10, Main Bazar, purkazi,
Muzaffarnagar, UP

SIDDHANT RAI JAIN

+91 6397355482

jain10siddhant@gmail.com

http://www.linkedin.com/in/siddhant02

PROFESSIONAL SUMMARY:

An optimistic and passionate graduate, possessing good problem solving skills, wish to use his technical knowledge to fulfill the needs of the company. I like to work with a diverse group of people to help in the company growth and see myself leading a group of people.

AREAS OF EXPERTISE:

- C ++
- Python
- HTML5
- CSS3
- Data Structures and Algorithms
- JavaScript
- Node.js
- Express
- Angular.js
- SQL
- Mongodb
- EDA
- Numpy
- Pandas
- Matplotlib
- AWS
- Bootstrap

EDUCATION:

2021- 2023 | Chandigarh University, Mohalli, Punjab

MCA- Computer applications | Pursuing

2018 - 2021 | Gurukul Kangri Deemed to be University, Haridwar (Uttarakhand)

B.Sc- Maths with computer Science | percentage: 74%

2017-2018| Greenway Modern Sr. Sec. School, Roorkee (Haridwar)

12th - PCM | percentage: 67.8 %

2015-2016| New Stepping Stones School, Purkazi, Muzaffarnagar (UP)

10th | CGPA: 9.5

CERTIFICATIONS :

- Technical Support Fundamentals- by Google
- Python Programming Certificate- via Coursera
- HTML Certificate Course- by Sololearn
- Diploma in Computer Applications(DCA)- by OAICTE

EXTRA CARRICULAR & CO-CURRICULAR ACTIVITIES

- Participated in Blood Donation Camp .
- Volunteered at Astronomy workshop at GKV.
- Secured 3rd position in ATTR-ACT organized by Orator club, CU.
- Solving HackerRank Problems.
- Done internship at younity.in as Campus Ambassador.

HOBBIES

- Travelling
- Reading Novels
- Listening podcasts

PROJECTS

- Zomato Data Analysis using EDA.
- Text Editor using Python.
- Front-end using Html, css

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.
NAAC GRADE A+
Accredited University

Subject Code: 21CAP722

Final LAB MST

Student Name:Km Rishika Jaiswal **UID: 21MCI1152**
Section/Group: 21MAM-1_B **Semester: 3rd**
Date of Submission: 14/11/2022 **Course: MCA (AIML)**
Subject: Machine Learning Lab

Aim/Overview of the practical:

Task to be done:

Twitter has become an important communication channel in times of emergency.
The ubiquitousness of smartphones enables people to announce an emergency they're observing in real-time. Because of this, more agencies are interested in programatically monitoring Twitter (i.e. disaster relief organizations and news agencies).

The author explicitly uses the word "ABLAZE" but means it metaphorically. This is clear to a human right away, especially with the visual aid. But it's less clear to a machine.

In this competition, you're challenged to build a machine learning model that predicts which Tweets are about real disasters and which one's aren't. You'll have access to a dataset of 10,000 tweets that were hand classified

Solution:

1. **Code for experiment/practical**

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
# visulization
import matplotlib.pyplot as plt
# %matplotlib inline
```

```
import seaborn as sns
# to detect languages
import langid

pip install langid

df_train=pd.read_csv('train.csv')

df_test=pd.read_csv('test.csv')

df_train.head()

df_test.head()

df=df_train

df.info()

df.shape

df.isna().sum()

df.duplicated().sum()

sns.countplot(df['target'])
plt.title("Countplot for target Labels")

df['target'].value_counts()

min_val_sum = min(df.text,key=len)
print("The minimum text is:\n" ,min_val_sum,"\nAnd his length",len(min_val_sum))

max_val_sum = max(df['text'], key=len)
print("The maximum text is:\n" ,max_val_sum,"\nAnd his length",len(max_val_sum))

ids_langid=df['text'].apply(langid.classify)
langs = ids_langid.apply(lambda tuple: tuple[0])
print("Number of tagged languages (estimated):")
print(len(langs.unique()))
print("Percent of data in English (estimated):")
print((sum(langs=="en")/len(langs))*100)

langs_df = pd.DataFrame(langs)
langs_count = langs_df.text.value_counts()
print(langs_count)

langs_count.plot.bar(figsize=(15,10), fontsize=15)

df['detect']=langs=="en"
```

```
non_english_text=df[df['detect'] == False]
non_english_text.head()

pip install pySpellChecker

pip install contractions

from re import sub
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from spellchecker import SpellChecker
import contractions as ct
import string
import nltk

df['text'][0:10].values

nltk.download('stopwords')

import nltk
nltk.download('punkt')

spell = SpellChecker()

corpus=[]
for i in df['text'].values:

    text=i

    #expand contraction in text
    text=[ct.fix(i) for i in text.split()]
    text=' '.join(text)

    #removing http and @
    text = sub(r"http\S+|@\S+", "", text)

    #Removing Punctuations
    text = sub("[^a-zA-Z]",' ',text)

        #Lowercasing
    text=text.lower()

    #Tokenization
    text=word_tokenize(text)

    #Spelling Correction
    text = [spell.correction(i) for i in text]
```

```
    #join text
    text = ' '.join(filter(lambda x: x if x is not None else '', text))


    corpus.append(text)

corpus[0:10]
```

```
df['clean_text']=corpus
df.head()

df=df[['clean_text','target']]
df
```

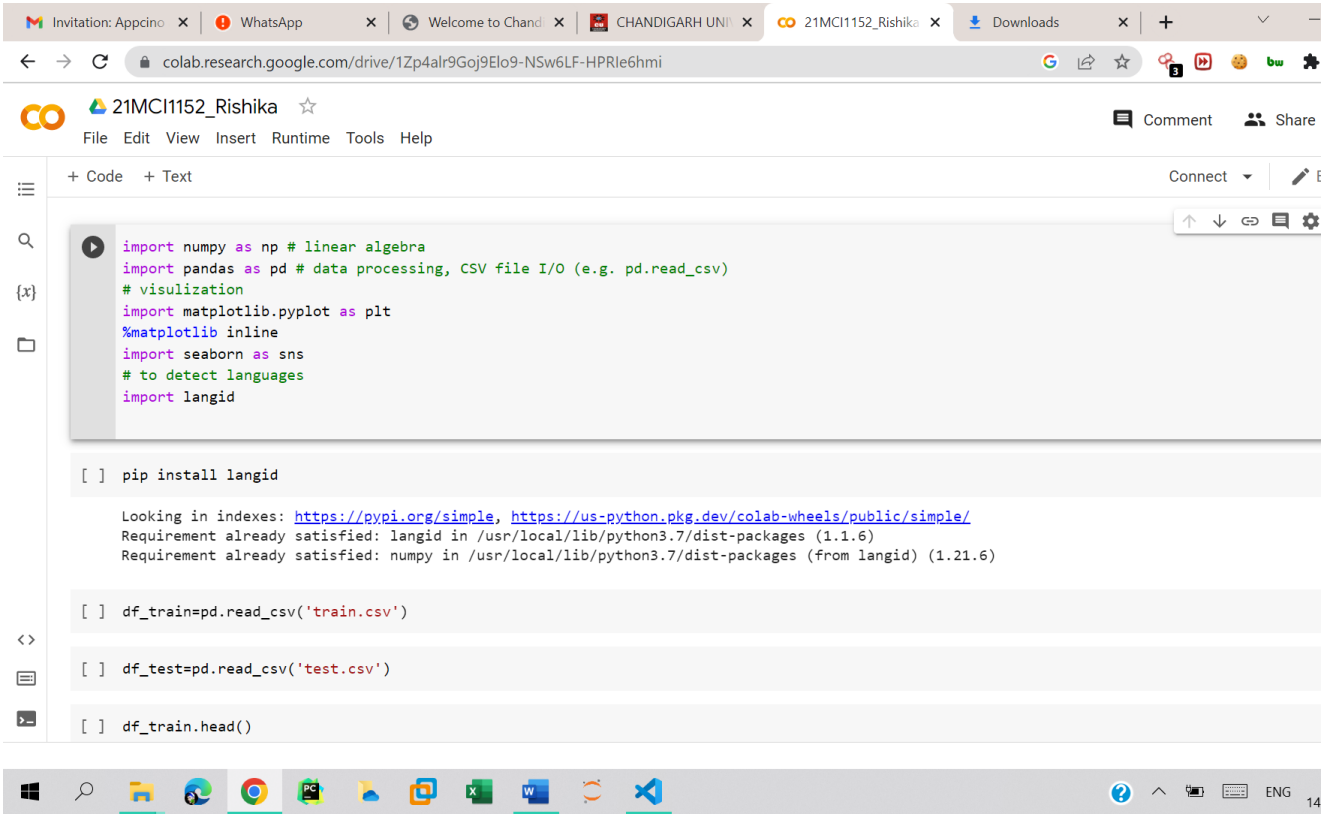
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
```

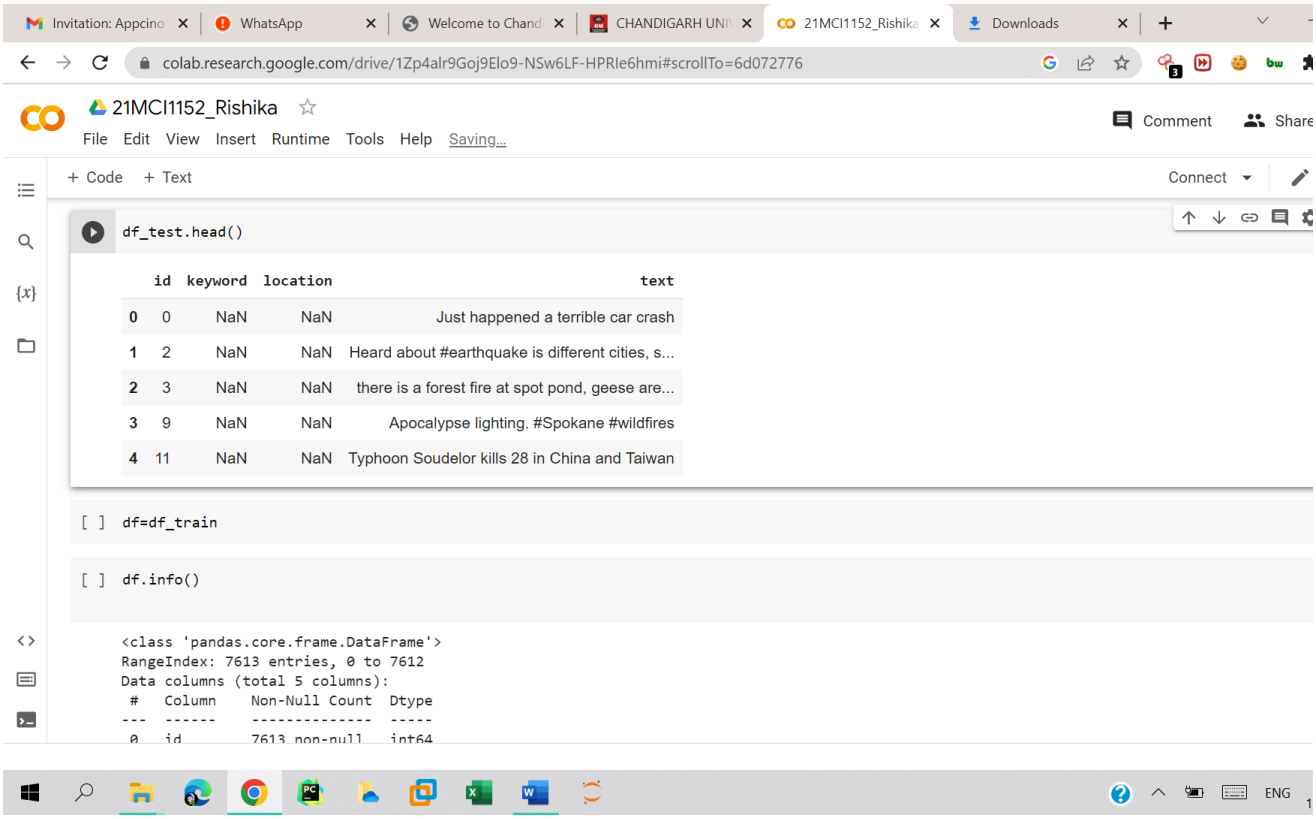
```
tfidf = TfidfVectorizer(stop_words='english')

X = df['clean_text']
y = df['target']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size= 0.2, random_state= 0)

#apply feature extraxtion on train set
X_train = tfidf.fit_transform(X_train)
X_test = tfidf.transform(X_test)

print(len(y_train))
print(len(y_test))

def classification(model, X, y):

    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    print('-Classification Report-\n')
    print(classification_report(y_test, y_pred))

    print('Accuracy= ', accuracy_score(y_test, y_pred)*100, '%\n')
```

```
    print('-Confusion Matrix-\n')
    cm=confusion_matrix(y_test, y_pred)
    sns.heatmap(cm, annot=True, fmt='d')
    plt.xlabel('Predicted')
    plt.ylabel('Truth')

from sklearn.ensemble import RandomForestClassifier

RF=RandomForestClassifier(n_estimators = 100, criterion = 'entropy' , random_state = 0)
classification(RF, X, y)

from sklearn.naive_bayes import MultinomialNB

NB=MultinomialNB()
classification(NB, X, y)

from sklearn.svm import SVC

SVM = SVC(kernel = 'rbf', random_state = 0)
classification(SVM, X, y)

from sklearn.svm import SVC

SVM = SVC(kernel = 'rbf', random_state = 0)
classification(SVM, X, y)

from sklearn.tree import DecisionTreeClassifier

DT= DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
classification(DT, X, y)

from sklearn.ensemble import RandomForestClassifier

RF=RandomForestClassifier(n_estimators = 100, criterion = 'entropy' , random_state = 0)
classification(RF, X, y)

from sklearn.linear_model import LogisticRegression

LR = LogisticRegression()
LR.fit(X_train, y_train)
y_pred= LR.predict(X_test)

print('-Classification Report-\n\n',classification_report(y_test, y_pred))

print('Accuracy= ', accuracy_score(y_test, y_pred)*100, '%\n')

print('-Confusion Matrix-\n')
cm=confusion_matrix(y_test, y_pred)
```

```
sns.heatmap(cm, annot=True, fmt='d')
plt.xlabel('Predicted')
plt.ylabel('Truth')

df_test=df_test[['id','text']]

df_test.head()

x = tfidf.transform(df_test['text'])

df_test['target'] = LR.predict(x)

df_test.head()

submission = df_test[['id','target']]
submission.to_csv('submission.csv', index = False)

submission
```

2. Result/Output:





Invitation: Appcino | WhatsApp | Welcome to Chand | CHANDIGARH UNI | 21MCI1152_Rishika | Downloads

colab.research.google.com/drive/1Zp4alr9Goj9Elo9-NSw6LF-HPRIe6hmi#scrollTo=6d072776

21MCI1152_Rishika

File Edit View Insert Runtime Tools Help Saving...

Comment Share

+ Code + Text

Connect

```
df_test.head()
```

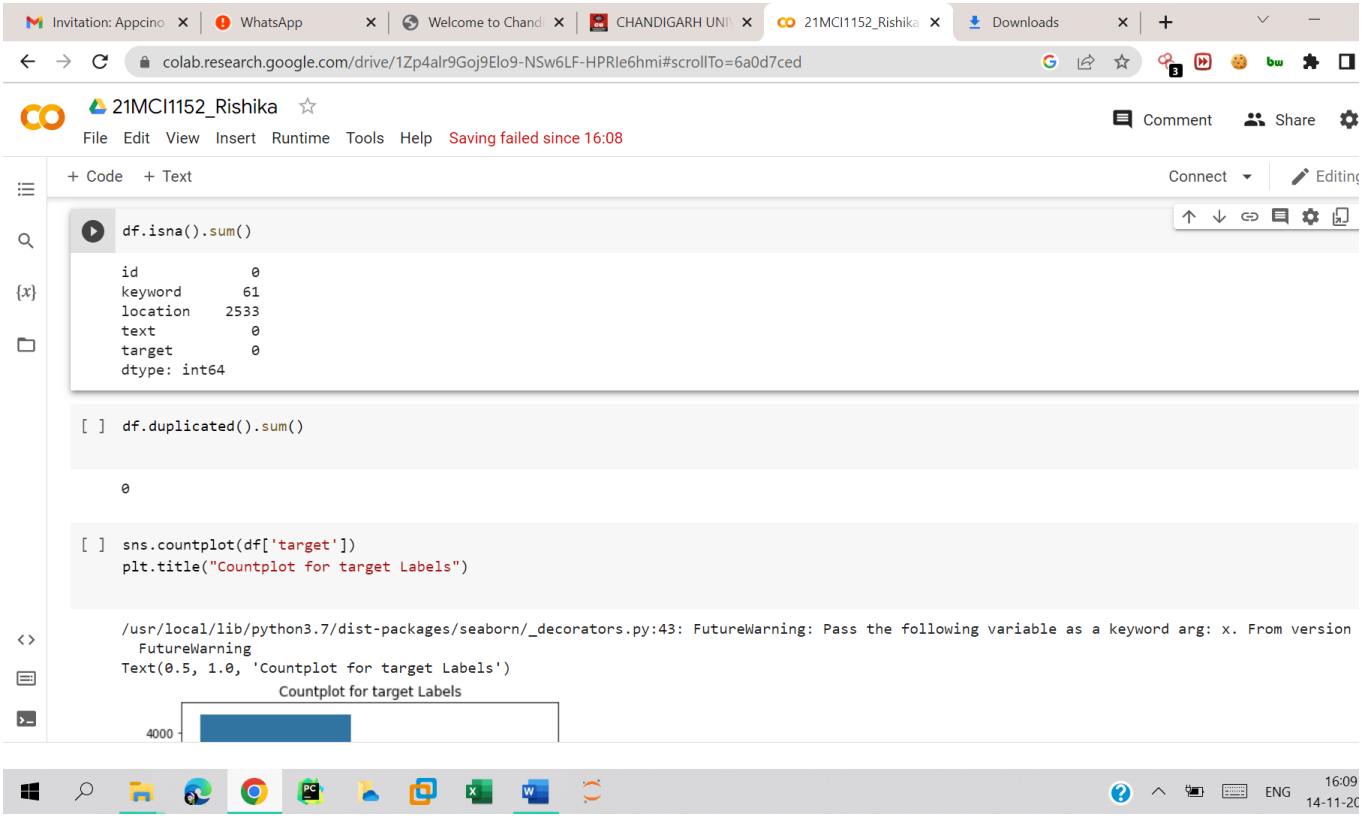
	id	keyword	location	text
0	0	NaN	NaN	Just happened a terrible car crash
1	2	NaN	NaN	Heard about #earthquake is different cities, s...
2	3	NaN	NaN	there is a forest fire at spot pond, geese are...
3	9	NaN	NaN	Apocalypse lighting. #Spokane #wildfires
4	11	NaN	NaN	Typhoon Soudelor kills 28 in China and Taiwan

```
[ ] df=df_train
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7613 entries, 0 to 7612
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   id        7613 non-null   int64
```

ENG



Invitation: Appcino | WhatsApp | Welcome to Chand | CHANDIGARH UNI | 21MCI1152_Rishika | Downloads

colab.research.google.com/drive/1Zp4alr9Goj9Elo9-NSw6LF-HPRIe6hmi#scrollTo=6a0d7ced

21MCI1152_Rishika

File Edit View Insert Runtime Tools Help Saving failed since 16:08

Comment Share

+ Code + Text

Connect Editing

```
df.isna().sum()
```

```
id             0
keyword       61
location    2533
text           0
target         0
dtype: int64
```

```
[ ] df.duplicated().sum()
```

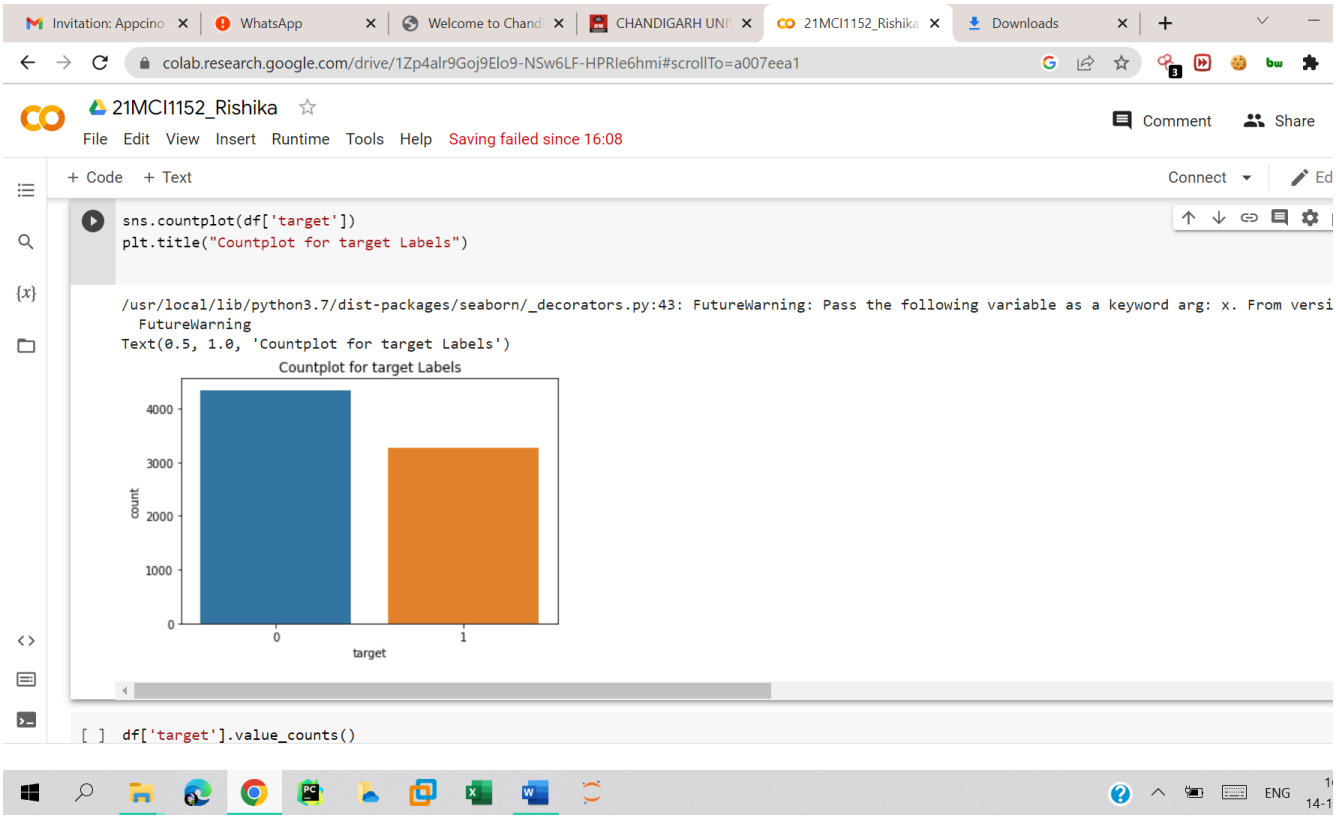
```
0
```

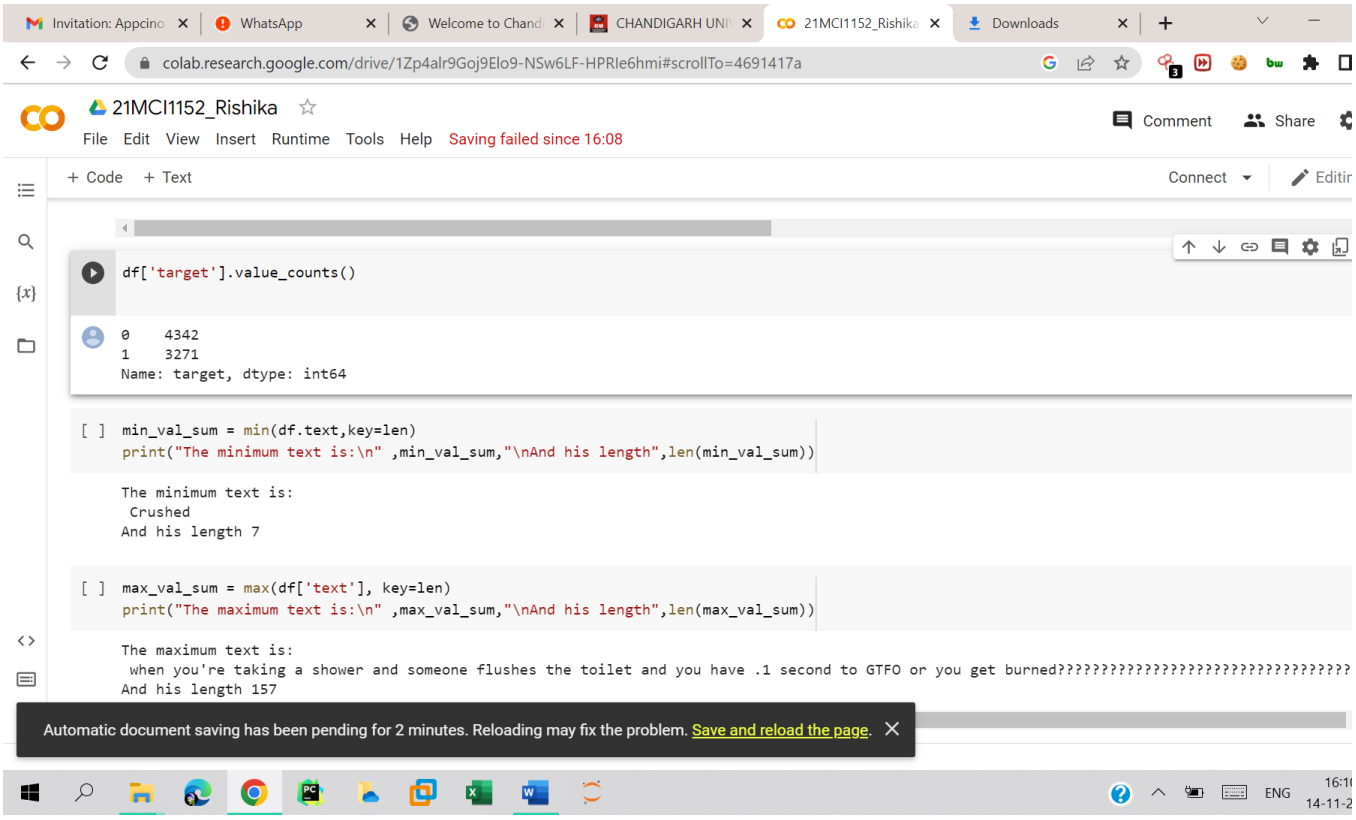
```
[ ] sns.countplot(df['target'])
    plt.title("Countplot for target Labels")
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version
  FutureWarning
Text(0.5, 1.0, 'Countplot for target Labels')
```

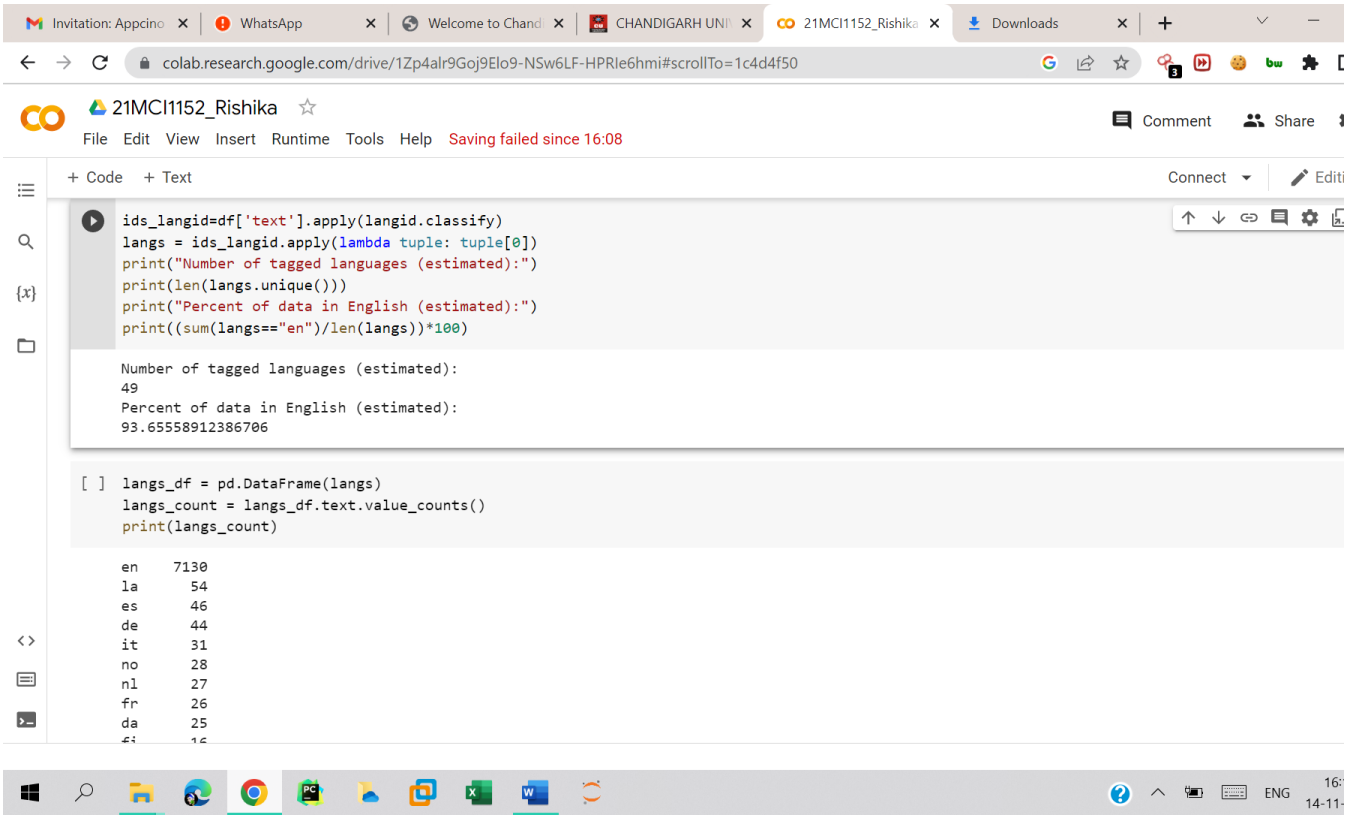

ENG 16:09 14-11-20

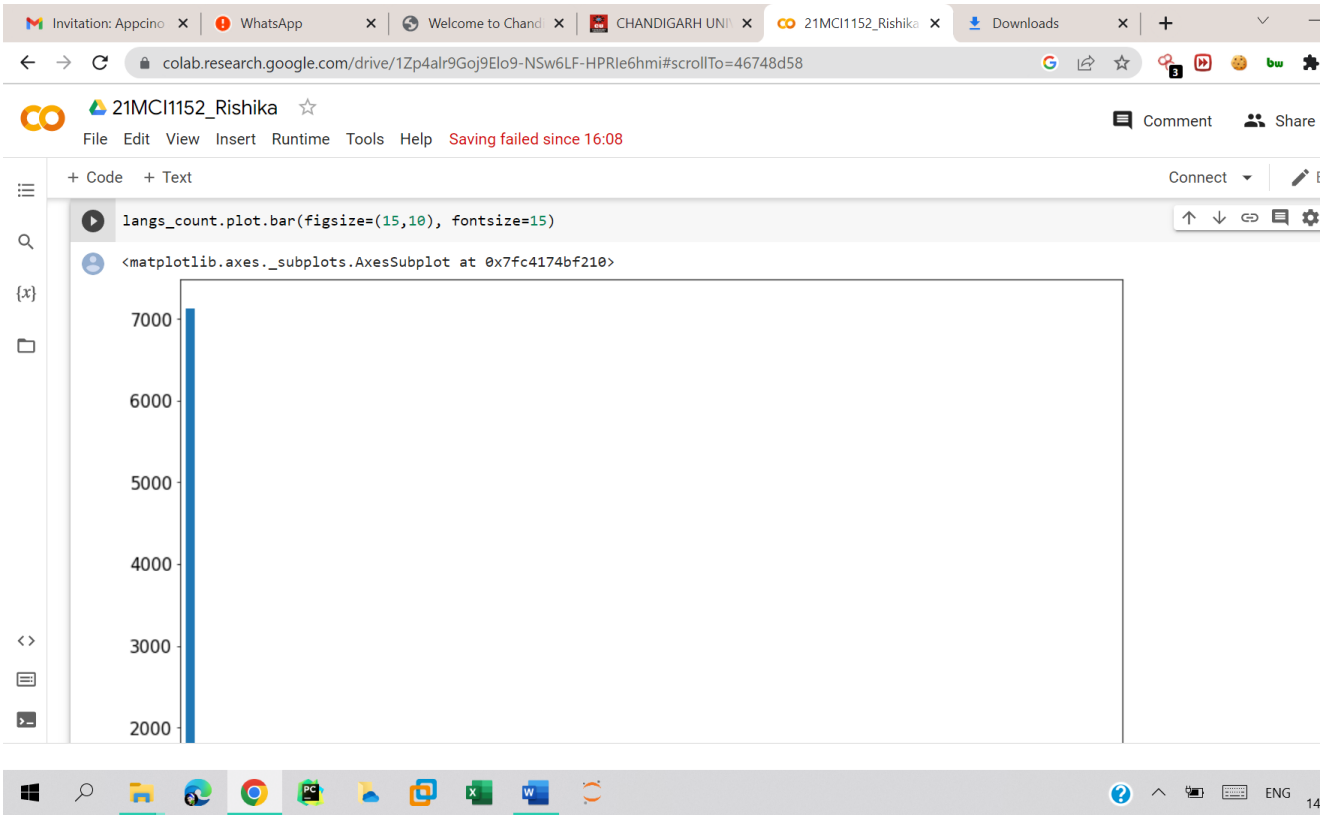
CHANDIGARH UNIVERSITY
Discover. Learn. Empower.
NAAC GRADE A+
Accredited University

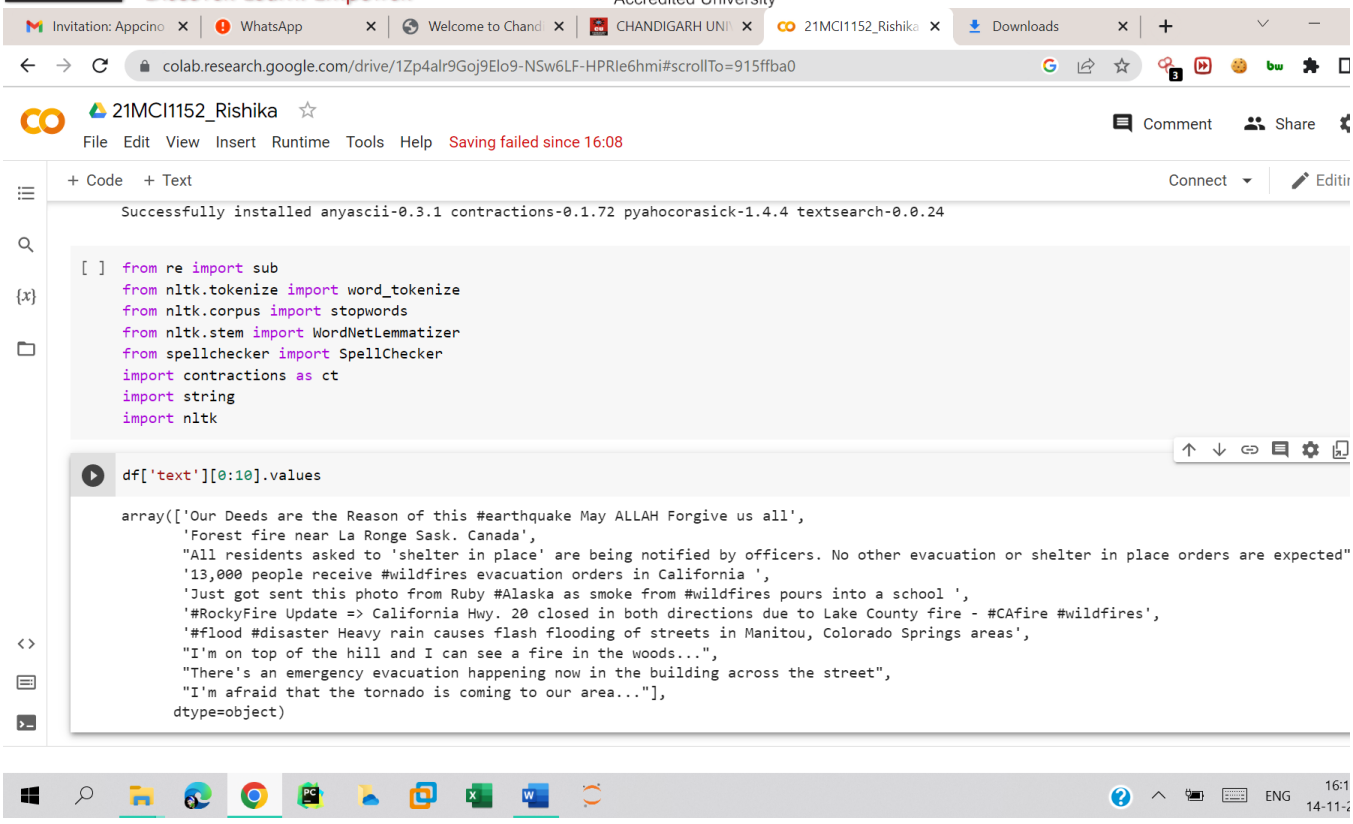


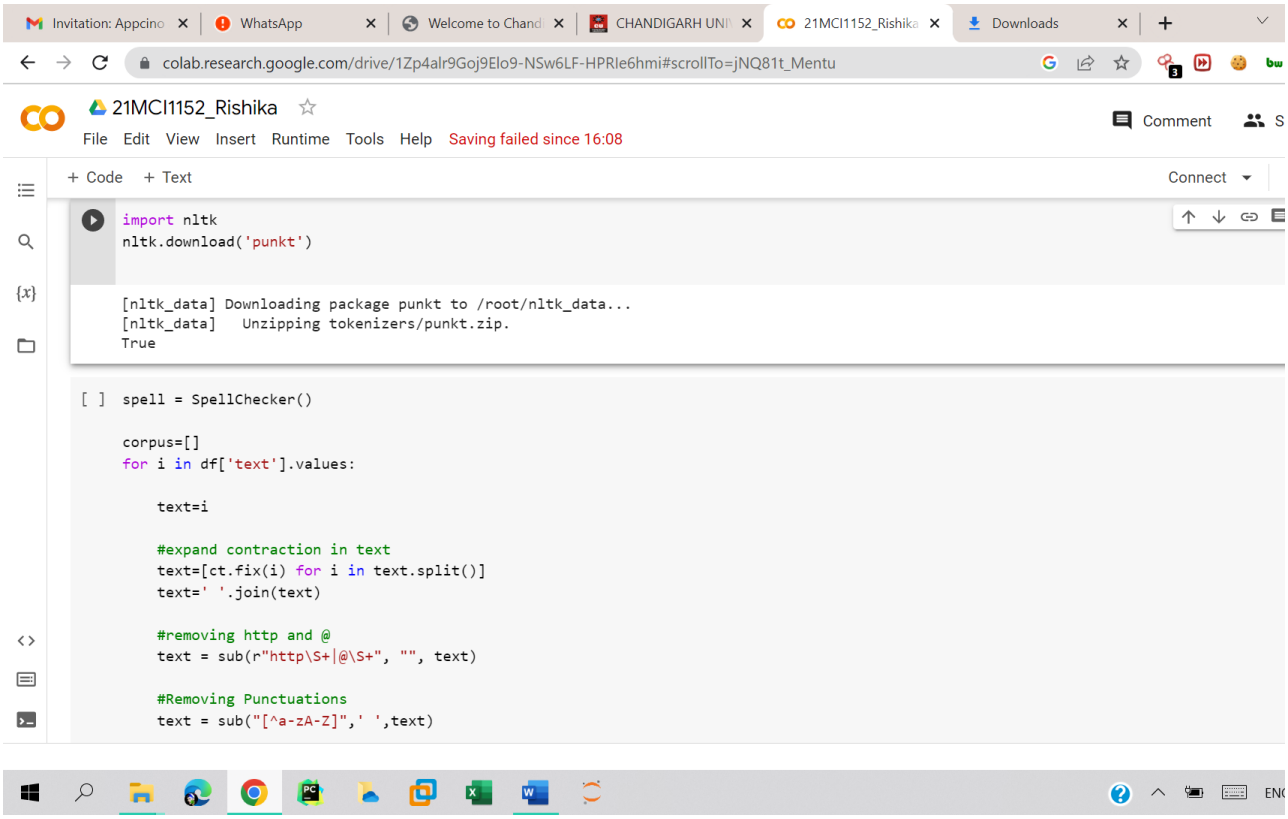


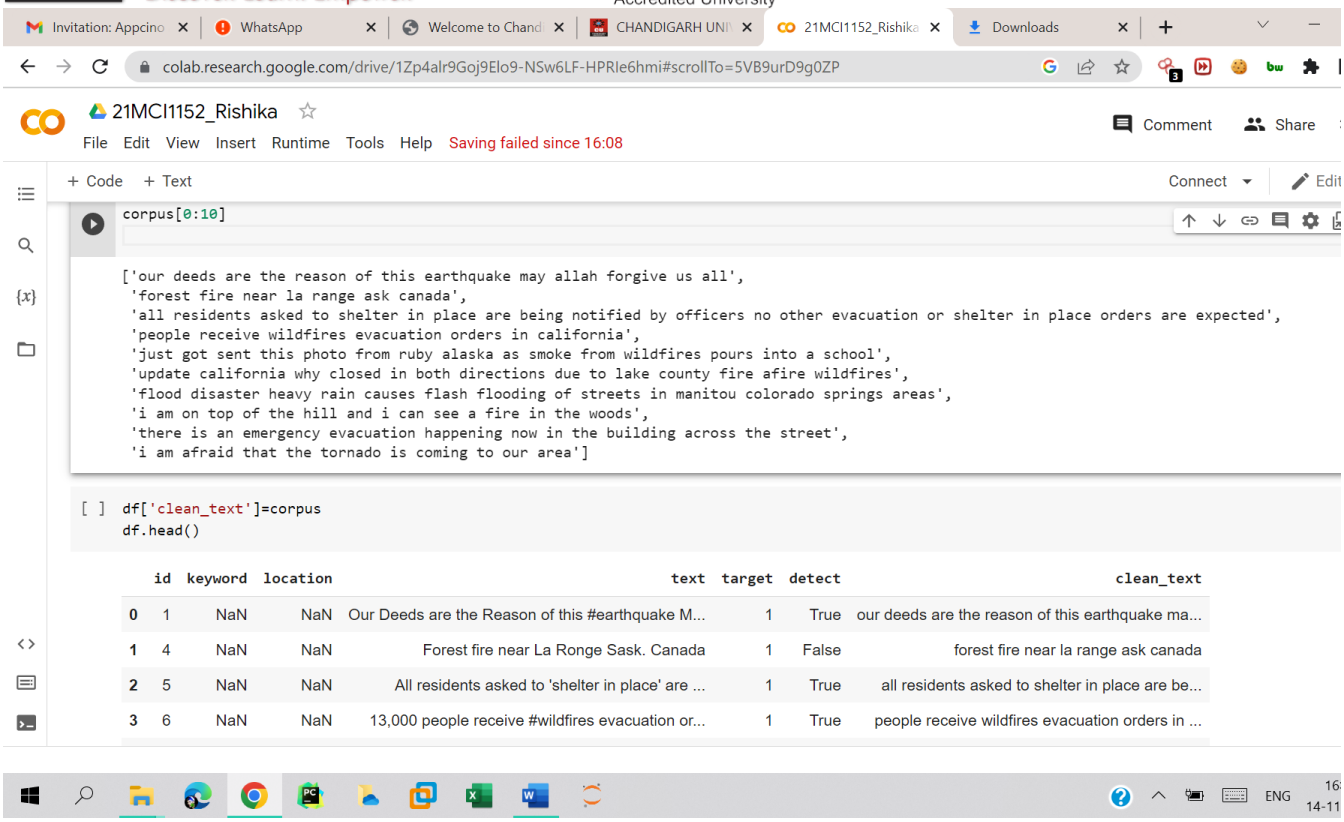
CHANDIGARH UNIVERSITY
Discover. Learn. Empower.
NAAC GRADE A+
Accredited University



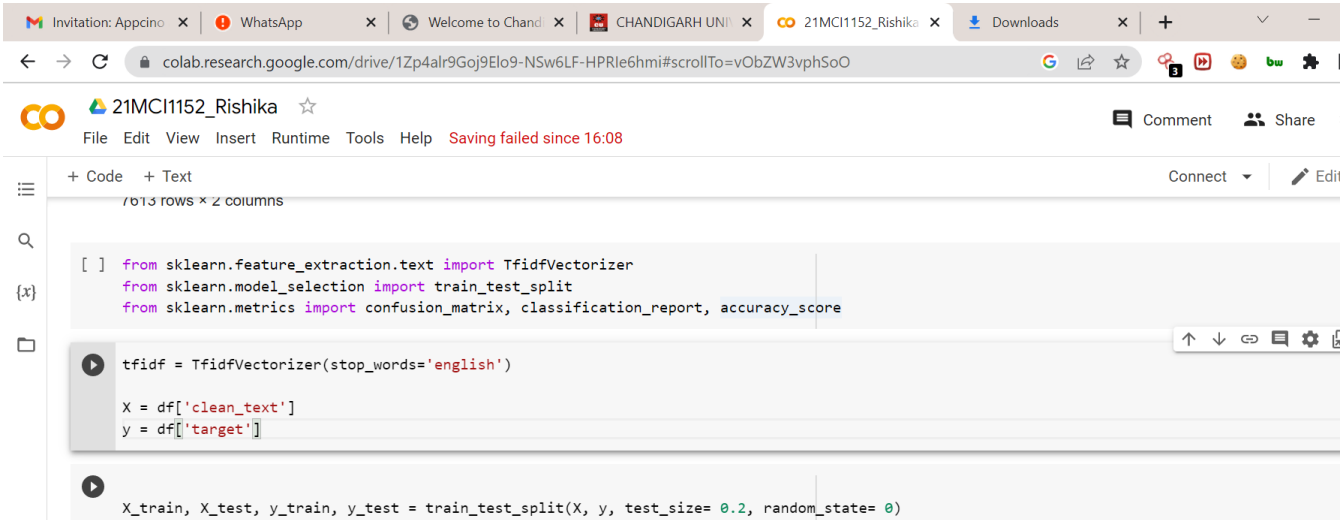








	id	keyword	location	text	target	detect	clean_text
0	1	NaN	NaN	Our Deeds are the Reason of this #earthquake M...	1	True	our deeds are the reason of this earthquake ma...
1	4	NaN	NaN	Forest fire near La Ronge Sask. Canada	1	False	forest fire near la range ask canada
2	5	NaN	NaN	All residents asked to 'shelter in place' are ...	1	True	all residents asked to shelter in place are be...
3	6	NaN	NaN	13,000 people receive #wildfires evacuation or...	1	True	people receive wildfires evacuation orders in ...



CHANDIGARH UNIVERSITY
Discover. Learn. Empower.
NAAC GRADE A+
Accredited University

Invitation: Appcino | WhatsApp | Welcome to Chand | CHANDIGARH UNI | 21MCI1152_Rishika | Downloads

colab.research.google.com/drive/1Zp4alr9Goj9Elo9-NSw6LF-HPRIe6hmi#scrollTo=49c2e69c

21MCI1152_Rishika

File Edit View Insert Runtime Tools Help Saving...

Comment

+ Code + Text

Connect

```
def classification(model, X, y):

    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    print('-Classification Report-\n')
    print(classification_report(y_test, y_pred))

    print('Accuracy= ', accuracy_score(y_test, y_pred)*100, '%\n')

    print('-Confusion Matrix-\n')
    cm=confusion_matrix(y_test, y_pred)
    sns.heatmap(cm, annot=True, fmt='d')
    plt.xlabel('Predicted')
    plt.ylabel('Truth')
```

```
from sklearn.ensemble import RandomForestClassifier

RF=RandomForestClassifier(n_estimators = 100, criterion = 'entropy' , random_state = 0)
classification(RF, X, y)
```

```
-Classification Report-

              precision    recall  f1-score   support
```

Invitation: Appcino | WhatsApp | Welcome to Chand | CHANDIGARH UNI | 21MCI1152_Rishika | Downloads

colab.research.google.com/drive/1Zp4alr9Goj9Elo9-NSw6LF-HPRIe6hmi#scrollTo=eLlnut9mkaWW

21MCI1152_Rishika

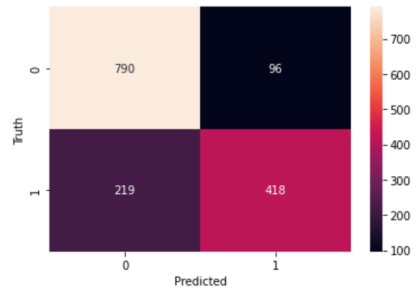
File Edit View Insert Runtime Tools Help Saving failed since 16:08

Comment Share

+ Code + Text

Connect Editin

```
    accuracy                           0.79      1523
   macro avg       0.80      0.77      0.78      1523
weighted avg       0.80      0.79      0.79      1523

Accuracy=  79.31713722915299 %

-Confusion Matrix-
```



Automatic document saving has been pending for 6 minutes. Reloading may fix the problem. Save and reload the page.

ENG 16:14 14-11-2

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.
NAAC GRADE A+
Accredited University

Invitation: Appcino | WhatsApp | Welcome to Chand | CHANDIGARH UNI | 21MCI1152_Rishika | Downloads

colab.research.google.com/drive/1Zp4alr9Goj9Elo9-NSw6LF-HPRIe6hmi#scrollTo=eLlnut9mkaWW

21MCI1152_Rishika

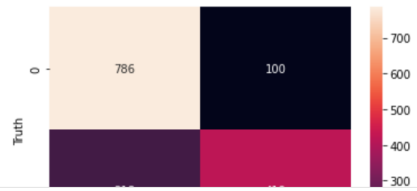
File Edit View Insert Runtime Tools Help Saving failed since 16:08

Comment Share

+ Code + Text Connect

```
NB=MultinomialNB()
classification(NB, X, y)
```

```
-Classification Report-

              precision    recall  f1-score   support

           0       0.78      0.89      0.83       886
           1       0.81      0.66      0.72       637

    accuracy                           0.79      1523
   macro avg       0.80      0.77      0.78      1523
weighted avg       0.79      0.79      0.79      1523

Accuracy=  79.12015758371635 %

-Confusion Matrix-
```



ENG

Invitation: Appcino | WhatsApp | Welcome to Chand | CHANDIGARH UNI | 21MCI1152_Rishika | Downloads

colab.research.google.com/drive/1Zp4alr9Goj9Elo9-NSw6LF-HPRIe6hmi#scrollTo=DFLSLdv7ksWH

21MCI1152_Rishika

File Edit View Insert Runtime Tools Help Saving failed since 16:08

Comment Share

+ Code + Text Connect

```
from sklearn.tree import DecisionTreeClassifier

DT= DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
classification(DT, X, y)
```

```
-Classification Report-

              precision    recall  f1-score   support

           0       0.79      0.80      0.80       886
           1       0.72      0.70      0.71       637

    accuracy                           0.76      1523
   macro avg       0.75      0.75      0.75      1523
weighted avg       0.76      0.76      0.76      1523

Accuracy=  76.03414313854235 %

-Confusion Matrix-
```

ENG

Invitation: Appcino | WhatsApp | Welcome to Chand | CHANDIGARH UNI | 21MCI1152_Rishika | Downloads

colab.research.google.com/drive/1Zp4alr9Goj9Elo9-NSw6LF-HPRIe6hmi#scrollTo=kNYqZPILk99c

21MCI1152_Rishika

File Edit View Insert Runtime Tools Help Saving failed since 16:08

Comment Share

+ Code + Text Connect Editi

Predicted

```
[ ] from sklearn.linear_model import LogisticRegression

    LR = LogisticRegression()
    LR.fit(X_train, y_train)
    y_pred= LR.predict(X_test)

    print('-Classification Report-\n\n',classification_report(y_test, y_pred))
```

```
-Classification Report-

              precision    recall  f1-score   support

           0       0.78      0.91      0.84       886
           1       0.84      0.65      0.73       637

    accuracy                           0.80      1523
   macro avg       0.81      0.78      0.78      1523
weighted avg       0.80      0.80      0.79      1523
```

```
print('Accuracy= ', accuracy_score(y_test, y_pred)*100, '%\n')

print('-Confusion Matrix-\n')
```

VMware Workstation Pro

ENG 16:
14-11-

Invitation: Appcino | WhatsApp | Welcome to Chand | CHANDIGARH UNI | 21MCI1152_Rishika | Downloads

colab.research.google.com/drive/1Zp4alr9Goj9Elo9-NSw6LF-HPRIe6hmi#scrollTo=kNYqZPILk99c

21MCI1152_Rishika

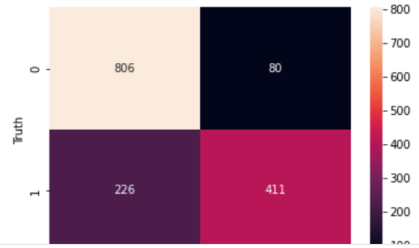
File Edit View Insert Runtime Tools Help Saving failed since 16:08

Comment

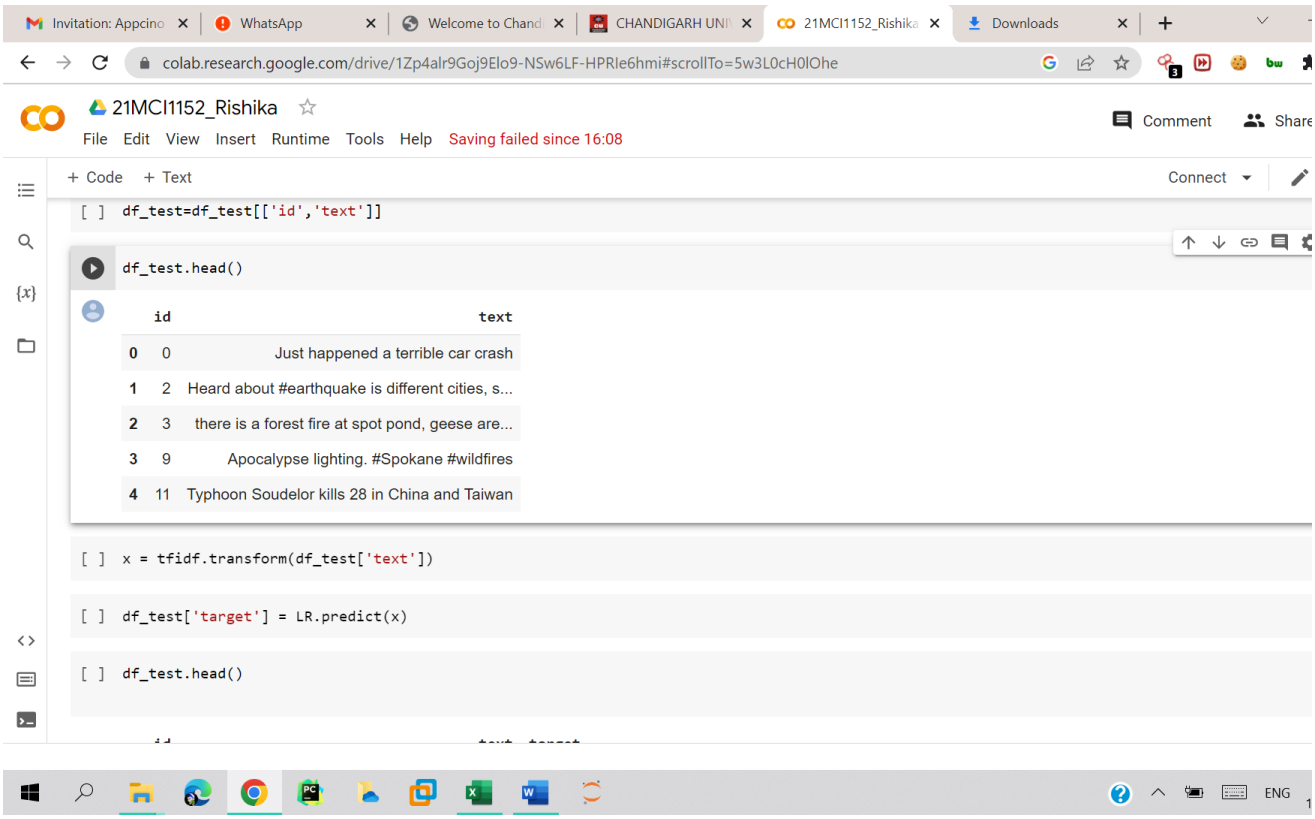
+ Code + Text Connect

```
print('Accuracy= ', accuracy_score(y_test, y_pred)*100, '%\n')

print('-Confusion Matrix-\n')
cm=confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d')
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

```
Accuracy=  79.9080761654629 %

-Confusion Matrix-

Text(33.0, 0.5, 'Truth')
```



EN



Invitation: Appcino | WhatsApp | Welcome to Chand | CHANDIGARH UNI | 21MCI1152_Rishika | Downloads

colab.research.google.com/drive/1Zp4alr9Goj9Elo9-NSw6LF-HPRIe6hmi#scrollTo=5w3L0cH0lOhe

21MCI1152_Rishika

File Edit View Insert Runtime Tools Help Saving failed since 16:08

Comment Share

+ Code + Text

Connect

```
[ ] df_test=df_test[['id','text']]
```

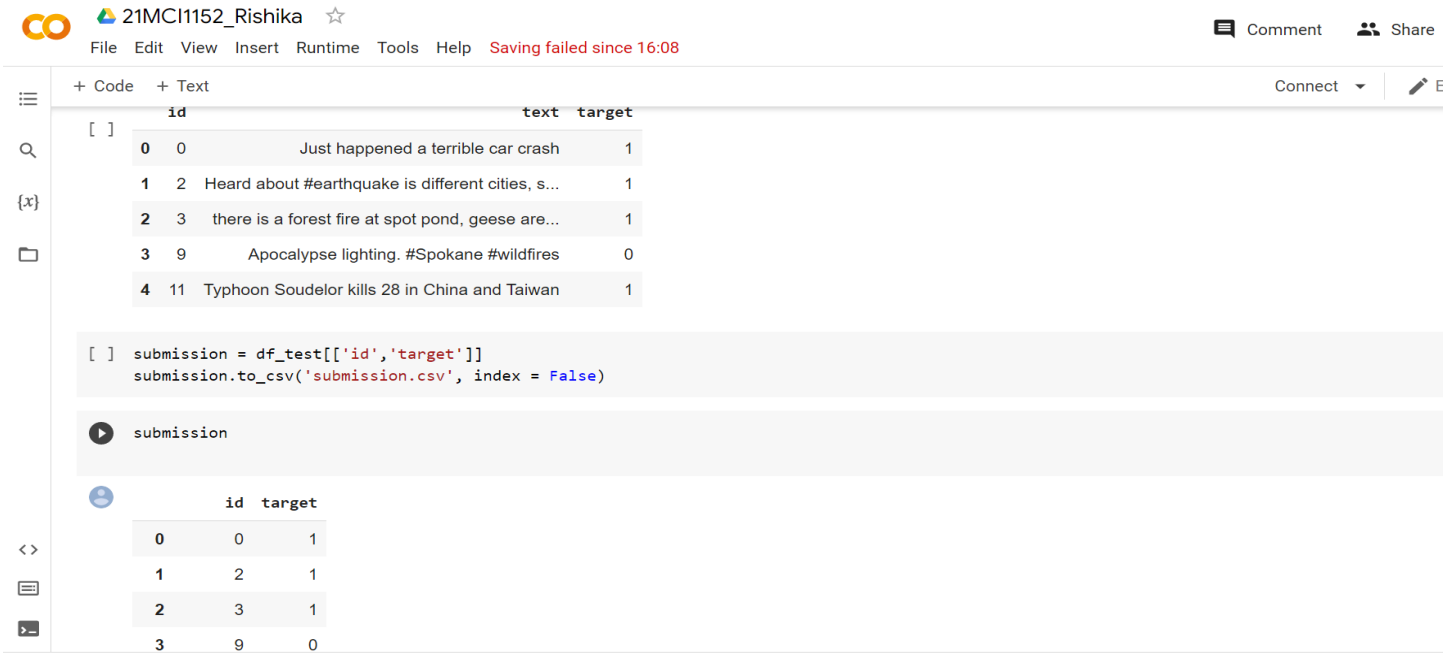
```
df_test.head()
```

	id	text
0	0	Just happened a terrible car crash
1	2	Heard about #earthquake is different cities, s...
2	3	there is a forest fire at spot pond, geese are...
3	9	Apocalypse lighting. #Spokane #wildfires
4	11	Typhoon Soudelor kills 28 in China and Taiwan

```
[ ] x = tfidf.transform(df_test['text'])
```

```
[ ] df_test['target'] = LR.predict(x)
```

```
[ ] df_test.head()
```



21MCI1152_Rishika

File Edit View Insert Runtime Tools Help Saving failed since 16:08

Comment Share

+ Code + Text

Connect

```
[ ]
```

	id	text	target
0	0	Just happened a terrible car crash	1
1	2	Heard about #earthquake is different cities, s...	1
2	3	there is a forest fire at spot pond, geese are...	1
3	9	Apocalypse lighting. #Spokane #wildfires	0
4	11	Typhoon Soudelor kills 28 in China and Taiwan	1

```
[ ] submission = df_test[['id','target']]
    submission.to_csv('submission.csv', index = False)
```

```
submission
```

	id	target
0	0	1
1	2	1
2	3	1
3	9	0

Learning outcomes (What I have learnt):

1. **I have learnt about the implementation and working of Decision Tree.**
2. **I have learnt about how to create Decision Tree Classifier and training on dataset.**
3. **I have learnt how to do prediction on validation data and check accuracy of model.**
4. **I have learnt to plot and visualize Decision Tree Classifier model.**

Evaluation Grid:

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Demonstration and Performance (Pre Lab Quiz)		5
2.	Worksheet		10
3.	Post Lab Quiz		5