

Model deployed with flask

Name : Mohamed Akrem Ben Jemia

- Batch code : LSMU12

-

Submission date : 5/9/2022

Submitted to : Data Glacier

Subject : create a model that predict the price of cares and deploy it on web application using flask framework

The model :

```
Entrée [1]: import pandas as pd
import pickle
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
Entrée [2]: from flask import Flask
```

```
Entrée [3]: CarData = pd.read_csv('CAR_DETAILS.csv')
```

```
Entrée [4]: x = CarData[['year', 'km_driven']]
y = CarData['selling_price']
```

```
Entrée [5]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size= 0.3, random_state=101)
```

```
Entrée [7]: linearModel = LinearRegression()
linearModel.fit(x, y)
```

```
Out[7]: LinearRegression()
```

```
Entrée [8]: pickle.dump(linearModel, open('model.pkl', 'wb'))
```

```
Entrée [ ]:
```

The web app :

```

Entrée [1]: import numpy as np
            from flask import Flask , request , render_template
            import pickle

Entrée [2]: app = Flask(__name__,template_folder='template')
            model = pickle.load(open('model.pkl','rb'))

Entrée [3]: @app.route("/")
            def home():
                return render_template('index.html')

Entrée [4]: @app.route('/predict',methods=['POST'])
            def predict():
                '''
                For rendring results in HTML GUI
                '''
                #save our features(that we get it from the form) in a List and convert them from a string to a int
                int_features =[int(x) for x in request.form.values()]
                final_features = [np.array(int_features)]
                prediction = model.predict(final_features)
                output = round(prediction[0],2)
                return render_template('index.html', prediction_text='Price of car should be $ {}'.format(output))

```

```

Entrée [*]: if __name__=='__main__':
            app.run(port=5000, debug=True,use_reloader=False)

* Serving Flask app "__main__" (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)
127.0.0.1 - - [25/Aug/2022 12:04:04] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Aug/2022 12:04:04] "GET /static/style.css HTTP/1.1" 200 -
127.0.0.1 - - [25/Aug/2022 12:04:04] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [25/Aug/2022 12:04:14] "POST /predict HTTP/1.1" 200 -

```

The web page :

predict price of used

2018

4500

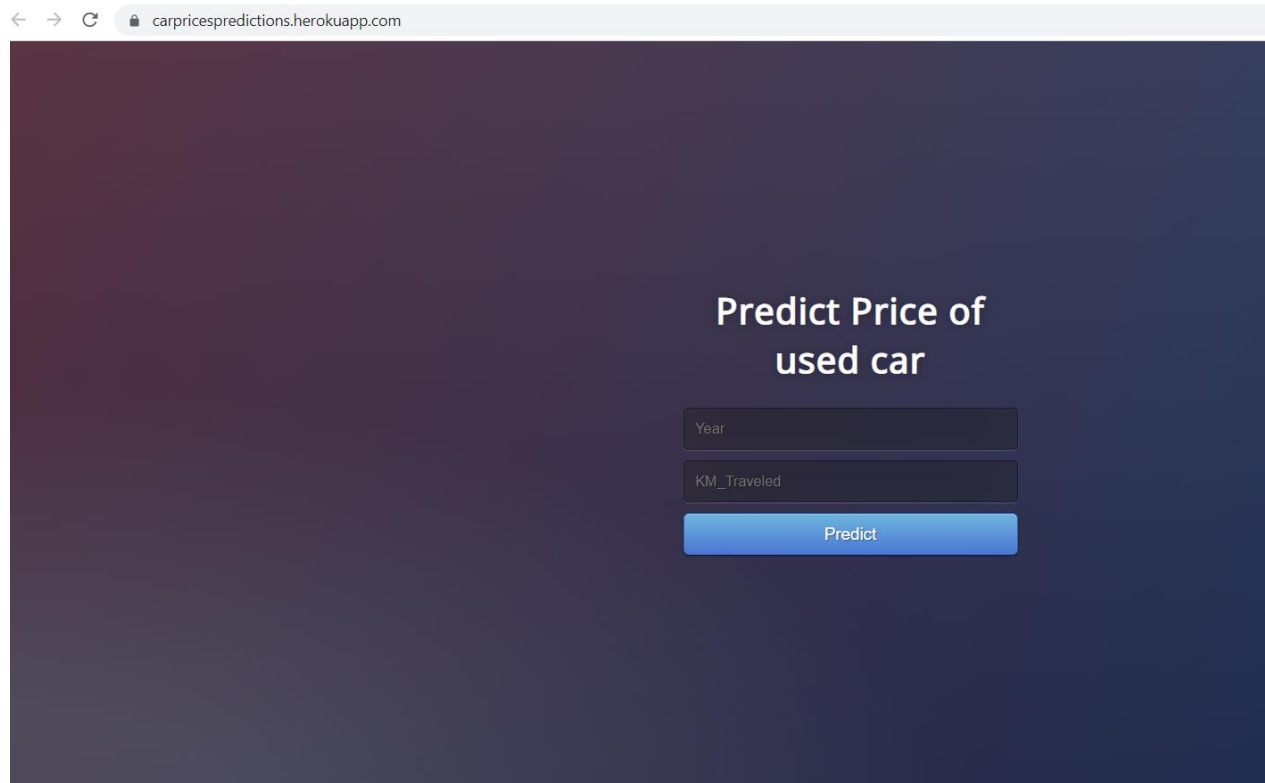
predict

Price of car should be \$ 793900.88

Activier Windows
Accédez aux paramètres pour activer Windows.

Deployment on cloud URL :

<https://carpricespredictions.herokuapp.com/>



The screenshot shows a web browser window with the address bar displaying "carpricespredictions.herokuapp.com". The main content area has a dark blue gradient background. In the center, the text "Predict Price of used car" is displayed in white. Below this text, there are two input fields: "Year" and "KM_Traveled", both with dark gray borders and light gray text. Below the input fields is a blue button with the text "Predict" in white.

Predict Price of
used car

Year

KM_Traveled

Predict