

Name: Parthkumar Sutariya

Batch Code: LISUM28

Submission Date: 27/12/2023

Submitted to: Data Glacier

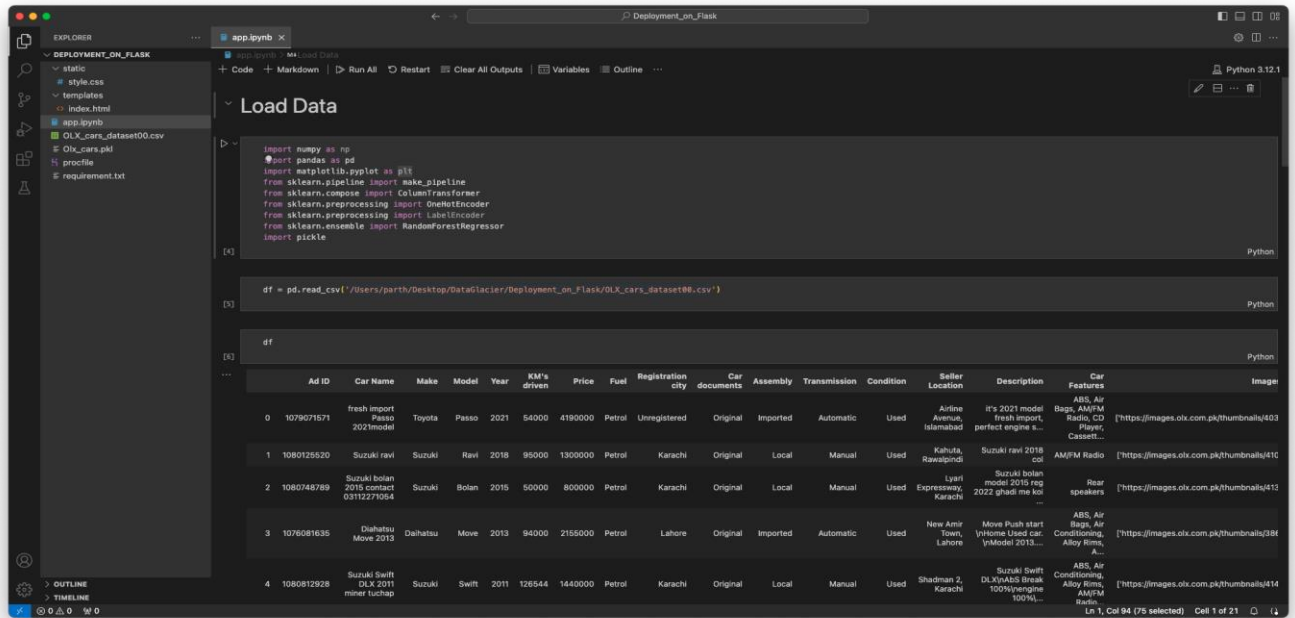
## Subject: Machine Learning Model Deployment on Heroku (API and Web based)

### Step 1: Selecting Toy Dataset from Kaggle

Ad ID	Car Name	Make	Model	Year	KM's driven	Price	Fuel	Registration	Car document	Assembly	Transmission	Condition	Seller Location	Description	Car Features	Images URI's	Car Profile
1079071571	fresh import R	Toyota	Passo	2021	54000	4190000	Petrol	Unregistered	Original	Imported	Automatic	Used	Airline Avenue,	it's 2021 mode	ABS, Air Bags,	<a href="https://image.https://www.ox.com.pk">https://image.https://www.ox.com.pk</a>	<a href="https://www.ox.com.pk">https://www.ox.com.pk</a>
1080125520	Suzuki ravi	Suzuki	Ravi	2018	95000	1300000	Petrol	Karachi	Original	Local	Manual	Used	Kahuta, Rawal	Suzuki ravi 201	AM/FM Radio	<a href="https://image.https://www.ox.com.pk">https://image.https://www.ox.com.pk</a>	<a href="https://www.ox.com.pk">https://www.ox.com.pk</a>
1080748789	Suzuki bolan 2	Suzuki	Bolan	2015	50000	800000	Petrol	Karachi	Original	Local	Manual	Used	Lyari Expressw	Suzuki bolan n	Rear speakers	<a href="https://image.https://www.ox.com.pk">https://image.https://www.ox.com.pk</a>	<a href="https://www.ox.com.pk">https://www.ox.com.pk</a>
1076081635	Daihatsu Move	Daihatsu	Move	2013	94000	2155000	Petrol	Lahore	Original	Imported	Automatic	Used	New Amir Tow	Move Push sta	ABS, Air Bags,	<a href="https://image.https://www.ox.com.pk">https://image.https://www.ox.com.pk</a>	<a href="https://www.ox.com.pk">https://www.ox.com.pk</a>
1080812928	Suzuki Swift D	Suzuki	Swift	2011	126544	1440000	Petrol	Karachi	Original	Local	Manual	Used	Shadman 2, Ka	Suzuki Swift D	ABS, Air Condi	<a href="https://image.https://www.ox.com.pk">https://image.https://www.ox.com.pk</a>	<a href="https://www.ox.com.pk">https://www.ox.com.pk</a>
1079496823	WagonR AGS	Suzuki	Wagon R	2020	54000	2830000	Petrol	Lahore	Original	Local	Automatic	Used	Abbot Road, Si	wagon r total g	ABS, Air Bags,	<a href="https://image.https://www.ox.com.pk">https://image.https://www.ox.com.pk</a>	<a href="https://www.ox.com.pk">https://www.ox.com.pk</a>
1080333426	Daihatsu Mira	Daihatsu	Mira	2013	140000	2150000	Petrol	Karachi	Original	Imported	Automatic	Used	North Nazim	bilkul okay con	ABS, Air Bags,	<a href="https://image.https://www.ox.com.pk">https://image.https://www.ox.com.pk</a>	<a href="https://www.ox.com.pk">https://www.ox.com.pk</a>
1080784758	Suzuki Bolan 2	Suzuki	Bolan	2016	32000	1050000	Petrol	Karachi	Original	Local	Manual	Used	Korangi Indust	SUZUKI BOLAN	Alloy Rims, AM	<a href="https://image.https://www.ox.com.pk">https://image.https://www.ox.com.pk</a>	<a href="https://www.ox.com.pk">https://www.ox.com.pk</a>
1076448625	2004 Honda C	Honda	City iDSi	2004	160000	1230000	Petrol	Sindh	Original	Local	Manual	Used	Alama Iqbal T	A very well mai	ABS, Air Condi	<a href="https://image.https://www.ox.com.pk">https://image.https://www.ox.com.pk</a>	<a href="https://www.ox.com.pk">https://www.ox.com.pk</a>
1080820574	car Bashir s 8C	Suzuki	Cultus VXR	2008	75000	950000	Petrol	Faisalabad	Original	Local	Manual	Used	Al Najaf Colon	Anders hotel j	Air Conditionin	<a href="https://image.https://www.ox.com.pk">https://image.https://www.ox.com.pk</a>	<a href="https://www.ox.com.pk">https://www.ox.com.pk</a>
1080792524	Suzuki Every	Suzuki	Every	2018	101628	1850000	Petrol	Islamabad	Original	Imported	Automatic	Used	Saddar, Rawal	Suzuki Every	Air Bags, Air C	<a href="https://image.https://www.ox.com.pk">https://image.https://www.ox.com.pk</a>	<a href="https://www.ox.com.pk">https://www.ox.com.pk</a>
1080676810	I am selling XLI	Toyota	Corolla XLI	2007	346000	1895000	Petrol	Lahore	Original	Local	Manual	Used	New Northern	Alloy rim	Air Conditionin	<a href="https://image.https://www.ox.com.pk">https://image.https://www.ox.com.pk</a>	<a href="https://www.ox.com.pk">https://www.ox.com.pk</a>

Toy Dataset

## Step 2: Model Building and Model Saving

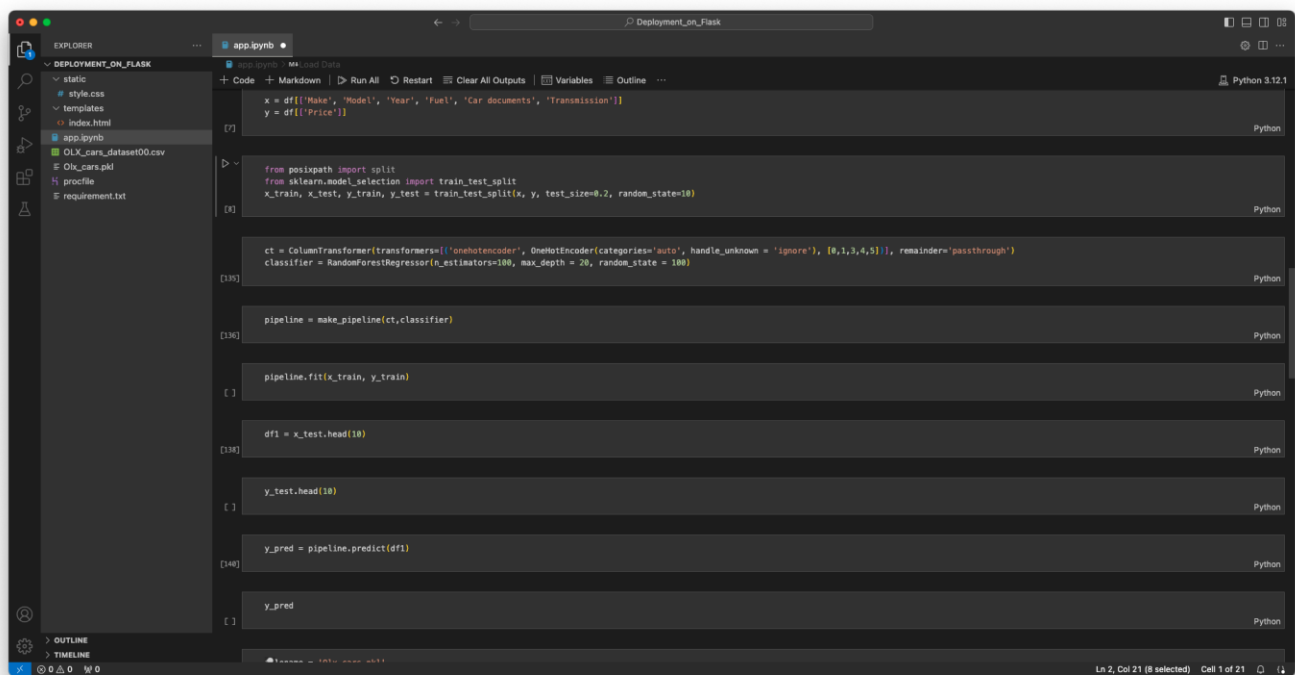


```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.pipeline import make_pipeline
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestRegressor
import pickle

df = pd.read_csv('Users\parth\Desktop\DataGlacier\Deployment_on_Flask\DLX_cars_dataset00.csv')

df
```

Ad ID	Car Name	Make	Model	Year	KM's driven	Price	Fuel	Registration	City	Car documents	Assembly	Transmission	Condition	Seller Location	Description	Car Features	Image			
0	fresh import	Paseo	2021	Model	Toyota	54000	4190000	Petrol	Unregistered	Original	Imported	Automatic	Used	Airline Avenue, Islamabad	It's 2021 model fresh import, perfect engine s...	ABS, Air Bags, AM/FM Radio, CD Player, Cassett...	[https://images.olx.com.pk/thumbnails/403			
1	1080125520	Suzuki ravi	Suzuki	Ravi	2018	95000	1300000	Petrol	Karachi	Original	Local	Manual	Used	Kahuta, Rawalpindi	Suzuki ravi 2018 col	AM/FM Radio	[https://images.olx.com.pk/thumbnails/410			
2	1080748789	Suzuki bolan	2015	contact	031227304	Suzuki	Bolan	2015	50000	800000	Petrol	Karachi	Original	Local	Manual	Used	Lyeri Expressway, Karachi	Suzuki bolan model 2015 reg 2022 ghadi me koi ...	Rear speakers	[https://images.olx.com.pk/thumbnails/412
3	1076081635	Daihatsu Move	2013	Move	2013	94000	2155000	Petrol	Lahore	Original	Imported	Automatic	Used	New Amir Town, Lahore	Move Push start VHome Used Car VnModel 2013...	ABS, Air Bags, Air Conditioning, Alloy Rims, A...	[https://images.olx.com.pk/thumbnails/386			
4	1080812928	Suzuki Swift	DLX 2011	minor lachup	Suzuki	Swift	2011	126544	1440000	Petrol	Karachi	Original	Local	Manual	Used	Shedman 2, Karachi	Suzuki Swift DLX 2011ABS break 100%(engine 100%...	ABS, Air Conditioning, Alloy Rims, AM/FM Radio...	[https://images.olx.com.pk/thumbnails/414	



```
x = df[['Make', 'Model', 'Year', 'Fuel', 'Car documents', 'Transmission']]
y = df[['Price']]

from posixpath import split
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=10)

ct = ColumnTransformer(transformers=[('onehotencoder', OneHotEncoder(categories='auto', handle_unknown='ignore'), [0,1,3,4,5])], remainder='passthrough')
classifier = RandomForestRegressor(n_estimators=100, max_depth = 20, random_state = 100)

pipeline = make_pipeline(ct, classifier)

pipeline.fit(x_train, y_train)

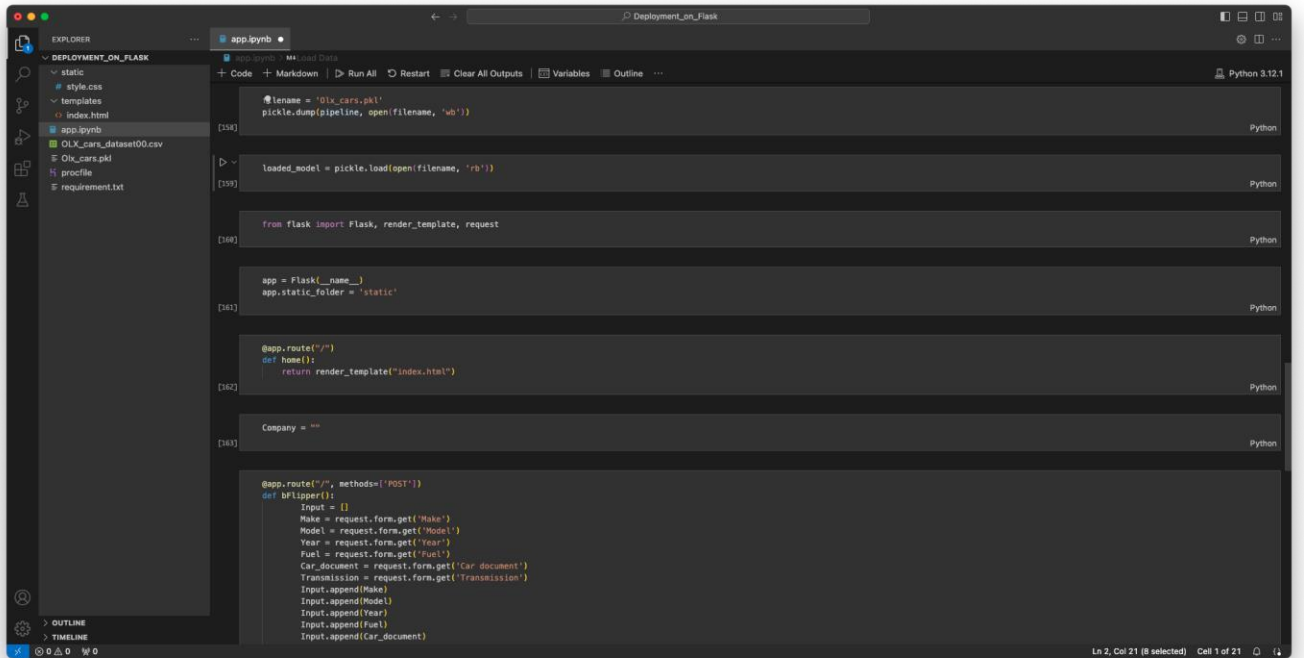
df1 = x_test.head(10)

y_test.head(10)

y_pred = pipeline.predict(df1)

y_pred
```

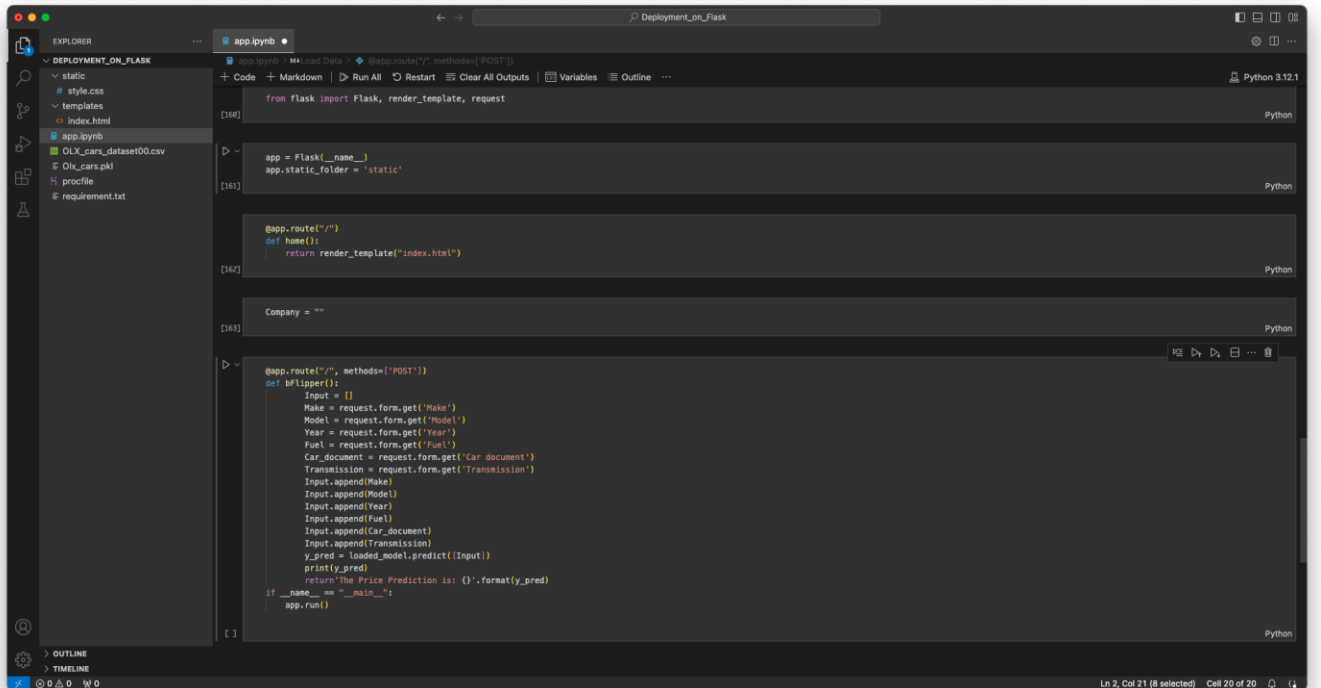
### Step 3: Building Flask App



The screenshot shows the VS Code editor with the file explorer on the left displaying the project structure for 'DEPLOYMENT\_ON\_FLASK'. The main editor window shows the 'app.py' file with the following code:

```
[154] filename = 'olx_cars.pkl'
[155] pickle.dump(pipeline, open(filename, 'wb'))
[156]
[157] loaded_model = pickle.load(open(filename, 'rb'))
[158]
[159] from flask import Flask, render_template, request
[160]
[161] app = Flask(__name__)
[162] app.static_folder = 'static'
[163]
[164] @app.route("/")
[165] def home():
[166]     return render_template("index.html")
[167]
[168] Company = ""
[169]
[170] @app.route("/", methods=['POST'])
[171] def bflipper():
[172]     Input = []
[173]     Make = request.form.get('Make')
[174]     Model = request.form.get('Model')
[175]     Year = request.form.get('Year')
[176]     Fuel = request.form.get('Fuel')
[177]     Car_document = request.form.get('Car document')
[178]     Transmission = request.form.get('Transmission')
[179]     Input.append(Make)
[180]     Input.append(Model)
[181]     Input.append(Year)
[182]     Input.append(Fuel)
[183]     Input.append(Car_document)
```

The status bar at the bottom indicates 'Ln 2, Col 21 (8 selected) Cell 1 of 21'.



The screenshot shows the VS Code editor with the file explorer on the left displaying the project structure for 'DEPLOYMENT\_ON\_FLASK'. The main editor window shows the 'app.py' file with the following code:

```
[154] filename = 'olx_cars.pkl'
[155] pickle.dump(pipeline, open(filename, 'wb'))
[156]
[157] loaded_model = pickle.load(open(filename, 'rb'))
[158]
[159] from flask import Flask, render_template, request
[160]
[161] app = Flask(__name__)
[162] app.static_folder = 'static'
[163]
[164] @app.route("/")
[165] def home():
[166]     return render_template("index.html")
[167]
[168] Company = ""
[169]
[170] @app.route("/", methods=['POST'])
[171] def bflipper():
[172]     Input = []
[173]     Make = request.form.get('Make')
[174]     Model = request.form.get('Model')
[175]     Year = request.form.get('Year')
[176]     Fuel = request.form.get('Fuel')
[177]     Car_document = request.form.get('Car document')
[178]     Transmission = request.form.get('Transmission')
[179]     Input.append(Make)
[180]     Input.append(Model)
[181]     Input.append(Year)
[182]     Input.append(Fuel)
[183]     Input.append(Car_document)
[184]     Input.append(Transmission)
[185]     y_pred = loaded_model.predict(Input)
[186]     print(y_pred)
[187]     return 'The Price Prediction is: {}'.format(y_pred)
[188] if __name__ == '__main__':
[189]     app.run()
```

The status bar at the bottom indicates 'Ln 2, Col 21 (8 selected) Cell 20 of 20'.

### Step 4: Building HTML file

The screenshot displays a VS Code editor window with the following components:

- Explorer (Left):** Shows the project structure. The 'index.html' file is selected under the 'templates' folder. Other files visible include 'app.py', 'OLX\_cars\_dataset00.csv', 'OLX\_cars.pkl', 'profile', and 'requirement.txt'.
- Main Editor:** Displays the content of 'index.html'. The code is a single HTML file that includes Bootstrap CSS and jQuery, and contains a form for car price prediction. The form has fields for Company name, Model, Year, Fuel, Car documents, Transmission, and a Predict button.
- Run and Debug (Right):** Shows the 'Run and Debug' sidebar, which is currently empty.

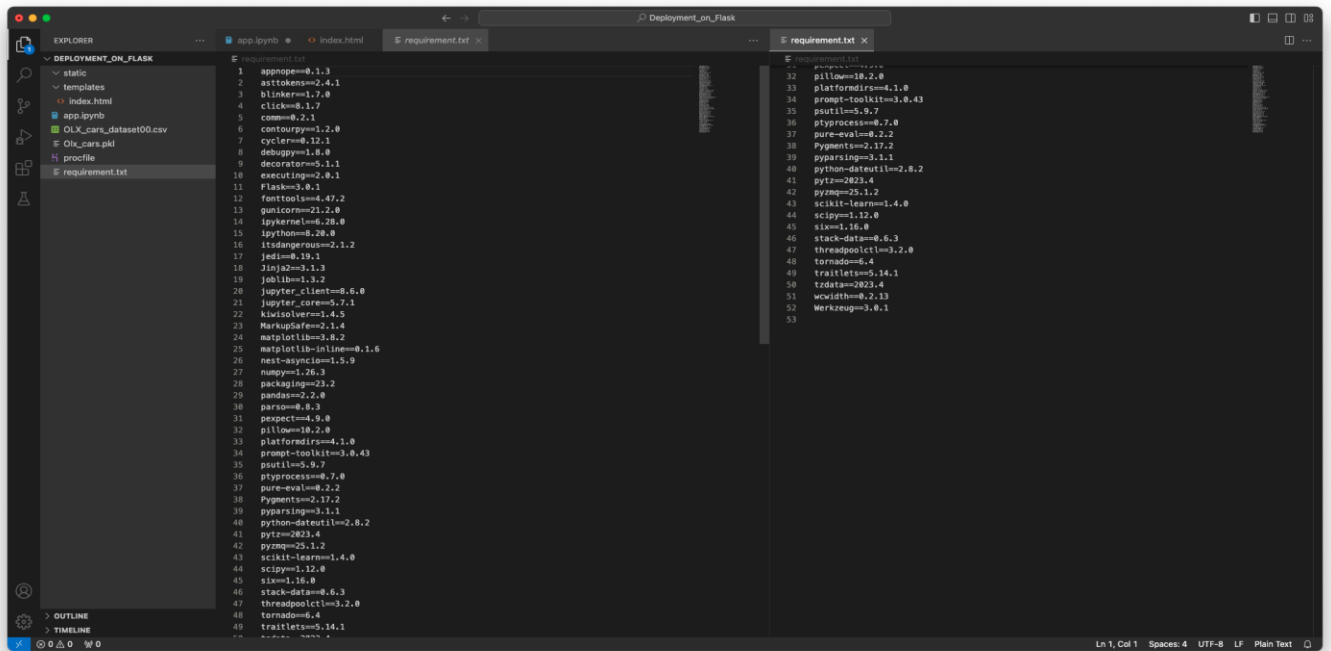
The code in 'index.html' is as follows:

```

1 <!-- index.html -->
2 <html>
3 <head>
4 <link rel="stylesheet" href="/Users/parth/Desktop/DataGlacier/Deployment_on_Flask/static/style.css">
5 <!-- Required meta tags -->
6 <meta charset="utf-8">
7 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
8 <!-- Bootstrap CSS -->
9 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css" integrity="sha384-1CbkR0CuGaJ0kS1kP0p0d77vePS1yEQEjAuZQ7gFLD2y1zuqKfKf16/eSrtXlkn" crossorigin="anonymous">
10 <title>OLX Car Price Prediction</title>
11 </head>
12 <body>
13 <div class="form-example">
14 <div class="form-example">
15 <label form="name">Enter Company name: </label>
16 <input type="text" name="name" id="name" required>
17 </div>
18 <div class="form-example">
19 <label form="Model">Model: </label>
20 <input type="text" name="Model" id="Model" required>
21 </div>
22 <div class="form-example">
23 <label form="Year">Year: </label>
24 <input type="text" name="Year" id="Year" required>
25 </div>
26 <div class="form-example">
27 <label form="Fuel">Fuel: </label>
28 <input type="text" name="Fuel" id="Fuel" required>
29 </div>
30 </div>
31 <div class="form-example">
32 <label form="Car documents">Car documents: </label>
33 <input type="text" name="Car documents" id="Car documents" required>
34 </div>
35 <div class="form-example">
36 <label form="Transmission">Transmission: </label>
37 <input type="text" name="Transmission" id="Transmission" required>
38 </div>
39 <div class="form-example">
40 <input class="btn btn-primary" type="submit" value="Predict">
41 </div>
42 </div>
43 <!-- jQuery and Bootstrap Bundle -->
44 <script src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js" integrity="sha384-DfXdzHtPb8Lys5ScT/jy4C40GanoFy38W/Bn4IbbYtWd0CkAfrk" crossorigin="anonymous">
45 <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-1QyJgMvRvqgW1871zRfS800qgAR9W0w/annp" crossorigin="anonymous">
46 </script>
47 </body>
48 </html>

```

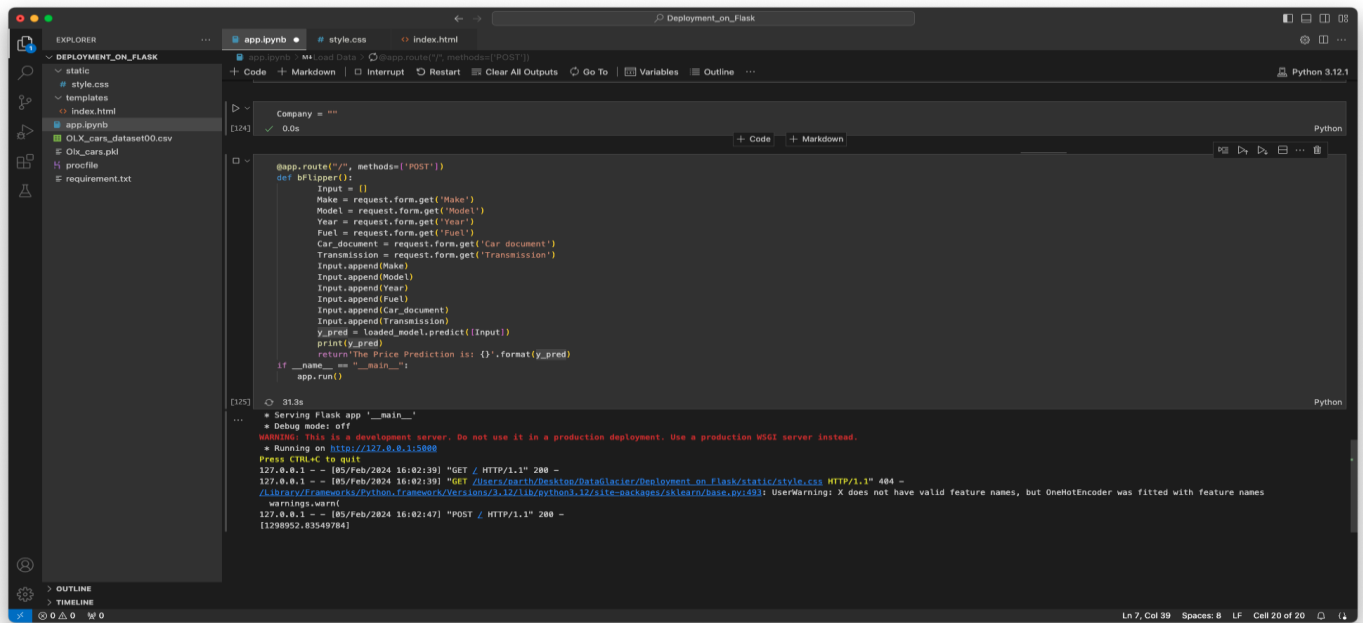
## Step 5: Creating Requirement.txt file for deploying Flask app



The screenshot shows a VS Code editor with a file explorer on the left and a code editor in the center. The file explorer shows a project named 'DEPLOYMENT\_ON\_FLASK' with files like 'static', 'templates', 'app.py', 'OLX\_cars\_dataset00.csv', 'Olx\_cars.pkl', 'profile', and 'requirement.txt'. The code editor displays the content of 'requirement.txt', which lists various Python packages and their versions. The packages listed are: appnope==0.1.3, asttokens==2.4.1, blinker==1.7.0, click==8.1.7, color==0.2.1, contourpy==1.2.0, cycler==0.12.1, debugpy==1.6.0, decorator==5.1.1, executing==2.0.1, Flask==3.0.1, fonttools==4.47.2, gunicorn==21.2.0, ipykernel==6.28.0, ipython==8.28.0, itsdangerous==2.1.2, jedi==0.19.1, Jinja2==3.1.3, joblib==1.3.2, Jupyter-Client==8.6.0, Jupyter-Core==5.7.1, Kwisolver==1.4.5, MarkupSafe==2.1.4, matplotlib==3.8.2, matplotlib-inline==0.1.6, nest-asyncio==1.5.9, numpy==1.26.3, packaging==23.2, pandas==2.2.0, parso==0.8.3, pexpect==4.9.0, pillow==10.2.0, platformdirs==4.1.0, prompt-toolkit==3.0.43, psutil==5.9.7, ptyprocess==0.7.0, pure-eval==0.2.2, Pygments==2.17.2, pyarsing==3.1.1, python-dateutil==2.8.2, pytz==2023.4, pyzmq==25.1.2, scikit-learn==1.4.0, scipy==1.12.0, size==1.16.0, stack-data==0.6.3, threadpoolctl==3.2.0, tornado==6.4, traitlets==5.14.1, tzdata==2023.4, wcwidth==0.2.13, Werkzeug==3.0.1.

```
1 appnope==0.1.3
2 asttokens==2.4.1
3 blinker==1.7.0
4 click==8.1.7
5 color==0.2.1
6 contourpy==1.2.0
7 cycler==0.12.1
8 debugpy==1.6.0
9 decorator==5.1.1
10 executing==2.0.1
11 Flask==3.0.1
12 fonttools==4.47.2
13 gunicorn==21.2.0
14 ipykernel==6.28.0
15 ipython==8.28.0
16 itsdangerous==2.1.2
17 jedi==0.19.1
18 Jinja2==3.1.3
19 joblib==1.3.2
20 Jupyter-Client==8.6.0
21 Jupyter-Core==5.7.1
22 Kwisolver==1.4.5
23 MarkupSafe==2.1.4
24 matplotlib==3.8.2
25 matplotlib-inline==0.1.6
26 nest-asyncio==1.5.9
27 numpy==1.26.3
28 packaging==23.2
29 pandas==2.2.0
30 parso==0.8.3
31 pexpect==4.9.0
32 pillow==10.2.0
33 platformdirs==4.1.0
34 prompt-toolkit==3.0.43
35 psutil==5.9.7
36 ptyprocess==0.7.0
37 pure-eval==0.2.2
38 Pygments==2.17.2
39 pyarsing==3.1.1
40 python-dateutil==2.8.2
41 pytz==2023.4
42 pyzmq==25.1.2
43 scikit-learn==1.4.0
44 scipy==1.12.0
45 size==1.16.0
46 stack-data==0.6.3
47 threadpoolctl==3.2.0
48 tornado==6.4
49 traitlets==5.14.1
50 tzdata==2023.4
51 wcwidth==0.2.13
52 Werkzeug==3.0.1
```

## Step 6: Running Flask App



The screenshot shows a VS Code editor with a file explorer on the left and a code editor in the center. The file explorer shows a project named 'DEPLOYMENT\_ON\_FLASK' with files like 'static', 'style.css', 'templates', 'app.py', 'OLX\_cars\_dataset00.csv', 'Olx\_cars.pkl', 'profile', and 'requirement.txt'. The code editor displays the content of 'app.py', which is a Flask application. The application has a route '/' with a POST method. The code defines a function 'b\_lipper()' that takes a request object and returns a response. The response is a dictionary with keys 'Make', 'Model', 'Year', 'Fuel', 'Car\_document', and 'Transmission'. The code also includes a 'print' statement to display the predicted price. The code is running in a terminal window, and the output shows the application running successfully. The terminal output includes the following lines: 'Serving Flask app 'main...' on http://127.0.0.1:5000', 'Press CTRL+C to quit', and a warning message: 'WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.' The terminal also shows the output of the 'print' statement: 'The Price Prediction is: 127.0.0.1'.

```
1 @app.route('/', methods=['POST'])
2 def b_lipper():
3     Input = {}
4     Make = request.form.get('Make')
5     Model = request.form.get('Model')
6     Year = request.form.get('Year')
7     Fuel = request.form.get('Fuel')
8     Car_document = request.form.get('Car document')
9     Transmission = request.form.get('Transmission')
10    Input.append(Make)
11    Input.append(Model)
12    Input.append(Year)
13    Input.append(Fuel)
14    Input.append(Car_document)
15    Input.append(Transmission)
16    y_pred = loaded_model.predict(Input)
17    print(y_pred)
18    return 'The Price Prediction is: {}'.format(y_pred)
19 if __name__ == '__main__':
20    app.run()
```

```
127.0.0.1 - - [05/Feb/2024 16:02:39] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [05/Feb/2024 16:02:39] "GET /static/style.css HTTP/1.1" 404 -
127.0.0.1 - - [05/Feb/2024 16:02:39] "GET /static/OLX_cars_dataset00.csv HTTP/1.1" 404 -
127.0.0.1 - - [05/Feb/2024 16:02:47] "POST / HTTP/1.1" 200 -
```

## Step 7: Deployed Flask App

### (1) Home Page

---

Enter Company name:

Model:

Year:

Fuel:

Car documents:

Transmission:

### (2) Predicted Result

---

The Price Prediction is: [1298952.83549784]