

Name: Parthkumar Sutariya

Batch Code: LISUM28

Submission Date: 27/12/2023

Submitted to: Data Glacier

## Subject: Machine Learning Model Deployment on Flask

### Step 1: Selecting Toy Dataset from Kaggle

Deployment\_on\_Flask

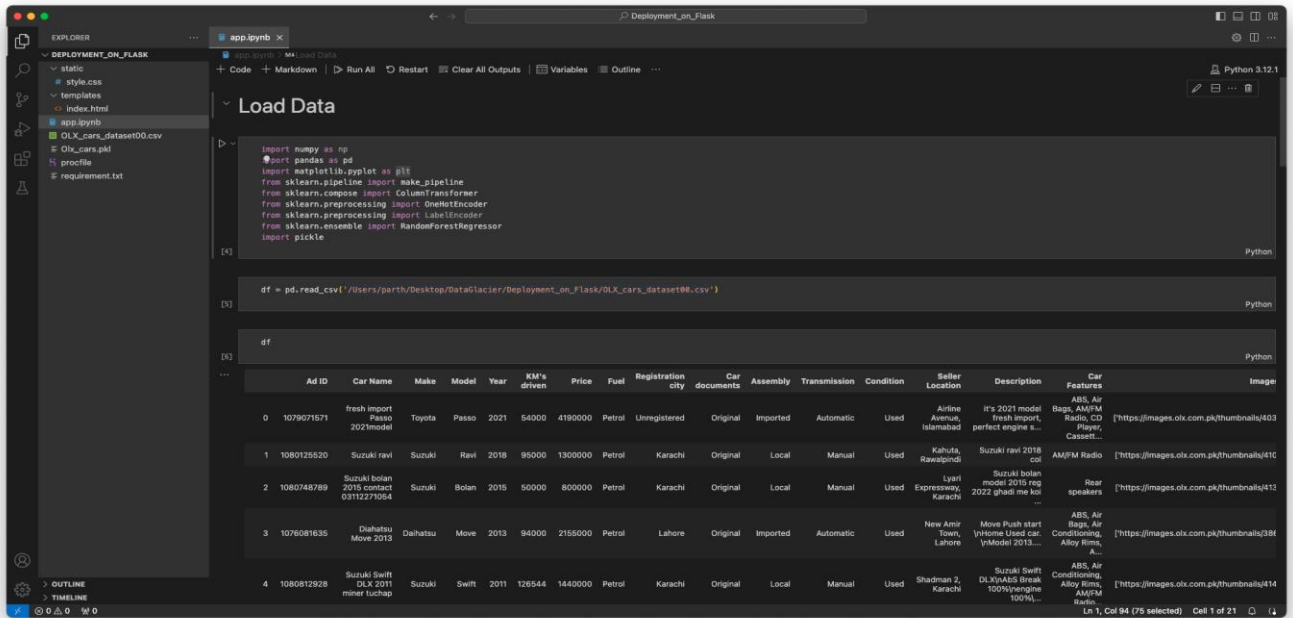
OLX\_cars\_dataset00.csv

OLX\_cars\_dataset00.csv

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,	[https://image	https://www.ox.com.pk,	
1080125520	Suzuki ravi	Suzuki	Ravi	2018	95000	1300000	Petrol	Karachi	Original	Local	Manual	Used	Kahuta, Rawal,	Suzuki ravi 201	AM/FM Radio	[https://image	https://www.ox.com.pk,	
1080748789	Suzuki bolan 2	Suzuki	Bolan	2015	50000	800000	Petrol	Karachi	Original	Local	Manual	Used	Lyari Expressw	Suzuki bolan n	Rear speakers	[https://image	https://www.ox.com.pk,	
1076081635	Daihatsu Move	Daihatsu	Move	2013	94000	2155000	Petrol	Lahore	Original	Imported	Automatic	Used	New Amir Tow	Move Push sta	ABS, Air Bags,	[https://image	https://www.ox.com.pk,	
													Home Used ca	Model 2013.	Registered in 2			
													All origina doc	Total Genuine	Never been int			
1080812928	Suzuki Swift D	Suzuki	Swift	2011	126544	1440000	Petrol	Karachi	Original	Local	Manual	Used	Shadman 2, Ka	Suzuki Swift D	ABS, Air Condi	[https://image	https://www.ox.com.pk,	
													AbS Break 100	engine 100%	suspension 100			
													Android panel	brand new tyre	2/3 piece Touc			
													all documents	cpic Clear	no work requir			
													contact no:0/3					
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,	[https://image	https://www.ox.com.pk,	
1080333426	Daihatsu Mira	Daihatsu	Mira	2013	140000	2150000	Petrol	Karachi	Original	Imported	Automatic	Used	North Nazimat	bikul okay con	ABS, Air Bags,	[https://image	https://www.ox.com.pk,	
1080784758	Suzuki Bolan 2	Suzuki	Bolan	2016	32000	1050000	Petrol	Karachi	Original	Local	Manual	Used	Korangi Indust	SUZUKI BOLAN	Alloy Rims, AM	[https://image	https://www.ox.com.pk,	
1076448625	2004 Honda C	Honda	City IDSI	2004	160000	1230000	Petrol	Sindh	Original	Local	Manual	Used	Alama Iqbal Tc	A very well mai	ABS, Air Condi	[https://image	https://www.ox.com.pk,	
													no mechanical	complete ,Engi	just buy and dr			
													original conditi	Neat and clean	only contact fo			
													No silly offers	Very good Con	10/10 conditior			
													Genuine Condi					
													-Honda City 24	-MODEL 2004	-MILAGE DRIV			
1080820574	car Bashir's 80	Suzuki	Cultus VXR	2008	75000	950000	Petrol	Faisalabad	Original	Local	Manual	Used	Al Najaf Color	Andar's hotel	Air Conditionin	[https://image	https://www.ox.com.pk,	
1080792524	Suzuki Every	Suzuki	Every	2018	101628	1850000	Petrol	Islamabad	Original	Imported	Automatic	Used	Saddar, Rawal	Suzuki Every	Model 2012 im	Air Bags, Air C	[https://image	https://www.ox.com.pk,
													Fully Genlon	White Color, h				
1080676810	I am selling XLI	Toyota	Corolla XLI	2007	346000	1895000	Petrol	Lahore	Original	Local	Manual	Used	New Northern	Alloy rim	new tyre	Air Conditionin	[https://image	https://www.ox.com.pk,
													chat bunnet de	sides p touchir				

Toy Dataset

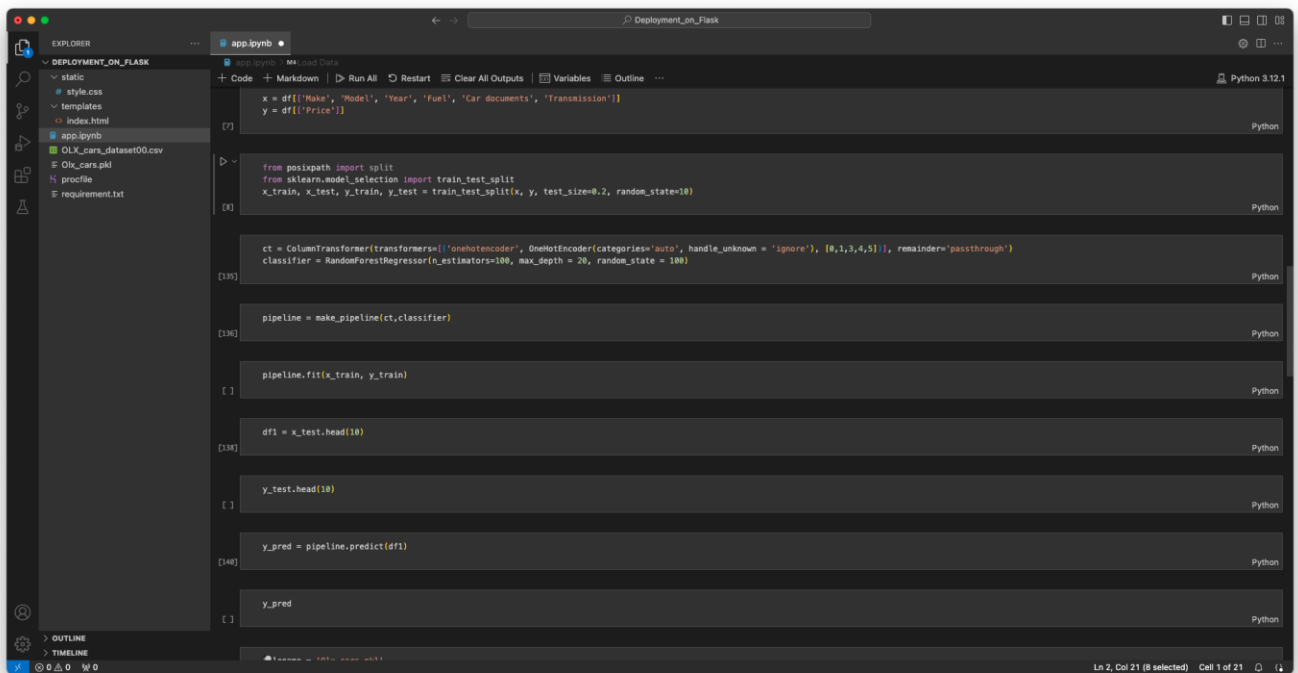
## Step 2: Model Building and Model Saving



```
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')
```

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 Passo 2021model	Toyota	Passo	2021	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...	<a href="https://images.olx.com.pk/thumbnails/403">[https://images.olx.com.pk/thumbnails/403]</a>	
1	1080125520 Suzuki ravi	Suzuki	Ravi	2018	95000	1300000	Petrol	Karachi	Original	Local	Manual	Used	Kahuta, Rawalpindi	Suzuki ravi 2018 col	AM/FM Radio	<a href="https://images.olx.com.pk/thumbnails/410">[https://images.olx.com.pk/thumbnails/410]</a>	
2	1080748789 Suzuki bolan 2015 contact 031527304	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	<a href="https://images.olx.com.pk/thumbnails/412">[https://images.olx.com.pk/thumbnails/412]</a>	
3	1076081635 Daihatsu Move 2013	Daihatsu	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...	<a href="https://images.olx.com.pk/thumbnails/386">[https://images.olx.com.pk/thumbnails/386]</a>	
4	1080812928 Suzuki Swift DLX 2011 minor lachup	Suzuki	Swift	2011	126544	1440000	Petrol	Karachi	Original	Local	Manual	Used	Shedman 2, Karachi	Suzuki Swift DLX1800S Break 100%(engine 100%...	ABS, Air Conditioning, Alloy Rims, AM/FM Radio...	<a href="https://images.olx.com.pk/thumbnails/414">[https://images.olx.com.pk/thumbnails/414]</a>	



```
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=18)
```

```
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 = 188)
```

```
pipeline = make_pipeline(ct, classifier)
```

```
pipeline.fit(x_train, y_train)
```

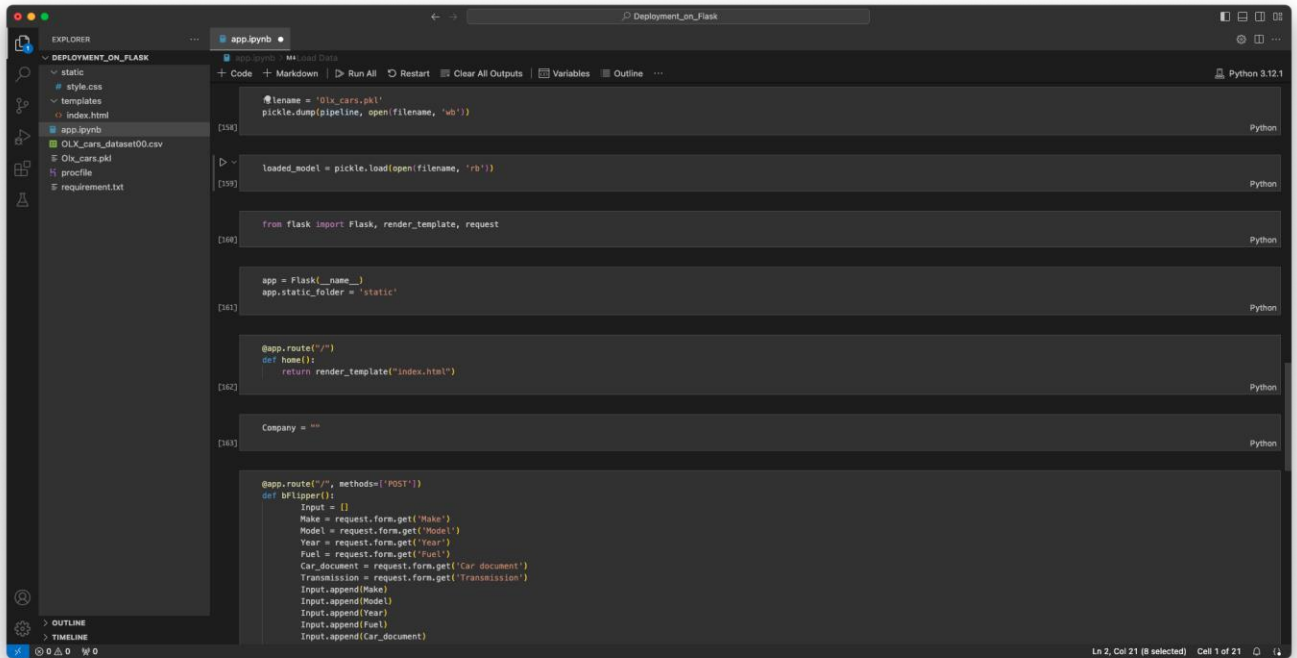
```
df1 = x_test.head(10)
```

```
y_test.head(10)
```

```
y_pred = pipeline.predict(df1)
```

```
y_pred
```

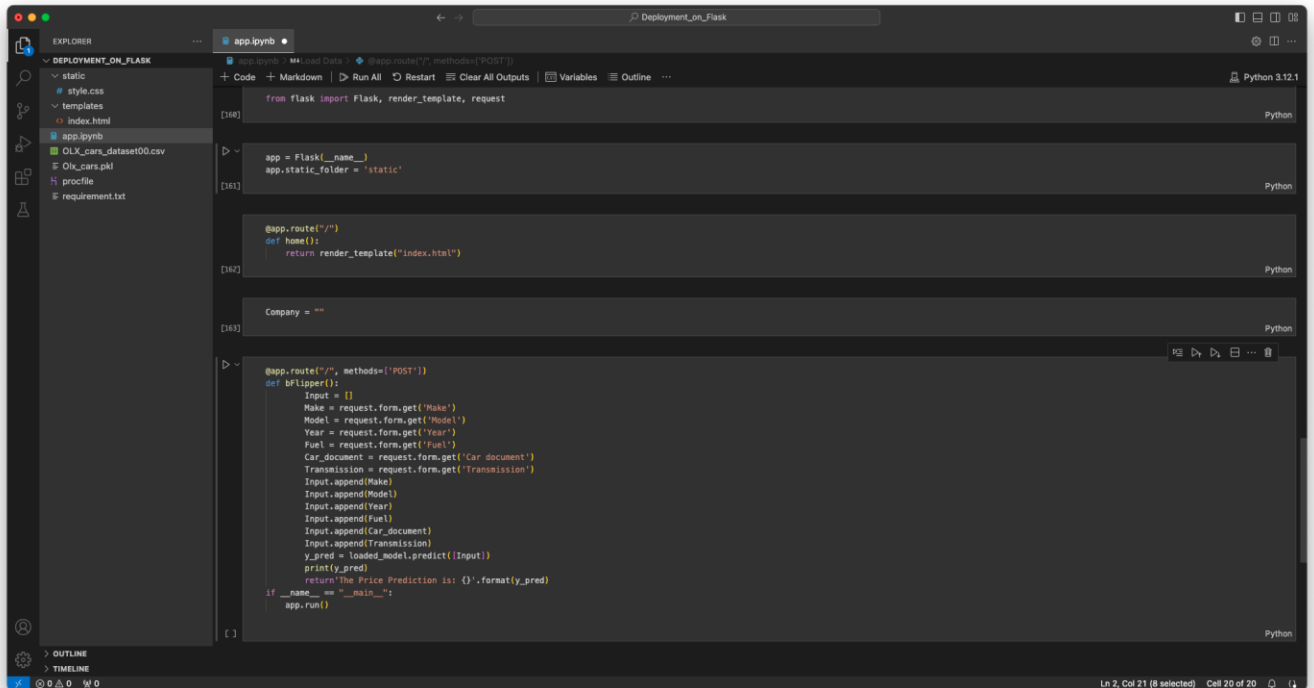
### Step 3: Building Flask App



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

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

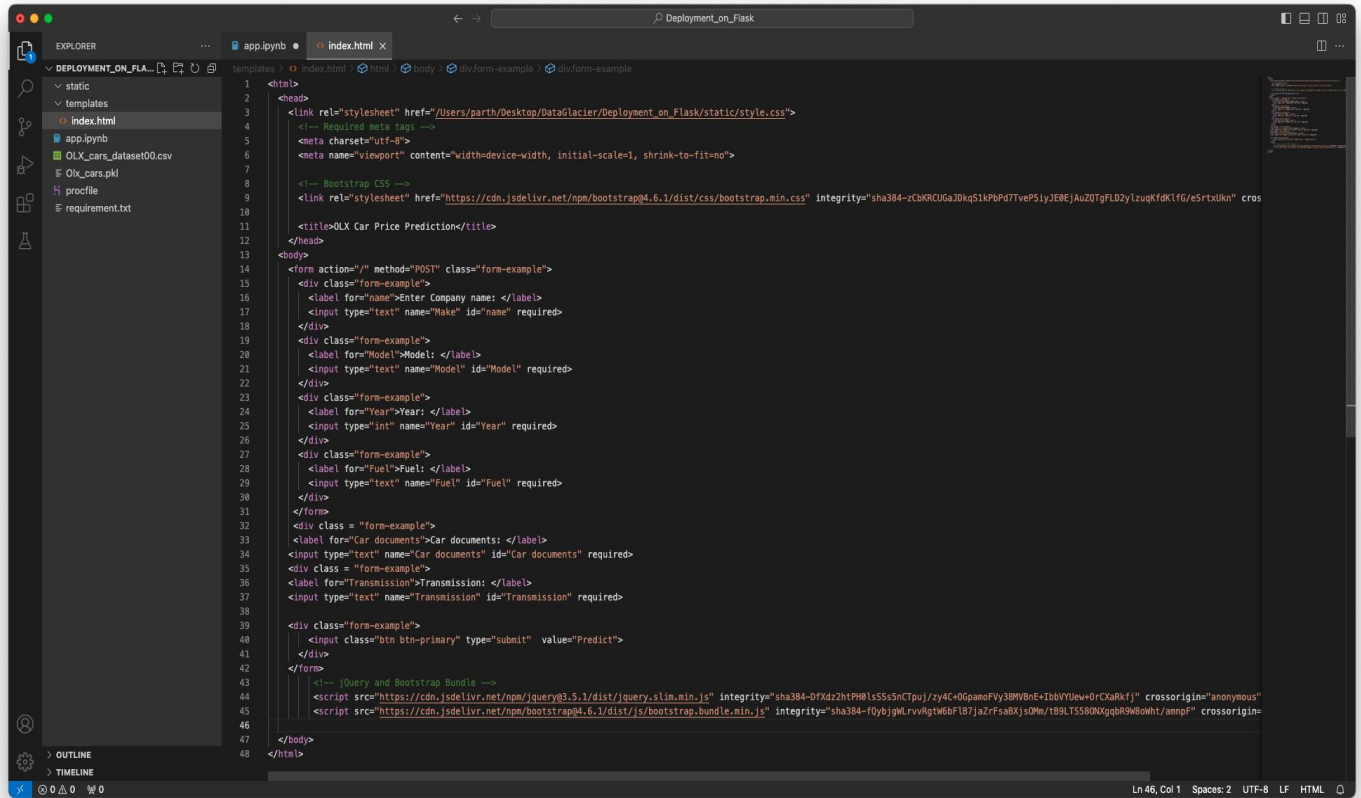


The screenshot shows the VS Code editor with the file explorer on the left displaying the project structure for 'DEPLOYMENT\_ON\_FLASK'. The main editor window shows the 'app.py' file with the following code:

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

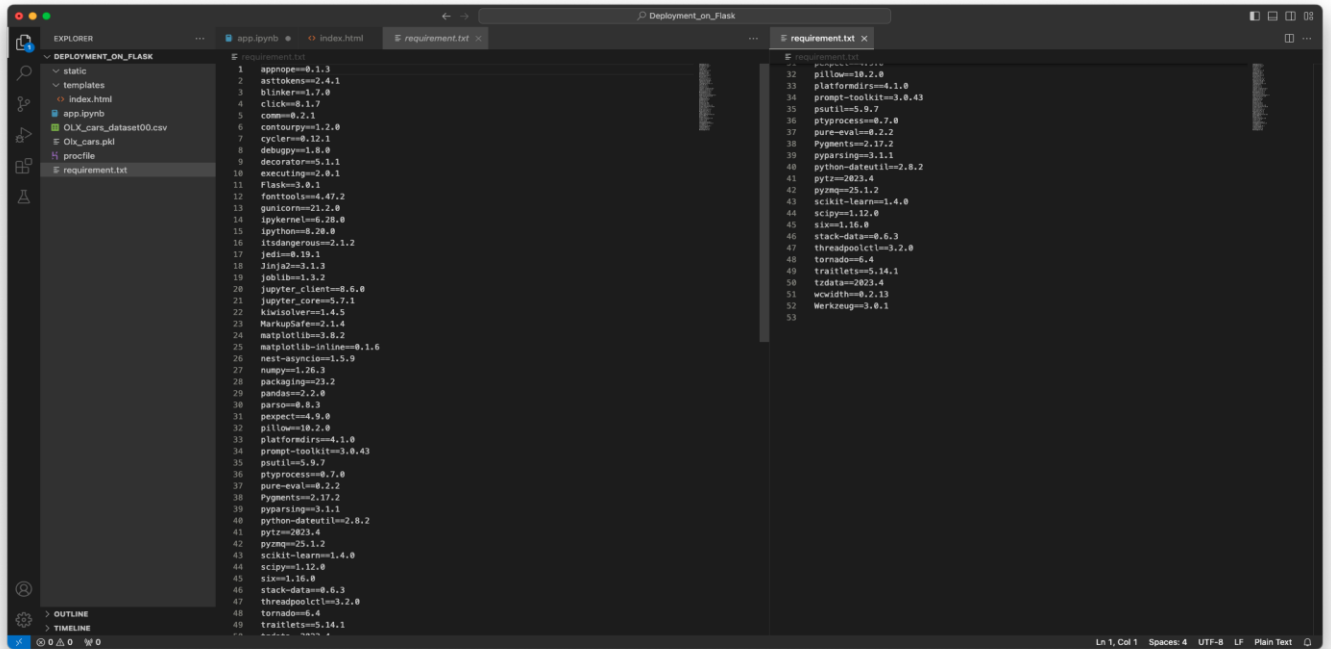
The status bar at the bottom indicates 'Ln 2, Col 21 (8 selected) Cell 20 of 20'.

## Step 4: Building HTML file



```
1 <html>
2 <head>
3   <link rel="stylesheet" href="/Users/parth/Desktop/DataGlacier/Deployment_on_Flask/static/style.css">
4   <!-- Required meta tags -->
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7   <!-- Bootstrap CSS -->
8   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css" integrity="sha384-7CkRCUGaJ0kq51kPbP977Vep51yJ8EjAuZ0TgFLD2y1zuqKfDkLfg/eSrtXkn" crossorigin="anonymous">
9
10  <title>OLX Car Price Prediction</title>
11 </head>
12 <body>
13   <form action="/" method="POST" 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   </form>
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 </body>
43 <!-- jQuery and Bootstrap Bundle -->
44 <script src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js" integrity="sha384-DfXd22HtP94L561K4FPPfsoC3k0L6p3CHWV8WUE10bVUew+0rCxRk1f" crossorigin="anonymous"></script>
45 <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-fQybjgMLrvvAgwtq6Bf1B7jaZrFsaBXs0Me/199LTS58OUKq6BR9w8wht/annpF" crossorigin="anonymous"></script>
46
47 </body>
48 </html>
```

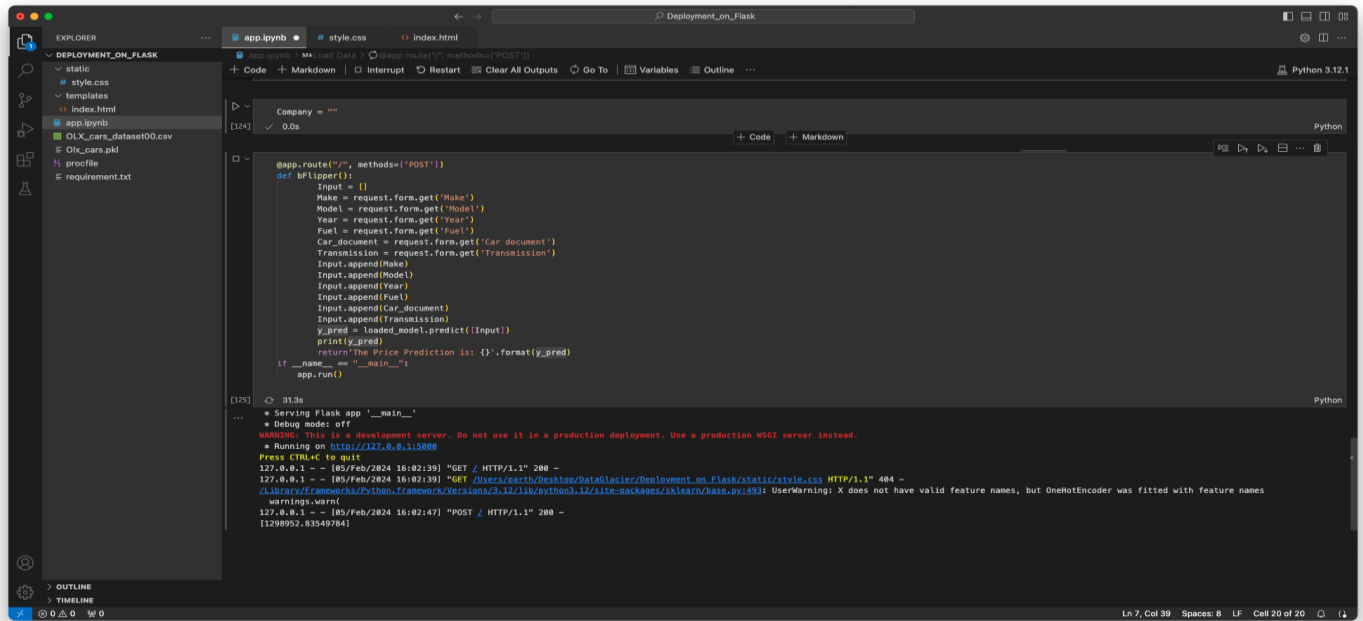
## Step 5: Creating Requirement.txt file for deploying Flask app



The screenshot shows the VS Code editor with the 'requirement.txt' file open. The file lists the following dependencies:

```
1 appnope==0.1.3
2 asttokens==2.4.1
3 blinker==1.7.0
4 click==8.1.7
5 colorama==0.2.1
6 contourpy==1.2.0
7 cycler==0.12.1
8 debugpy==1.8.0
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 kiwisolver==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 pyrsing==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 the VS Code editor with the 'app.py' file open. The code defines a Flask application with a single route for POST requests. The terminal output shows the application running successfully on port 5000.

```
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()
```

Terminal Output:

```
125 * Serving Flask app "app"
126 * Debug mode: off
127 WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
128 * Running on http://127.0.0.1:5000
129 Press CTRL+C to quit
130 127.0.0.1 - - [05/Feb/2024 16:02:39] "GET / HTTP/1.1" 200 -
131 127.0.0.1 - - [05/Feb/2024 16:02:39] "GET /static/style.css HTTP/1.1" 404 -
132 /usr/local/lib/python3.12/site-packages/sklearn/base.py:491: UserWarning: X does not have valid feature names, but OneHotEncoder was fitted with feature names
133 warnings.warn(
134 127.0.0.1 - - [05/Feb/2024 16:02:47] "POST / HTTP/1.1" 200 -
135 127.0.0.1 - - [05/Feb/2024 16:02:47]
```

## 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]