NAME: RESHMA JAYAPALAN

BATCH CODE: LISUM02

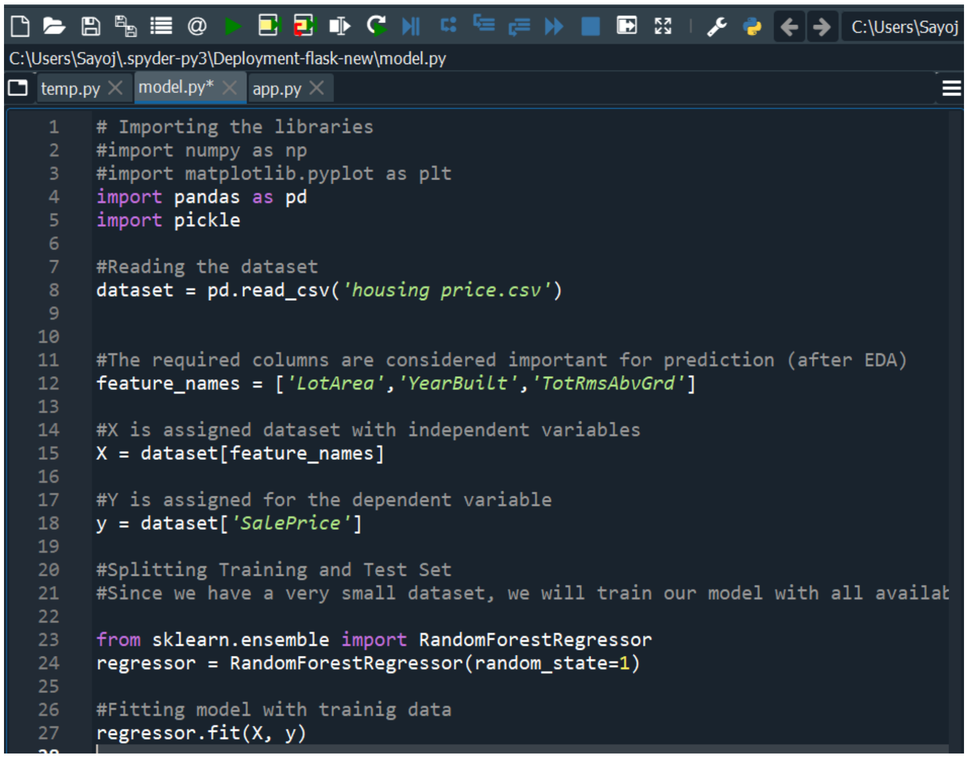
SUBMISSION DATE: 15/08/2021

DEPLOYMENT OF ML MODEL ON FLASK

1. Select the dataset

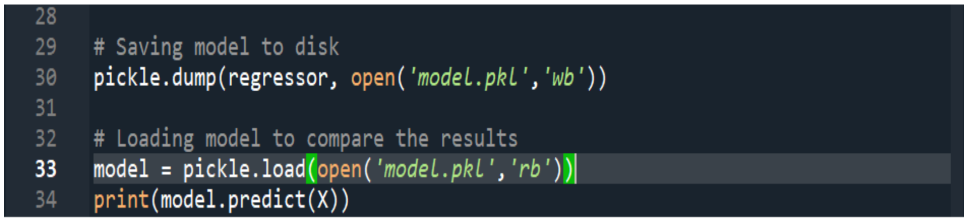
I have selected a small dataset for housing price prediction from Kaggle

2. Open IDE Spyder and work on a file named model.py, cleaning the data, conducting EDA on the dataset and finally model building from the dataset



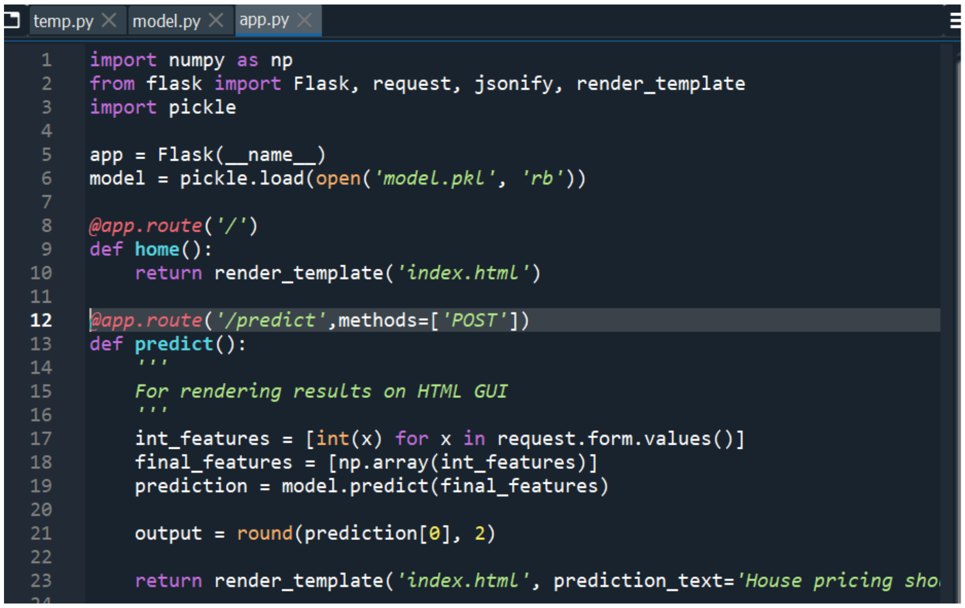
```
# Importing the libraries
#import numpy as np
#import matplotlib.pyplot as plt
import pandas as pd
import pickle

#Reading the dataset
dataset = pd.read_csv('housing price.csv')


#The required columns are considered important for prediction (after EDA)
feature_names = ['LotArea','YearBuilt','TotRmsAbvGrd']

#X is assigned dataset with independent variables
X = dataset[feature_names]

#Y is assigned for the dependent variable
y = dataset['SalePrice']

#Splitting Training and Test Set
#Since we have a very small dataset, we will train our model with all availat

from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(random_state=1)

#Fitting model with trainig data
regressor.fit(X, y)
```

3. Load the model to the disk using pickle (model.pkl) and compare the results



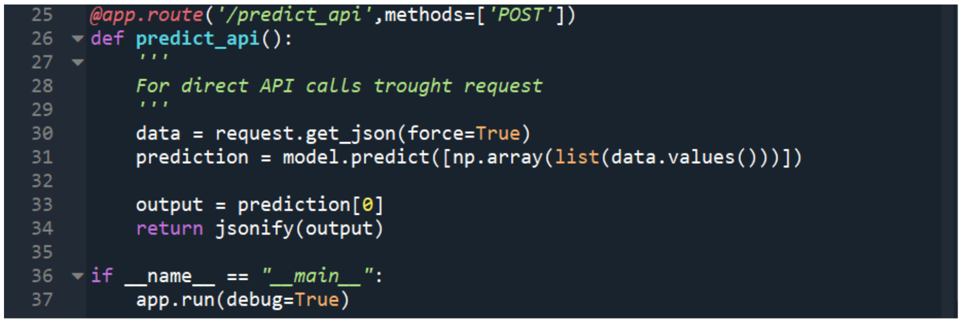
```python

# Saving model to disk
pickle.dump(regressor, open('model.pkl','wb'))

# Loading model to compare the results
model = pickle.load(open('model.pkl','rb'))
print(model.predict(X))
```

4. The app.py file is created, app.py file contains the python code that can redirect to execute the model from model.py



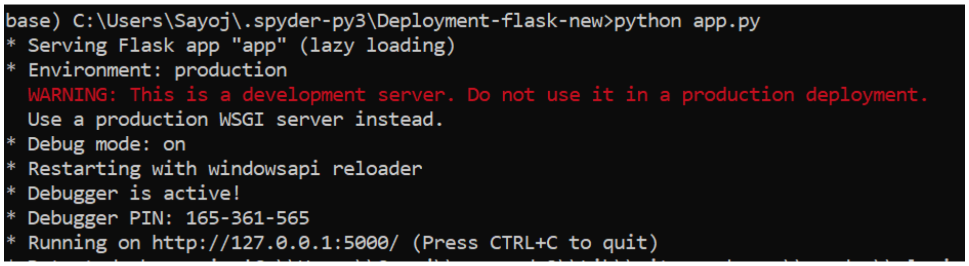
temp.py model.py app.py

```python
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

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

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [int(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='House pricing sho
```



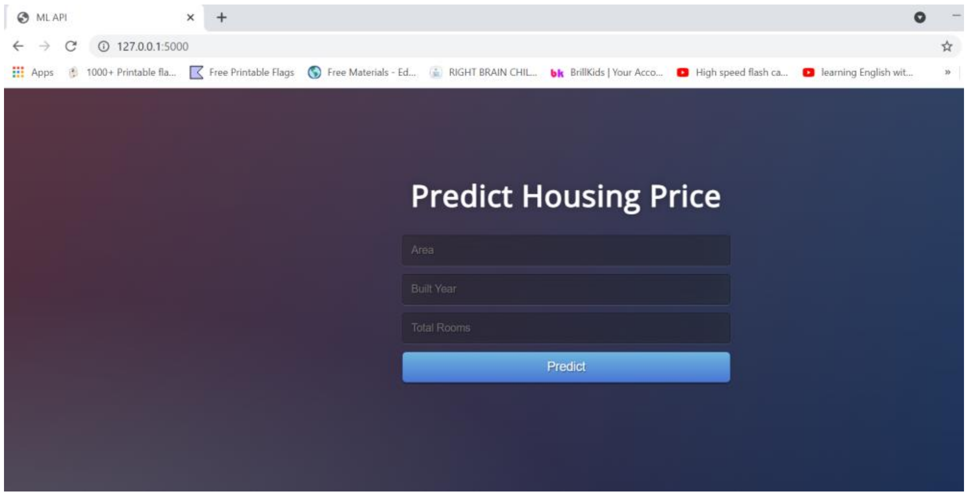
```python
@app.route('/predict_api',methods=['POST'])
def predict_api():
    '''
    For direct API calls trought request
    '''
    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)

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

5. The static folder contains the css file (style file), these are used to control the appearance of the UI
6. The templates folder contains the html pages which will be used to create the UI.
7. On Anaconda Prompt, ensure that you are in the project home directory. Create the machine learning model by running below command - `python model.py`This would create a serialized version of our model into a file model.pkl
8. Run app.py using `python app.py` command to start Flask API



```
base) C:\Users\Sayoj\.spyder-py3\Deployment-flask-new>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
* Restarting with windowsapi reloader
* Debugger is active!
* Debugger PIN: 165-361-565
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

9. Use the url received in the browser and the result page as below:


ML API
127.0.0.1:5000
Apps
1000+ Printable fla...
Free Printable Flags
Free Materials - Ed...
RIGHT BRAIN CHIL...
BrillKids | Your Acco...
High speed flash ca...
learning English wit...
Predict Housing Price
Area
Built Year
Total Rooms
Predict