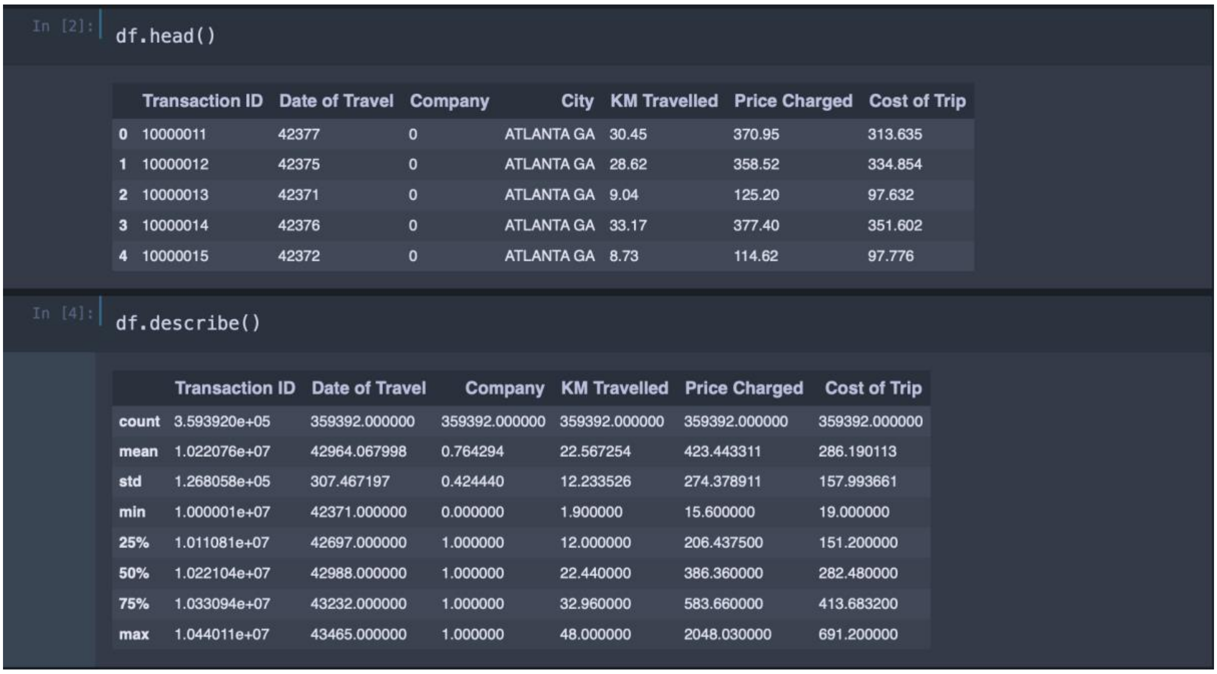
Name: **Deployment on Flask**
Submission Date: **1-07-2022**
Internship Batch: **LISUM11**
Data Intake and Model Creation by: **Kemal Cagin Sertkaya**
Submitted to: **https://github.com/cagin24/Data-Glacier-Week-4**

## Creating a Machine Learning Model and Deployment on Flask

Cab Data (**Cab_Data.csv)** selected as the dataset. Information and structure of the dataset can be seen below.



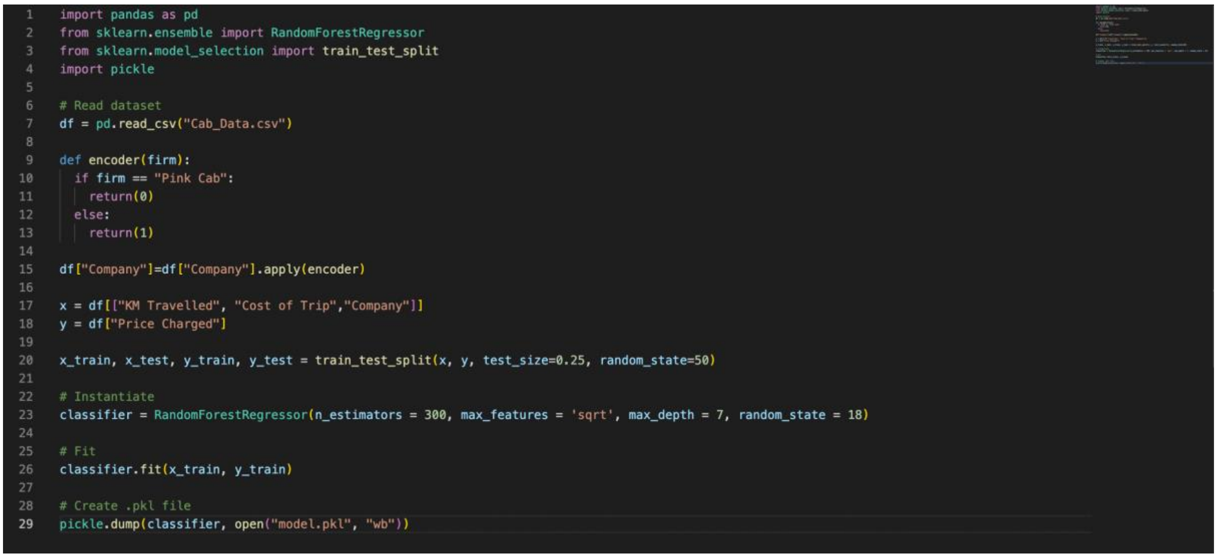
```
In [2]: df.head()
```

| | Transaction ID | Date of Travel | Company | City | KM Travelled | Price Charged | Cost of Trip |
|---|---|---|---|---|---|---|---|
| **0** | 10000011 | 42377 | 0 | ATLANTA GA | 30.45 | 370.95 | 313.635 |
| **1** | 10000012 | 42375 | 0 | ATLANTA GA | 28.62 | 358.52 | 334.854 |
| **2** | 10000013 | 42371 | 0 | ATLANTA GA | 9.04 | 125.20 | 97.632 |
| **3** | 10000014 | 42376 | 0 | ATLANTA GA | 33.17 | 377.40 | 351.602 |
| **4** | 10000015 | 42372 | 0 | ATLANTA GA | 8.73 | 114.62 | 97.776 |

```
In [4]: df.describe()
```

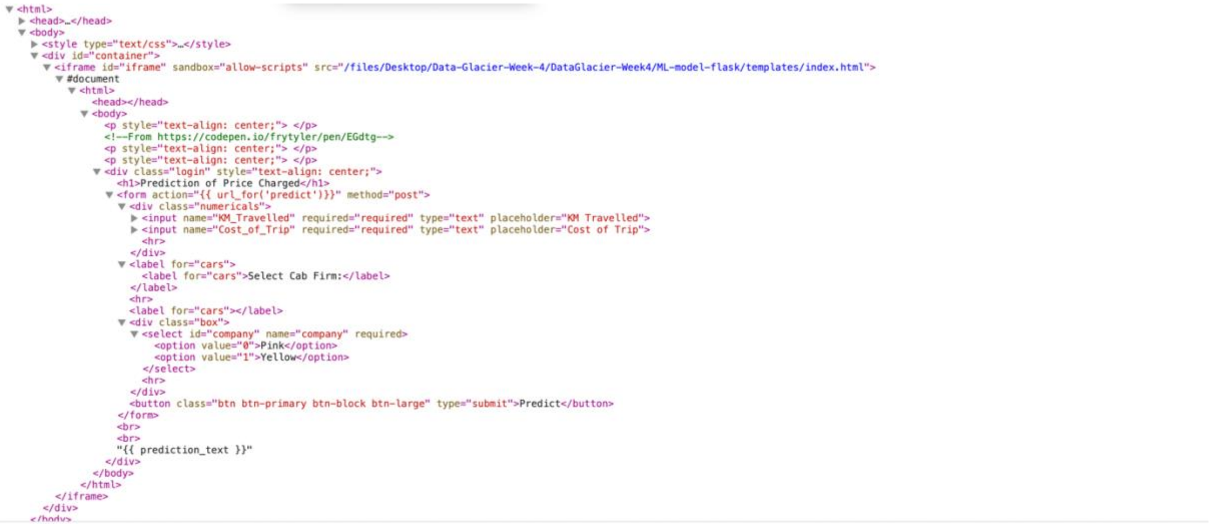
| | Transaction ID | Date of Travel | Company | KM Travelled | Price Charged | Cost of Trip |
|---|---|---|---|---|---|---|
| **count** | 3.593920e+05 | 359392.000000 | 359392.000000 | 359392.000000 | 359392.000000 | 359392.000000 |
| **mean** | 1.022076e+07 | 42964.067998 | 0.764294 | 22.567254 | 423.443311 | 286.190113 |
| **std** | 1.268058e+05 | 307.467197 | 0.424440 | 12.233526 | 274.378911 | 157.993661 |
| **min** | 1.000001e+07 | 42371.000000 | 0.000000 | 1.900000 | 15.600000 | 19.000000 |
| **25%** | 1.011081e+07 | 42697.000000 | 1.000000 | 12.000000 | 206.437500 | 151.200000 |
| **50%** | 1.022104e+07 | 42988.000000 | 1.000000 | 22.440000 | 386.360000 | 282.480000 |
| **75%** | 1.033094e+07 | 43232.000000 | 1.000000 | 32.960000 | 583.660000 | 413.683200 |
| **max** | 1.044011e+07 | 43465.000000 | 1.000000 | 48.000000 | 2048.030000 | 691.200000 |

**model.py** file is as follows. Random Forest Regressor is used as seen in line 23. Here, the model is fit and the pickle file is created.



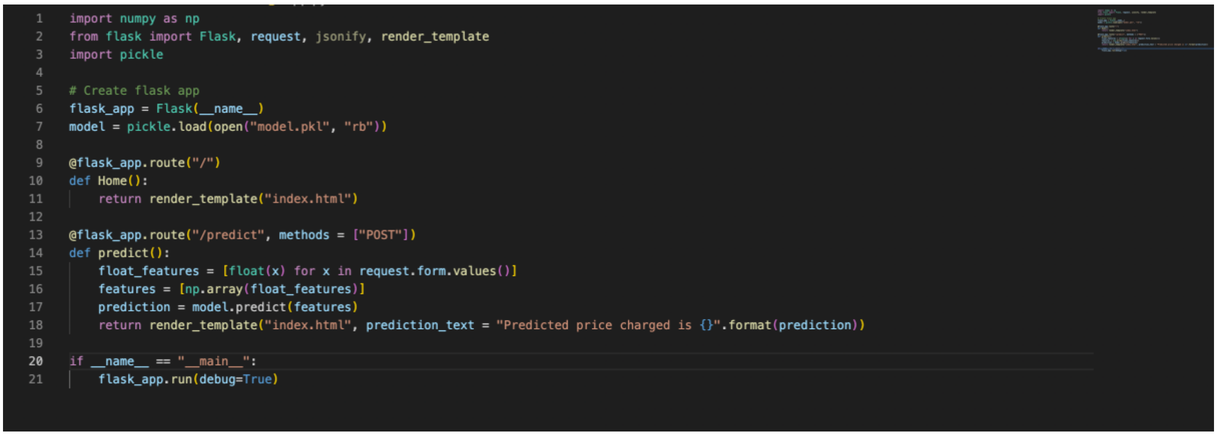
```
1   import pandas as pd
2   from sklearn.ensemble import RandomForestRegressor
3   from sklearn.model_selection import train_test_split
4   import pickle
5
6   # Read dataset
7   df = pd.read_csv("Cab_Data.csv")
8
9   def encoder(firm):
10    if firm == "Pink Cab":
11      return(0)
12    else:
13      return(1)
14
15  df["Company"]=df["Company"].apply(encoder)
16
17  x = df[["KM Travelled", "Cost of Trip","Company"]]
18  y = df["Price Charged"]
19
20  x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=50)
21
22  # Instantiate
23  classifier = RandomForestRegressor(n_estimators = 300, max_features = 'sqrt', max_depth = 7, random_state = 18)
24
25  # Fit
26  classifier.fit(x_train, y_train)
27
28  # Create .pkl file
29  pickle.dump(classifier, open("model.pkl", "wb"))
```

**index.html** code of the website
Created a simple webpage without a stylish design. Only necessary buttons, text fields and headings are provided.



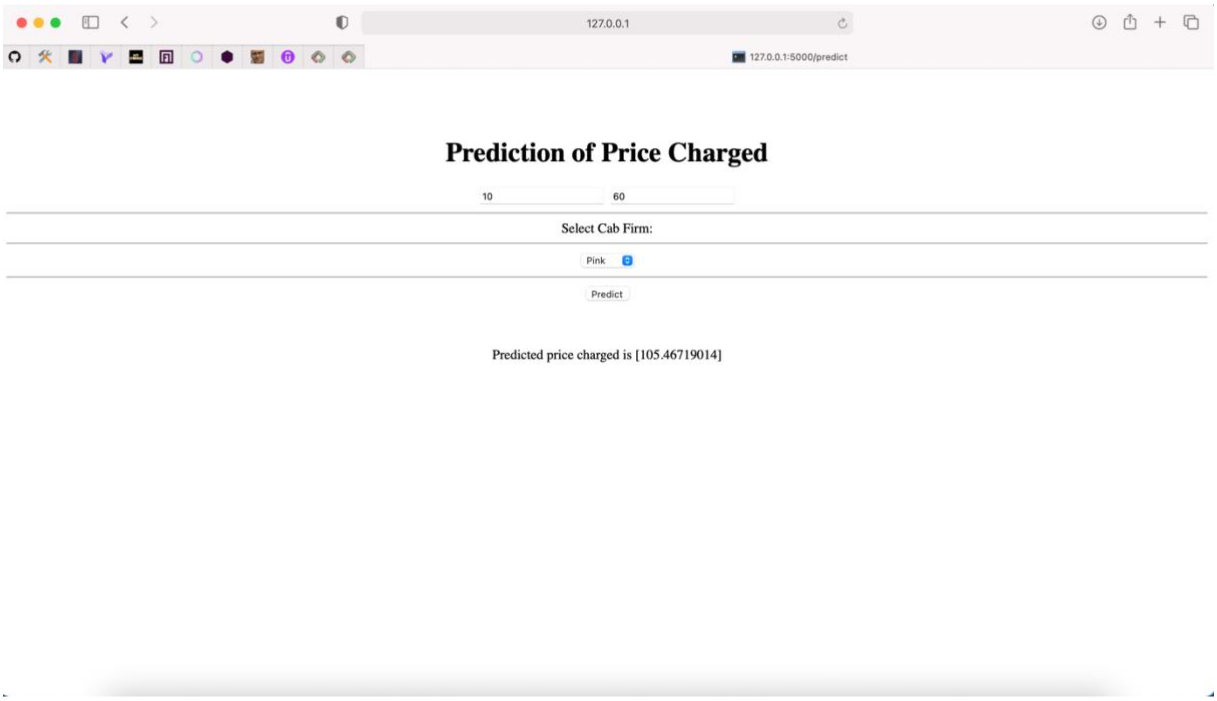
```
▼ <html>
  ▶ <head>…</head>
  ▼ <body>
    ▶ <style type="text/css">…</style>
    ▼ <div id="container">
      ▼ <iframe id="iframe" sandbox="allow-scripts" src="/files/Desktop/Data-Glacier-Week-4/DataGlacier-Week4/ML-model-flask/templates/index.html">
        ▼ #document
          ▼ <html>
              <head></head>
            ▼ <body>
                <p style="text-align: center;"> </p>
                <!--From https://codepen.io/frytyler/pen/EGdtg-->
                <p style="text-align: center;"> </p>
                <p style="text-align: center;"> </p>
              ▼ <div class="login" style="text-align: center;">
                  <h1>Prediction of Price Charged</h1>
                ▼ <form action="{{ url_for('predict')}}" method="post">
                  ▼ <div class="numericals">
                    ▶ <input name="KM_Travelled" required="required" type="text" placeholder="KM Travelled">
                    ▶ <input name="Cost_of_Trip" required="required" type="text" placeholder="Cost of Trip">
                      <hr>
                    </div>
                  ▼ <label for="cars">
                      <label for="cars">Select Cab Firm:</label>
                    </label>
                    <hr>
                    <label for="cars"></label>
                  ▼ <div class="box">
                    ▼ <select id="company" name="company" required>
                        <option value="0">Pink</option>
                        <option value="1">Yellow</option>
                      </select>
                      <hr>
                    </div>
                    <button class="btn btn-primary btn-block btn-large" type="submit">Predict</button>
                  </form>
                  <br>
                  <br>
                  "{{ prediction_text }}"
                </div>
              </body>
            </html>
        </iframe>
      </div>
    </body>
```

***app.py** file*
Created flask application here. Since the application gets input from the user and **POST** the request, line 17 basically calls the model and predict the value with the provided data.



```
1   import numpy as np
2   from flask import Flask, request, jsonify, render_template
3   import pickle
4
5   # Create flask app
6   flask_app = Flask(__name__)
7   model = pickle.load(open("model.pkl", "rb"))
8
9   @flask_app.route("/")
10  def Home():
11      return render_template("index.html")
12
13  @flask_app.route("/predict", methods = ["POST"])
14  def predict():
15      float_features = [float(x) for x in request.form.values()]
16      features = [np.array(float_features)]
17      prediction = model.predict(features)
18      return render_template("index.html", prediction_text = "Predicted price charged is {}".format(prediction))
19
20  if __name__ == "__main__":
21      flask_app.run(debug=True)
```

The deployed application can be seen below.



## Prediction of Price Charged

10 60

Select Cab Firm:

Pink

Predicted price charged is [105.46719014]