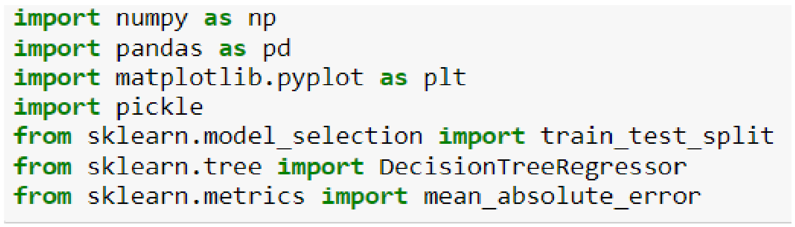
Week4: Deployment on Flask

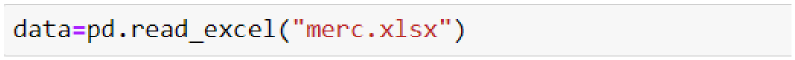
Name: *Muhammed Sefa Sözer*
Report date: 29 june 2022
Internship Batch: *LISUM10*

Project
I taught the machine the car data I got from Kaggle.
will be able to make a price estimation for the vehicle features we have entered.

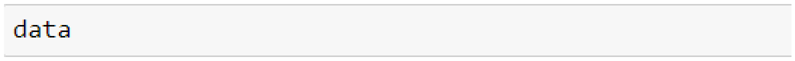
1.First, we start by importing the necessary libraries and then pulling our data.



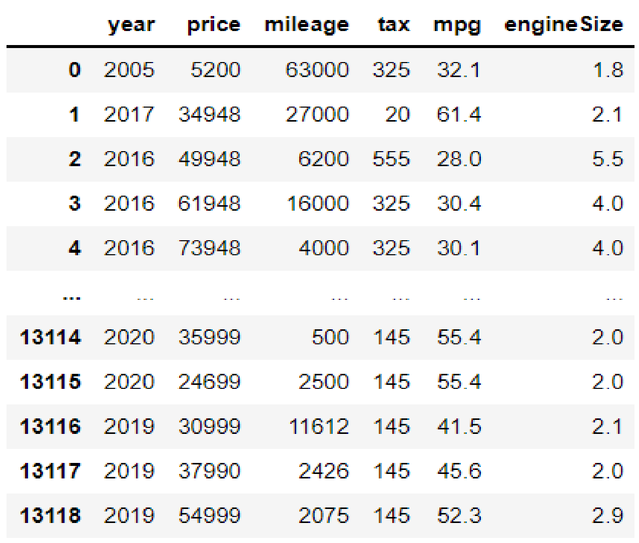
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import pickle
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error
```



```
data=pd.read_excel("merc.xlsx")
```

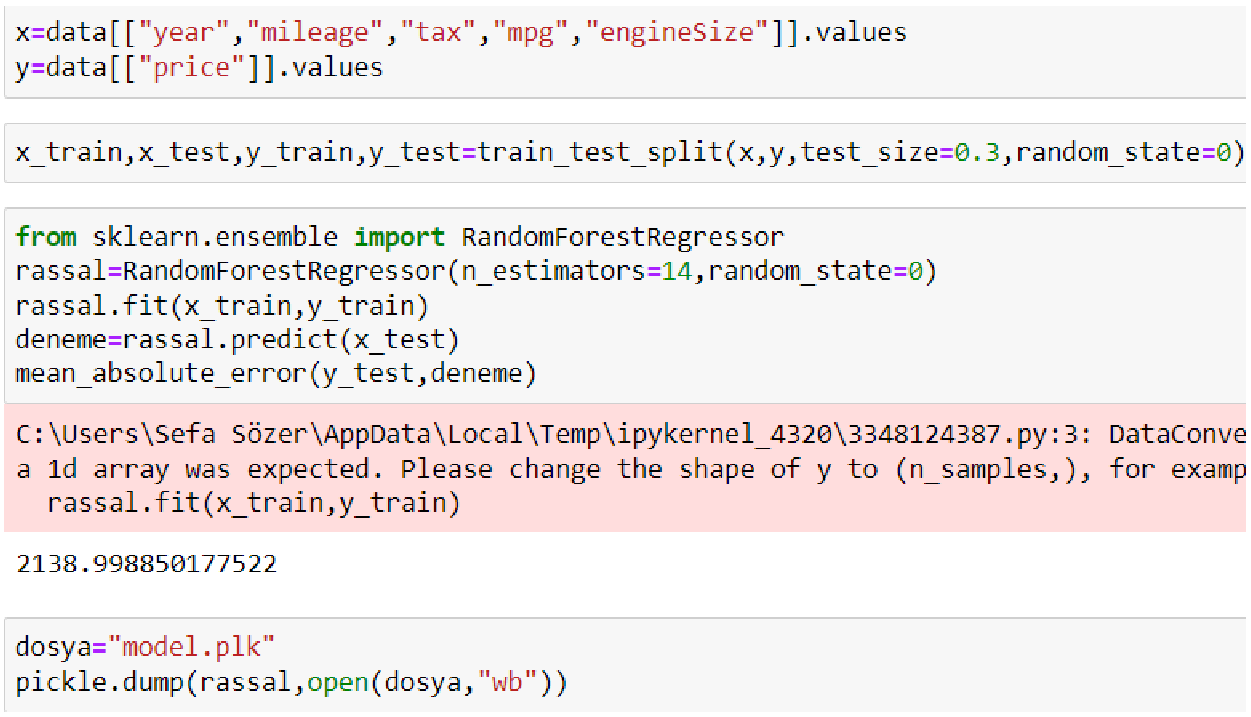


```
data
```



	year	price	mileage	tax	mpg	engineSize
0	2005	5200	63000	325	32.1	1.8
1	2017	34948	27000	20	61.4	2.1
2	2016	49948	6200	555	28.0	5.5
3	2016	61948	16000	325	30.4	4.0
4	2016	73948	4000	325	30.1	4.0
...	...	...	...	...	...	...
13114	2020	35999	500	145	55.4	2.0
13115	2020	24699	2500	145	55.4	2.0
13116	2019	30999	11612	145	41.5	2.1
13117	2019	37990	2426	145	45.6	2.0
13118	2019	54999	2075	145	52.3	2.9

2.We teach our machine with a random forest model by separating your input and output columns. Then we save our model.



```
x=data[["year","mileage","tax","mpg","engineSize"]].values
y=data[["price"]].values
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

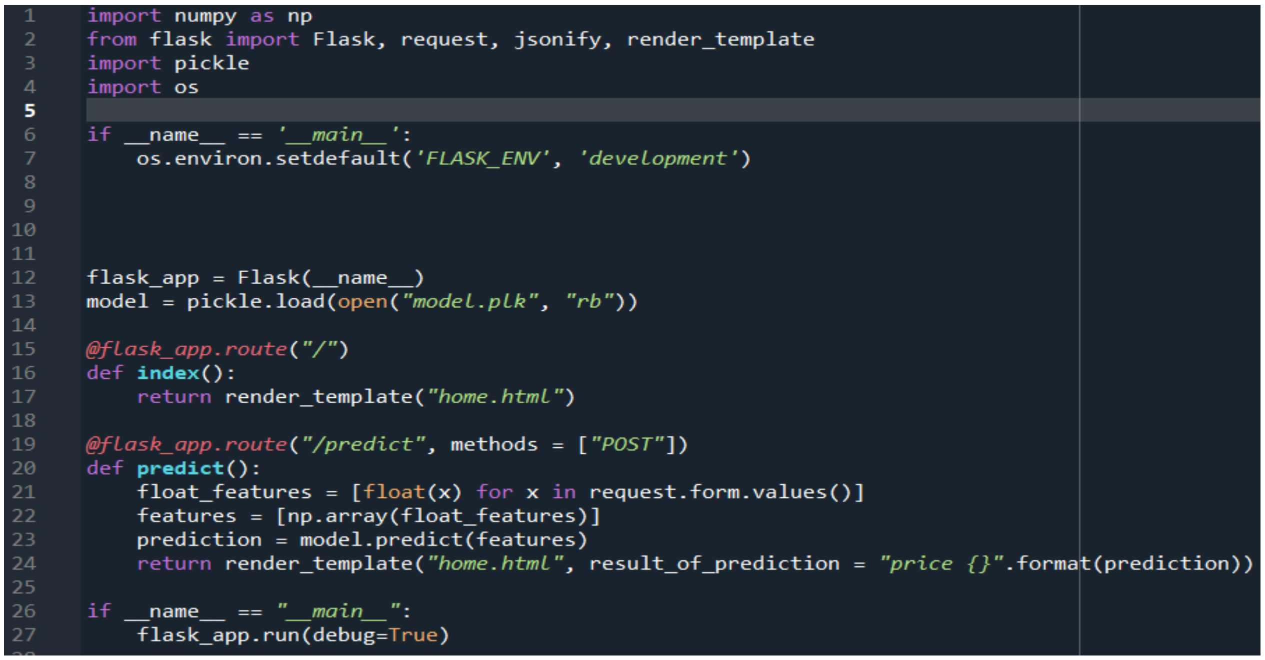
```
from sklearn.ensemble import RandomForestRegressor
rassal=RandomForestRegressor(n_estimators=14,random_state=0)
rassal.fit(x_train,y_train)
deneme=rassal.predict(x_test)
mean_absolute_error(y_test,deneme)
```

```
C:\Users\Sefa Sözer\AppData\Local\Temp\ipykernel_4320\3348124387.py:3: DataConve
a 1d array was expected. Please change the shape of y to (n_samples,), for examp
  rassal.fit(x_train,y_train)
```

```
2138.998850177522
```

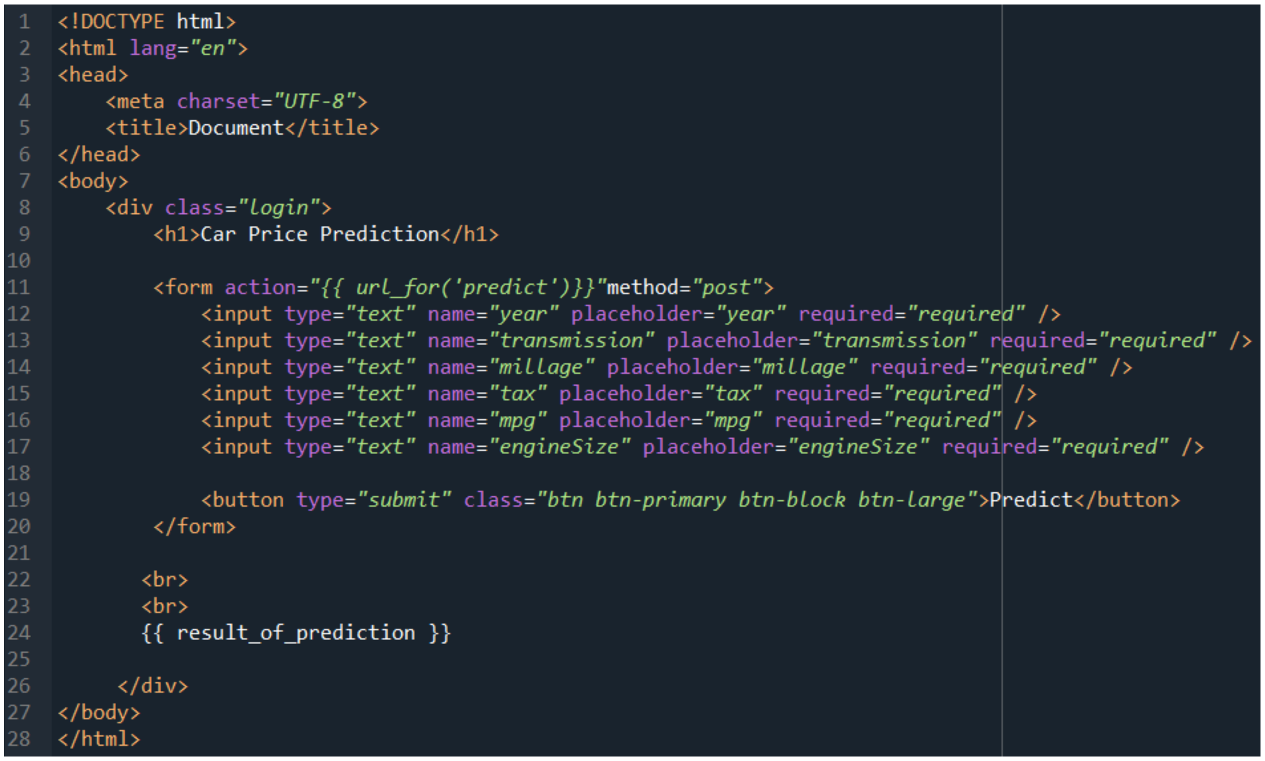
```
dosya="model.plk"
pickle.dump(rassal,open(dosya,"wb"))
```

3.Create an app.py



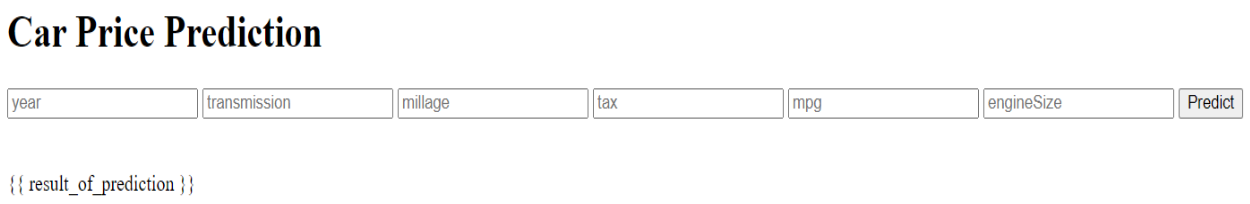
```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
import os

if __name__ == '__main__':
    os.environ.setdefault('FLASK_ENV', 'development')




flask_app = Flask(__name__)
model = pickle.load(open("model.plk", "rb"))

@flask_app.route("/")
def index():
    return render_template("home.html")

@flask_app.route("/predict", methods = ["POST"])
def predict():
    float_features = [float(x) for x in request.form.values()]
    features = [np.array(float_features)]
    prediction = model.predict(features)
    return render_template("home.html", result_of_prediction = "price {}".format(prediction))

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

4.Create an home.html



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body>
    <div class="login">
        <h1>Car Price Prediction</h1>

        <form action="{{ url_for('predict')}}"method="post">
            <input type="text" name="year" placeholder="year" required="required" />
            <input type="text" name="transmission" placeholder="transmission" required="required" />
            <input type="text" name="millage" placeholder="millage" required="required" />
            <input type="text" name="tax" placeholder="tax" required="required" />
            <input type="text" name="mpg" placeholder="mpg" required="required" />
            <input type="text" name="engineSize" placeholder="engineSize" required="required" />

            <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
        </form>

       <br>
       <br>
       {{ result_of_prediction }}

     </div>
</body>
</html>
```

5.Conclusion



Car Price Prediction

{{ result_of_prediction }}