

NAME: RESHMA JAYAPALAN

BATCH CODE: [LISUM02](#)

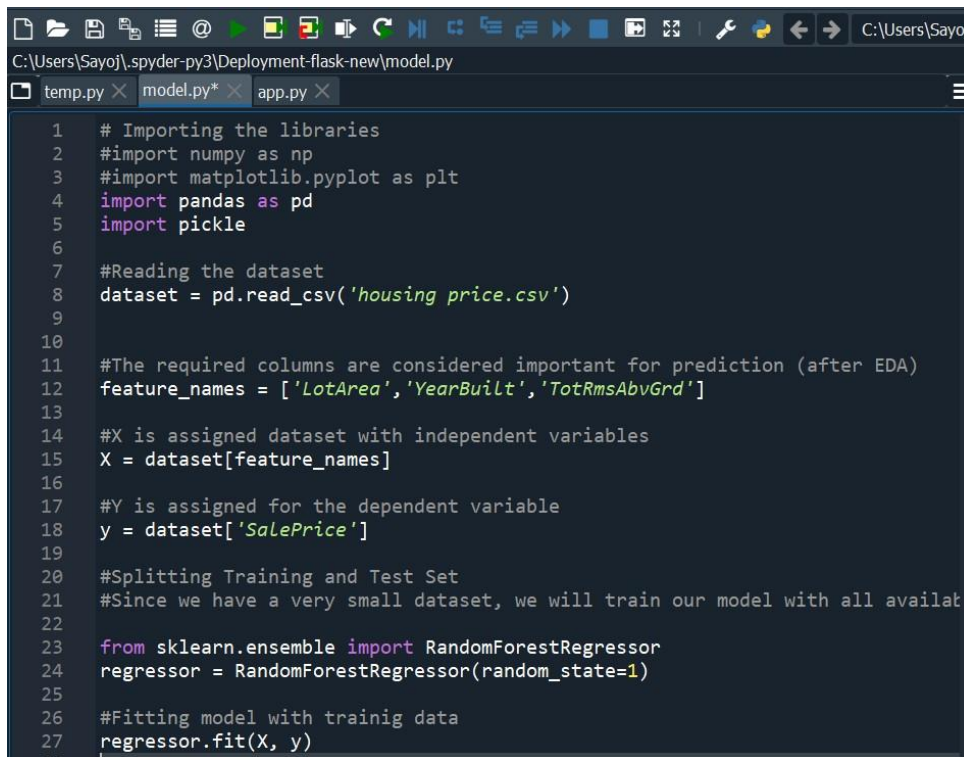
SUBMISSION DATE: 22/08/2021

DEPLOYMENT OF ML MODEL ON HEROKU

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



```
1  # Importing the libraries
2  #import numpy as np
3  #import matplotlib.pyplot as plt
4  import pandas as pd
5  import pickle
6
7  #Reading the dataset
8  dataset = pd.read_csv('housing price.csv')
9
10
11 #The required columns are considered important for prediction (after EDA)
12 feature_names = ['LotArea', 'YearBuilt', 'TotRmsAbvGrd']
13
14 #X is assigned dataset with independent variables
15 X = dataset[feature_names]
16
17 #Y is assigned for the dependent variable
18 y = dataset['SalePrice']
19
20 #Splitting Training and Test Set
21 #Since we have a very small dataset, we will train our model with all available data
22
23 from sklearn.ensemble import RandomForestRegressor
24 regressor = RandomForestRegressor(random_state=1)
25
26 #Fitting model with training data
27 regressor.fit(X, y)
```

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

```

28
29 # Saving model to disk
30 pickle.dump(regressor, open('model.pkl', 'wb'))
31
32 # Loading model to compare the results
33 model = pickle.load(open('model.pkl', 'rb'))
34 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

```

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 should be')

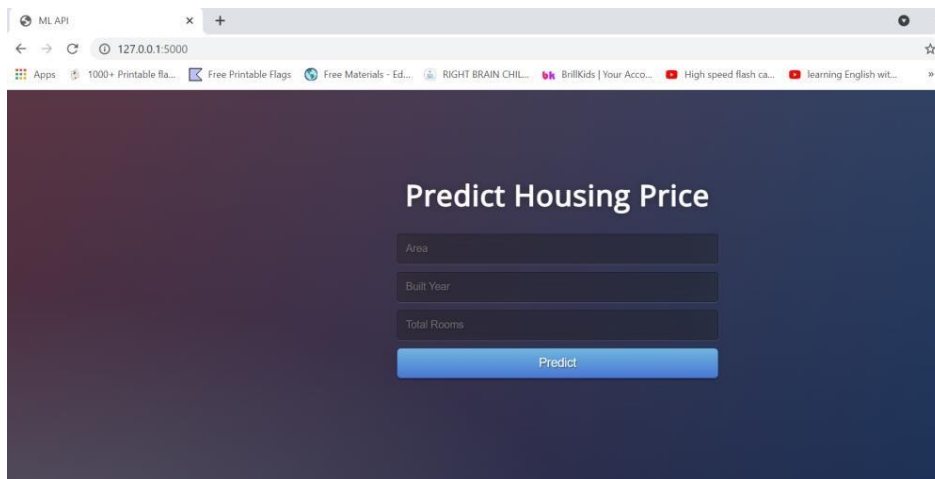
if __name__ == "__main__":
    app.run(debug=True)

```

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:



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000'. The page has a dark blue background with a white title 'Predict Housing Price'. Below the title, there are three input fields labeled 'Area', 'Built Year', and 'Total Rooms'. At the bottom of the form is a blue button labeled 'Predict'.

10. Create account in heroku, type 'heroku login' in Anaconda prompt or in your terminal
11. After login, create new app, and name it " housingpriceprediction-dg "
12. Choose the deployment method as Github and connect the Github account


13. Meanwhile, create a repository on Github and add all the related files

	Reshma-Jayapalan	Add files via upload
	.gitignore	Add files via upload
	LICENSE	Initial commit
	Procfile	Add files via upload
	README.md	Initial commit
	app.py	Add files via upload
	housing price.csv	Add files via upload
	index.html	Add files via upload
	model.pkl	Add files via upload
	model.py	Add files via upload
	request.py	Add files via upload
	requirements.txt	Add files via upload
	runtime.txt	Add files via upload


14. On Heroku, connect to the repository created on Github and manually deploy the "main" branch

15. On successful build, the app can be viewed on the browser

Choose a branch to deploy

 main

Deploy Branch

Receive code from GitHub 

Build main 55a9ecad 

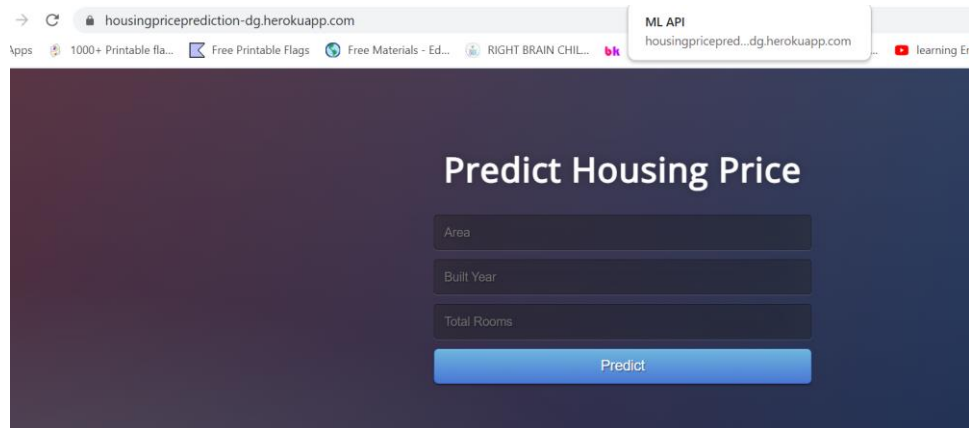
Release phase 

Deploy to Heroku 

Your app was successfully deployed.

 View

16. On clicking view, the app is opened as below



→ ↻ housingpriceprediction-dg.herokuapp.com ML API housingpricepred...dg.herokuapp.com

Apps 1000+ Printable fla... Free Printable Flags Free Materials - Ed... RIGHT BRAIN CHIL... bk learning En

Predict Housing Price

Area

Built Year

Total Rooms

Predict