10, Main Bazar, purkazi,
Muzaffarnagar, UP 📍

SIDDHANT RAI JAIN

+91 6397355482 ☎

jain10siddhant@gmail.com ✉

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

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

Practical WORKSHEET

Student Name: Siddhant Rai Jain	**UID:** 21MCI1182
Branch: MCA (AIML)	**Section/Group:** 21MAM1/B
Semester: III	**Date of Performance:**14/11/22
Subject Name: Machine Learning Lab	**Subject Code:** 21CAP-722

1) 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

Steps for experiment/practical:

```
# Importing Libraries
import pandas as pd
import numpy as np
import sys
import re
import string
import contractions
from sklearn.model_selection import train_test_split
import ktrain
import tensorflow as tf
from ktrain import text
df_train = pd.read_csv('/content/train_data_cleaning.csv')
df_train
df_train.dtypes
```

```
df_val = pd.read_csv('/content/test_data_cleaning.csv')
df_train['target'].value_counts(normalize=True)
sum(df_train.keyword.isna())
sum(df_train.location.isna())
# Droping keyword and location columns
df_train.drop(columns=['keyword', 'location' ,'id'], inplace=True)
df_train

#We'll remove hashtags(#example), @username and links(starting with h
ttp:// or https://) only.
# As we are going to use BERT, we are not removing emoticons as it wi
ll help BERT in prediction.
#We will again do text pre-processing later using BERT.
def pre_process(tweet):
    tweet = ' '.join(re.sub("(@[A-Za-z0-9_]+)|(#[A-Za-z0-
9]+)", " ", tweet).split())  # remove #tags and @usernames
    tweet = ' '.join(re.sub("(\w+:\/\/\S+)", " ", tweet).split()) # r
emove urls
    return(tweet)

def pre_process1(tweet):
    tweet = ' '.join(re.sub("(\w+:\/\/\S+)", " ", tweet).split()) # r
emove urls
    return(tweet)
#@title Handling constractions: Below funnction will replace constact
ions (e.g. wouldn't to would not).
def fn_contractions(tweet):
    expanded_words = []
    for word in tweet.split():
        expanded_words.append(contractions.fix(word))
    return(' '.join(expanded_words))
df_train['text'] = df_train['text'].apply(lambda x:pre_process(x))
df_train
df_train['text'] = df_train['text'].apply(lambda x:fn_contractions(x)
)
df_train
df_val['text'] = df_val['text'].apply(lambda x:pre_process(x))
df_val['text'] = df_val['text'].apply(lambda x:fn_contractions(x))
df_val
#@title split data for train and test
train, test = train_test_split(df_train, test_size=0.2)
X_train = train.text.tolist()
```

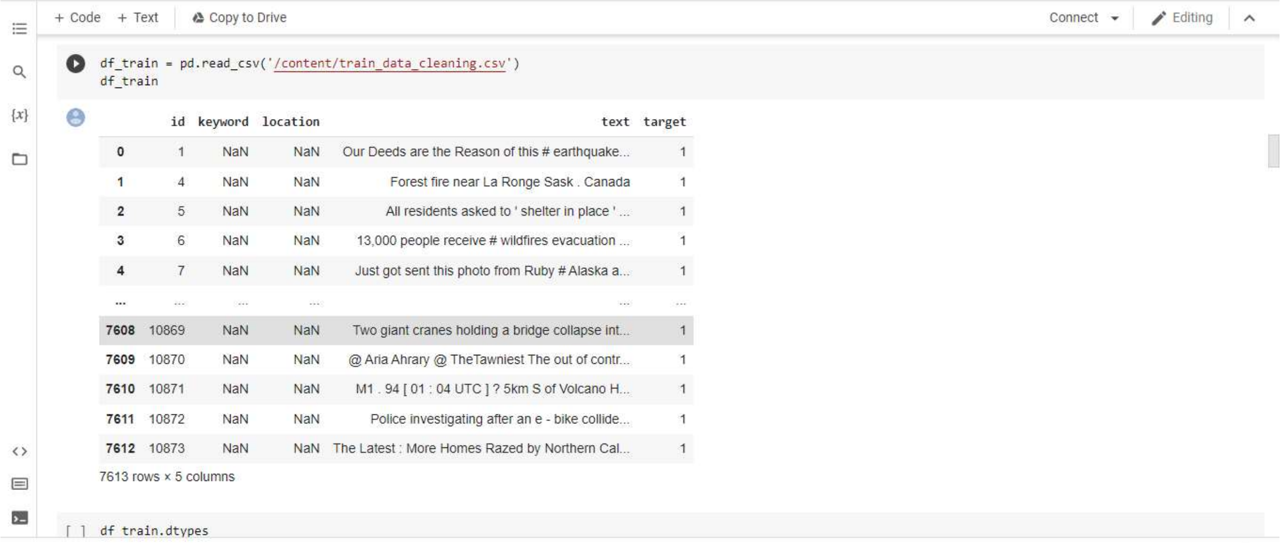
```
X_test = test.text.tolist()
y_train = train.target.tolist()
y_test = test.target.tolist()
X_train[:10]
y_train[:10]
print(len(X_train),len(X_test),len(y_train),len(y_test))
#@title Model building using BERT
# We are using bert-base-uncased model. You can choose any other model. I am selecting maxlen of tokenization as 512 (it's max for BERT).
model_arch ='bert-base-uncased'
factors = [0,1] # We have two factors to predict.
MAXLEN = 512
trans = text.Transformer(model_arch, maxlen=MAXLEN, class_names= factors)
train_data = trans.preprocess_train(X_train,y_train)
test_data = trans.preprocess_test(X_test,y_test)
model = trans.get_classifier()
learner = ktrain.get_learner(model, train_data=train_data, val_data=test_data, batch_size=10)
learner.fit_onecycle(3e-5, 4)
learner.validate(val_data=test_data, class_names=factors)
predictor = ktrain.get_predictor(learner.model, preproc=trans)
```

#Prediction

```
df_val['target'] = predictor.predict(df_val.text.tolist())
df_val
df_val.to_csv('/working/test_result_final.csv', index=False)
df_submission = df_val[['id','target']]
df_submission.to_csv('/working/submission5.csv', index=False)
```

3) Output

Data Output

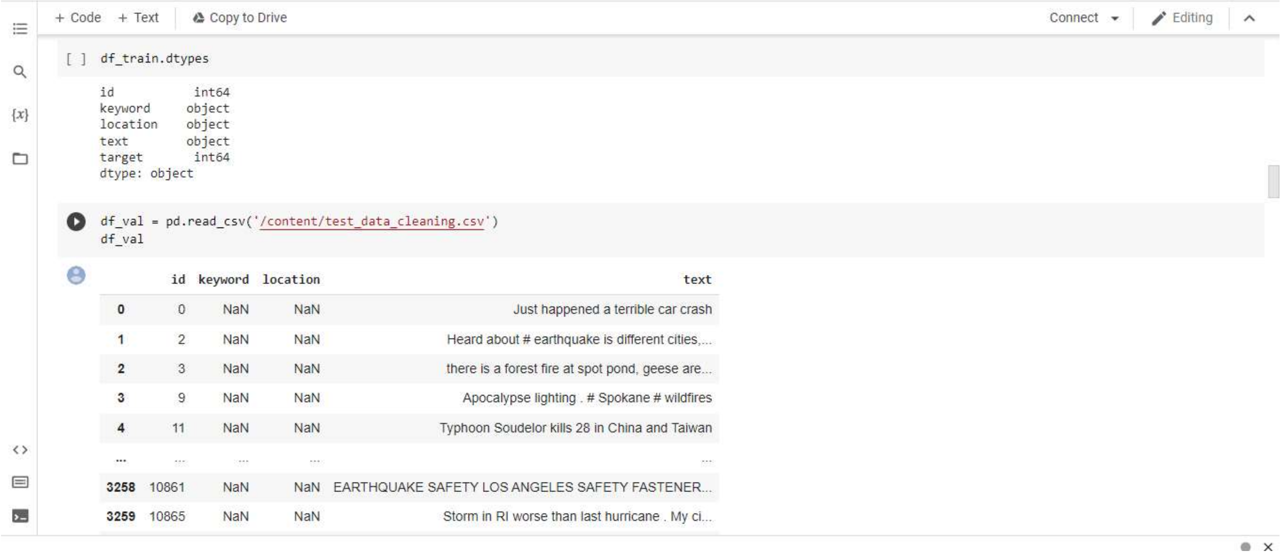


```
df_train = pd.read_csv('/content/train_data_cleaning.csv')
df_train
```

	id	keyword	location	text	target
0	1	NaN	NaN	Our Deeds are the Reason of this # earthquake...	1
1	4	NaN	NaN	Forest fire near La Ronge Sask . Canada	1
2	5	NaN	NaN	All residents asked to ' shelter in place ' ...	1
3	6	NaN	NaN	13,000 people receive # wildfires evacuation ...	1
4	7	NaN	NaN	Just got sent this photo from Ruby # Alaska a...	1
...	...	...	...	...	...
7608	10869	NaN	NaN	Two giant cranes holding a bridge collapse int...	1
7609	10870	NaN	NaN	@ Aria Ahrary @ TheTawniest The out of contr...	1
7610	10871	NaN	NaN	M1 . 94 [01 : 04 UTC] ? 5km S of Volcano H...	1
7611	10872	NaN	NaN	Police investigating after an e - bike collide...	1
7612	10873	NaN	NaN	The Latest : More Homes Razed by Northern Cal...	1

7613 rows × 5 columns

```
[ ] df_train.dtypes
```

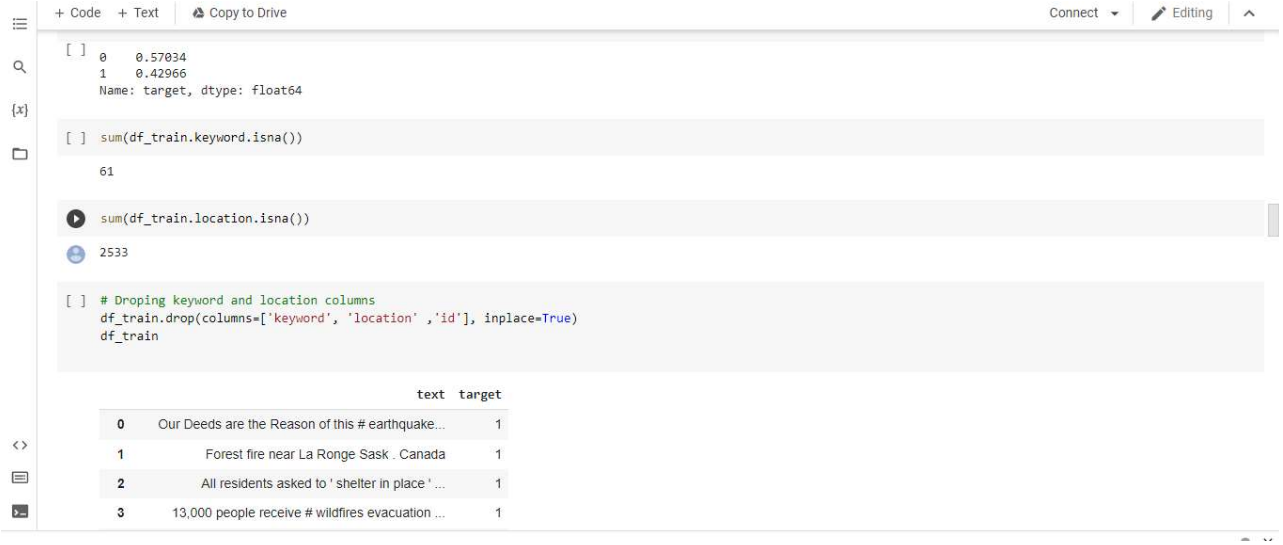


```
[ ] df_train.dtypes
```

```
id           int64
keyword     object
location    object
text        object
target       int64
dtype: object
```

```
df_val = pd.read_csv('/content/test_data_cleaning.csv')
df_val
```

	id	keyword	location	text
0	0	NaN	NaN	Just happened a terrible car crash
1	2	NaN	NaN	Heard about # earthquake is different cities,...
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
...	...	...	...	...
3258	10861	NaN	NaN	EARTHQUAKE SAFETY LOS ANGELES SAFETY FASTENER...
3259	10865	NaN	NaN	Storm in RI worse than last hurricane . My ci...



```
0    0.57034
1    0.42966
Name: target, dtype: float64
```

```
sum(df_train.keyword.isna())
```

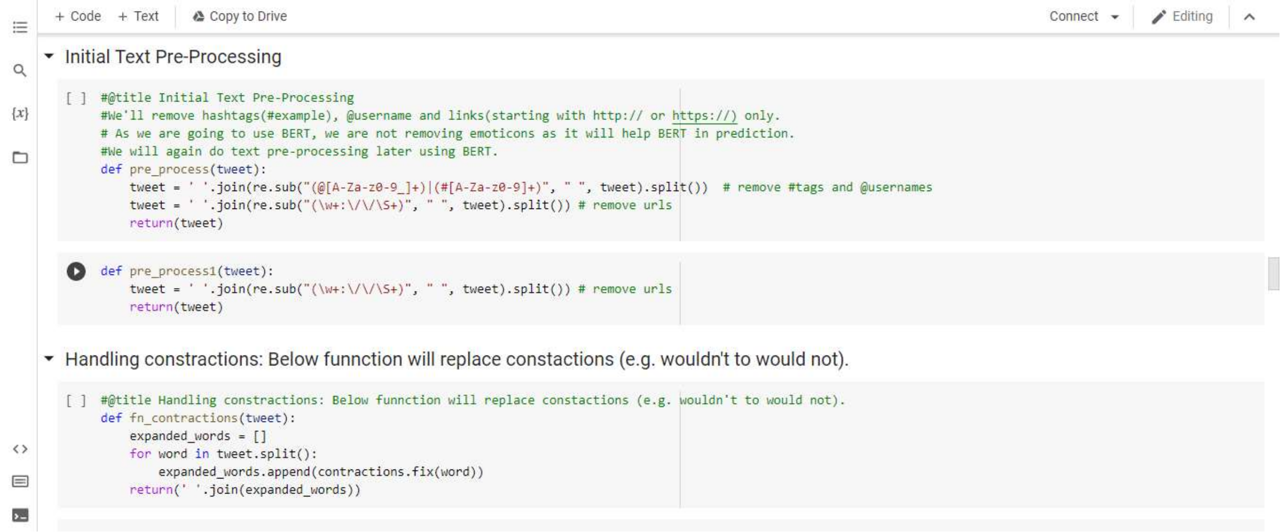
```
61
```

```
sum(df_train.location.isna())
```

```
2533
```

```
# Droping keyword and location columns
df_train.drop(columns=['keyword', 'location' ,'id'], inplace=True)
df_train
```

	text	target
0	Our Deeds are the Reason of this # earthquake...	1
1	Forest fire near La Ronge Sask . Canada	1
2	All residents asked to ' shelter in place ' ...	1
3	13,000 people receive # wildfires evacuation ...	1



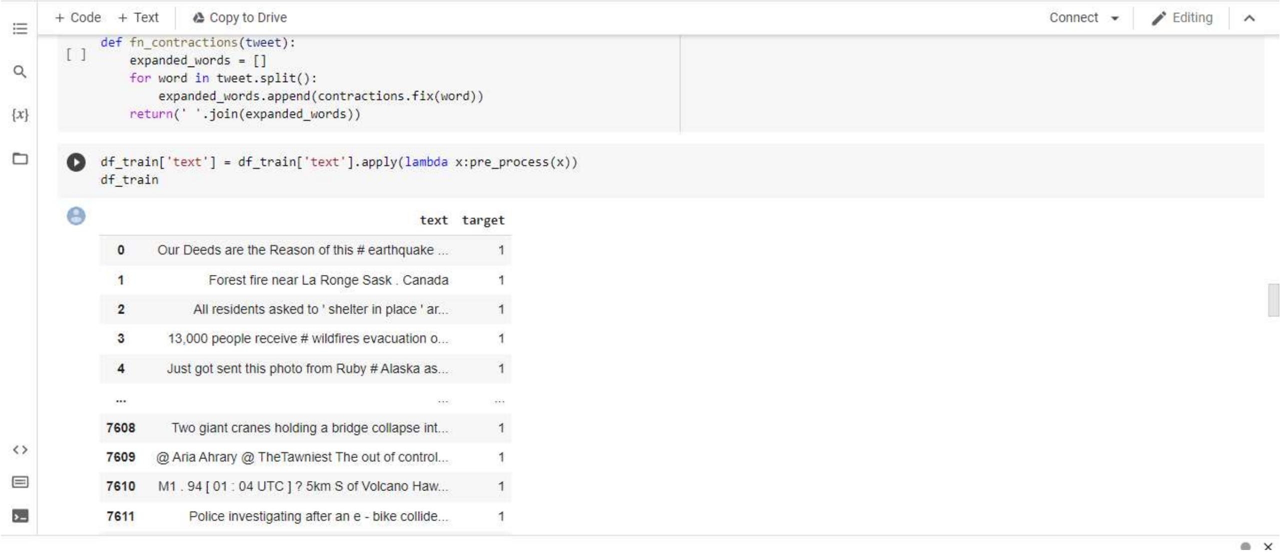
Initial Text Pre-Processing

```
#@title Initial Text Pre-Processing
#We'll remove hashtags(#example), @username and links(starting with http:// or https://) only.
# As we are going to use BERT, we are not removing emoticons as it will help BERT in prediction.
#We will again do text pre-processing later using BERT.
def pre_process(tweet):
    tweet = ' '.join(re.sub("(@[A-Za-z0-9_]+)|(#[A-Za-z0-9]+)", " ", tweet).split())  # remove #tags and @usernames
    tweet = ' '.join(re.sub("(\w+:\/\/\S+)", " ", tweet).split()) # remove urls
    return(tweet)
```

```
def pre_process1(tweet):
    tweet = ' '.join(re.sub("(\w+:\/\/\S+)", " ", tweet).split()) # remove urls
    return(tweet)
```

Handling constractions: Below funnction will replace constactions (e.g. wouldn't to would not).

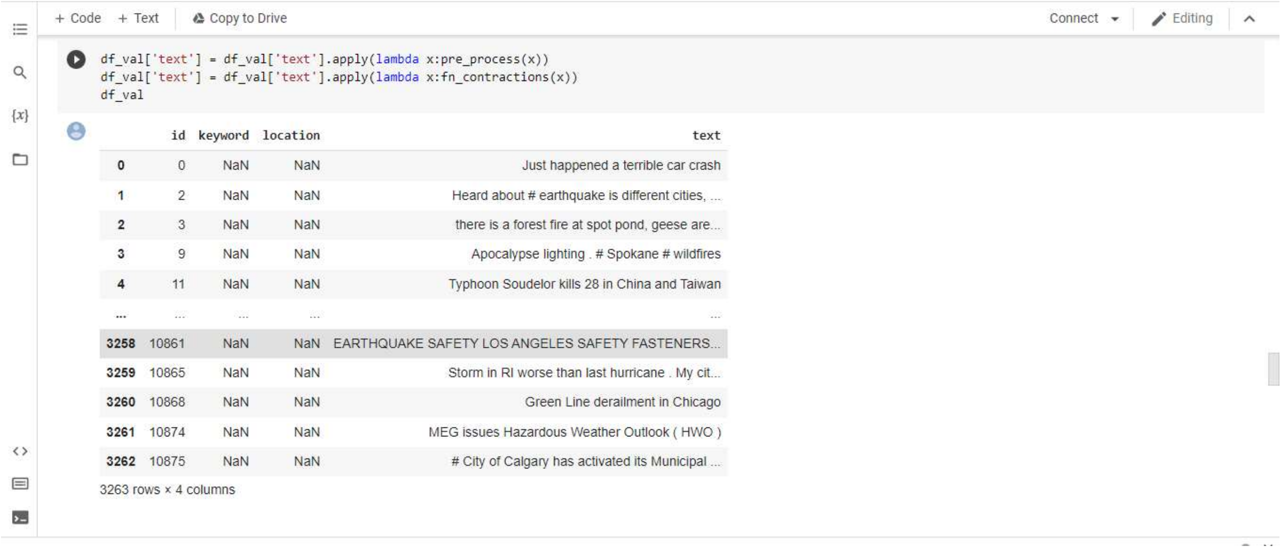
```
#@title Handling constractions: Below funnction will replace constactions (e.g. wouldn't to would not).
def fn_contractions(tweet):
    expanded_words = []
    for word in tweet.split():
        expanded_words.append(contractions.fix(word))
    return(' '.join(expanded_words))
```



```
[ ]  def fn_contractions(tweet):
         expanded_words = []
         for word in tweet.split():
             expanded_words.append(contractions.fix(word))
         return(' '.join(expanded_words))
```

```
df_train['text'] = df_train['text'].apply(lambda x:pre_process(x))
df_train
```

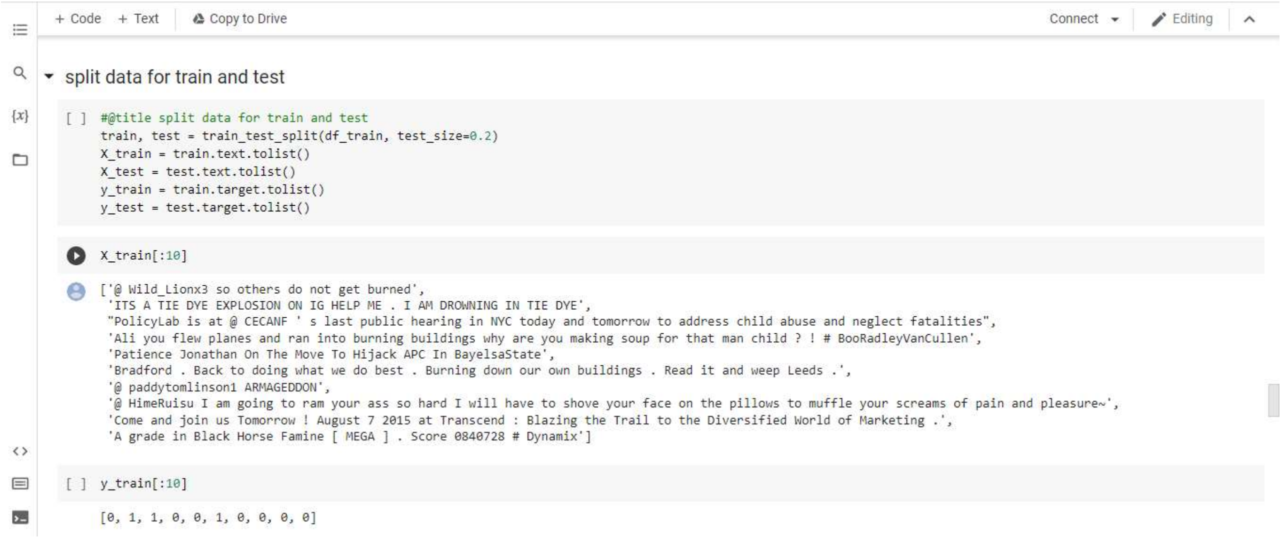
```
                                                   text  target
0        Our Deeds are the Reason of this # earthquake ...       1
1                   Forest fire near La Ronge Sask . Canada       1
2             All residents asked to ' shelter in place ' ar...       1
3          13,000 people receive # wildfires evacuation o...       1
4           Just got sent this photo from Ruby # Alaska as...       1
...                                                     ...     ...
7608        Two giant cranes holding a bridge collapse int...       1
7609     @ Aria Ahrary @ TheTawniest The out of control...       1
7610     M1 . 94 [ 01 : 04 UTC ] ? 5km S of Volcano Haw...       1
7611         Police investigating after an e - bike collide...       1
```



```
df_val['text'] = df_val['text'].apply(lambda x:pre_process(x))
df_val['text'] = df_val['text'].apply(lambda x:fn_contractions(x))
df_val
```

```
         id  keyword  location                                                text
0         0      NaN       NaN                           Just happened a terrible car crash
1         2      NaN       NaN                     Heard about # earthquake is different cities, ...
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
...     ...      ...       ...                                                                 ...
3258  10861      NaN       NaN  EARTHQUAKE SAFETY LOS ANGELES SAFETY FASTENERS...
3259  10865      NaN       NaN                    Storm in RI worse than last hurricane . My cit...
3260  10868      NaN       NaN                                 Green Line derailment in Chicago
3261  10874      NaN       NaN                   MEG issues Hazardous Weather Outlook ( HWO )
3262  10875      NaN       NaN                      # City of Calgary has activated its Municipal ...

3263 rows × 4 columns
```



split data for train and test

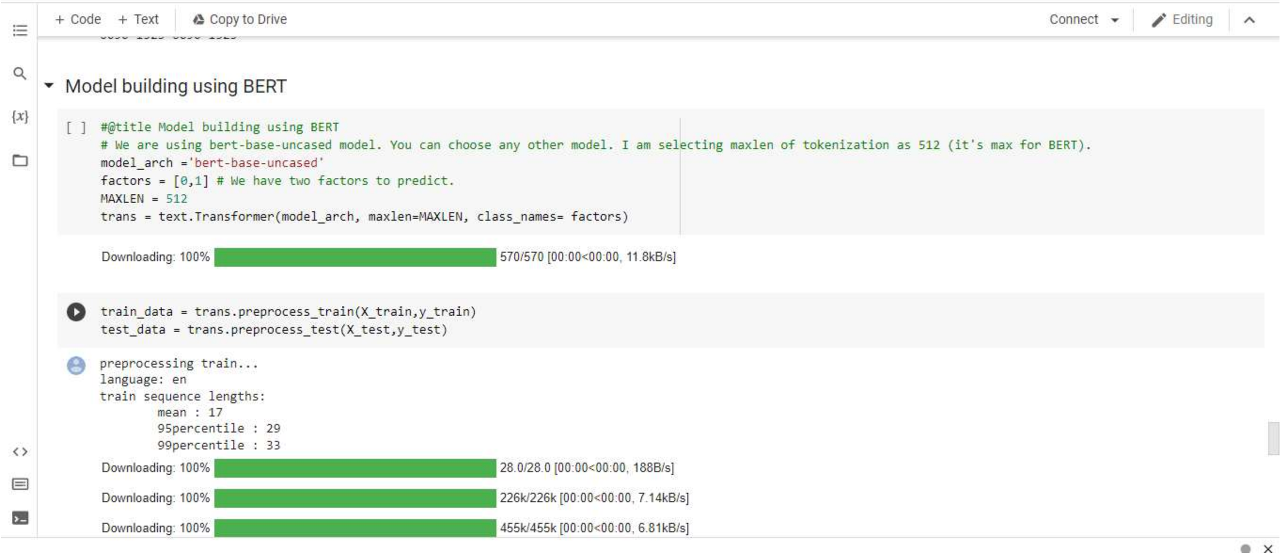
```
[ ] #@title split data for train and test
    train, test = train_test_split(df_train, test_size=0.2)
    X_train = train.text.tolist()
    X_test = test.text.tolist()
    y_train = train.target.tolist()
    y_test = test.target.tolist()
```

```
X_train[:10]
```

```
['@ Wild_Lionx3 so others do not get burned',
 'ITS A TIE DYE EXPLOSION ON IG HELP ME . I AM DROWNING IN TIE DYE',
 "PolicyLab is at @ CECANF ' s last public hearing in NYC today and tomorrow to address child abuse and neglect fatalities",
 'Ali you flew planes and ran into burning buildings why are you making soup for that man child ? ! # BooRadleyVanCullen',
 'Patience Jonathan On The Move To Hijack APC In BayelsaState',
 'Bradford . Back to doing what we do best . Burning down our own buildings . Read it and weep Leeds .',
 '@ paddytomlinson1 ARMAGEDDON',
 '@ HimeRuisu I am going to ram your ass so hard I will have to shove your face on the pillows to muffle your screams of pain and pleasure~',
 'Come and join us Tomorrow ! August 7 2015 at Transcend : Blazing the Trail to the Diversified World of Marketing .',
 'A grade in Black Horse Famine [ MEGA ] . Score 0840728 # Dynamix']
```

```
[ ] y_train[:10]
```

```
[0, 1, 1, 0, 0, 1, 0, 0, 0, 0]
```



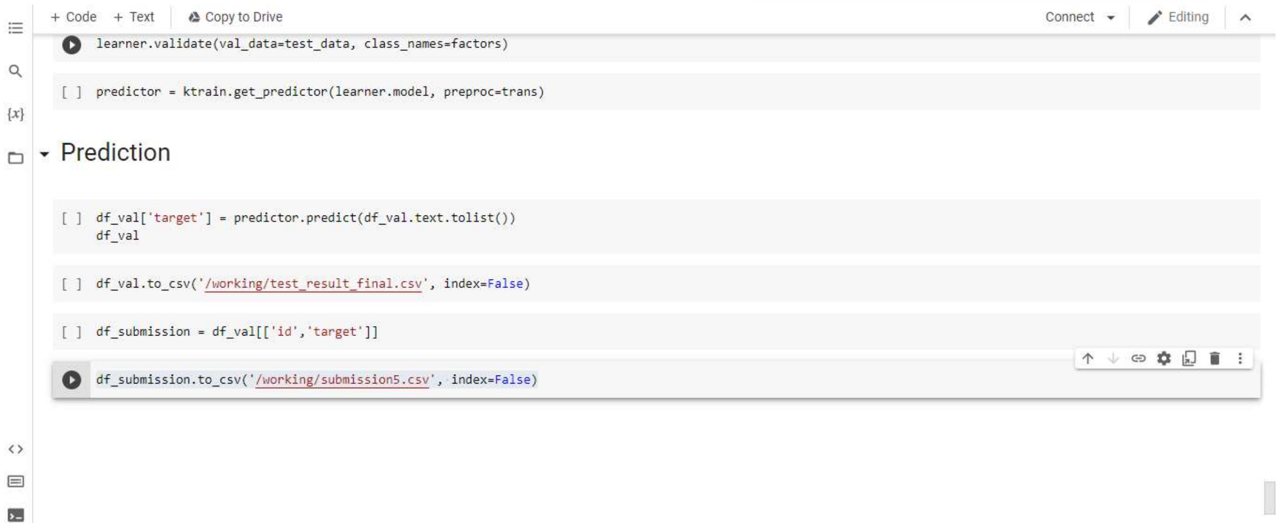
Model building using BERT

```
[ ] #@title Model building using BERT
    # We are using bert-base-uncased model. You can choose any other model. I am selecting maxlen of tokenization as 512 (it's max for BERT).
    model_arch ='bert-base-uncased'
    factors = [0,1] # We have two factors to predict.
    MAXLEN = 512
    trans = text.Transformer(model_arch, maxlen=MAXLEN, class_names= factors)
```

```
Downloading: 100% 570/570 [00:00<00:00, 11.8kB/s]
```

```
train_data = trans.preprocess_train(X_train,y_train)
test_data = trans.preprocess_test(X_test,y_test)
```

```
preprocessing train...
language: en
train sequence lengths:
        mean : 17
        95percentile : 29
        99percentile : 33
Downloading: 100% 28.0/28.0 [00:00<00:00, 188B/s]
Downloading: 100% 226k/226k [00:00<00:00, 7.14kB/s]
Downloading: 100% 455k/455k [00:00<00:00, 6.81kB/s]
```



```
learner.validate(val_data=test_data, class_names=factors)
```

```
[ ] predictor = ktrain.get_predictor(learner.model, preproc=trans)
```

Prediction

```
[ ] df_val['target'] = predictor.predict(df_val.text.tolist())
    df_val
```

```
[ ] df_val.to_csv('/working/test_result_final.csv', index=False)
```

```
[ ] df_submission = df_val[['id','target']]
```

```
df_submission.to_csv('/working/submission5.csv', index=False)
```

4) Learning outcomes (What I have learnt):

1. Learn the concept of transformer
2. Learn to implement linear regression on data.