

# Data Intake Report

Name: Dummy Model Deployment on Flask

Report date: 1/28/2023

Internship Batch: LISUM17

Version: 1.0

Data intake by: Sarah Littell

Data intake reviewer: Sarah Littell

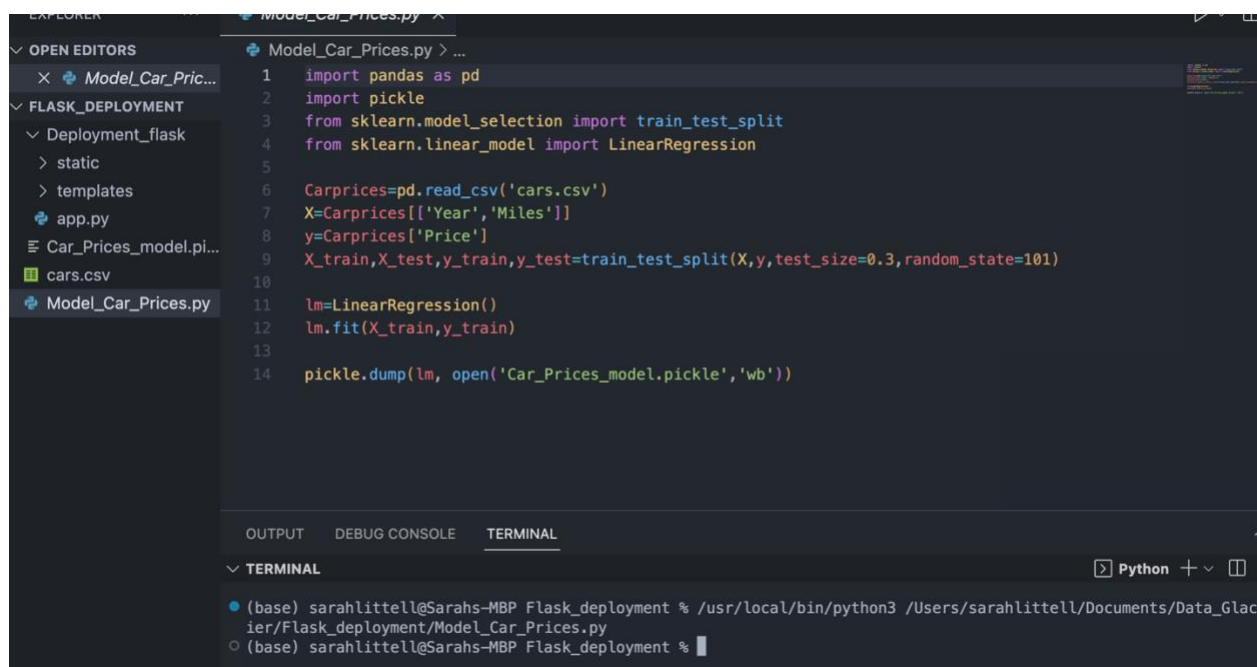
Data storage location: [https://github.com/sarahlittell/Flask\\_Deployment\\_Dummy\\_Model](https://github.com/sarahlittell/Flask_Deployment_Dummy_Model)

## Tabular data details:

<b>Total number of observations</b>	2201
<b>Total number of files</b>	1
<b>Total number of features</b>	4
<b>Base format of the file</b>	.CSV
<b>Size of the data</b>	731 KB

## Proposed Approach:

- Assuming all data is real and relevant (i.e. no outliers to significantly skew results)
- Not doing dedup because it is possible to have valid duplicates with limited information given, and this is just a dummy model



The screenshot displays a code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project structure with folders like 'FLASK\_DEPLOYMENT' and 'static', and files like 'app.py', 'Car\_Prices\_model.pi...', 'cars.csv', and 'Model\_Car\_Prices.py'. The main editor window shows the code for 'Model\_Car\_Prices.py', which imports pandas, pickle, and sklearn, reads a CSV file, splits the data, trains a linear regression model, and saves it as a pickle file. The terminal window shows the command to run the script using Python 3.

```
1 import pandas as pd
2 import pickle
3 from sklearn.model_selection import train_test_split
4 from sklearn.linear_model import LinearRegression
5
6 Carprices=pd.read_csv('cars.csv')
7 X=Carprices[['Year','Miles']]
8 y=Carprices['Price']
9 X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=101)
10
11 lm=LinearRegression()
12 lm.fit(X_train,y_train)
13
14 pickle.dump(lm, open('Car_Prices_model.pickle','wb'))
```

Terminal output:

```
(base) sarahlittell@Sarahs-MBP Flask_deployment % /usr/local/bin/python3 /Users/sarahlittell/Documents/Data_Glac
ier/Flask_deployment/Model_Car_Prices.py
(base) sarahlittell@Sarahs-MBP Flask_deployment %
```

Deployment\_flask > app.py > ...

```
1 import numpy as np
2 from flask import Flask, request, render_template
3 import pickle
4
5 app=Flask(__name__)
6 model=pickle.load(open('Car_Prices_model.pickle', 'rb'))
7
8 @app.route('/')
9 def home():
10     return render_template('index.html')
11
12 @app.route('/predict',methods=['POST'])
13 def predict():
14     """For rendering results on HTML GUI"""
15     int_features=[int(x) for x in request.form.values()]
16     final_features=np.array(int_features)
17     prediction=model.predict(final_features)
18     output=round(prediction[0],2)
19
20     return render_template('index.html', prediction_text=f'Car price is predicted to be {output}')
```

OUTPUT DEBUG CONSOLE TERMINAL

▼ TERMINAL

Python + ▾ □ □

```
/usr/local/bin/python3 /Users/sarahlittell/Documents/Data_Glacier/Flask_deployment/Deployment_flask/app.py
(base) sarahlittell@Sarahs-MBP Flask_deployment % /usr
/local/bin/python3 /Users/sarahlittell/Documents/Data_
Glacier/Flask_deployment/Deployment_flask/app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server in
stead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 700-316-446
```

## Predict Car Price

## Predict Car Price

Car price is predicted to be \$ 3438.63