NAME :- MANALI YASHWANT BHASKAR.

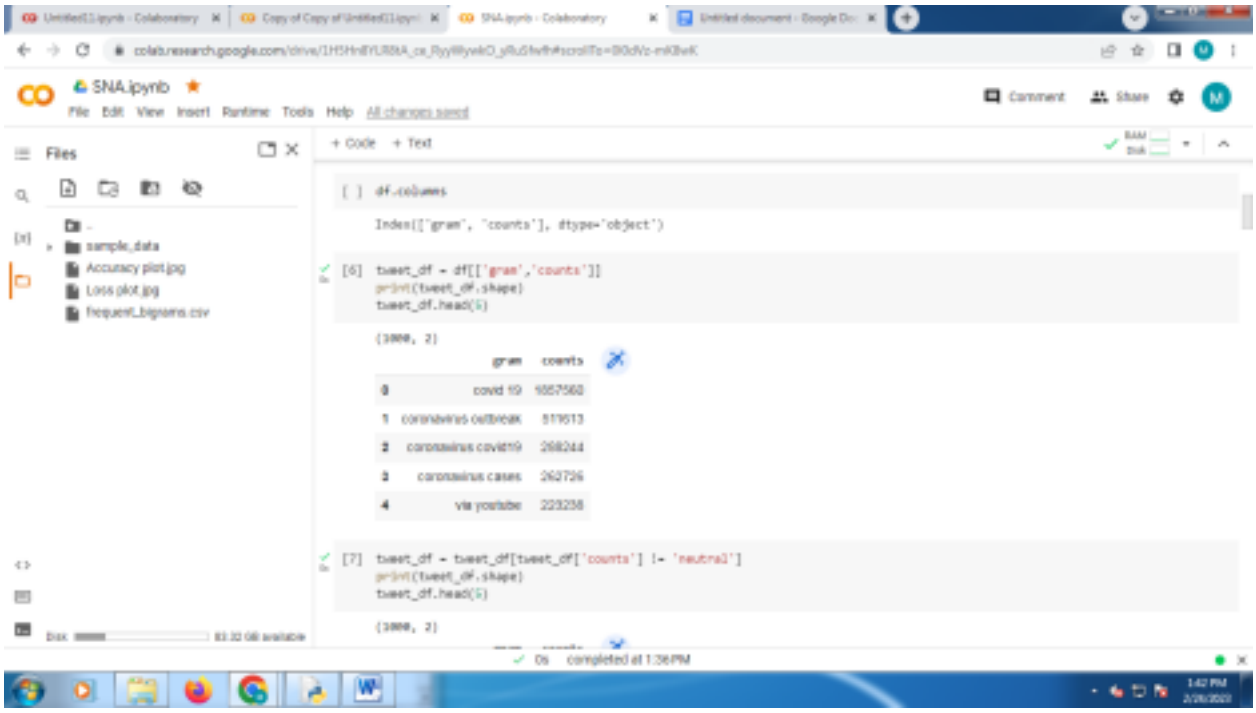
SUBJECT :- SNA

Introduction:-

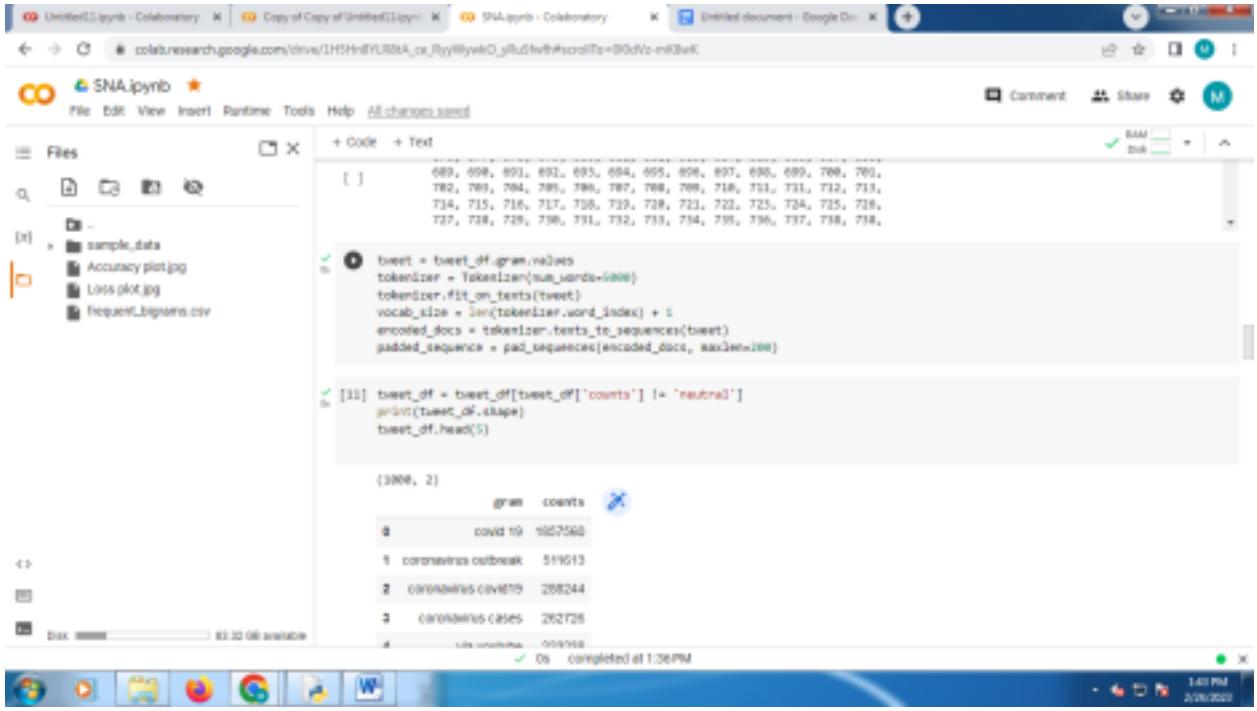
We can clearly see that sentiment analysis is becoming more popular as e-commerce, SaaS solutions, and digital technologies advance. We'll go through how this works and look at some of the most common corporate applications. We'll also discuss the analysis' existing issues and limitations.

Sentiment analysis examines how a text expresses emotion. Customer feedback, survey replies, and product reviews are all frequent uses. This can be useful in various situations, including social media monitoring, reputation management, and customer service. For example, Analyzing thousands of product reviews might provide important feedback on pricing and product features.

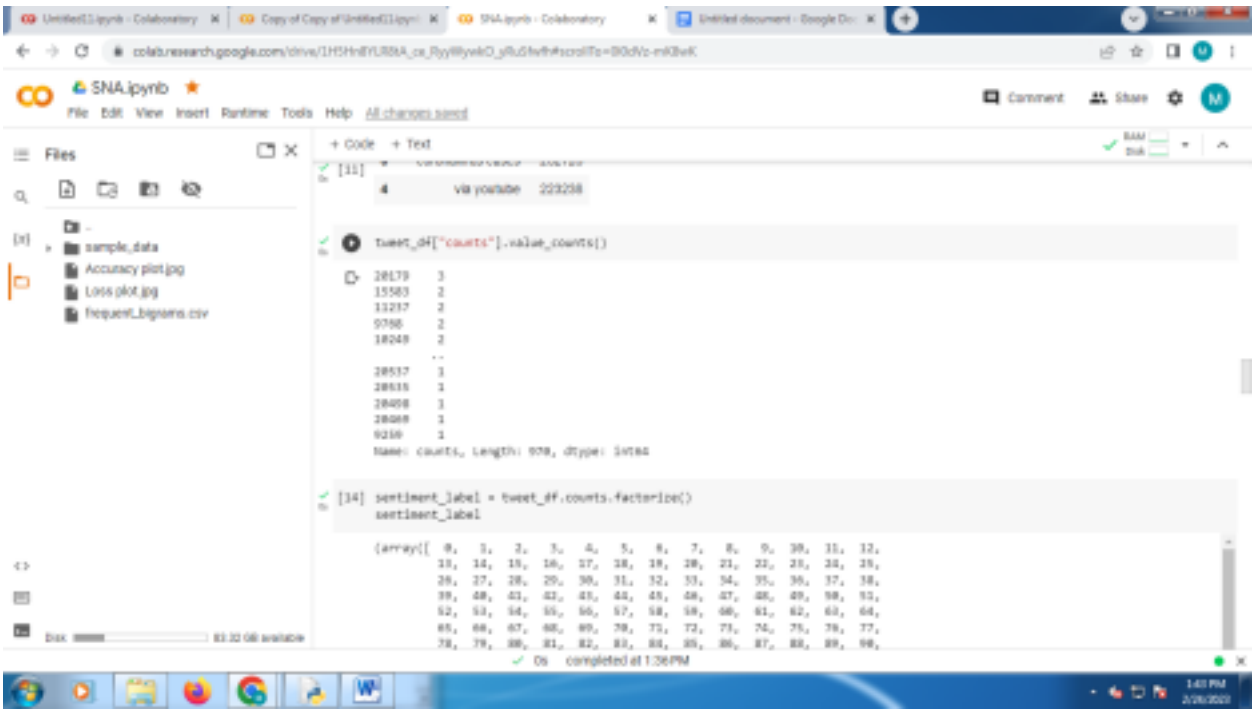
STEP :- 1


SNA.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
+ Code + Text
sample_data
Accuracy plot.jpg
Loss plot.jpg
frequent_bigrams.csv
[ ] df.columns
Index(['gram', 'counts'], dtype='object')
[6] tweet_df = df[['gram','counts']]
print(tweet_df.shape)
tweet_df.head(5)
(1000, 2)
gram counts
0 covid 19 1057560
1 coronavirus outbreak 519513
2 coronavirus covid19 268244
3 coronavirus cases 262726
4 via youtube 223258
[7] tweet_df = tweet_df[tweet_df['counts'] != 'neutral']
print(tweet_df.shape)
tweet_df.head(5)
(1000, 2)
0s completed at 1:36 PM

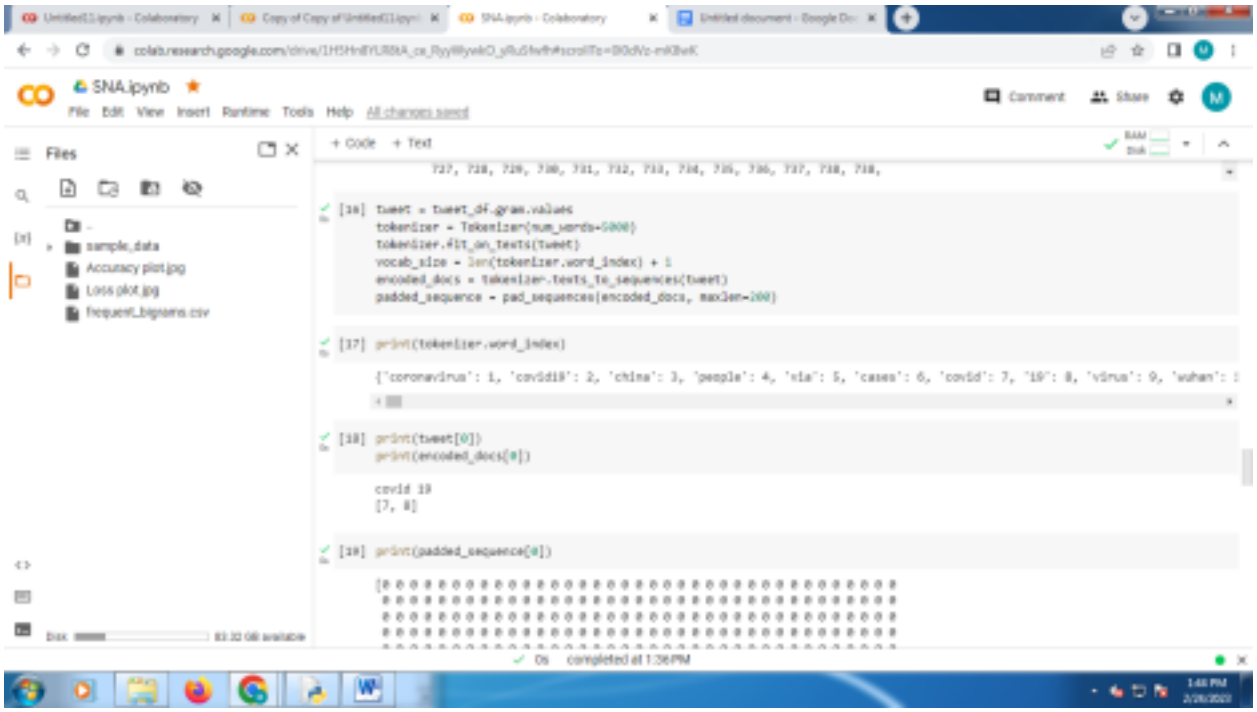
STEP :- 2


SNA.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
+ Code + Text
sample_data
Accuracy plot.jpg
Loss plot.jpg
frequent_bigrams.csv
[ ] 689, 690, 691, 692, 695, 694, 695, 696, 697, 698, 699, 700, 701,
702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 731, 712, 713,
714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726,
727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 738,
tweet = tweet_df.gram.values
tokenizer = Tokenizer(num_words=5000)
tokenizer.fit_on_texts(tweet)
vocab_size = len(tokenizer.word_index) + 1
encoded_docs = tokenizer.texts_to_sequences(tweet)
padded_sequence = pad_sequences(encoded_docs, maxlen=200)
[11] tweet_df = tweet_df[tweet_df['counts'] != 'neutral']
print(tweet_df.shape)
tweet_df.head(5)
(1000, 2)
gram counts
0 covid 19 1057560
1 coronavirus outbreak 519513
2 coronavirus covid19 268244
3 coronavirus cases 262726
0s completed at 1:36 PM

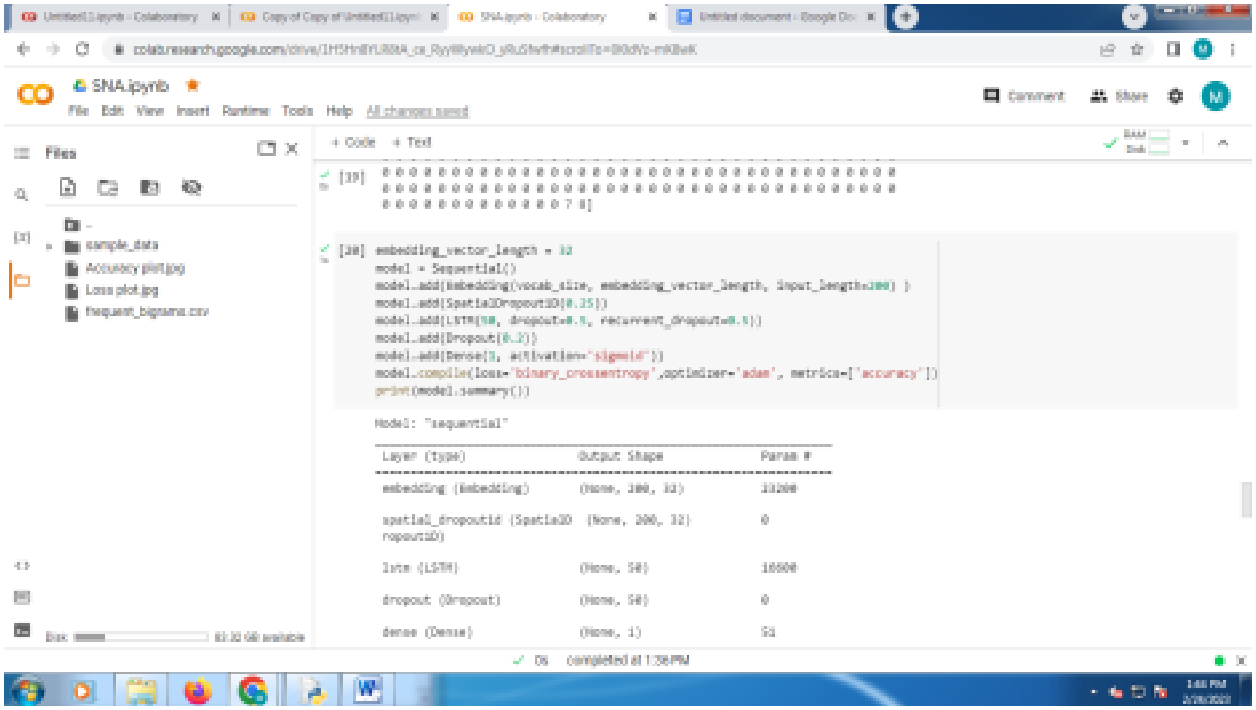
STEP :- 3


SNA.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
+ Code + Text
sample_data
Accuracy plot.jpg
Loss plot.jpg
frequent_bigrams.csv
tweet_df["counts"].value_counts()
sentiment_label = tweet_df.counts.factorize()
sentiment_label

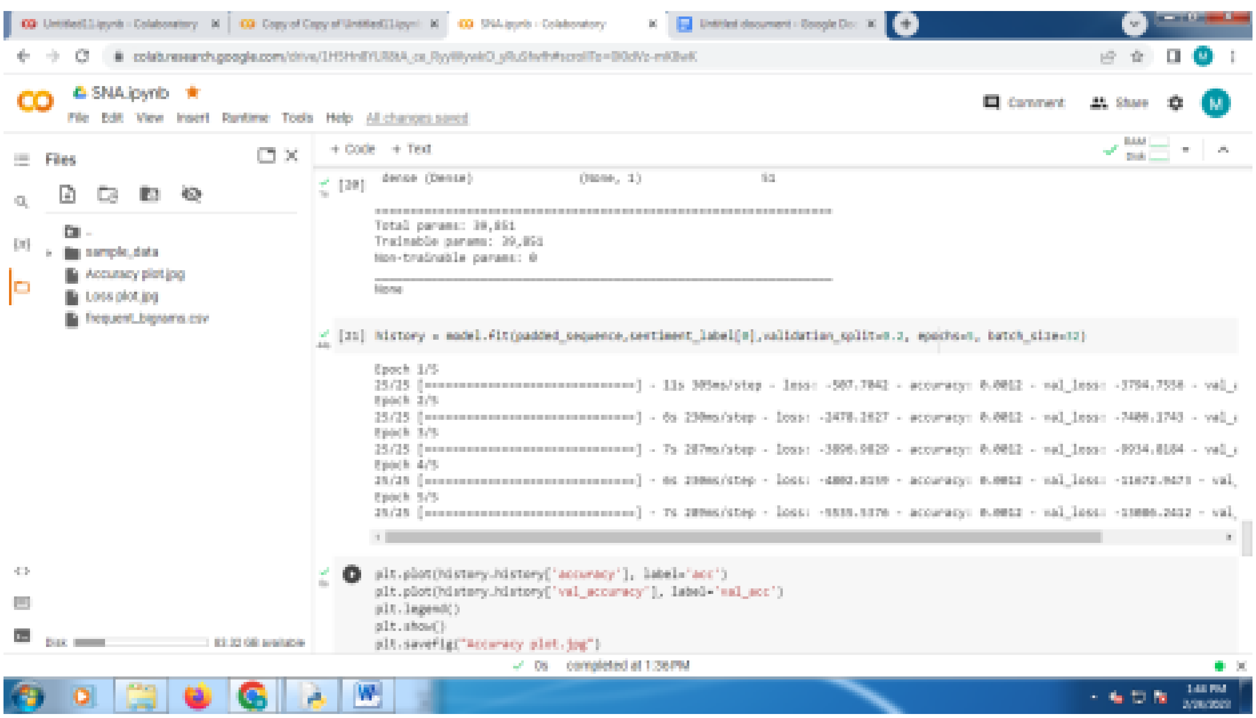
STEP :- 4


SNA.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
+ Code + Text
sample_data
Accuracy plot.jpg
Loss plot.jpg
frequent_bigrams.csv
tweet = tweet_df.grams.values
tokenizer = Tokenizer(num_words=5000)
tokenizer.fit_on_texts(tweet)
vocab_size = len(tokenizer.word_index) + 1
encoded_docs = tokenizer.texts_to_sequences(tweet)
padded_sequence = pad_sequences(encoded_docs, maxlen=200)
print(tokenizer.word_index)
print(tweet[0])
print(encoded_docs[0])
print(padded_sequence[0])

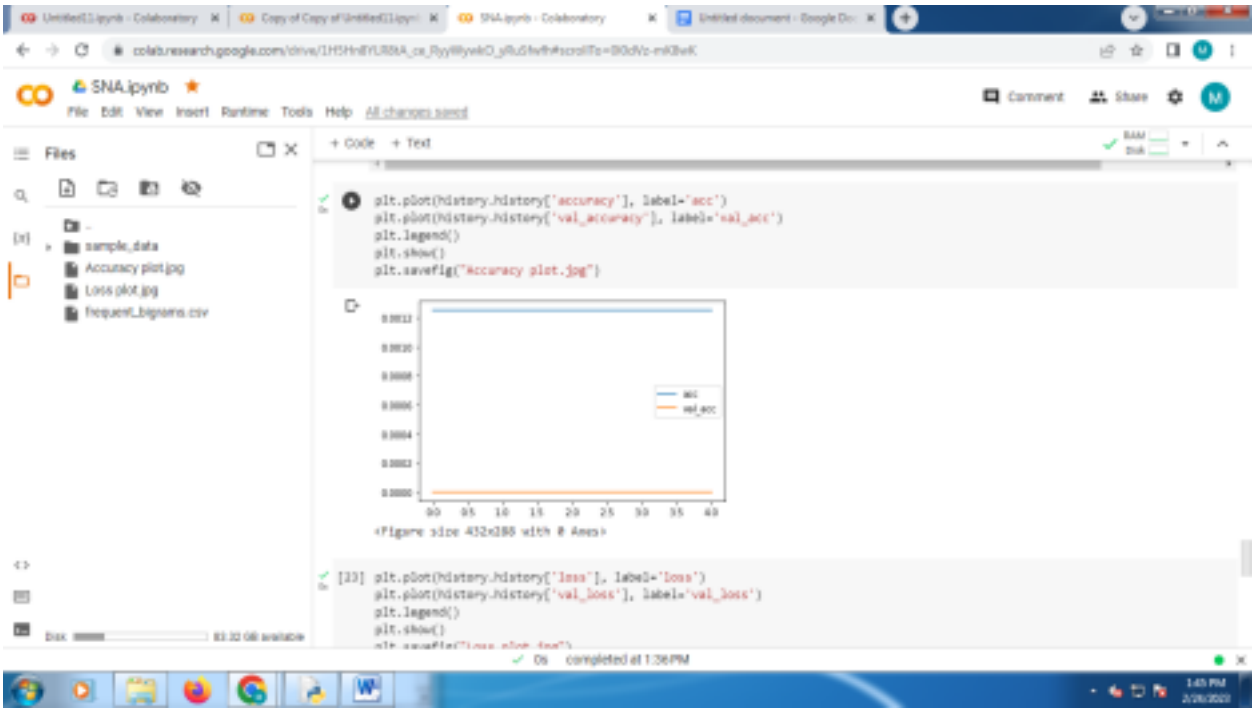
STEP :- 5



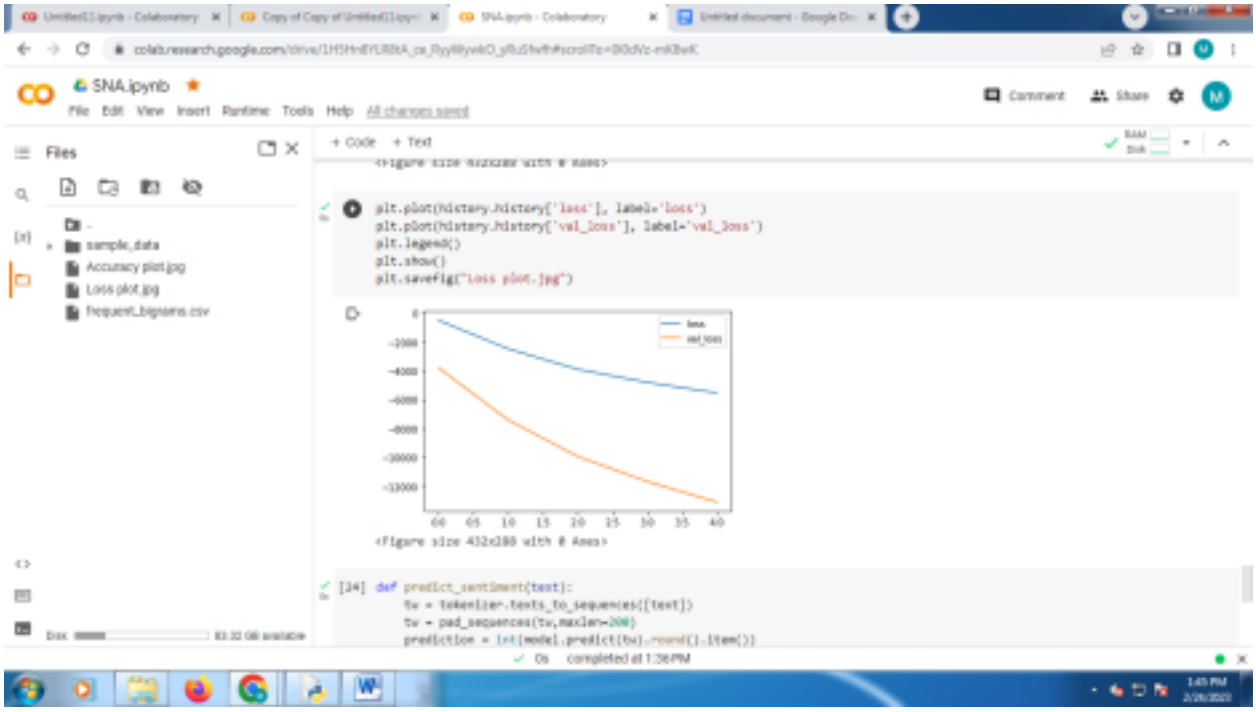
STEP :- 6



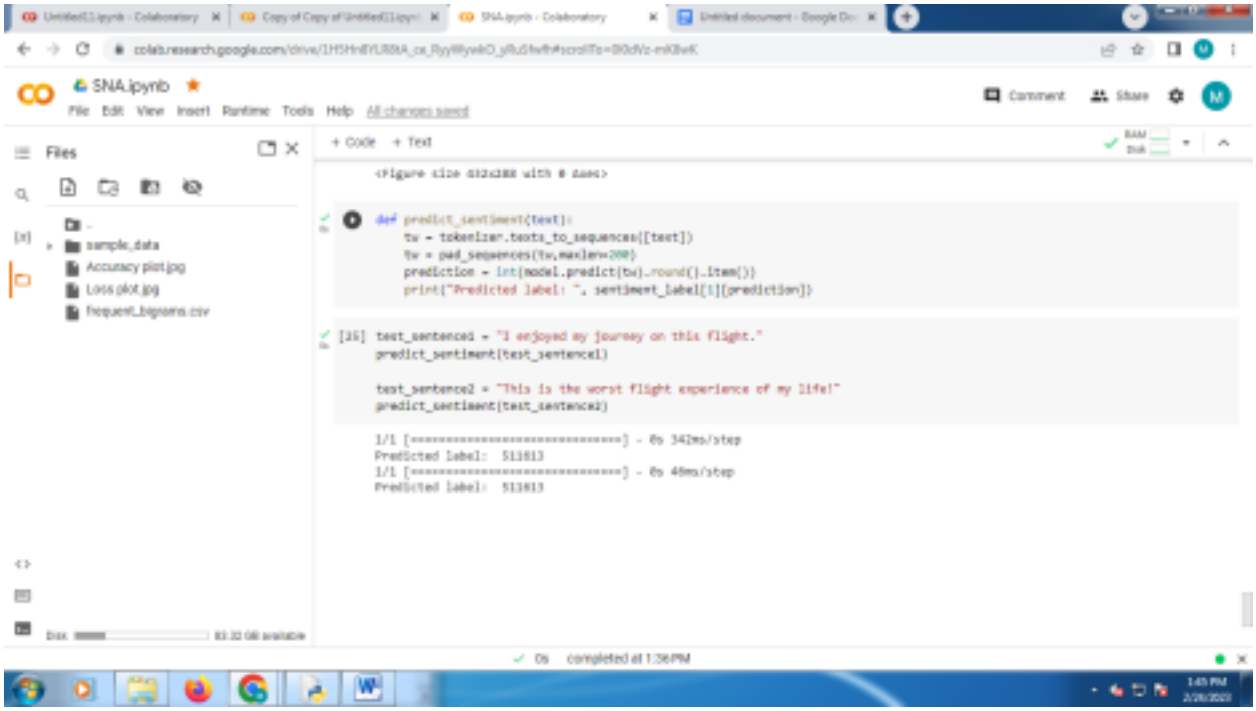
STEP :- 7


SNA.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
+ Code + Text
sample_data
Accuracy plot.jpg
Loss plot.jpg
frequent_bigrams.csv
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 432x288 with 0 Axes>
[13] plt.plot(history.history['loss'], label='loss')
plt.plot(history.history['val_loss'], label='val_loss')
plt.legend()
plt.show()

STEP :- 8


SNA.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
Files
+ Code + Text
sample_data
Accuracy plot.jpg
Loss plot.jpg
frequent_bigrams.csv
plt.plot(history.history['loss'], label='loss')
plt.plot(history.history['val_loss'], label='val_loss')
plt.legend()
plt.show()
plt.savefig("Loss plot.jpg")
<Figure size 432x288 with 0 Axes>
[14] def predict_sentiment(text):
tw = tokenizer.texts_to_sequences([text])
tw = pad_sequences(tw,maxlen=200)

STEP :- 9


SNA.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
sample_data
Accuracy plot.jpg
Loss plot.jpg
frequent_bigrams.csv
+ Code + Text
<Figure size 612x288 with 0 Axes>
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])
[26] 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 342ms/step
Predicted label: 511613
1/1 [==============================] - 0s 40ms/step
Predicted label: 511613
Disk 83.22 GB available
0s completed at 1:36 PM