 'frommadmantohappyfarm']
7	hankwasiak	5/3/2023	0:01:08	invite to sorority dinner thank you to annie for asking me	['invit', 'soror', 'dinner', 'thank', 'anni', 'ask', 'usc', 'soror', 'frommadmantohappyfarm']
8	will_camardella	5/2/2023	23:56:06	my view from the legends suites at angel stadium where usc will battle long beach state in about an hour	['view', 'legend', 'suit', 'angel', 'stadium', 'usc', 'battl', 'long', 'beach', 'state', 'hour']
9	InsideUSC	5/2/2023	23:49:48	it was a big deal when usc reclaimed the victory bell in 1952	['big', 'deal', 'usc', 'reclaim', 'victori', 'bell', '1952']
10	GGreensss	5/2/2023	23:00:51	why sc must go after keon coleman	['sc', 'must', 'go', 'keon', 'coleman', 'make', 'special', 'mycah', 'pittman', 'usc', 'smoke', 'usc', 'fighton', '
11	USC_Rivals	5/2/2023	22:57:55	our jeff mcculloch checked in with fourstar rivals100 priority usc fighton ol target deandre carter for the	['jeff', 'mcculloch', 'check', 'fourstar', 'rivals100', 'prioriti', 'usc', 'fighton', 'ol', 'target', 'deandr', 'carter
12	MicrotelGarners	5/2/2023	22:32:08	is there really a city lost in the seas depths catch the lost city between art and science and other exhibitions	['reallî', 'citi', 'lost', 'sea', 'depth', 'catch', 'lost', 'citi', 'art', 'scienc', 'exhibit', 'dailî', 'usc', 'mckissick', 'mu
13	bvbiahouston	5/2/2023	22:30:05	game highlight	['game', 'highlight', 'goal', '15', 'jake', 'g', 'sharp', 'quick', 'get', 'onto', 'rebound', '12', 'junior', 'shot', 'blo
14	gustavokralj	5/2/2023	22:29:45	analysis us washington state bill takes away confession secrecy on abuse reporting	['analysi', 'u', 'washington', 'state', 'bill', 'take', 'away', 'confess', 'secreci', 'abus', 'report', 'bill', 'would'
15	gustavokralj	5/2/2023	22:08:21	for the north american synod report the synod on synodality is a generational project	['north', 'american', 'synod', 'report', 'synod', 'synod', 'gener', 'project', 'day', 'public', 'final', 'documen
16	TrojansWire	5/2/2023	22:00:25	when usc goes to the b1g it will have another source of big plays	['usc', 'goe', 'b1g', 'anoth', 'sourc', 'big', 'play']
17	WashsCade	5/2/2023	21:16:55	check out usc 2008 commencement 1299	['check', 'usc', '2008', 'commenc', '1299', 'ebay', 'via', 'ebay']
18	Gil_InUrCorner	5/2/2023	20:20:11	lebron facing curry amp the warriors in the playoffs for the 1st time in a lakers uni	['lebron', 'face', 'curri', 'amp', 'warrior', 'playoff', '1st', 'time', 'laker', 'uni', 'also', 'first', 'time', 'noncha
19	PaddleHistory	5/2/2023	20:00:05	university of southern california losangeles california fighton usc trojans	['univers', 'southern', 'california', 'losangel', 'california', 'fighton', 'usc', 'trojan']
20	Dallasfanindc1	5/2/2023	19:56:14	watch michigan take this l live stream michigan	['watch', 'michigan', 'take', 'l', 'live', 'stream', 'michigan', 'goblu', 'ohiost', 'buckey', 'usc', 'lsu', 'freeman
21	On3USC	5/2/2023	19:33:12	impact of 5star dl aydin brelands first usc visit since november	['impact', '5star', 'dl', 'aydin', 'breland', 'first', 'usc', 'visit', 'sinc', 'novemb', 'via', 'fighton']

Figure 3.1 Screenshot of the preprocessed tweets

Feature Extraction:

The code extracts bag-of-words and TF-IDF features from preprocessed tweet data using CountVectorizer and TfidfVectorizer from scikit-learn. Bag-of-words is a simple text representation method that involves counting the frequency of each word in a document, while TF-IDF is a more advanced method that considers the importance of each word in a document and in a corpus of documents. The bag-of-words feature matrix contains the count of each word in each

tweet, while the TF-IDF feature matrix contains the weighted frequency of each word in each tweet, taking into account the importance of the word in the corpus.

Sentiment Analysis using Transformer-based model:

First, the preprocessed data is loaded into a Pandas DataFrame from a CSV file. The text data from the 'Tweet' column is extracted and stored in a list.

The sentiment analysis model is loaded using the hugging face transformers library. The specific model used is 'distilbert-base-uncased-finetuned-sst-2-english', which is pre-trained on a sentiment analysis task on the Stanford Sentiment Treebank dataset. The tokenizer for the model is also loaded.

A pipeline for sentiment analysis is created using the loaded model and tokenizer. The pipeline is then used to perform sentiment analysis on the list of tweets. The resulting sentiments are stored in a list.

Finally, the predicted sentiments are added as a new column to the original DataFrame and saved back to a CSV file.

Sample file after sentiment Analysis:

	A	B	C	D	E	F
1	Username	Date	Time	Tweet	tokens	sentiment
2	gustavokralj	5/3/2023	1:10:40	for the north american synod report the synod on synodality is a	['north', 'american', 'synod', 'report', 'synod', 'synod', 'gener', 'project', 'd	NEGATIVE
3	InsideUSC	5/3/2023	1:01:01	usc playing long beach state at angel stadium tonight	['usc', 'play', 'long', 'beach', 'state', 'angel', 'stadium', 'tonight']	POSITIVE
4	jacki_essays	5/3/2023	0:41:13	hmu for essaysassignments	['hmu', 'essaysassign', 'gramfam', 'msu', 'utrgv', 'clemsonunivers', 'tsu', 'w	NEGATIVE
5	hankwasiak	5/3/2023	0:20:23	part 3 of sorority dinner had a great time	['part', '3', 'soror', 'dinner', 'great', 'time', 'usc', 'soror', 'frommadmantoh	POSITIVE
6	hankwasiak	5/3/2023	0:10:16	part 2 of sorority dinner invite	['part', '2', 'soror', 'dinner', 'invit', 'usc', 'soror', 'frommadmantohappyfar	POSITIVE
7	hankwasiak	5/3/2023	0:01:08	invite to sorority dinner thank you to annie for asking me	['invit', 'soror', 'dinner', 'thank', 'anni', 'ask', 'usc', 'soror', 'frommadmantc	POSITIVE
8	will_camardella	5/2/2023	23:56:06	my view from the legends suites at angel stadium where usc will batt	['view', 'legend', 'suit', 'angel', 'stadium', 'usc', 'battl', 'long', 'beach', 'state	POSITIVE
9	InsideUSC	5/2/2023	23:49:48	it was a big deal when usc reclaimed the victory bell in 1952	['big', 'deal', 'usc', 'reclaim', 'victori', 'bell', '1952']	POSITIVE
10	GGreensss	5/2/2023	23:00:51	why sc must go after keon coleman	['sc', 'must', 'go', 'keon', 'coleman', 'make', 'special', 'mycah', 'pittman', 'u	NEGATIVE
11	USC_Rivals	5/2/2023	22:57:55	our jeff mcculloch checked in with fourstar rivals100 priority usc	['jeff', 'mcculloch', 'check', 'fourstar', 'rivals100', 'prioriti', 'usc', 'fighton',	POSITIVE
12	MicrotelGarners	5/2/2023	22:32:08	is there really a city lost in the seas depths catch the lost city	['realli', 'citi', 'lost', 'sea', 'depth', 'catch', 'lost', 'citi', 'art', 'scienc', 'exhibit	POSITIVE
13	bvbiahouston	5/2/2023	22:30:05	game highlight	['game', 'highlight', 'goal', '15', 'jake', 'g', 'sharp', 'quick', 'get', 'onto', 'rebc	POSITIVE
14	gustavokralj	5/2/2023	22:29:45	analysis us washington state bill takes away confession secrecy on	['analysi', 'u', 'washington', 'state', 'bill', 'take', 'away', 'confess', 'secreci',	NEGATIVE
15	gustavokralj	5/2/2023	22:08:21	for the north american synod report the synod on synodality is a	['north', 'american', 'synod', 'report', 'synod', 'synod', 'gener', 'project', 'd	NEGATIVE
16	TrojansWire	5/2/2023	22:00:25	when usc goes to the b1g it will have another source of big plays	['usc', 'goe', 'b1g', 'anoth', 'sourc', 'big', 'play']	POSITIVE
17	WashsCade	5/2/2023	21:16:55	check out usc 2008 commencement 1299	['check', 'usc', '2008', 'commenc', '1299', 'ebay', 'via', 'ebay']	NEGATIVE
18	Gil_InUrCorner	5/2/2023	20:20:11	lebron facing curry amp the warriors in the playoffs for the 1st time	['lebron', 'face', 'curri', 'amp', 'warrior', 'playoff', '1st', 'time', 'laker', 'uni',	POSITIVE
19	PaddleHistory	5/2/2023	20:00:05	university of southern california losangeles california fighton usc troj	['univers', 'southern', 'california', 'losangel', 'california', 'fighton', 'usc', 'tr	NEGATIVE
20	Dallasfanindc1	5/2/2023	19:56:14	watch michigan take this l live stream michigan	['watch', 'michigan', 'take', 'l', 'live', 'stream', 'michigan', 'goblu', 'ohiost',	POSITIVE
21	On3USC	5/2/2023	19:33:12	impact of 5star dl aydin brelands first usc visit since november	['impact', '5star', 'dl', 'aydin', 'breland', 'first', 'usc', 'visit', 'sinc', 'novemb',	POSITIVE
22	gustavokralj	5/2/2023	19:29:19	analysis us washington state bill takes away confession secrecy on	['analysi', 'u', 'washington', 'state', 'bill', 'take', 'away', 'confess', 'secreci',	NEGATIVE
23	USCCenter4PR	5/2/2023	19:20:38	the 2023 global communication report new reputation incorporates	['2023', 'global', 'commun', 'report', 'new', 'reput', 'incorpor', 'theme', 'ex	POSITIVE
24	Ale_LtdEdition	5/2/2023	19:17:50	i can pleasantly say that usc wireless connection sucks	['pleasantli', 'say', 'usc', 'wireless', 'connect', 'suck']	NEGATIVE
25	gustavokralj	5/2/2023	19:08:06	for the north american synod report the synod on synodality is a	['north', 'american', 'synod', 'report', 'synod', 'synod', 'gener', 'project', 'd	NEGATIVE
26	IESportsRadio	5/2/2023	19:01:04	tune in live for setpoint with	['tune', 'live', 'setpoint', 'ncaavolleybal', 'ncaamvb', 'hawaii', 'ucla', 'ohios	NEGATIVE

Figure 3.2: Screenshot after sentiment analysis

4. Model Architecture:

- Coming to the model architecture code, I am training the sentiment analysis model using a sequential neural network architecture in Keras.
- The preprocessed tweet data is loaded from a CSV file and transformed using a tokenizer to convert each tweet into a sequence of integer tokens.
- The sequences are then padded to ensure they all have the same length, which is required by the neural network.
- The resulting padded sequences and corresponding sentiment labels are split into training and testing sets.
- The neural network consists of an embedding layer, two recurrent layers (LSTM and GRU), a convolutional layer, and two dense layers with dropout regularization.
- The output layer uses a sigmoid activation function to produce a binary sentiment classification.
- The model is compiled with the binary cross-entropy loss function, Adam optimizer with a learning rate of 0.001, and accuracy metric.
- The model is trained for 25 epochs using a batch size of 64 and evaluated on the validation set after each epoch. The model summary is displayed at the end of the training.

5. Results and evaluation metrics

The classification report is a summary of the performance of a machine learning model that is used for binary or multi-class classification.

In the classification report provided, we have two classes, 0 and 1, which represent the negative and positive sentiments in the given dataset. The precision score measures how many of the predicted positive labels are actually positive, while recall measures how many of the true positive labels are correctly predicted by the model. F1-score is the harmonic mean of precision and recall, and it is a useful metric for assessing the overall performance of the model. Support indicates the number of samples in each class.

From the report, we can see that the model has an overall accuracy of 82%, which is the percentage of correctly predicted labels. The precision score for class 0 (negative sentiment) is 0.85, which means that 85% of the predicted negative labels are actually negative, and the recall score for class 0 is 0.79, which means that 79% of the true negative labels are correctly predicted by the model. Similarly, the precision and recall scores for class 1 (positive sentiment) are 0.80 and 0.85, respectively.

The macro-avg F1-score is the average F1-score across all classes, and the weighted-avg F1-score is the weighted average of the F1-score across all classes, weighted by the number of samples in each class. In this case, both macro-avg and weighted-avg F1-scores are 0.82, which indicates that the model has performed equally well in both classes.

```
In [32]: print("Classification report: ")
         print(classification_report(y_test, y_pred))
```

```
Classification report:
              precision    recall  f1-score   support

           0       0.85      0.79      0.82       183
           1       0.80      0.85      0.82       178

    accuracy                           0.82       361
   macro avg       0.82      0.82      0.82       361
weighted avg       0.82      0.82      0.82       361
```

Figure 5.1 Classification Matrix

Confusion Matrix:

The confusion matrix is used to evaluate the performance of a classification model by comparing the actual values of the target variable with the predicted values.

The confusion matrix is computed using the confusion_matrix() function from scikit-learn. The resulting matrix is then visualized as a heatmap using the sns.heatmap() function from the Seaborn library.

The significance of the confusion matrix provides a visual representation of how well the model is performing in terms of predicting the sentiment of tweets. It allows us to see how many tweets were classified correctly (true positives and true negatives) and how many were misclassified (false positives and false negatives).

```
# Compute the confusion matrix
cm = confusion_matrix(y_test, y_pred)

# Plot the confusion matrix using a heatmap
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d", cbar=False, annot_kws={"size": 16},
            xticklabels=label_encoder.classes_, yticklabels=label_encoder.classes_)
plt.xlabel('Predicted', fontsize=14)
plt.ylabel('Actual', fontsize=14)
plt.show()
```

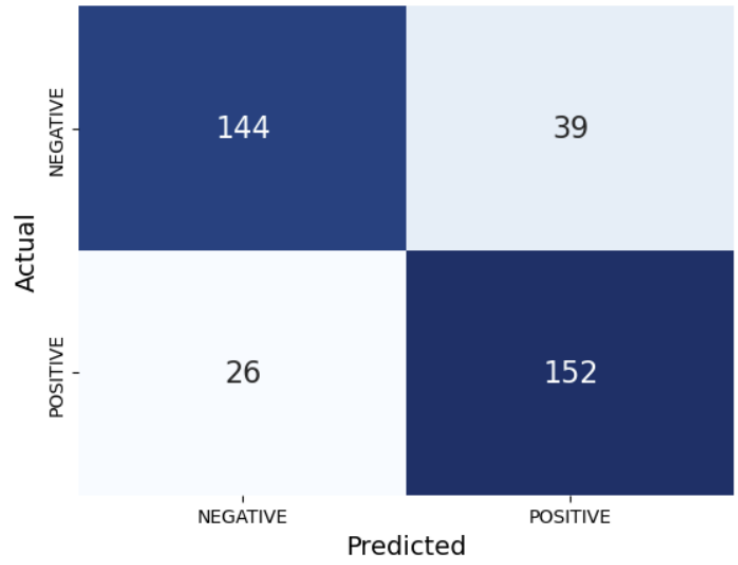


Figure 5.2 Confusion Matrix

Model Accuracy:

The first plot shows the training and validation accuracy over the epochs, which can be used to evaluate the performance of the model during training. We cans see that as the training data is increasing the validation accuracy is also increasing.

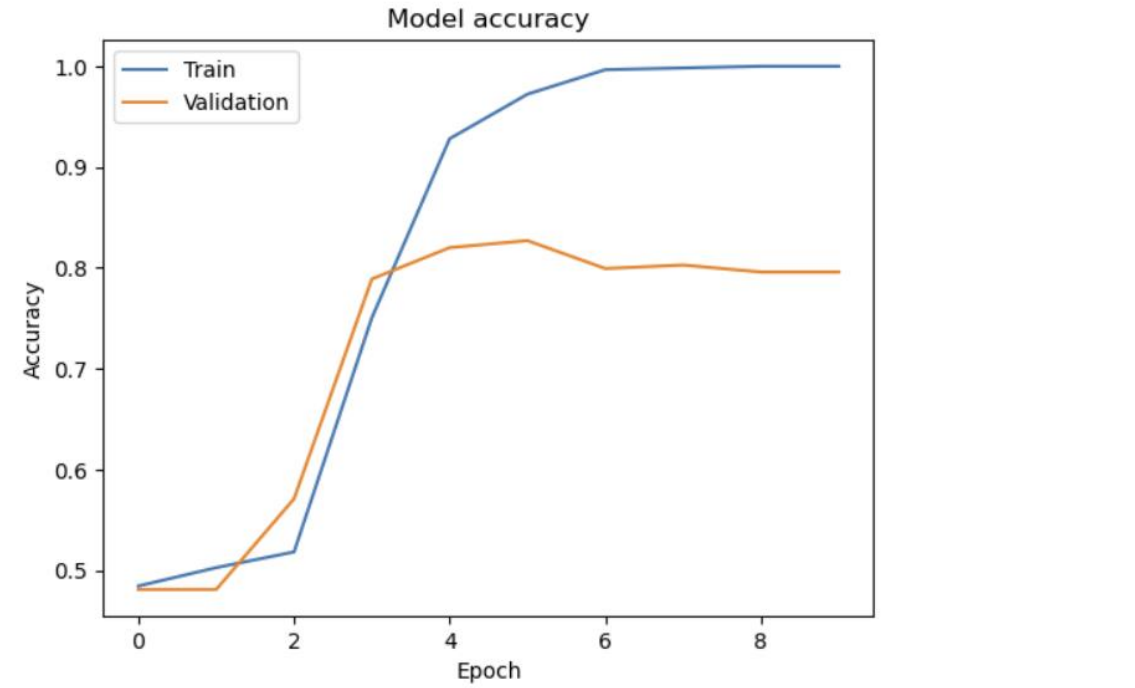


Figure 5.3: Model Accuracy

Using a Trained Model for Sentiment Analysis on New Sentences:

For testing I am taking a list of sentences and using the trained sentiment analysis model to predict the sentiment of each sentence. The predicted sentiment can be either "Positive", "Negative", or "Neutral". For each sentence in the list, the code prints the sentence and the predicted sentiment as output. The Model predicted almost all the sentences correctly.

```
1/1 [==============================] - 0s 79ms/step
Sentence:  I love this movie
Predicted Sentiment:  Positive

Sentence:  The food at the restaurant was horrible and gave me bad food poisoning.
Predicted Sentiment:  Neutral

Sentence:  The food at the restaurant was amazing
Predicted Sentiment:  Positive

Sentence:  I can't believe how bad the service was
Predicted Sentiment:  Neutral

Sentence:  This book is so amazing
Predicted Sentiment:  Negative

Sentence:  The concert was incredible
Predicted Sentiment:  Negative

Sentence:  I'm feeling really exhausted
Predicted Sentiment:  Neutral

Sentence:  The traffic jam ruined my day
Predicted Sentiment:  Negative

Sentence:  The party was excellent yesterday
Predicted Sentiment:  Positive

Sentence:  I had a terrible experience at the theme park
Predicted Sentiment:  Neutral

Sentence:  The weather today is irritating to spend time outside.
Predicted Sentiment:  Negative

Sentence:  The sky is blue and the grass is green.
Predicted Sentiment:  Neutral
```

Figure 5.4: Sentiment Analysis on new sentences

6. Limitations

One limitation of the model is that it may be computationally expensive. Another limitation is that the model may not be able to handle variations or noise in the data. If the data contains outliers, errors, or other types of noise, the model may not be able to accurately classify the data. But in this case, it is not possible because we have already preprocessed the data and removed the noise. This is particularly relevant in real-world scenarios where data may not always be clean and consistent.

One potential solution to address the limitations of the model is to use a larger and more diverse dataset. A larger dataset would provide the model with more examples to learn from, which could help it to generalize better to new examples. Additionally, a more diverse dataset could expose the model to a wider range of language styles and contexts, which would also help it to perform better on a wider range of inputs. Another potential solution is to fine-tune the model on domain-specific data. For example, if the model is being used to classify sentiment in customer reviews, fine-tuning the model on a dataset of customer reviews from the specific industry or domain of interest could help to improve its performance on that specific task.

7. Conclusion

In Summary I have developed a sentiment analysis model in this project, and I am able to successfully predict sentiment labels for a given text dataset with decent accuracy. In this project I have performed various preprocessing steps and the model implementation involved various machine learning techniques such as tokenization, padding, and the use of deep learning models like LSTM, Conv1D, MaxPooling1D, and GRU.s. However, the model had some limitations such as the model is computationally expensive. Overall, this sentiment analysis project demonstrates the potential of machine learning models in analyzing text data and provides insights into the importance of continuous improvement and adaptation in natural language processing tasks.