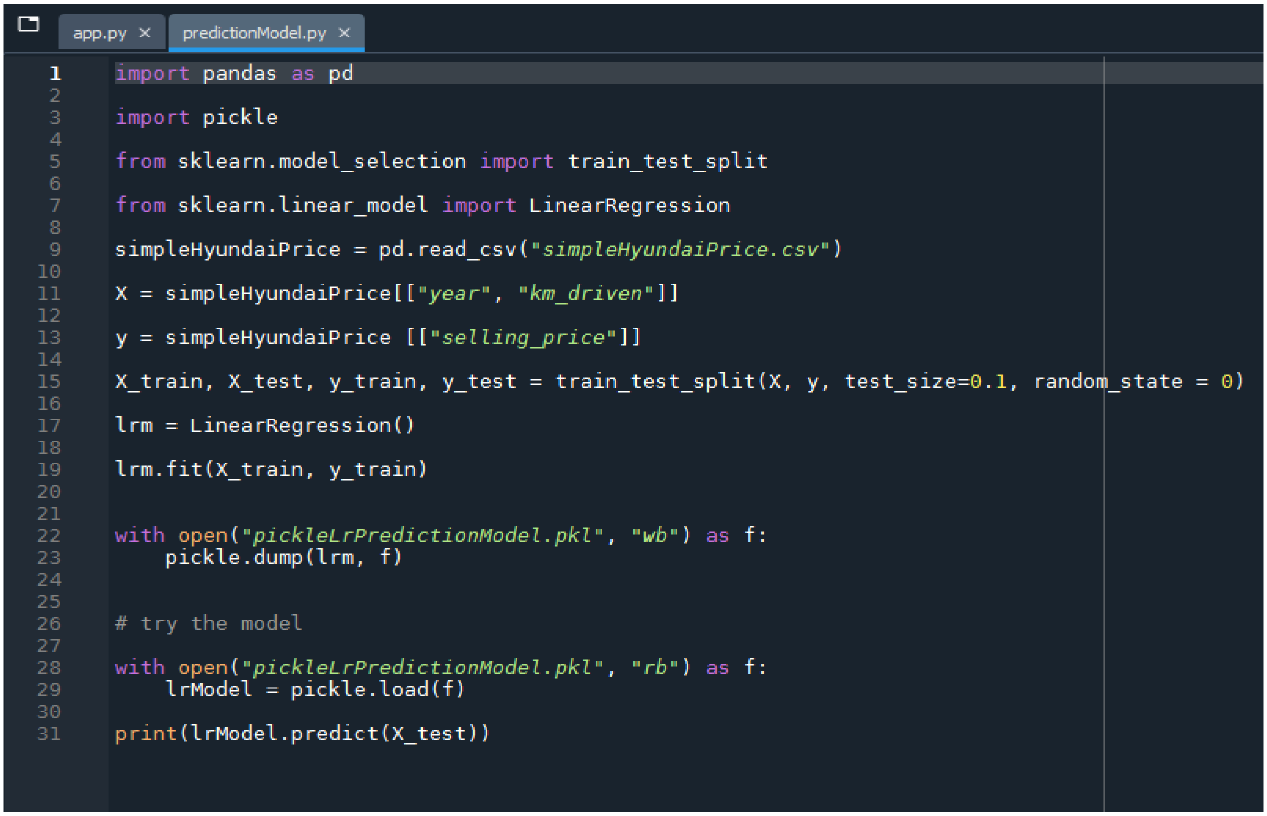
Flask Deployment

Name: Tolga Yaz

Batch Code: LISUM04

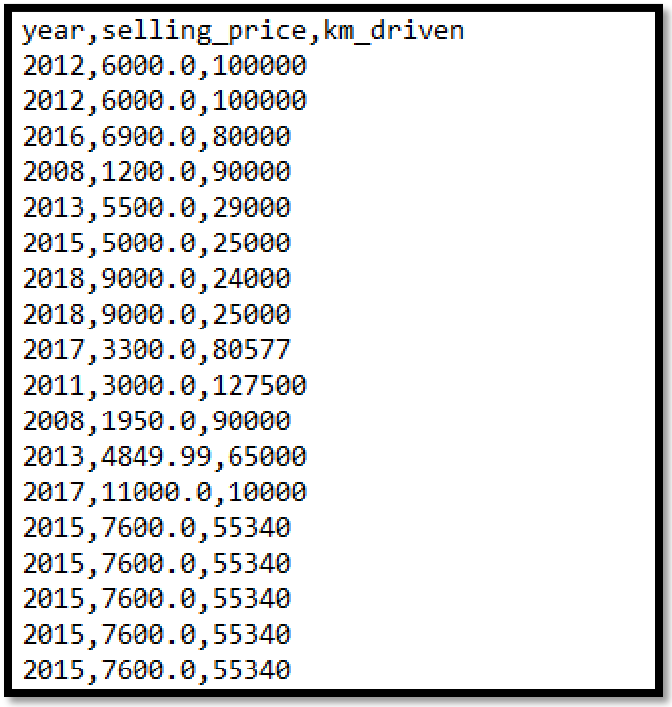
Submission Date: 18.10.2021

Submitted to: Canvas



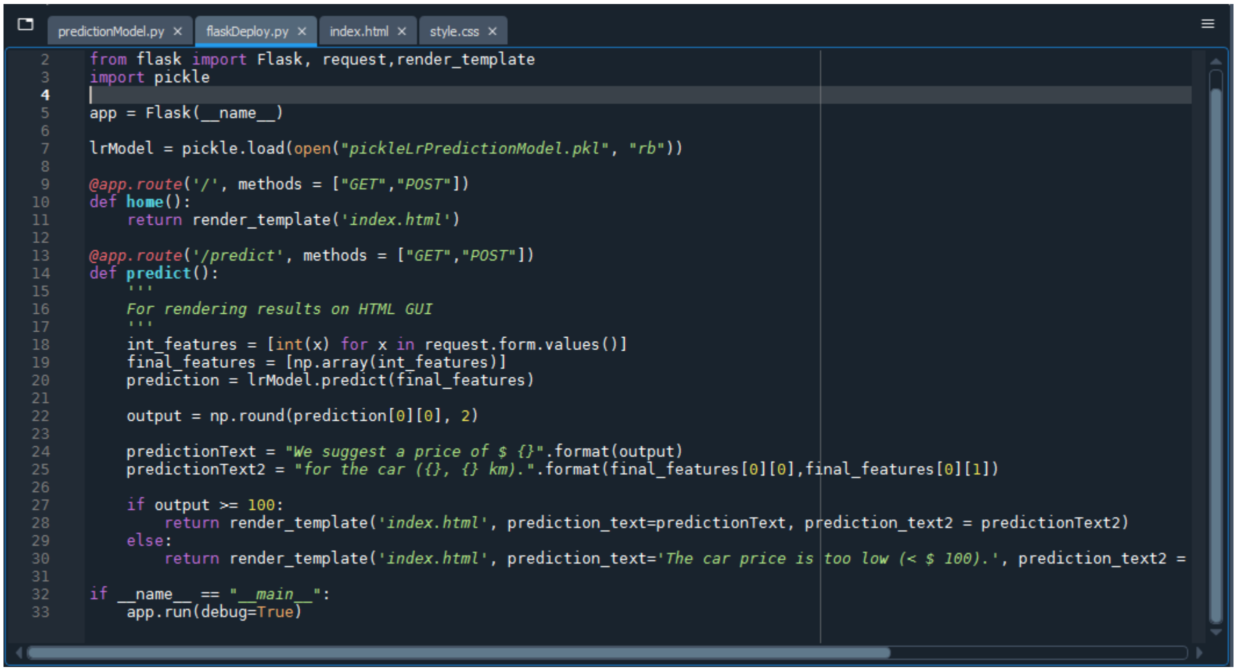
```python
import pandas as pd

import pickle

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

simpleHyundaiPrice = pd.read_csv("simpleHyundaiPrice.csv")

X = simpleHyundaiPrice[["year", "km_driven"]]

y = simpleHyundaiPrice [["selling_price"]]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state = 0)

lrm = LinearRegression()

lrm.fit(X_train, y_train)


with open("pickleLrPredictionModel.pkl", "wb") as f:
    pickle.dump(lrm, f)


# try the model

with open("pickleLrPredictionModel.pkl", "rb") as f:
    lrModel = pickle.load(f)

print(lrModel.predict(X_test))
```

Prediction Model



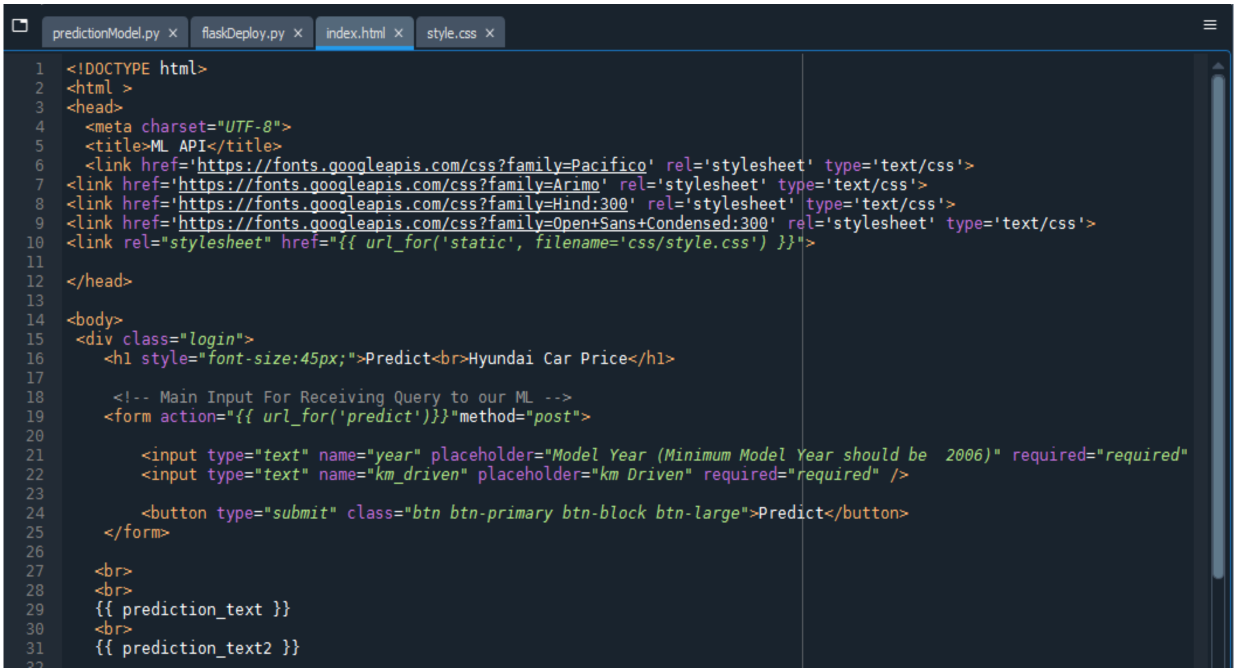
```
year,selling_price,km_driven
2012,6000.0,100000
2012,6000.0,100000
2016,6900.0,80000
2008,1200.0,90000
2013,5500.0,29000
2015,5000.0,25000
2018,9000.0,24000
2018,9000.0,25000
2017,3300.0,80577
2011,3000.0,127500
2008,1950.0,90000
2013,4849.99,65000
2017,11000.0,10000
2015,7600.0,55340
2015,7600.0,55340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

A Snapshot from Data File (.csv)



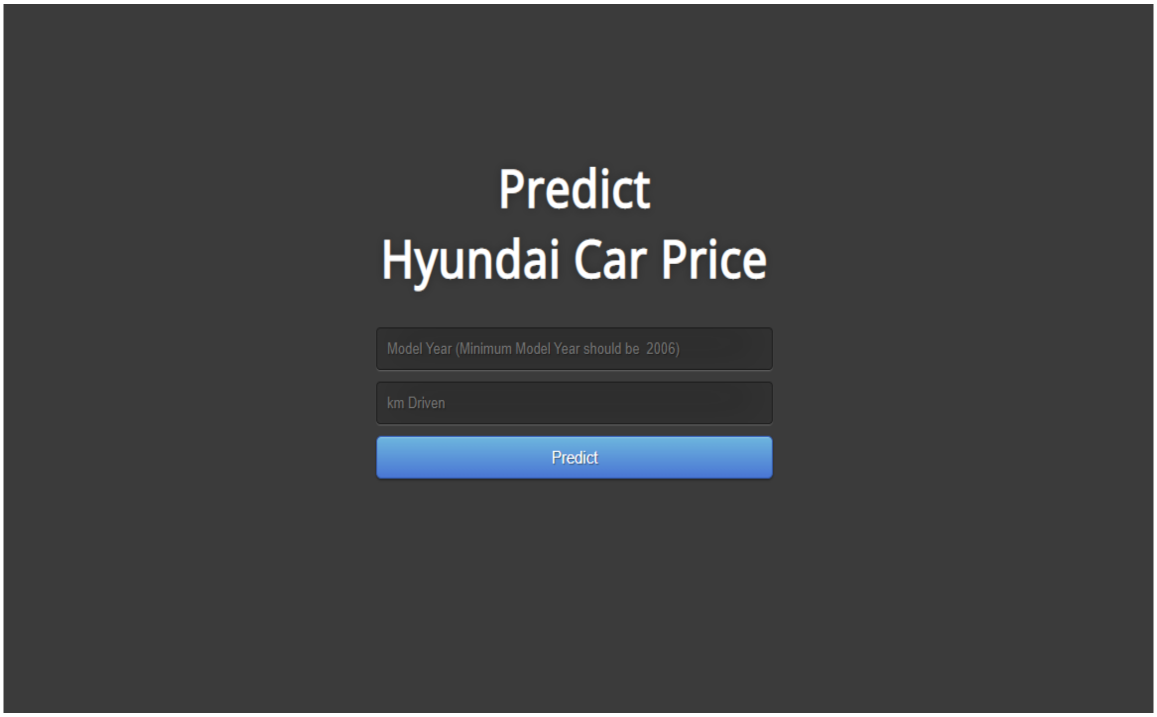
```
from flask import Flask, request,render_template
import pickle

app = Flask(__name__)

lrModel = pickle.load(open("pickleLrPredictionModel.pkl", "rb"))

@app.route('/', methods = ["GET","POST"])
def home():
    return render_template('index.html')

@app.route('/predict', methods = ["GET","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 = lrModel.predict(final_features)

    output = np.round(prediction[0][0], 2)

    predictionText = "We suggest a price of $ {}".format(output)
    predictionText2 = "for the car ({}, {} km).".format(final_features[0][0],final_features[0][1])

    if output >= 100:
        return render_template('index.html', prediction_text=predictionText, prediction_text2 = predictionText2)
    else:
        return render_template('index.html', prediction_text='The car price is too low (< $ 100).', prediction_text2 =

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

A Snapshot from Deployment Code



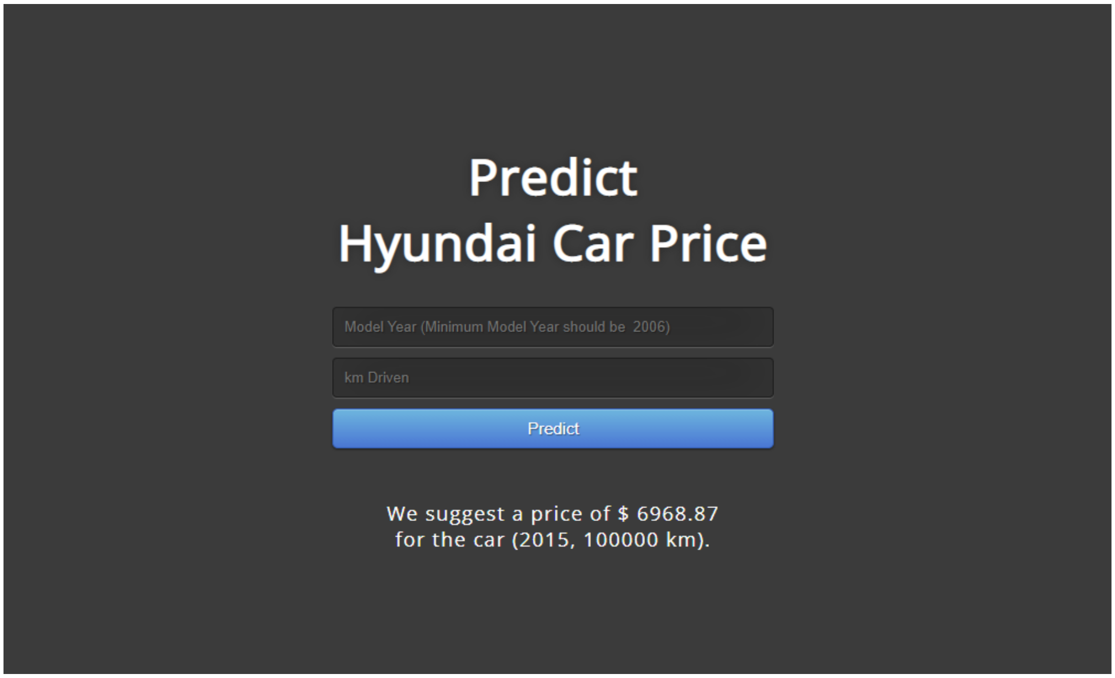
```html
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">

</head>

<body>
 <div class="login">
    <h1 style="font-size:45px;">Predict<br>Hyundai Car Price</h1>

     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">

        <input type="text" name="year" placeholder="Model Year (Minimum Model Year should be  2006)" required="required"
        <input type="text" name="km_driven" placeholder="km Driven" required="required" />

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

   <br>
   <br>
   {{ prediction_text }}
   <br>
   {{ prediction_text2 }}
```

A Snapshot from html Code



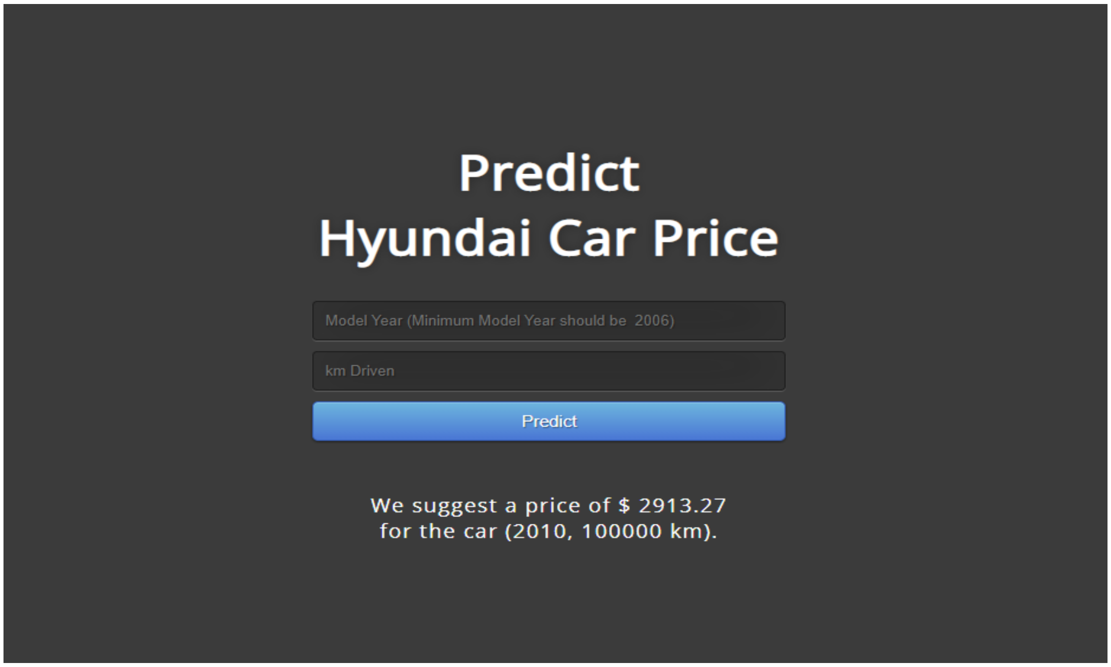
Predict
Hyundai Car Price

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