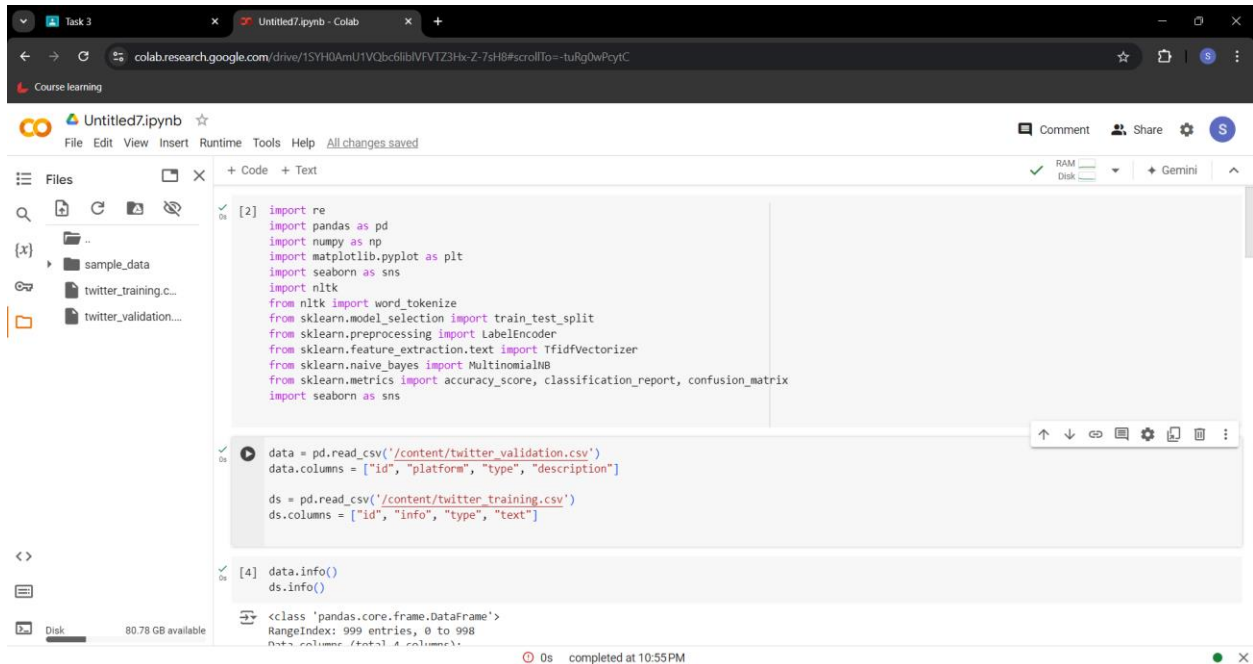


# Syed Faris Hussain Naqvi



This screenshot shows the first two code cells of a Google Colab notebook. The first cell imports various libraries including re, pandas, numpy, matplotlib, seaborn, nltk, and sklearn. The second cell loads two CSV files: 'twitter\_validation.csv' and 'twitter\_training.csv', and assigns them to 'data' and 'ds' respectively. The notebook interface includes a file explorer on the left, a top menu bar, and a status bar at the bottom indicating 80.78 GB of available disk space.

```
[2] import re
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
from nltk import word_tokenize
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import seaborn as sns
```

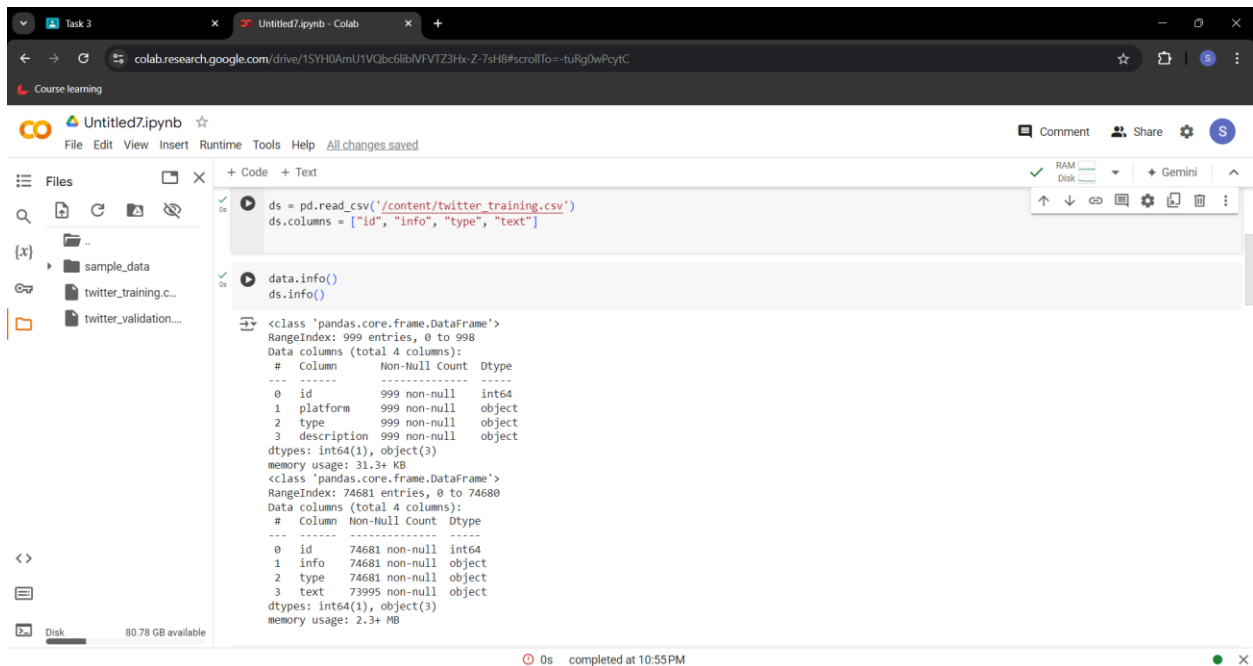
```
data = pd.read_csv('/content/twitter_validation.csv')
data.columns = ["id", "platform", "type", "description"]

ds = pd.read_csv('/content/twitter_training.csv')
ds.columns = ["id", "info", "type", "text"]
```

```
[4] data.info()
ds.info()
```

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 999 entries, 0 to 998  
Data columns (total 4 columns):

0s completed at 10:55PM



This screenshot shows the same notebook after execution. The second code cell is now highlighted, and its output is displayed below it. The output shows the details of the 'ds' DataFrame, including its range index (0 to 74680), column names, and data types. The first code cell's output is also visible, showing the details of the 'data' DataFrame. The notebook interface remains the same, with the file explorer and status bar.

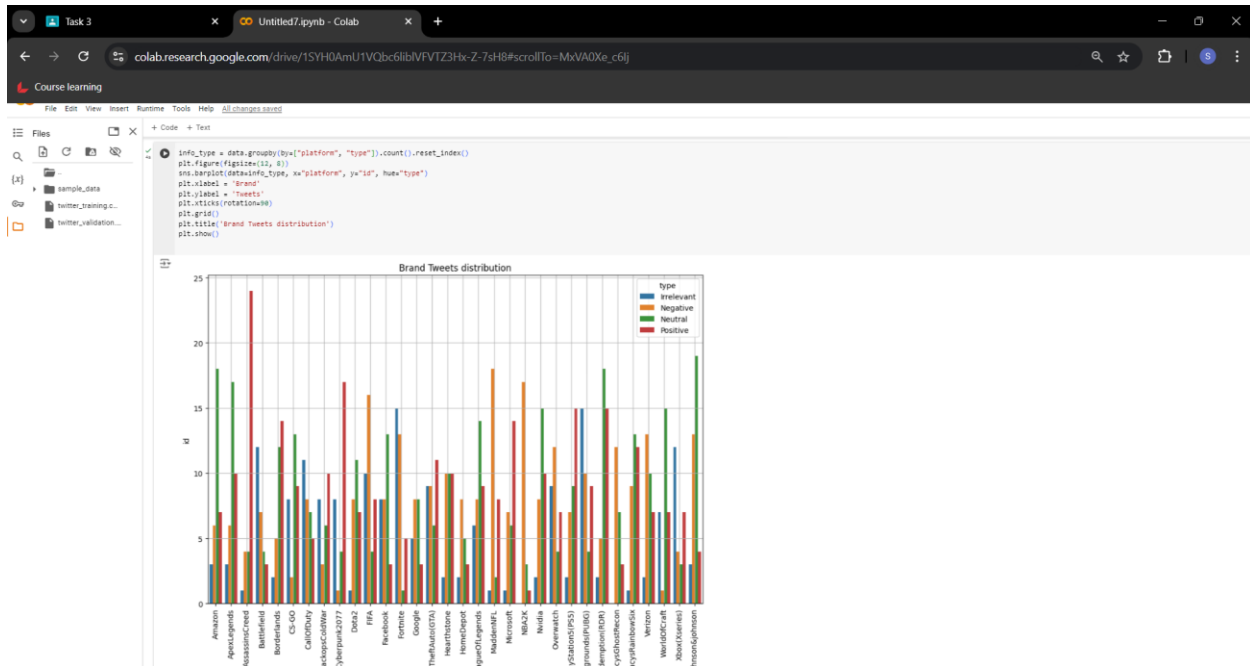
```
ds = pd.read_csv('/content/twitter_training.csv')
ds.columns = ["id", "info", "type", "text"]
```

```
data.info()
ds.info()
```

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 999 entries, 0 to 998  
Data columns (total 4 columns):  
# Column Non-Null Count Dtype  
---  
0 id 999 non-null int64  
1 platform 999 non-null object  
2 type 999 non-null object  
3 description 999 non-null object  
dtypes: int64(1), object(3)  
memory usage: 31.3+ KB  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 74681 entries, 0 to 74680  
Data columns (total 4 columns):  
# Column Non-Null Count Dtype  
---  
0 id 74681 non-null int64  
1 info 74681 non-null object  
2 type 74681 non-null object  
3 text 73995 non-null object  
dtypes: int64(1), object(3)  
memory usage: 2.3+ MB

0s completed at 10:55PM

Syed Faris Hussain Naqvi



The screenshot shows a Google Colab notebook titled "Untitled7.ipynb". The notebook is open to a cell containing the following code:

```
[6] nltk.download('punkt')
    nltk.download('stopwords')

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   unzipping corpora/stopwords.zip.
True

[8] data["lower"] = data.description.astype(str).str.lower().apply(lambda x: re.sub('[^A-Za-z0-9 ]+', ' ', x))
    ds["lower"] = ds.text.astype(str).str.lower().apply(lambda x: re.sub('[^A-Za-z0-9 ]+', ' ', x))

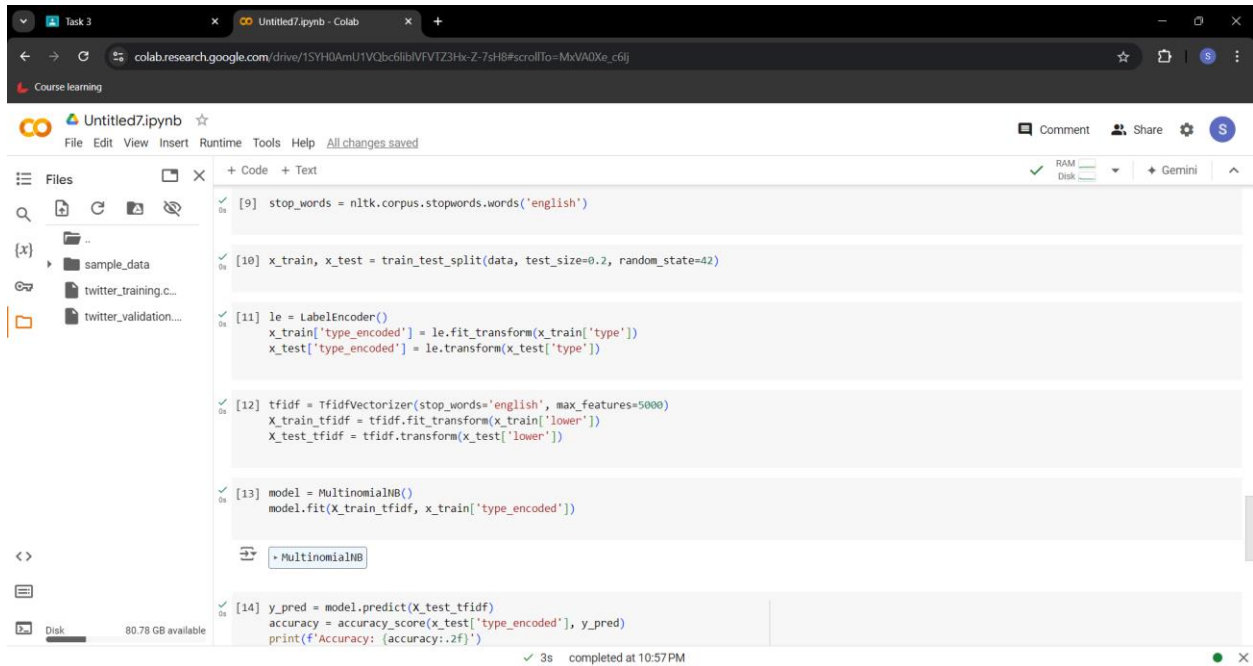
data.head()
ds.head()
```

The output of the code shows the progress of downloading and unzipping the NLTK corpora. Below the code, a table view displays the first few rows of the processed data:

	id	info	type	text	lower
0	2401	Borderlands	Positive	I am coming to the borders and I will kill you...	i am coming to the borders and i will kill you...
1	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...	im getting on borderlands and i will kill you ...
2	2401	Borderlands	Positive	im coming on borderlands and i will murder you...	im coming on borderlands and i will murder you...
3	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...	im getting on borderlands 2 and i will murder ...
4	2401	Borderlands	Positive	im getting into borderlands and i can murder y...	im getting into borderlands and i can murder y...

At the bottom of the notebook, there are links for "Generate code with ds", "View recommended plots", and "New interactive sheet". The status bar indicates "3s completed at 10:57 PM".

# Syed Faris Hussain Naqvi



Colab notebook titled 'Untitled7.ipynb' showing code for text classification using MultinomialNB. The code includes importing libraries, splitting data, encoding labels, vectorizing text, fitting the model, and predicting accuracy.

```
[9] stop_words = nltk.corpus.stopwords.words('english')

[10] x_train, x_test = train_test_split(data, test_size=0.2, random_state=42)

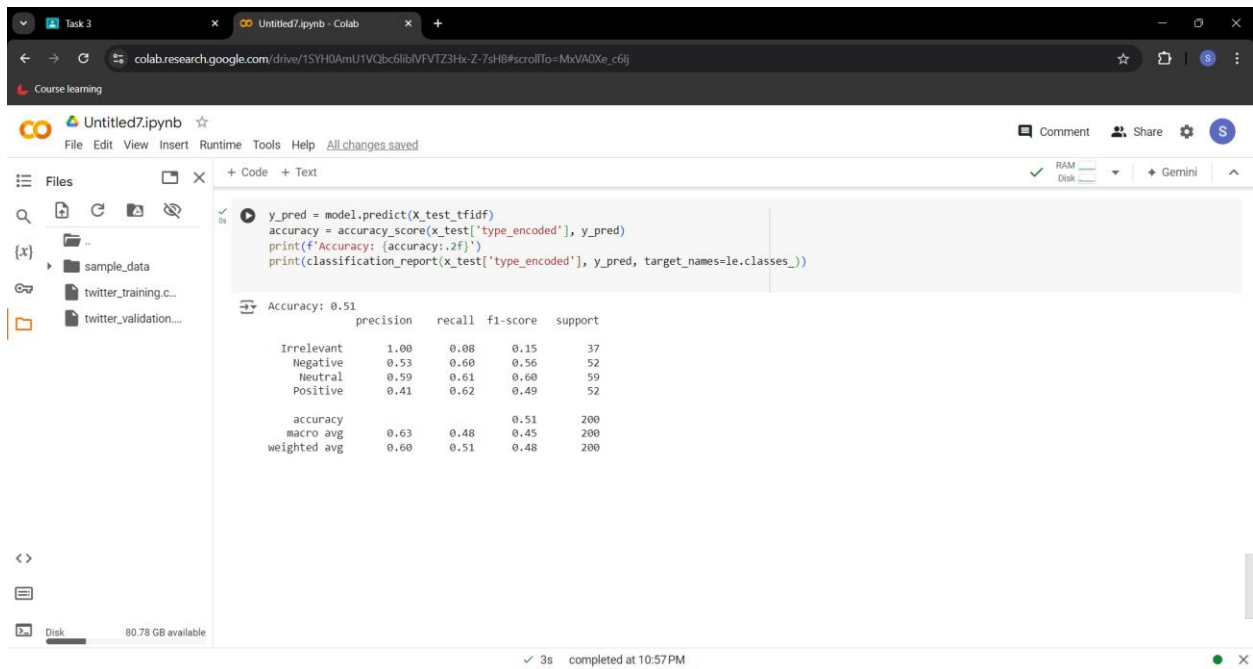
[11] le = LabelEncoder()
x_train['type_encoded'] = le.fit_transform(x_train['type'])
x_test['type_encoded'] = le.transform(x_test['type'])

[12] tfidf = TfidfVectorizer(stop_words='english', max_features=5000)
x_train_tfidf = tfidf.fit_transform(x_train['lower'])
x_test_tfidf = tfidf.transform(x_test['lower'])

[13] model = MultinomialNB()
model.fit(x_train_tfidf, x_train['type_encoded'])

[14] y_pred = model.predict(x_test_tfidf)
accuracy = accuracy_score(x_test['type_encoded'], y_pred)
print(f'Accuracy: {accuracy:.2f}')
```

3s completed at 10:57PM



Colab notebook titled 'Untitled7.ipynb' showing the classification report for the MultinomialNB model. The code includes predicting on the test set and printing the accuracy and classification report.

```
y_pred = model.predict(x_test_tfidf)
accuracy = accuracy_score(x_test['type_encoded'], y_pred)
print(f'Accuracy: {accuracy:.2f}')
print(classification_report(x_test['type_encoded'], y_pred, target_names=le.classes_))
```

Accuracy: 0.51

	precision	recall	f1-score	support
Irrelevant	1.00	0.08	0.15	37
Negative	0.53	0.60	0.56	52
Neutral	0.59	0.61	0.60	59
Positive	0.41	0.62	0.49	52
accuracy			0.51	200
macro avg	0.63	0.48	0.45	200
weighted avg	0.60	0.51	0.48	200

3s completed at 10:57PM