

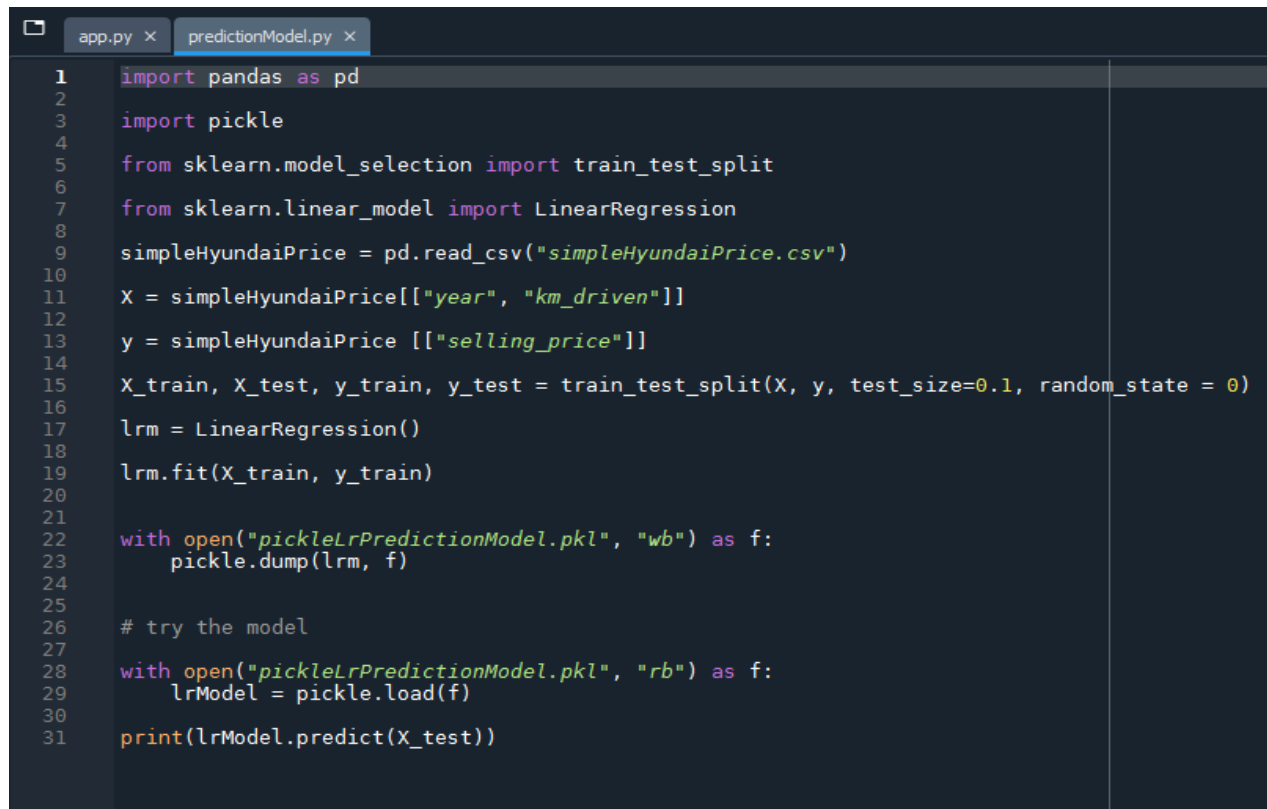
Flask Deployment

Name: Tolga Yaz

Batch Code: LISUM04

Submission Date: 18.10.2021

Submitted to: Canvas

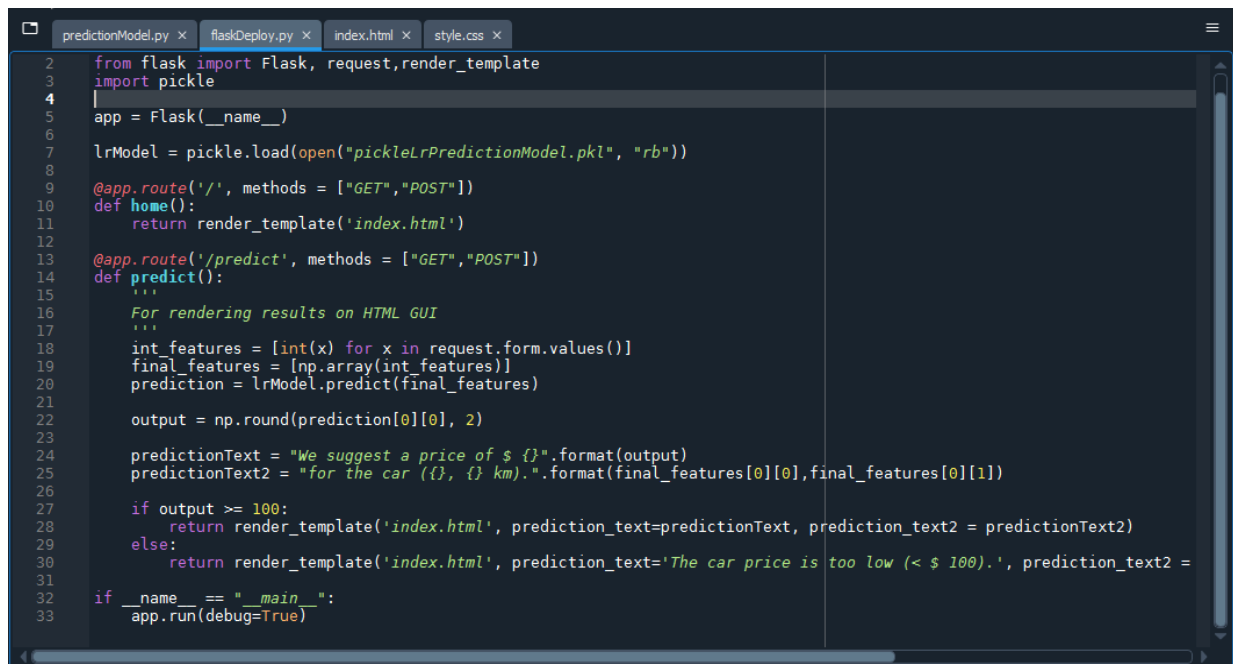
A screenshot of a code editor with two tabs: 'app.py' and 'predictionModel.py'. The 'predictionModel.py' tab is active, showing Python code for training and testing a linear regression model. The code imports pandas, pickle, and sklearn modules, reads a CSV file, splits the data, trains a LinearRegression model, saves it to a pickle file, and then loads it to make predictions on test data.

```
1 import pandas as pd
2
3 import pickle
4
5 from sklearn.model_selection import train_test_split
6
7 from sklearn.linear_model import LinearRegression
8
9 simpleHyundaiPrice = pd.read_csv("simpleHyundaiPrice.csv")
10
11 X = simpleHyundaiPrice[["year", "km_driven"]]
12
13 y = simpleHyundaiPrice[["selling_price"]]
14
15 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state = 0)
16
17 lrm = LinearRegression()
18
19 lrm.fit(X_train, y_train)
20
21
22 with open("pickleLrPredictionModel.pkl", "wb") as f:
23     pickle.dump(lrm, f)
24
25
26 # try the model
27
28 with open("pickleLrPredictionModel.pkl", "rb") as f:
29     lrModel = pickle.load(f)
30
31 print(lrModel.predict(X_test))
```

Prediction Model

```
year,selling_price,km_driven
2012,6000.0,100000
2012,6000.0,100000
2016,6900.0,80000
2008,1200.0,90000
2013,5500.0,29000
2015,5000.0,25000
2018,9000.0,24000
2018,9000.0,25000
2017,3300.0,80577
2011,3000.0,127500
2008,1950.0,90000
2013,4849.99,65000
2017,11000.0,10000
2015,7600.0,55340
2015,7600.0,55340
2015,7600.0,55340
2015,7600.0,55340
2015,7600.0,55340
```

A Snapshot from Data File (.csv)

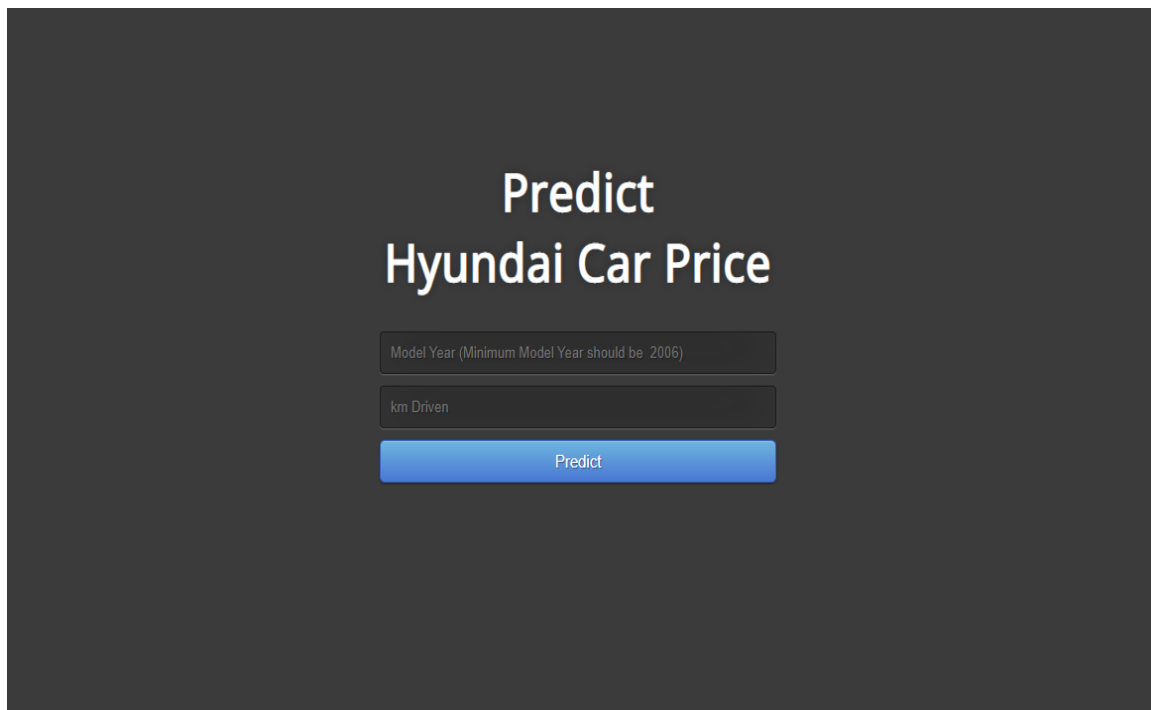


```
1 from flask import Flask, request, render_template
2 import pickle
3
4
5 app = Flask(__name__)
6
7 lrModel = pickle.load(open("pickleLrPredictionModel.pkl", "rb"))
8
9 @app.route('/', methods = ["GET", "POST"])
10 def home():
11     return render_template('index.html')
12
13 @app.route('/predict', methods = ["GET", "POST"])
14 def predict():
15     '''
16     For rendering results on HTML GUI
17     '''
18     int_features = [int(x) for x in request.form.values()]
19     final_features = [np.array(int_features)]
20     prediction = lrModel.predict(final_features)
21
22     output = np.round(prediction[0][0], 2)
23
24     predictionText = "We suggest a price of $ {}".format(output)
25     predictionText2 = "for the car ({} , {} km)".format(final_features[0][0], final_features[0][1])
26
27     if output >= 100:
28         return render_template('index.html', prediction_text=predictionText, prediction_text2 = predictionText2)
29     else:
30         return render_template('index.html', prediction_text='The car price is too low (< $ 100).', prediction_text2 =
31
32 if __name__ == "__main__":
33     app.run(debug=True)
```

A Snapshot from Deployment Code

```
predictionModel.py x flaskDeploy.py x index.html x style.css x
1 <!DOCTYPE html>
2 <html >
3 <head>
4 <meta charset="UTF-8">
5 <title>ML API</title>
6 <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
7 <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
8 <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
9 <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
10 <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
11
12 </head>
13
14 <body>
15 <div class="login">
16 <h1 style="font-size:45px;">Predict<br>Hyundai Car Price</h1>
17
18 <!-- Main Input For Receiving Query to our ML -->
19 <form action="{{ url_for('predict') }}" method="post">
20
21 <input type="text" name="year" placeholder="Model Year (Minimum Model Year should be 2006)" required="required"
22 <input type="text" name="km_driven" placeholder="km Driven" required="required" />
23
24 <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
25 </form>
26
27 <br>
28 <br>
29 {{ prediction_text }}
30 <br>
31 {{ prediction_text2 }}
```

A Snapshot from html Code



First Snapshot from the Web App

Predict Hyundai Car Price

We suggest a price of \$ 6968.87
for the car (2015, 100000 km).

Second Snapshot from the Web App

Predict Hyundai Car Price

We suggest a price of \$ 2913.27
for the car (2010, 100000 km).

Third Snapshot from the Web App