

# Model deployment using Flask

**Name:** Maha Haj Meftah

**Batch code:** LISUM05

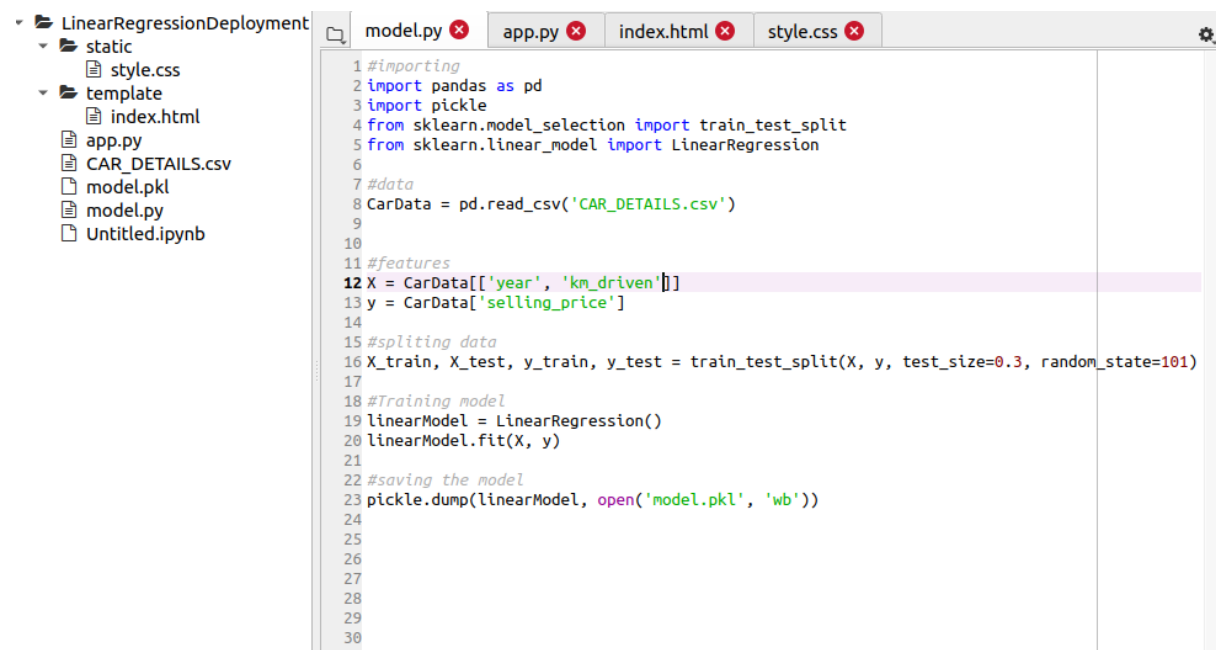
**Submission date:** 15 Jan 2022

**Submitted to:** Data Glacier

## overview:

deploy a linear regression model using Flask. The model predicts the price of used cars based on the year and the km traveled.

## first step : building the model and saving it



```
1 #importing
2 import pandas as pd
3 import pickle
4 from sklearn.model_selection import train_test_split
5 from sklearn.linear_model import LinearRegression
6
7 #data
8 CarData = pd.read_csv('CAR_DETAILS.csv')
9
10
11 #features
12 X = CarData[['year', 'km_driven']]
13 y = CarData['selling_price']
14
15 #splitting data
16 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
17
18 #Training model
19 linearModel = LinearRegression()
20 linearModel.fit(X, y)
21
22 #saving the model
23 pickle.dump(linearModel, open('model.pkl', 'wb'))
24
25
26
27
28
29
30
```

## second step:writing the code of the the web app

### app.py

```
project explorer
└─ LinearRegressionDeployment
   └─ static
      └─ style.css
   └─ template
      └─ index.html
      └─ app.py
      └─ CAR_DETAILS.csv
      └─ model.pkl
      └─ model.py
      └─ Untitled.ipynb

Editor - /home/maha/Projects/LinearRegressionDeployment/app.py
model.py x app.py x index.html x style.css x

1 # import Flask
2 import numpy as np
3 from flask import Flask, request, render_template
4 import pickle
5
6 app = Flask(__name__, template_folder='template')
7 model = pickle.load(open('model.pkl', 'rb'))
8 @app.route("/") #home
9 def home():
10     return render_template('index.html')
11
12
13 @app.route('/predict', methods=['POST'])
14 def predict():
15     '''
16     For rendering results in HTML GUI
17     '''
18     #save our features(that we get it from the form) in a list and convert them from a string to integer
19     int_features = [int(x) for x in request.form.values()]
20     final_features = [np.array(int_features)]
21     prediction = model.predict(final_features)
22     output = round(prediction[0],2)
23     return render_template('index.html', prediction_text='Price of car should be $ {}'.format(output))
24
25 if __name__ == '__main__':
26     app.run(port=5000, debug=True)
```

### html.index

```
project explorer
└─ LinearRegressionDeployment
   └─ static
      └─ style.css
   └─ template
      └─ index.html
      └─ app.py
      └─ CAR_DETAILS.csv
      └─ model.pkl
      └─ model.py
      └─ Untitled.ipynb

Editor - /home/maha/Projects/LinearRegressionDeployment/template/index.html
model.py x app.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 rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
7
8 </head>
9 <body>
10 <div class="login">
11     <h1>Predict Price of used car</h1>
12     <form action="{{url_for('predict')}}" method="post">
13         <input type="text" name="year" placeholder="Year" required="required" />
14         <input type="text" name="KM_Traveled" placeholder="KM_Traveled" required="required" />
15         <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
16     </form>
17     <br>
18     <br>
19     {{prediction_text}}
20 </div>
21 </body>
22 </html>
23
```

## style.css

```
1 @import url(https://fonts.googleapis.com/css?family=Open+Sans);
2 .btn { display: inline-block; *display: inline; *zoom: 1; padding: 4px 10px 4px; margin-b
3 .btn:hover, .btn.active, .btn.active, .btn.disabled, .btn[disabled] { background-color: #
4 .btn-large { padding: 9px 14px; font-size: 15px; line-height: normal; -webkit-border-radi
5 .btn:hover { color: #333333; text-decoration: none; background-color: #e6e6e6; background
6 .btn-primary, .btn-primary:hover { text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25); color: #fff
7 .btn-primary.active { color: rgba(255, 255, 255, 0.75); }
8 .btn-primary { background-color: #4a77d4; background-image: -moz-linear-gradient(top, #6e
9 .btn-primary:hover, .btn-primary.active, .btn-primary.active, .btn-primary.disabled, .btn
10 .btn-block { width: 100%; display: block; }
11
12 * { -webkit-box-sizing: border-box; -moz-box-sizing: border-box; -ms-box-sizing: border-box;
13
14 html { width: 100%; height: 100%; overflow: hidden; }
15
16 body {
17     width: 100%;
18     height: 100%;
19     font-family: 'Open Sans', sans-serif;
20     background: #092756;
21     background: -moz-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4) 10%,rgb
22     background: -webkit-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4) 10%,
23     background: -o-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4) 10%,rgba(
24     background: -ms-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4) 10%,rgb
25     background: -webkit-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4) 10%,
26     filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#3E1D6D', endColor
27 }
28 .login {
29     position: absolute;
30     top: 50%;
31     left: 50%;
32     margin: -150px 0 0 -150px;
33     width: 300px;
34     height: 300px;
```

## third step :running the app.py

```
maha@maha-Inspiron-3537:~/Projects/LinearRegressionDeployment$ python3 app.py
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 113-914-564
```

**a web page is generated**

127.0.0.1:5000

## Predict Price of used car

# Prediction

→ ↻ ⓘ 127.0.0.1:5000/predict ☆

**Predict Price of  
used car**

2015

50000

Predict

Price of car should be \$ 614644.78