In [ ]:

```
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from nltk.tokenize import word_tokenize
from nltk import NaiveBayesClassifier
import numpy as np
import nltk
import re
import random
import string
```

In [ ]:

```
!pip install swachhdata -q
```

In [ ]:

```
from swachhdata.text import TweetExtractor
```

IMPORTING DATA

In [ ]:

```
keys = {'consumer_key': 'fZhACw4QvjRUHgqFt7GCffCWT',
        'consumer_secret': '4wfdmkTxt5FuzDsOvO4PZp9Jk05DwAg5KAkKbZZi8B9FjCD2LK',
        'access_token': '2276034169-3P9mubPsEn2sab3AkWzuWY1SSanYs6LBuKsfSG3',
        'access_token_secret': 'CtfgyPmKUGCJOxcNZKzUAprVd0XyCreneiJ9QQL0tNBl2'}
te = TweetExtractor(keys)
```

In [ ]:

```
df = te.extract(keyword='covid19', count=300)
```

In [ ]:

```
df
```

Out[ ]:

	ID	User	Tweets	fav_count	rt_count	tweet_date
0	1359850836738859013	Assem Sweidan	Consqequnces of @Conservatives #ToryScumbags L...	0	0	2021-02-11 13:04:38
1	1359850835119865859	Neighborhood Health Centers Lehigh Valley	RT @PAHealthDept: None of the #COVID19 vaccine...	0	2	2021-02-11 13:04:38
2	1359850831546376195	Ye Htet Myint	RT @MayWongCNA: #Myanmar #military chief speak...	0	638	2021-02-11 13:04:37
3	1359850830313082882	Nikhil Singh	RT @htTweets: CAA will be implemented after #C...	0	4	2021-02-11 13:04:37
4	1359850829809922049	Lovescience	RT @HarbRimah: Sharing Valuable Data Will Resu...	0	14	2021-02-11 13:04:37
...	...	...	...	...	...	...
295	1359850318809415683	Ward Jolles	Sources: https://t.co/rxLzePCZTE	0	0	2021-02-11 13:02:35
296	1359850317958045696	Neurosurgery at NM	RT @CraigHorbinski: #Pituitary infarct caused ...	0	6	2021-02-11 13:02:35
297	1359850314870972420	Averlon Toussaint	RT @AdamBowers7: Women scientists are playing ...	0	3	2021-02-11 13:02:34
298	1359850310810877953	Dominic Pritchard #FBPE #BlackLivesMatter 3.5%	RT @DrEricDing: I feel like the guy on the rig...	0	1408	2021-02-11 13:02:33

	ID	User	Tweets	fav_count	rt_count	tweet_date
299	1359850308910809090	OMG TEA	5 star review from Timothy B. https://t....	0	0	2021-02-11 13:02:33

300 rows × 6 columns

In [ ]:

```
import pandas as pd

df=pd.read_excel("my_tweets.xlsx")
```

In [ ]:

```
df['Tweets'] = df['Tweets'].str.lower()
```

In [ ]:

```
from swachhdata.text import *
```

In [ ]:

```
url = urlRecast(process='remove', verbose=1)
df['Tweets'] = url.setup_recast(df['Tweets'])
```

In [ ]:

```
html = htmlRecast(verbose=1)
df['Tweets'] = html.setup_recast(df['Tweets'])
```

In [ ]:

```
esr = EscapeSequenceRecast(verbose=1)
df['Tweets'] = esr.setup_recast(df['Tweets'])
```

In [ ]:

```
mention = MentionRecast(process='remove', verbose=1)
df['Tweets'] = mention.setup_recast(df['Tweets'])
```

In [ ]:

```
contraction = ContractionsRecast(verbose=1)
df['Tweets'] = contraction.setup_recast(df['Tweets'])
```

In [ ]:

```
emoji = EmojiRecast(process='remove', verbose=1)
df['Tweets'] = emoji.setup_recast(df['Tweets'])
```

In [ ]:

```
hashtag = HashtagRecast(process='remove', verbose=1)
df['Tweets'] = hashtag.setup_recast(df['Tweets'])
```

In [ ]:

```
stopwords = StopWordsRecast(package='nltk', verbose=1)
df['Tweets'] = stopwords.setup_recast(df['Tweets'])
```

In [ ]:

```
num = NumberRecast(process='remove', verbose=1)
df['Tweets'] = num.setup_recast(df['Tweets'])
```

In [ ]:

```
alp = AlphabetRecast(verbose=1)
df['Tweets'] = alp.setup_recast(df['Tweets'])
```

In [ ]:

```
punc = PunctuationRecast(verbose=1)
df['Tweets'] = punc.setup_recast(df['Tweets'])
```

In [ ]:

```
lem = LemmatizationRecast(verbose=1)
df['Tweets'] = lem.setup_recast(df['Tweets'])
```

In [ ]:

```
lem = LemmatizationRecast(verbose=1)
df['Tweets'] = lem.setup_recast(df['Tweets'])
```

In [ ]:

```
from textblob import TextBlob
import nltk
#nltk.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer
nltk.download('vader_lexicon', quiet=True)
sid = SentimentIntensityAnalyzer()

from operator import add
from itertools import starmap
import pandas as pd


def sentiment_model(**kwargs):

    if 'data' in kwargs and 'col_index' in kwargs:

        text = kwargs['data']

        if (text.isnull().sum() > 0)[0]:
            print(f'{text.isnull().sum()} NULL rows found!')
            text.dropna(inplace=True)

        text = list(text.iloc[:,kwargs['col_index']])

        #TextBlob Model
        if kwargs['textblob']:

            sent_tb = []
            for sentence in text:
                sent_tb.append(TextBlob(sentence).sentiment)

            sent_tb = pd.DataFrame(sent_tb)

        #NLTK Model
```

```
        if kwargs['nltk']:

            sent_nltk = []
            for sentence in text:
                sent_nltk.append(sid.polarity_scores(sentence))

            sent_nltk = pd.DataFrame(sent_nltk)

        if kwargs['nltk'] and kwargs['textblob']:

            if kwargs['add_text']:
                sent_data = pd.concat([pd.DataFrame(text).reset_index(drop=True),
                                       sent_tb.reset_index(drop=True),
                                       sent_nltk.reset_index(drop=True)],
                                       axis=1)

            else:
                sent_data = pd.concat([sent_tb.reset_index(drop=True),
                                       sent_nltk.reset_index(drop=True)], axis=1)

        elif kwargs['add_text']:

            if kwargs['textblob']:
                sent_data = pd.concat([pd.DataFrame(text).reset_index(drop=True),
                                       sent_tb.reset_index(drop=True)],
                                       axis=1)

            elif kwargs['nltk']:
                sent_data = pd.concat([pd.DataFrame(text).reset_index(drop=True),
                                       sent_nltk.reset_index(drop=True)],
                                       axis=1)

        else:

            if kwargs['textblob']:
                return sent_tb

            elif kwargs['nltk']:
                return sent_nltk

        return sent_data

    else:

        print('Arguments Required: name= \'data\',\t type= pandas.DataFrame \n\t\t    na
me= \'col_index\', type= int')
```

In [ ]:

```
sentiment = sentiment_model(data = df,
                            col_index = 3,
                            textblob = True,
                            nltk = True,
                            add_text = True)
```

In [ ]:

```
sentiment
```

Out[ ]:

	0	polarity	subjectivity	neg	neu	pos	compound
0	consqequnces lie lie passenger locator form ye...	0.000000	0.000000	0.231	0.769	0.000	-0.5106
1	rt none vaccine currently authorize use u use ...	0.068182	0.450000	0.000	1.000	0.000	0.0000
2	rt chief speaks nation nd time via statement s...	0.000000	0.000000	0.118	0.882	0.000	-0.2500
3	rt caa implement vaccination end say hm	0.000000	0.000000	0.000	1.000	0.000	0.0000
4	rt share valuable data result new business model	0.136364	0.454545	0.000	0.531	0.469	0.6486

	...	...	...	...	...	...	...
	0	polarity	subjectivity	neg	neu	pos	compound
295	source	0.000000	0.000000	0.000	1.000	0.000	0.0000
296	rt infarct cause patient multiple infection in...	0.000000	0.000000	0.000	1.000	0.000	0.0000
297	rt woman scientist play vital role fight under...	0.100000	0.400000	0.160	0.429	0.411	0.4767
298	rt feel like guy right day honestly enjoy twee...	0.342857	0.517857	0.000	0.408	0.592	0.8910
299	star review timothy b	0.000000	0.000000	0.000	1.000	0.000	0.0000

300 rows × 7 columns

In [ ]:

```
sentiment.to_excel("covid1.xlsx")
```

In [ ]:

```
df = pd.read_excel("/content/covid19.xlsx")
df = pd.DataFrame(data= df)
```

In [ ]:

```
import nltk
word_data= df['Tweets']
nltk_tokens = word_data.apply(nltk.word_tokenize)
```

In [ ]:

```
nltk_tokens
```

Out[ ]:

```
0       [rt, crisis, first, unfolded, last, year, big,...
1       [rt, feel, like, guy, right, day, honestly, en...
2       [rt, test, could, use, school, ltcs, meat, pac...
3       [rt, pm, kyriakos, mitsotakis, tuesday, announ...
4                           [rt, a, course, big, driver]
                              ...
167     [rt, ontario, use, rapid, test, provide, feder...
168     [rt, chief, speaks, nation, nd, time, via, sta...
169     [rt, today, international, day, woman, scienti...
170     [rt, covid, highlight, perennial, pressure, gr...
171     [ontario, head, towards, rd, wave, predict, ex...
Name: Tweets, Length: 172, dtype: object
```

In [ ]:

```
frames= [nltk_tokens,df['Sentiment']]
dataset = pd.concat(frames,axis=1)
```

In [ ]:

```
X=dataset['Tweets']
Y=dataset['Sentiment']
```

LSTM Model

In [ ]:

```
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
tokenizer =Tokenizer(num_words=None, split=' ')
tokenizer.fit_on_texts(df['Tweets'].values)
vocabSize = len(tokenizer.word_index) + 1
X = tokenizer.texts_to_sequences(df['Tweets'])
X = pad_sequences(X)
Y=  df['Sentiment']
```

In [ ]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.3, random_state=69)
```

In [ ]:

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
from keras.layers import Embedding
model = Sequential()
model.add(Embedding(vocabSize, 128, input_length = 24))
model.add(LSTM(500, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(1, activation='sigmoid'))
```

In [ ]:

```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(X_train, Y_train, epochs =50)
```

```
Epoch 1/50
WARNING:tensorflow:Model was constructed with shape (None, 24) for input KerasTensor(type
_spec=TensorSpec(shape=(None, 24), dtype=tf.float32, name='embedding_1_input'), name='emb
edding_1_input', description="created by layer 'embedding_1_input'"), but it was called o
n an input with incompatible shape (None, 27).
WARNING:tensorflow:Model was constructed with shape (None, 24) for input KerasTensor(type
_spec=TensorSpec(shape=(None, 24), dtype=tf.float32, name='embedding_1_input'), name='emb
edding_1_input', description="created by layer 'embedding_1_input'"), but it was called o
n an input with incompatible shape (None, 27).
4/4 [==============================] - 6s 755ms/step - loss: 0.6712 - accuracy: 0.5921
Epoch 2/50
4/4 [==============================] - 3s 770ms/step - loss: 1.0940 - accuracy: 0.7058
Epoch 3/50
4/4 [==============================] - 3s 750ms/step - loss: 0.6520 - accuracy: 0.6996
Epoch 4/50
4/4 [==============================] - 3s 736ms/step - loss: 0.6452 - accuracy: 0.7277
Epoch 5/50
4/4 [==============================] - 3s 744ms/step - loss: 0.6080 - accuracy: 0.7538
Epoch 6/50
4/4 [==============================] - 3s 739ms/step - loss: 0.5415 - accuracy: 0.7538
Epoch 7/50
4/4 [==============================] - 3s 742ms/step - loss: 0.4943 - accuracy: 0.7454
Epoch 8/50
4/4 [==============================] - 3s 745ms/step - loss: 0.4930 - accuracy: 0.7133
Epoch 9/50
4/4 [==============================] - 4s 1s/step - loss: 0.4130 - accuracy: 0.7754
Epoch 10/50
4/4 [==============================] - 3s 785ms/step - loss: 0.3111 - accuracy: 0.7827
Epoch 11/50
4/4 [==============================] - 3s 768ms/step - loss: 0.2563 - accuracy: 0.9285
Epoch 12/50
4/4 [==============================] - 3s 782ms/step - loss: 0.1520 - accuracy: 0.9337
Epoch 13/50
4/4 [==============================] - 3s 762ms/step - loss: 0.1319 - accuracy: 0.9569
Epoch 14/50
4/4 [==============================] - 3s 775ms/step - loss: 0.1537 - accuracy: 0.9413
Epoch 15/50
4/4 [==============================] - 3s 737ms/step - loss: 0.1358 - accuracy: 0.9235
Epoch 16/50
4/4 [==============================] - 3s 730ms/step - loss: 0.1022 - accuracy: 0.9592
Epoch 17/50
4/4 [==============================] - 3s 746ms/step - loss: 0.0922 - accuracy: 0.9654
Epoch 18/50
4/4 [==============================] - 3s 728ms/step - loss: 0.0837 - accuracy: 0.9698
Epoch 19/50
4/4 [==============================] - 3s 742ms/step - loss: 0.1000 - accuracy: 0.9358
Epoch 20/50
4/4 [==============================] - 3s 745ms/step - loss: 0.0925 - accuracy: 0.9331
Epoch 21/50
4/4 [==============================] - 3s 744ms/step - loss: 0.0971 - accuracy: 0.9308
```

```
Epoch 22/50
4/4 [==============================] - 3s 737ms/step - loss: 0.0761 - accuracy: 0.9621
Epoch 23/50
4/4 [==============================] - 3s 745ms/step - loss: 0.0872 - accuracy: 0.9571
Epoch 24/50
4/4 [==============================] - 3s 741ms/step - loss: 0.0939 - accuracy: 0.9190
Epoch 25/50
4/4 [==============================] - 3s 741ms/step - loss: 0.1092 - accuracy: 0.9394
Epoch 26/50
4/4 [==============================] - 3s 748ms/step - loss: 0.0965 - accuracy: 0.9383
Epoch 27/50
4/4 [==============================] - 3s 777ms/step - loss: 0.0899 - accuracy: 0.9542
Epoch 28/50
4/4 [==============================] - 3s 739ms/step - loss: 0.0960 - accuracy: 0.9465
Epoch 29/50
4/4 [==============================] - 3s 751ms/step - loss: 0.0715 - accuracy: 0.9740
Epoch 30/50
4/4 [==============================] - 3s 740ms/step - loss: 0.0604 - accuracy: 0.9760
Epoch 31/50
4/4 [==============================] - 3s 740ms/step - loss: 0.0893 - accuracy: 0.9215
Epoch 32/50
4/4 [==============================] - 3s 737ms/step - loss: 0.1072 - accuracy: 0.9329
Epoch 33/50
4/4 [==============================] - 3s 739ms/step - loss: 0.0781 - accuracy: 0.9665
Epoch 34/50
4/4 [==============================] - 3s 767ms/step - loss: 0.0711 - accuracy: 0.9517
Epoch 35/50
4/4 [==============================] - 3s 751ms/step - loss: 0.1125 - accuracy: 0.9350
Epoch 36/50
4/4 [==============================] - 3s 742ms/step - loss: 0.1012 - accuracy: 0.9475
Epoch 37/50
4/4 [==============================] - 3s 777ms/step - loss: 0.1034 - accuracy: 0.9467
Epoch 38/50
4/4 [==============================] - 3s 752ms/step - loss: 0.0902 - accuracy: 0.9444
Epoch 39/50
4/4 [==============================] - 3s 847ms/step - loss: 0.0971 - accuracy: 0.9465
Epoch 40/50
4/4 [==============================] - 3s 824ms/step - loss: 0.0903 - accuracy: 0.9475
Epoch 41/50
4/4 [==============================] - 3s 767ms/step - loss: 0.0671 - accuracy: 0.9592
Epoch 42/50
4/4 [==============================] - 3s 740ms/step - loss: 0.0893 - accuracy: 0.9348
Epoch 43/50
4/4 [==============================] - 3s 742ms/step - loss: 0.0820 - accuracy: 0.9679
Epoch 44/50
4/4 [==============================] - 3s 779ms/step - loss: 0.0708 - accuracy: 0.9715
Epoch 45/50
4/4 [==============================] - 3s 757ms/step - loss: 0.0889 - accuracy: 0.9477
Epoch 46/50
4/4 [==============================] - 3s 739ms/step - loss: 0.0748 - accuracy: 0.9498
Epoch 47/50
4/4 [==============================] - 3s 787ms/step - loss: 0.0747 - accuracy: 0.9508
Epoch 48/50
4/4 [==============================] - 4s 876ms/step - loss: 0.0976 - accuracy: 0.9425
Epoch 49/50
4/4 [==============================] - 3s 771ms/step - loss: 0.0680 - accuracy: 0.9706
Epoch 50/50
4/4 [==============================] - 3s 770ms/step - loss: 0.0795 - accuracy: 0.9446
```

Out[ ]:

```
<tensorflow.python.keras.callbacks.History at 0x7fa8f0af0860>
```

In [ ]:

```
model.evaluate(X_test,Y_test)
```

```
WARNING:tensorflow:Model was constructed with shape (None, 24) for input KerasTensor(type
_spec=TensorSpec(shape=(None, 24), dtype=tf.float32, name='embedding_1_input'), name='emb
edding_1_input', description="created by layer 'embedding_1_input'"), but it was called o
n an input with incompatible shape (None, 27).
2/2 [==============================] - 1s 129ms/step - loss: 1.2943 - accuracy: 0.6731
```

Out[ ]:

```
[1.2942601442337036, 0.6730769276618958]
```

MULTINOMIAL NAIVE BAYES MODEL

In [ ]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer()
vector = tfidf.fit_transform(df['Tweets'])
X = vector.toarray()
Y= df['Sentiment']
```

In [ ]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.3, random_state=69)
```

In [ ]:

```
from sklearn.naive_bayes import MultinomialNB
multinom = MultinomialNB()
multinom.fit(X_train,Y_train)
ypred = multinom.predict(X_test)
```

In [ ]:

```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
print(accuracy_score(Y_test,ypred))
print(confusion_matrix(Y_test,ypred))
print(classification_report(Y_test,ypred))
```

```
0.7692307692307693
[[ 0 10]
 [ 2 40]]
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        10
           1       0.80      0.95      0.87        42

    accuracy                           0.77        52
   macro avg       0.40      0.48      0.43        52
weighted avg       0.65      0.77      0.70        52
```

GAUSSIAN NAIVE BAYES

In [ ]:

```
from sklearn.naive_bayes import GaussianNB
gauss = GaussianNB()
gauss.fit(X_train,Y_train)
ypred = gauss.predict(X_test)
```

In [ ]:

```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
print(accuracy_score(Y_test,ypred))
print(confusion_matrix(Y_test,ypred))
print(classification_report(Y_test,ypred))
```

```
0.6538461538461539
[[ 3  7]
 [11 31]]
              precision    recall  f1-score   support

           0       0.21      0.30      0.25        10
```

```
           1       0.82      0.74      0.78        42

    accuracy                           0.65        52
   macro avg       0.52      0.52      0.51        52
weighted avg       0.70      0.65      0.67        52
```

XGB CLASSIFIER

In [ ]:

```
from xgboost import XGBClassifier
model_xgb = XGBClassifier(max_depth=10,random_state=1,learning_rate=0.05,seed=1)
model_xgb.fit(X_train, Y_train)
y_pred=model_xgb.predict(X_test)

from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
print(accuracy_score(Y_test,y_pred))
print(confusion_matrix(Y_test,y_pred))
print(classification_report(Y_test,y_pred))
```

```
0.7692307692307693
[[ 3  7]
 [ 5 37]]
              precision    recall  f1-score   support

           0       0.38      0.30      0.33        10
           1       0.84      0.88      0.86        42

    accuracy                           0.77        52
   macro avg       0.61      0.59      0.60        52
weighted avg       0.75      0.77      0.76        52
```

FLASK

In [ ]:

```
model.save('/content/covid.h5')
```

In [ ]:

```
!pip install flask gevent requests pillow
```

```
Requirement already satisfied: flask in /usr/local/lib/python3.6/dist-packages (1.1.2)
Requirement already satisfied: gevent in /usr/local/lib/python3.6/dist-packages (21.1.2)
Requirement already satisfied: requests in /usr/local/lib/python3.6/dist-packages (2.23.0
)
Requirement already satisfied: pillow in /usr/local/lib/python3.6/dist-packages (7.0.0)
Requirement already satisfied: Jinja2>=2.10.1 in /usr/local/lib/python3.6/dist-packages (
from flask) (2.11.3)
Requirement already satisfied: itsdangerous>=0.24 in /usr/local/lib/python3.6/dist-packag
es (from flask) (1.1.0)
Requirement already satisfied: Werkzeug>=0.15 in /usr/local/lib/python3.6/dist-packages (
from flask) (1.0.1)
Requirement already satisfied: click>=5.1 in /usr/local/lib/python3.6/dist-packages (from
flask) (7.1.2)
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-packages (from
gevent) (53.0.0)
Requirement already satisfied: zope.interface in /usr/local/lib/python3.6/dist-packages (
from gevent) (5.2.0)
Requirement already satisfied: zope.event in /usr/local/lib/python3.6/dist-packages (from
gevent) (4.5.0)
Requirement already satisfied: greenlet<2.0,>=0.4.17; platform_python_implementation == "
CPython" in /usr/local/lib/python3.6/dist-packages (from gevent) (1.0.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.6/dist-packages (fr
om requests) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-packag
```

```
es (from requests) (2020.12.5)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.6/dist-package
s (from requests) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/
python3.6/dist-packages (from requests) (1.24.3)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.6/dist-packages
(from Jinja2>=2.10.1->flask) (1.1.1)
```

In [ ]:

```
!pip install flask-ngrok
from flask_ngrok import run_with_ngrok
from flask import Flask
```

```
Requirement already satisfied: flask-ngrok in /usr/local/lib/python3.6/dist-packages (0.0
.25)
Requirement already satisfied: Flask>=0.8 in /usr/local/lib/python3.6/dist-packages (from
flask-ngrok) (1.1.2)
Requirement already satisfied: requests in /usr/local/lib/python3.6/dist-packages (from f
lask-ngrok) (2.23.0)
Requirement already satisfied: click>=5.1 in /usr/local/lib/python3.6/dist-packages (from
Flask>=0.8->flask-ngrok) (7.1.2)
Requirement already satisfied: itsdangerous>=0.24 in /usr/local/lib/python3.6/dist-packag
es (from Flask>=0.8->flask-ngrok) (1.1.0)
Requirement already satisfied: Jinja2>=2.10.1 in /usr/local/lib/python3.6/dist-packages (
from Flask>=0.8->flask-ngrok) (2.11.3)
Requirement already satisfied: Werkzeug>=0.15 in /usr/local/lib/python3.6/dist-packages (
from Flask>=0.8->flask-ngrok) (1.0.1)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.6/dist-packages (fr
om requests->flask-ngrok) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/
python3.6/dist-packages (from requests->flask-ngrok) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-packag
es (from requests->flask-ngrok) (2020.12.5)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.6/dist-package
s (from requests->flask-ngrok) (3.0.4)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.6/dist-packages
(from Jinja2>=2.10.1->Flask>=0.8->flask-ngrok) (1.1.1)
```

In [ ]:

```
import pickle
with open('tokenizer.pickle', 'wb') as handle:
    pickle.dump(tokenizer, handle, protocol=pickle.HIGHEST_PROTOCOL)
```

In [ ]:

```
```

Webpage

In [ ]:

```
html = '''<!DOCTYPE HTML>
<html lang="en">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>Sentiment Analysis</title>
    </head>

    <body>

        <br>
        <center><h1 id="title">Sentiment Analysis</h1></center>
        <br>
        <form name="image_form" action="/" method="POST" enctype="multipart/form-data">
            <div >

                <div >
```

```
                    <textarea id="tweet" name="tweet" rows="4"></textarea>
                </div>

                <div>
                    <input id="predict" class="cust_button" name="button" type="submit"
value="Predict"/>
                </div>

                <div >
                    <span id="num" style="">{{pred}}</span>
                </div>
            </div>
        </form>

    </body>

</html>
'''
```

In [ ]:

```
!mkdir templates
!mkdir uploads
HTML_file = open('/content/templates/index.html', 'w')
HTML_file.write(html)
HTML_file.close()
```

```
mkdir: cannot create directory ‘templates’: File exists
mkdir: cannot create directory ‘uploads’: File exists
```

In [ ]:

```
import os
import pickle
import keras
import tensorflow as tf
from flask import Flask, render_template, request
from flask_ngrok import run_with_ngrok
from swachhdata.text import TextRecast

app = Flask(__name__, static_folder='/content/templates')
run_with_ngrok(app)

model = keras.models.load_model('/content/covid.h5')
with open('/content/tokenizer.pickle', 'rb') as handle:
    tokenizer = pickle.load(handle)

def process(tweet):
    return TextRecast(tweet, urlRecast = {'process': 'remove'},
                      htmlRecast = True,
                      EscapeSequenceRecast = True,
                      MentionRecast = {'process': 'extract_remove'},
                      ContractionsRecast = True,
                      CaseRecast = {'process': 'lower'},
                      EmojiRecast = {'process': 'remove', 'space_out': False},
                      HashtagRecast = {'process': 'remove'},
                      StopWordsRecast = {'package': 'nltk', 'stopwords': None},
                      NumberRecast = {'process': 'remove', 'seperator': None},
                      PunctuationRecast = True,
                      LemmatizationRecast = {'package':'nltk'})

def sentiment(tweet):
    tweet = tokenizer.texts_to_sequences(tweet)
    tweet = tf.keras.preprocessing.sequence.pad_sequences(tweet)
    return model.predict_classes(tweet)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/', methods=['GET', 'POST'])
```

```
def predict():
    if request.method == 'POST':
        tweet = request.form.to_dict(flat=True)
        tweet = tweet['tweet']
        tweet = process(tweet)
        tweet = sentiment([tweet])
        if tweet[0][0] == 0:
            prediction = 'Positive'
        else:
            prediction = 'Negative'
    return render_template('index.html', pred=prediction)

if __name__ == '__main__':
    app.run()
```

```
 * Serving Flask app "__main__" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
```

```
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

```
 * Running on http://0f13cb975c71.ngrok.io
 * Traffic stats available on http://127.0.0.1:4040
```

```
127.0.0.1 - - [11/Feb/2021 13:08:13] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [11/Feb/2021 13:08:14] "GET /favicon.ico HTTP/1.1" 404 -
```

```
WARNING:tensorflow:Model was constructed with shape (None, 24) for input KerasTensor(type
_spec=TensorSpec(shape=(None, 24), dtype=tf.float32, name='embedding_1_input'), name='emb
edding_1_input', description="created by layer 'embedding_1_input'"), but it was called o
n an input with incompatible shape (None, 1).
```

```
/usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/sequential.py:450:
UserWarning: `model.predict_classes()` is deprecated and will be removed after 2021-01-01
. Please use instead:* `np.argmax(model.predict(x), axis=-1)`,   if your model does multi
-class classification   (e.g. if it uses a `softmax` last-layer activation).* `(model.pre
dict(x) > 0.5).astype("int32")`,   if your model does binary classification   (e.g. if it
uses a `sigmoid` last-layer activation).
  warnings.warn('`model.predict_classes()` is deprecated and '
127.0.0.1 - - [11/Feb/2021 13:08:23] "POST / HTTP/1.1" 200 -
```

```
WARNING:tensorflow:Model was constructed with shape (None, 24) for input KerasTensor(type
_spec=TensorSpec(shape=(None, 24), dtype=tf.float32, name='embedding_1_input'), name='emb
edding_1_input', description="created by layer 'embedding_1_input'"), but it was called o
n an input with incompatible shape (None, 0).
```

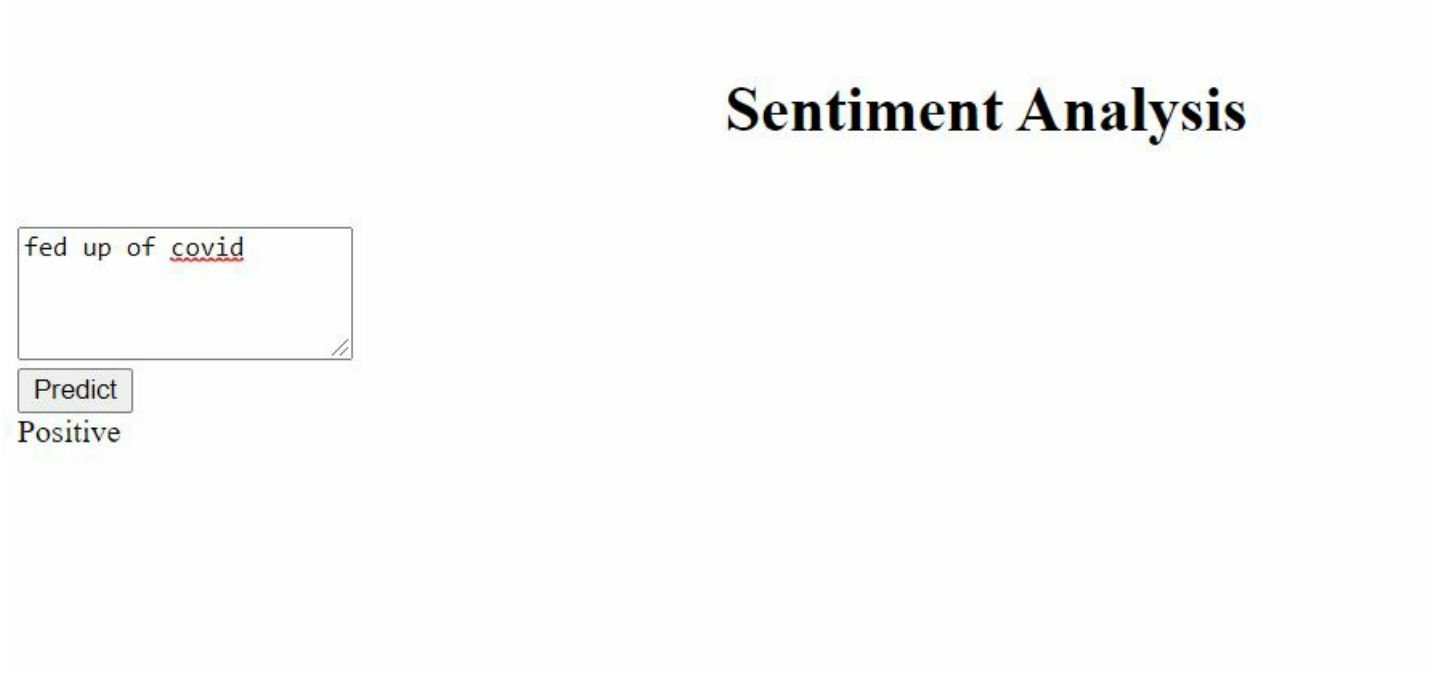
```
[2021-02-11 13:08:34,127] ERROR in app: Exception on / [POST]
Traceback (most recent call last):
  File "/usr/local/lib/python3.6/dist-packages/flask/app.py", line 2447, in wsgi_app
    response = self.full_dispatch_request()
  File "/usr/local/lib/python3.6/dist-packages/flask/app.py", line 1952, in full_dispatch
_request
    rv = self.handle_user_exception(e)
  File "/usr/local/lib/python3.6/dist-packages/flask/app.py", line 1821, in handle_user_e
xception
    reraise(exc_type, exc_value, tb)
  File "/usr/local/lib/python3.6/dist-packages/flask/_compat.py", line 39, in reraise
    raise value
  File "/usr/local/lib/python3.6/dist-packages/flask/app.py", line 1950, in full_dispatch
_request
    rv = self.dispatch_request()
  File "/usr/local/lib/python3.6/dist-packages/flask/app.py", line 1936, in dispatch_requ
est
    return self.view_functions[rule.endpoint](**req.view_args)
  File "<ipython-input-107-3a446e632f80>", line 45, in predict
    tweet = sentiment([tweet])
  File "<ipython-input-107-3a446e632f80>", line 33, in sentiment
    return model.predict_classes(tweet)
  File "/usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/sequential.
```

```
py", line 459, in predict_classes
    proba = self.predict(x, batch_size=batch_size, verbose=verbose)
  File "/usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/training.py
", line 1629, in predict
    tmp_batch_outputs = self.predict_function(iterator)
  File "/usr/local/lib/python3.6/dist-packages/tensorflow/python/eager/def_function.py",
line 828, in __call__
    result = self._call(*args, **kwds)
  File "/usr/local/lib/python3.6/dist-packages/tensorflow/python/eager/def_function.py",
line 862, in _call
    results = self._stateful_fn(*args, **kwds)
  File "/usr/local/lib/python3.6/dist-packages/tensorflow/python/eager/function.py", line
2941, in __call__
    filtered_flat_args) = self._maybe_define_function(args, kwargs)
  File "/usr/local/lib/python3.6/dist-packages/tensorflow/python/eager/function.py", line
3358, in _maybe_define_function
    args, kwargs, flat_args, filtered_flat_args, cache_key_context)
  File "/usr/local/lib/python3.6/dist-packages/tensorflow/python/eager/function.py", line
3280, in _define_function_with_shape_relaxation
    args, kwargs, override_flat_arg_shapes=relaxed_arg_shapes)
  File "/usr/local/lib/python3.6/dist-packages/tensorflow/python/eager/function.py", line
3206, in _create_graph_function
    capture_by_value=self._capture_by_value),
  File "/usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/func_graph.py"
, line 990, in func_graph_from_py_func
    func_outputs = python_func(*func_args, **func_kwargs)
  File "/usr/local/lib/python3.6/dist-packages/tensorflow/python/eager/def_function.py",
line 634, in wrapped_fn
    out = weak_wrapped_fn().__wrapped__(*args, **kwds)
  File "/usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/func_graph.py"
, line 977, in wrapper
    raise e.ag_error_metadata.to_exception(e)
ValueError: in user code:

    /usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/training.py:147
8 predict_function  *
        return step_function(self, iterator)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/training.py:146
8 step_function  **
        outputs = model.distribute_strategy.run(run_step, args=(data,))
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/distribute/distribute_lib.py
:1259 run
        return self._extended.call_for_each_replica(fn, args=args, kwargs=kwargs)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/distribute/distribute_lib.py
:2730 call_for_each_replica
        return self._call_for_each_replica(fn, args, kwargs)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/distribute/distribute_lib.py
:3417 _call_for_each_replica
        return fn(*args, **kwargs)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/training.py:146
1 run_step  **
        outputs = model.predict_step(data)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/training.py:143
4 predict_step
        return self(x, training=False)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/base_layer.py:1
012 __call__
        outputs = call_fn(inputs, *args, **kwargs)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/sequential.py:3
75 call
        return super(Sequential, self).call(inputs, training=training, mask=mask)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/functional.py:4
25 call
        inputs, training=training, mask=mask)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/functional.py:5
60 _run_internal_graph
        outputs = node.layer(*args, **kwargs)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/layers/recurrent.py:66
0 __call__
        return super(RNN, self).__call__(inputs, **kwargs)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/base_layer.py:1
012   call
```

```
    __    __
        outputs = call_fn(inputs, *args, **kwargs)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/layers/recurrent_v2.py
:1185 call
        zero_output_for_mask=self.zero_output_for_mask)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/util/dispatch.py:201 wrapper
        return target(*args, **kwargs)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/backend.py:4345 rnn
        [inp[0] for inp in flatted_inputs])
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/backend.py:4345 <listc
omp>
        [inp[0] for inp in flatted_inputs])
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/util/dispatch.py:201 wrapper
        return target(*args, **kwargs)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/array_ops.py:1047 _slice
_helper
        name=name)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/util/dispatch.py:201 wrapper
        return target(*args, **kwargs)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/array_ops.py:1219 stride
d_slice
        shrink_axis_mask=shrink_axis_mask)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/gen_array_ops.py:10479 s
trided_slice
        shrink_axis_mask=shrink_axis_mask, name=name)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/op_def_library.py:
750 _apply_op_helper
        attrs=attr_protos, op_def=op_def)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/func_graph.py:592
_create_op_internal
        compute_device)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/ops.py:3536 _creat
e_op_internal
        op_def=op_def)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/ops.py:2016 __init
__
        control_input_ops, op_def)
    /usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/ops.py:1856 _creat
e_c_op
        raise ValueError(str(e))

    ValueError: slice index 0 of dimension 0 out of bounds. for '{{node sequential_1/lstm
_1/strided_slice_2}} = StridedSlice[Index=DT_INT32, T=DT_FLOAT, begin_mask=0, ellipsis_ma
sk=0, end_mask=0, new_axis_mask=0, shrink_axis_mask=1](sequential_1/lstm_1/transpose, seq
uential_1/lstm_1/strided_slice_2/stack, sequential_1/lstm_1/strided_slice_2/stack_1, sequ
ential_1/lstm_1/strided_slice_2/stack_2)' with input shapes: [0,?,128], [1], [1], [1] and
with computed input tensors: input[1] = <0>, input[2] = <1>, input[3] = <1>.

127.0.0.1 - - [11/Feb/2021 13:08:34] "POST / HTTP/1.1" 500 -
127.0.0.1 - - [11/Feb/2021 13:08:44] "GET / HTTP/1.1" 200 -

127.0.0.1 - - [11/Feb/2021 13:08:56] "POST / HTTP/1.1" 200 -

127.0.0.1 - - [11/Feb/2021 13:09:10] "GET / HTTP/1.1" 200 -

127.0.0.1 - - [11/Feb/2021 13:09:24] "POST / HTTP/1.1" 200 -

127.0.0.1 - - [11/Feb/2021 13:09:36] "GET / HTTP/1.1" 200 -

127.0.0.1 - - [11/Feb/2021 13:09:57] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [11/Feb/2021 13:12:05] "GET / HTTP/1.1" 200 -

127.0.0.1 - - [11/Feb/2021 13:12:24] "POST / HTTP/1.1" 200 -
```

Q5 I improved the model by eliminating features with extremely low frequency and also increasing no of tweets.

Adopting the improved model helped reverse some missclassification errors encountered before.



After improved model