Assignment no: 07

practicedsassignment7.ipynb - C

colab.research.google.com/drive/1Ce2hwYlU3fp7v6l45VuzBGi_DGw90ZzN

practicedsassignment7.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

Connected Editing

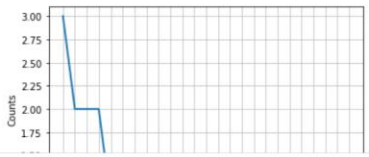
```
[ ] from nltk.probability import FreqDist
    fdist = FreqDist(tokenized_word)
    print(fdist)
```

```
<FreqDist with 25 samples and 30 outcomes>
```

```
[ ] fdist.most_common(2)
```

```
[('is', 3), (',', 2)]
```

```
import matplotlib.pyplot as plt
fdist.plot(30,cumulative=False)
plt.show()
```



Type here to search

40°C Partly sunny ENG 09:23 29-04-2022

practicedsassignment7.ipynb - C

colab.research.google.com/drive/1Ce2hwYlU3fp7v6l45VuzBGi_DGw90ZzN

practicedsassignment7.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

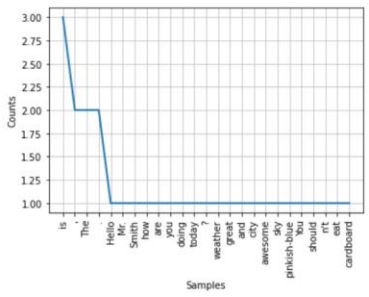
+ Code + Text

RAM Disk Editing

```
[ ] fdist.most_common(2)
```

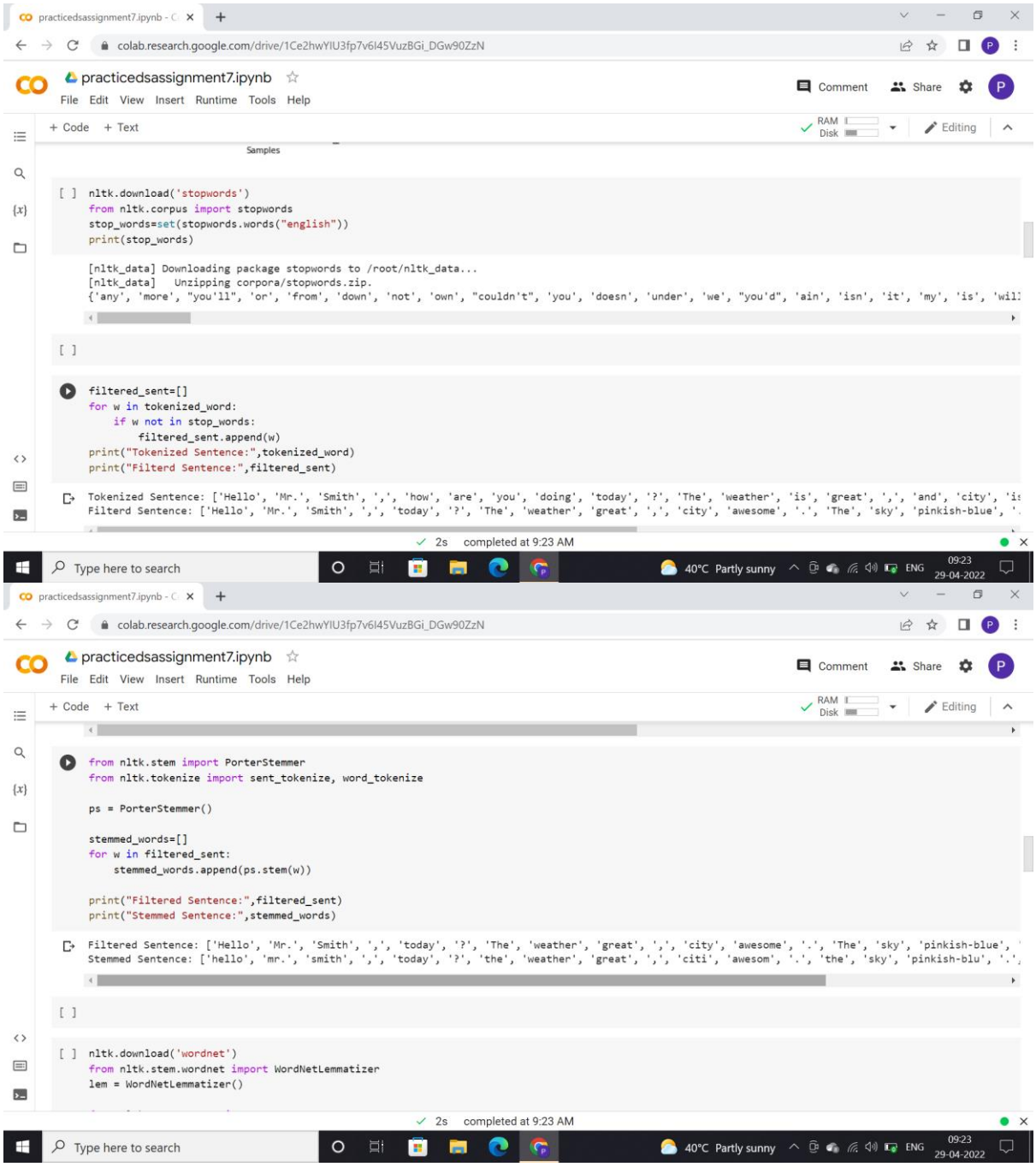
```
[('is', 3), (',', 2)]
```

```
import matplotlib.pyplot as plt
fdist.plot(30,cumulative=False)
plt.show()
```

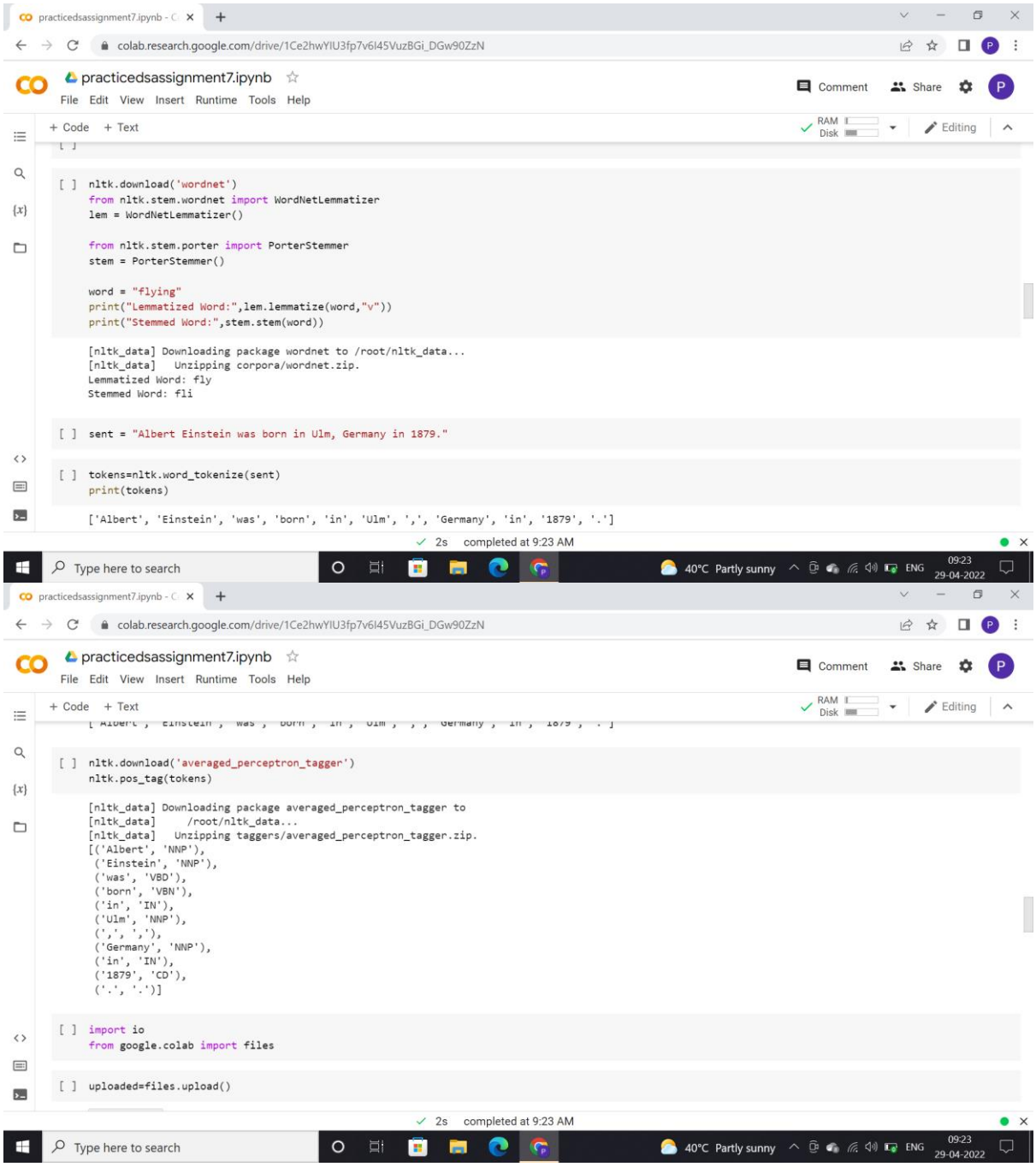


Type here to search

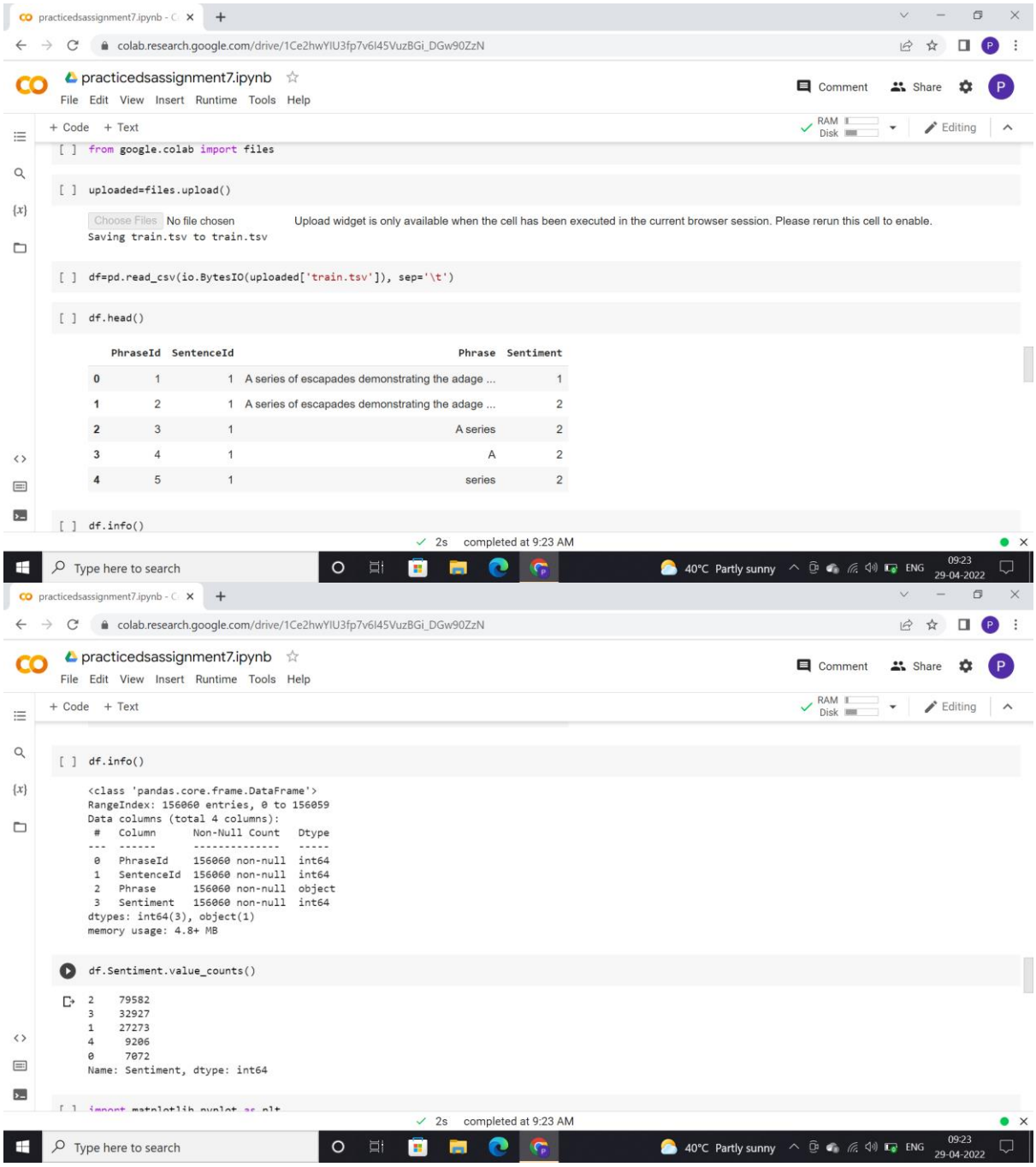
40°C Partly sunny ENG 09:23 29-04-2022


practicedsassignment7.ipynb
colab.research.google.com/drive/1Ce2hwYIU3fp7v6l45VuzBGi_DGw90ZzN
practicedsassignment7.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing
Samples
[ ] nltk.download('stopwords')
from nltk.corpus import stopwords
stop_words=set(stopwords.words("english"))
print(stop_words)
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
{'any', 'more', "you'll", 'or', 'from', 'down', 'not', 'own', "couldn't", 'you', 'doesn', 'under', 'we', "you'd", 'ain', 'isn', 'it', 'my', 'is', 'wil
[ ]
filtered_sent=[]
for w in tokenized_word:
if w not in stop_words:
filtered_sent.append(w)
print("Tokenized Sentence:",tokenized_word)
print("Filterd Sentence:",filtered_sent)
Tokenized Sentence: ['Hello', 'Mr.', 'Smith', ',', 'how', 'are', 'you', 'doing', 'today', '?', 'The', 'weather', 'is', 'great', ',', 'and', 'city', 'is
Filterd Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.
2s completed at 9:23 AM
Type here to search
40°C Partly sunny
ENG
09:23
29-04-2022

practicedsassignment7.ipynb
colab.research.google.com/drive/1Ce2hwYIU3fp7v6l45VuzBGi_DGw90ZzN
practicedsassignment7.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize
ps = PorterStemmer()
stemmed_words=[]
for w in filtered_sent:
stemmed_words.append(ps.stem(w))
print("Filtered Sentence:",filtered_sent)
print("Stemmed Sentence:",stemmed_words)
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue',
Stemmed Sentence: ['hello', 'mr.', 'smith', ',', 'today', '?', 'the', 'weather', 'great', ',', 'citi', 'awesom', '.', 'the', 'sky', 'pinkish-blu', '.',
[ ]
[ ] nltk.download('wordnet')
from nltk.stem.wordnet import WordNetLemmatizer
lem = WordNetLemmatizer()
2s completed at 9:23 AM
Type here to search
40°C Partly sunny
ENG
09:23
29-04-2022


practicedsassignment7.ipynb
colab.research.google.com/drive/1Ce2hwYIU3fp7v6l45VuzBGi_DGw90ZzN
practicedsassignment7.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing
[ ] nltk.download('wordnet')
from nltk.stem.wordnet import WordNetLemmatizer
lem = WordNetLemmatizer()
from nltk.stem.porter import PorterStemmer
stem = PorterStemmer()
word = "flying"
print("Lemmatized Word:",lem.lemmatize(word,"v"))
print("Stemmed Word:",stem.stem(word))
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Unzipping corpora/wordnet.zip.
Lemmatized Word: fly
Stemmed Word: fli
[ ] sent = "Albert Einstein was born in Ulm, Germany in 1879."
[ ] tokens=nltk.word_tokenize(sent)
print(tokens)
['Albert', 'Einstein', 'was', 'born', 'in', 'Ulm', ',', 'Germany', 'in', '1879', '.']
2s completed at 9:23 AM
Type here to search
40°C Partly sunny
ENG
09:23
29-04-2022

practicedsassignment7.ipynb
colab.research.google.com/drive/1Ce2hwYIU3fp7v6l45VuzBGi_DGw90ZzN
practicedsassignment7.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing
[ ] nltk.download('averaged_perceptron_tagger')
nltk.pos_tag(tokens)
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data] /root/nltk_data...
[nltk_data] Unzipping taggers/averaged_perceptron_tagger.zip.
[('Albert', 'NNP'),
('Einstein', 'NNP'),
('was', 'VBD'),
('born', 'VBN'),
('in', 'IN'),
('Ulm', 'NNP'),
(',', ','),
('Germany', 'NNP'),
('in', 'IN'),
('1879', 'CD'),
('.', '.')]
[ ] import io
from google.colab import files
[ ] uploaded=files.upload()
2s completed at 9:23 AM
Type here to search
40°C Partly sunny
ENG
09:23
29-04-2022



practicedsassignment7.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

```
[ ] from google.colab import files
```

```
[ ] uploaded=files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving train.tsv to train.tsv
```

```
[ ] df=pd.read_csv(io.BytesIO(uploaded['train.tsv']), sep='\t')
```

```
[ ] df.head()
```

	PhraseId	SentenceId	Phrase	Sentiment
0	1	1	A series of escapades demonstrating the adage ...	1
1	2	1	A series of escapades demonstrating the adage ...	2
2	3	1	A series	2
3	4	1	A	2
4	5	1	series	2

```
[ ] df.info()
```

2s completed at 9:23 AM

Type here to search 40°C Partly sunny ENG 09:23 29-04-2022

practicedsassignment7.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

```
[ ] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156060 entries, 0 to 156059
Data columns (total 4 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   PhraseId    156060 non-null  int64
 1   SentenceId  156060 non-null  int64
 2   Phrase      156060 non-null  object
 3   Sentiment   156060 non-null  int64
dtypes: int64(3), object(1)
memory usage: 4.8+ MB
```

```
df.Sentiment.value_counts()
```

```
2    79582
3    32927
1    27273
4     9206
0     7072
Name: Sentiment, dtype: int64
```

```
[ ] import matplotlib.pyplot as plt
```

2s completed at 9:23 AM

Type here to search 40°C Partly sunny ENG 09:23 29-04-2022

practicedsassignment7.ipynb - C

colab.research.google.com/drive/1Ce2hwYIU3fp7v6l45VuzBGi_DGw90ZzN

practicedsassignment7.ipynb

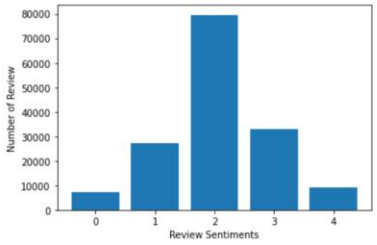
File Edit View Insert Runtime Tools Help All changes saved

Comment Share P

+ Code + Text

RAM Disk Editing

```
import matplotlib.pyplot as plt
Sentiment_count=df.groupby('Sentiment').count()
plt.bar(Sentiment_count.index.values, Sentiment_count['Phrase'])
plt.xlabel('Review Sentiments')
plt.ylabel('Number of Review')
plt.show()
```




```
from sklearn.feature_extraction.text import CountVectorizer
from nltk.tokenize import RegexpTokenizer
```

2s completed at 9:23 AM

Type here to search

40°C Partly sunny ENG 09:23 29-04-2022

practicedsassignment7.ipynb - C

colab.research.google.com/drive/1Ce2hwYIU3fp7v6l45VuzBGi_DGw90ZzN

practicedsassignment7.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share P

+ Code + Text

RAM Disk Editing

Review Sentiments

```
[ ] from sklearn.feature_extraction.text import CountVectorizer
    from nltk.tokenize import RegexpTokenizer
    #tokenizer to remove unwanted elements from out data like symbols and numbers
    token = RegexpTokenizer(r'[a-zA-Z0-9]+')
    cv = CountVectorizer(lowercase=True,stop_words='english',ngram_range = (1,1),tokenizer = token.tokenize)
    text_counts= cv.fit_transform(df['Phrase'])
```

```
[ ] from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(
        text_counts, df['Sentiment'], test_size=0.3, random_state=1)
```

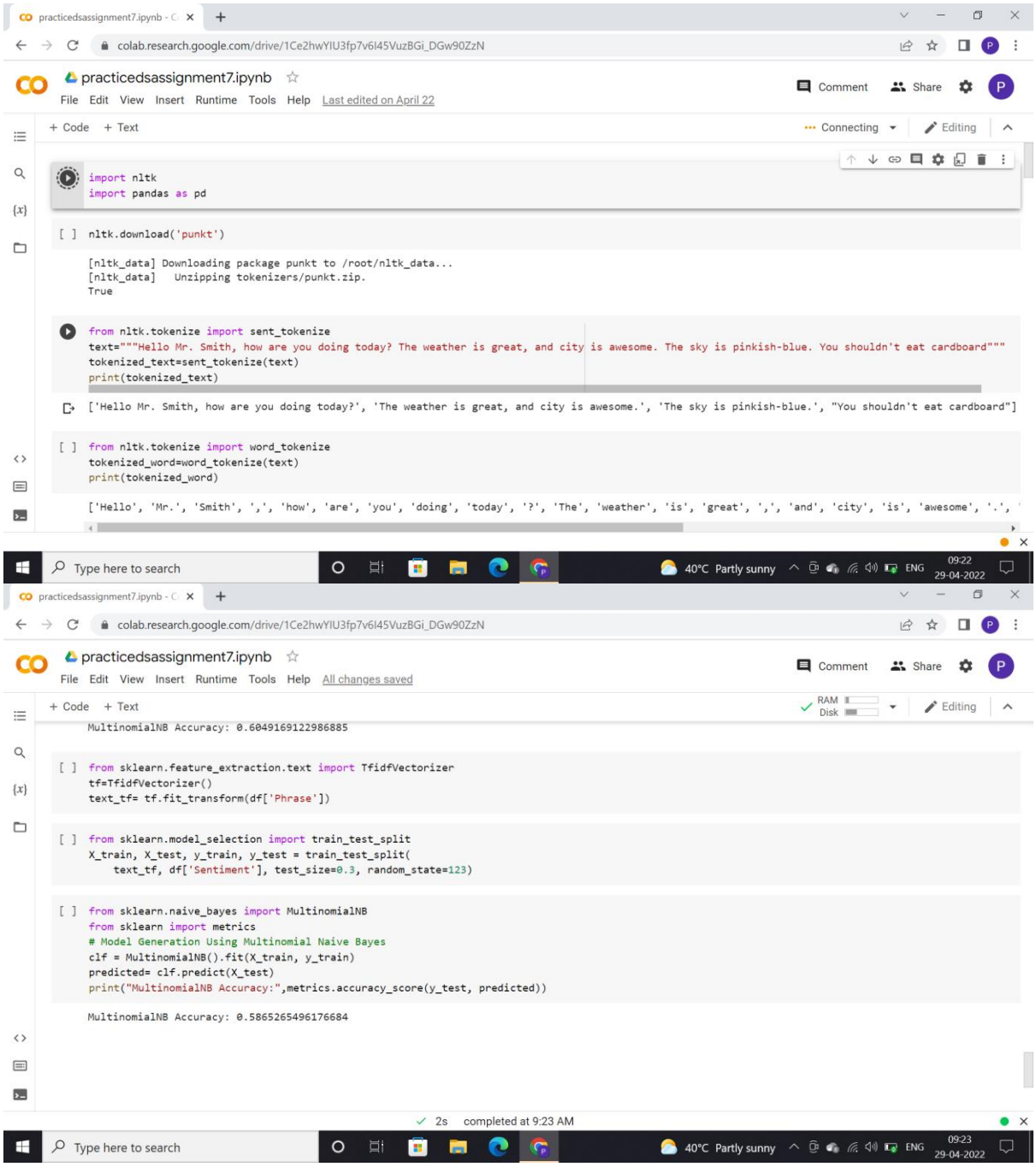
```
from sklearn.naive_bayes import MultinomialNB
#Import scikit-learn metrics module for accuracy calculation
from sklearn import metrics
# Model Generation Using Multinomial Naive Bayes
clf = MultinomialNB().fit(X_train, y_train)
predicted= clf.predict(X_test)
print("MultinomialNB Accuracy:",metrics.accuracy_score(y_test, predicted))
```

```
MultinomialNB Accuracy: 0.6049169122986885
```

2s completed at 9:23 AM

Type here to search

40°C Partly sunny ENG 09:23 29-04-2022


practicedsassignment7.ipynb - C
colab.research.google.com/drive/1Ce2hwYIU3fp7v6l45VuzBGi_DGw90ZzN
practicedsassignment7.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 22
Comment
Share
+ Code + Text
Connecting
Editing
import nltk
import pandas as pd
[ ] nltk.download('punkt')
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt.zip.
True
from nltk.tokenize import sent_tokenize
text="""Hello Mr. Smith, how are you doing today? The weather is great, and city is awesome. The sky is pinkish-blue. You shouldn't eat cardboard"""
tokenized_text=sent_tokenize(text)
print(tokenized_text)
['Hello Mr. Smith, how are you doing today?', 'The weather is great, and city is awesome.', 'The sky is pinkish-blue.', "You shouldn't eat cardboard"]
[ ] from nltk.tokenize import word_tokenize
tokenized_word=word_tokenize(text)
print(tokenized_word)
['Hello', 'Mr.', 'Smith', ',', 'how', 'are', 'you', 'doing', 'today', '?', 'The', 'weather', 'is', 'great', ',', 'and', 'city', 'is', 'awesome', '.',
Type here to search
40°C Partly sunny
ENG
09:22
29-04-2022

practicedsassignment7.ipynb - C
colab.research.google.com/drive/1Ce2hwYIU3fp7v6l45VuzBGi_DGw90ZzN
practicedsassignment7.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
MultinomialNB Accuracy: 0.6049169122986885
[ ] from sklearn.feature_extraction.text import TfidfVectorizer
tf=TfidfVectorizer()
text_tf= tf.fit_transform(df['Phrase'])
[ ] from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
text_tf, df['Sentiment'], test_size=0.3, random_state=123)
[ ] from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics
# Model Generation Using Multinomial Naive Bayes
clf = MultinomialNB().fit(X_train, y_train)
predicted= clf.predict(X_test)
print("MultinomialNB Accuracy:",metrics.accuracy_score(y_test, predicted))
MultinomialNB Accuracy: 0.5865265496176684
2s completed at 9:23 AM
Type here to search
40°C Partly sunny
ENG
09:23
29-04-2022