MODEL DEPLOYMENT ON A WEB APPLICATION USING FLASK

MAIN FILE (MODEL SAVING)

Extension - .py

This file makes generates 2 files with the extension of .pkl

- *Dumped vectorization model*
- *Dumped machine learning model*

```
import pandas as pd
import numpy as np
import re
import string
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import VotingClassifier
from sklearn.pipeline import Pipeline
import pickle

df_fake=pd.read_csv("Fake.csv")
df_true=pd.read_csv("True.csv")

df_fake["class"]=0
df_true["class"]=1

df_fake_manual_testing=df_fake.tail(10)
df_fake.drop([23470,23480],axis=0,inplace=True)
df_true_manual_testing=df_true.tail(10)
df_true.drop([21406,21416],axis=0,inplace=True)
df_manual_testing=pd.concat([df_fake_manual_testing,df_true_manual_testing],a
xis=0)
df_manual_testing.to_csv("manual_testing.csv")

df_merge=pd.concat([df_fake,df_true],axis=0)

df=df_merge.drop(["subject","date"],axis=1)

df = df.sample(frac = 1)

def conversion(title):
```

```
 title = title.lower()
 title = re.sub('\[.*?\]', '', title)
 title = re.sub("\\W"," ",title)
 title = re.sub('https?://\S+|www\.\S+', '', title)
 title = re.sub('<.*?>+', '', title)
 title = re.sub('[%s]' % re.escape(string.punctuation), '', title)
 title = re.sub('\n', '', title)
 title = re.sub('\w*\d\w*', '', title)
 return title

df["title"] = df["title"].apply(conversion)

x = df.iloc[0:5000, 0]
y = df.iloc[0:5000, -1]
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)

vectorization = TfidfVectorizer()
xv_train = vectorization.fit_transform(x_train)
xv_test = vectorization.transform(x_test)

pickle.dump(vectorization, open('transform.pkl','wb'))

knn=KNeighborsClassifier(n_neighbors=3)
knn.fit(xv_train, y_train)

LR = LogisticRegression()
LR.fit(xv_train, y_train)

svc = SVC()
svc.fit(xv_train, y_train)

models = list()

logistic_regression = Pipeline([('m', LogisticRegression())])
models.append(('logistic', logistic_regression))

svc = Pipeline([('m', SVC())])
models.append(('svc', svc))

k_n_n = Pipeline([('m', KNeighborsClassifier(n_neighbors=3))])
models.append(('knn', k_n_n))

ensemble = VotingClassifier(estimators=models, voting='hard')
ensemble.fit(xv_train, y_train)

filename='nlp_model.pkl'
pickle.dump(ensemble, open(filename, 'wb'))
```

FILE HAVING THE FLASK FRAMEWORK (FOR WEB APP CREATION)

Extension - .py

```
from flask import Flask, render_template, url_for, request
import pickle
import re
import string

filename = 'nlp_model.pkl'
ensemble = pickle.load(open(filename, 'rb'))
vectorization=pickle.load(open('transform.pkl', 'rb'))

app = Flask(__name__)

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

@app.route('/predict',methods=['POST'])
def predict():
    if request.method == 'POST':
        news = request.form['news']
        news = news.lower()
        news = re.sub('\[.*?\]', '', news)
        news = re.sub("\\W", " ", news)
        news = re.sub('https?://\S+|www\.\S+', '', news)
        news = re.sub('<.*?>+', '', news)
        news = re.sub('[%s]' % re.escape(string.punctuation), '', news)
        news = re.sub('\n', '', news)
        news = re.sub('\w*\d\w*', '', news)
        data = [news]
        vect = vectorization.transform(data).toarray()
        my_prediction = ensemble.predict(vect)
    return render_template('result.html', prediction=my_prediction)

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

HTML FILES (FOR GIVING LAYOUT FOR THE WEB APP)

Extension - .html

1. For home page

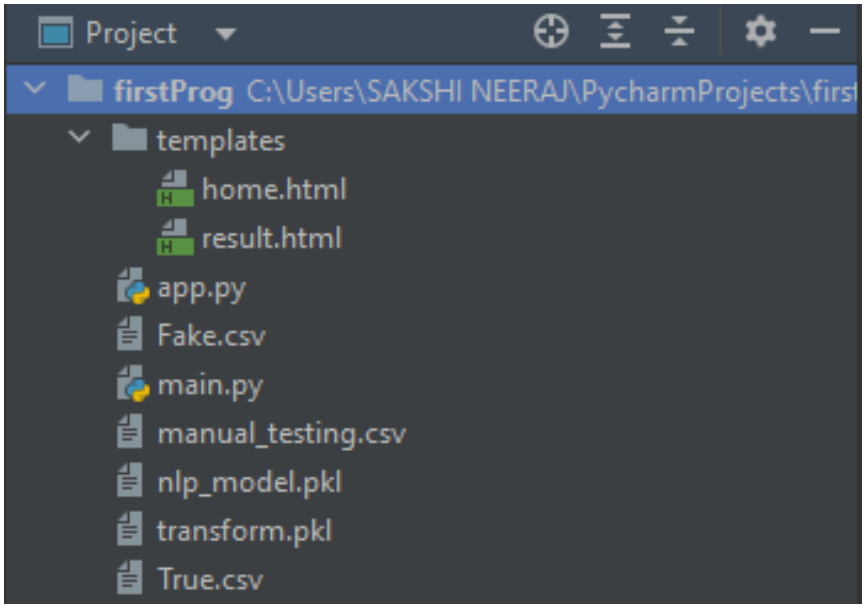
```
<!DOCTYPE html>
<html>
<head>

</head>
<body bgcolor="#DBD1FC">

   <header>
      <div class="container">
      <div id="brandname">
         DETECTION OF FAKE NEWS USING MACHINE LEARNING
      </div>

   </div>
   </header>

   <div class="ml-container">

      <form action="{{ url_for('predict')}}" method="POST">
      <p>ENTER THE NEWS HERE</p>
      <!-- <input type="text" name="comment"/> -->
      <textarea name="news" rows="6" cols="50"></textarea>
      <br/>

      <input type="submit" class="btn-info" value="predict">

   </form>

   </div>




</body>
</html>
```

2. For result page

```
<!DOCTYPE html>
<html>

<body bgcolor="#DBD1FC">
```

```
   <header>
      <div class="container">
      <div id="brandname">
         ML App
      </div>
      <h2>FAKE NEWS DETECTOR</h2>

   </div>
   </header>
   <div class="results">



   {% if prediction == 1%}
   <h2 style="color:red;">TRUE NEWS</h2>
   {% elif prediction == 0%}
   <h2 style="color:blue;">FAKE NEWS</h2>
   {% endif %}

   </div>

</body>
</html>
```

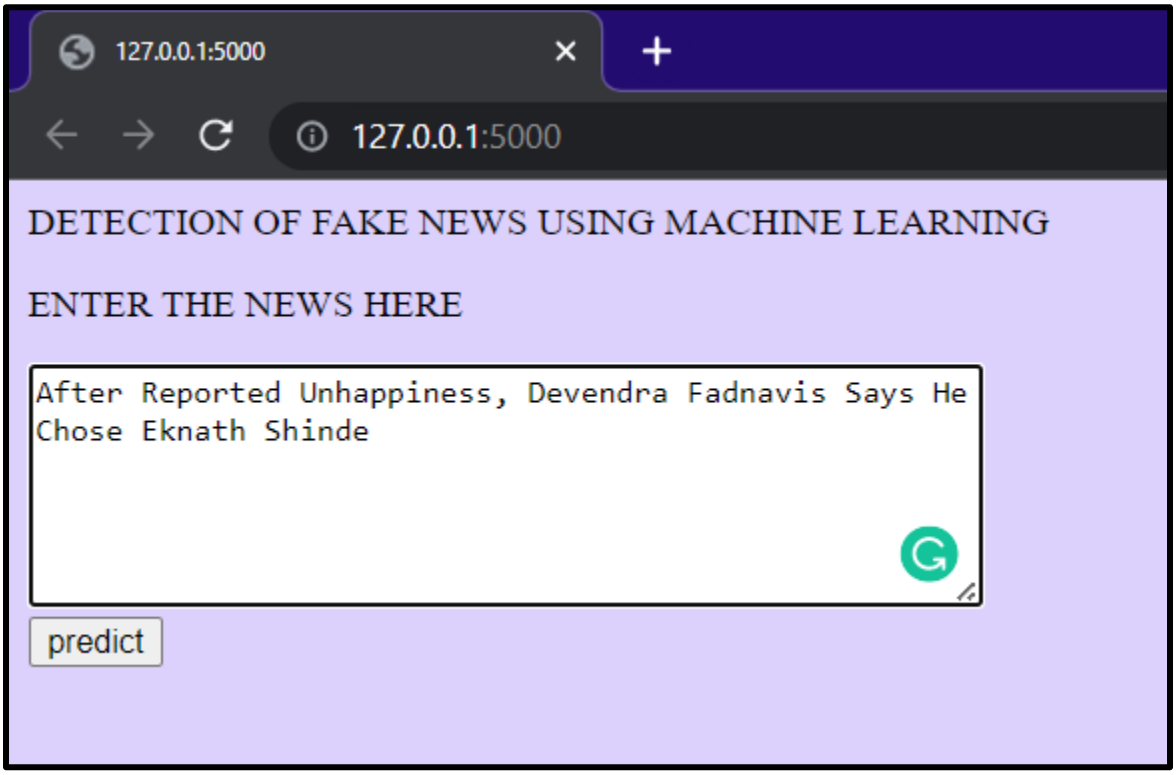
FILE TREE IN THE DIRECTORY FOR DEPLOYMENT

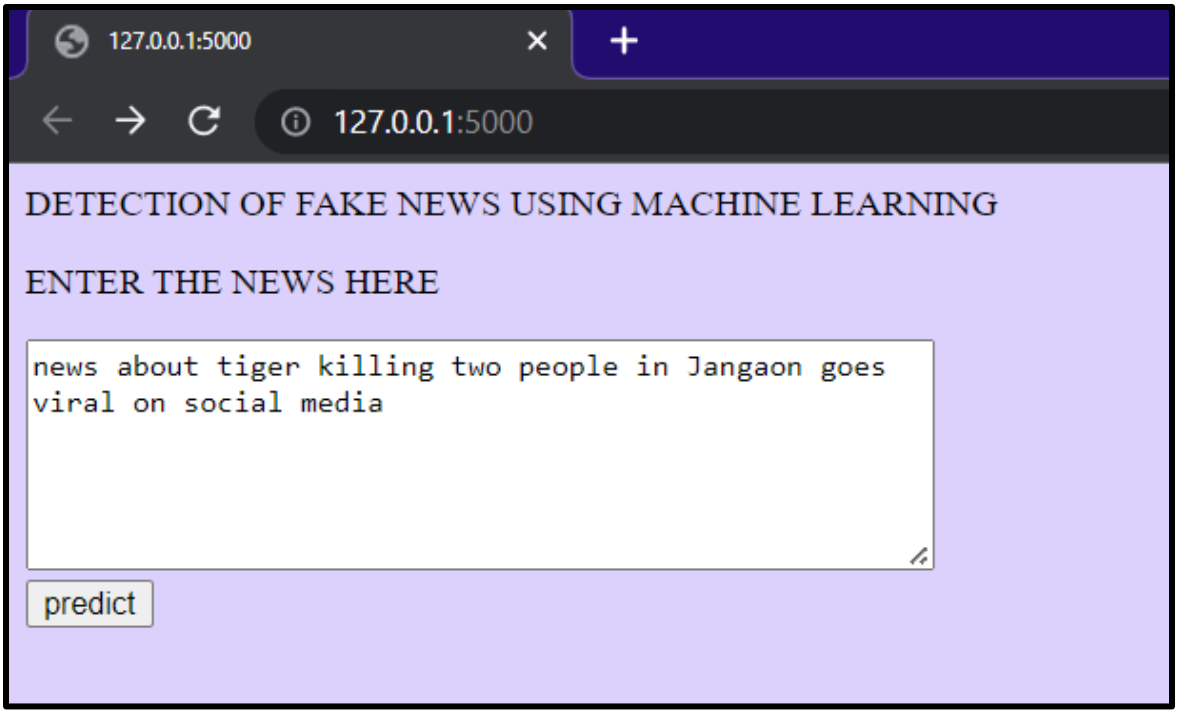


OUTPUT ON THE CONSOLE

```
PS C:\Users\SAKSHI NEERAJ\PycharmProjects\firstProg> python app.py
 * Serving Flask app 'app' (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: on
 * Running on http://127.0.0.1:5000 (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 862-429-275
127.0.0.1 - - [05/Jul/2022 14:31:55] "GET / HTTP/1.1" 200 -
```

WEB APP OUTPUT




127.0.0.1:5000
127.0.0.1:5000
DETECTION OF FAKE NEWS USING MACHINE LEARNING
ENTER THE NEWS HERE
news about tiger killing two people in Jangaon goes viral on social media
predict


127.0.0.1:5000/predict
127.0.0.1:5000/predict
ML App
FAKE NEWS DETECTOR
FAKE NEWS