Data Glacier

Week 4 – Deployment on Flask

Name : Farheen Fatima

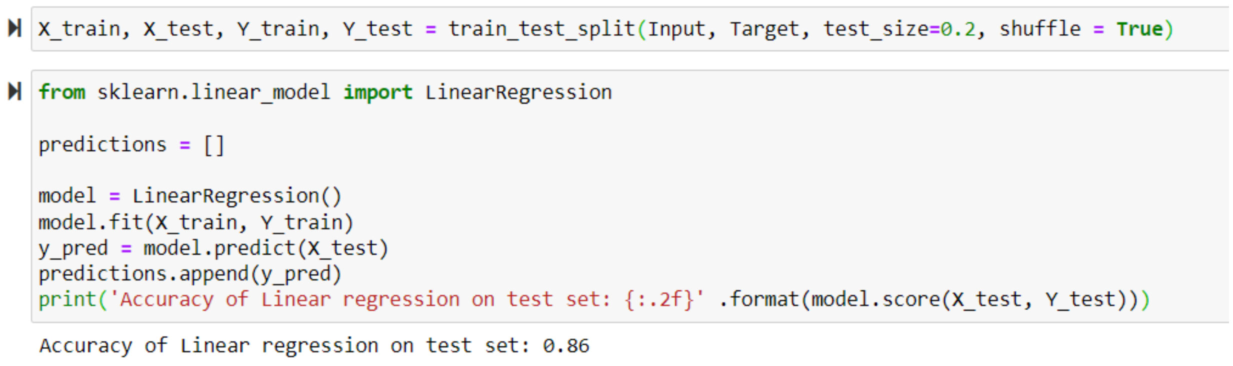
Batch code: LISUM12

Submission Date : 28-Aug-2022

Submitted to: Data Glacier

Model Deployment on Flask:

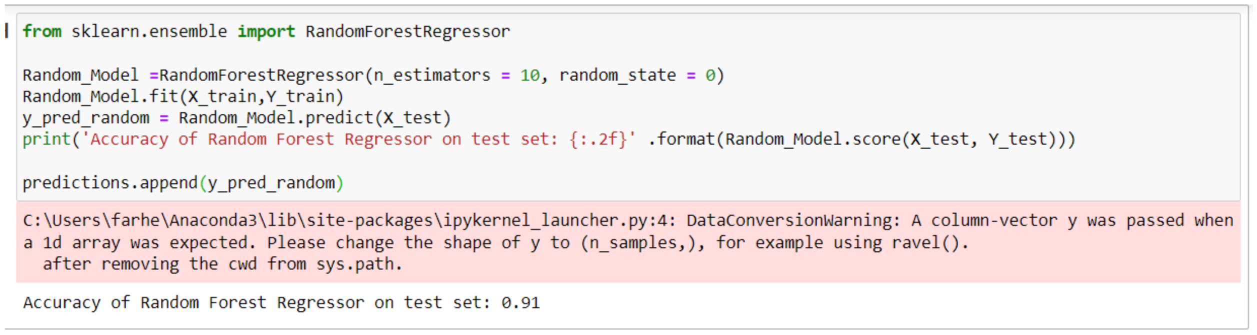
Model.py:



```
X_train, X_test, Y_train, Y_test = train_test_split(Input, Target, test_size=0.2, shuffle = True)
```

```
from sklearn.linear_model import LinearRegression

predictions = []

model = LinearRegression()
model.fit(X_train, Y_train)
y_pred = model.predict(X_test)
predictions.append(y_pred)
print('Accuracy of Linear regression on test set: {:.2f}' .format(model.score(X_test, Y_test)))
```

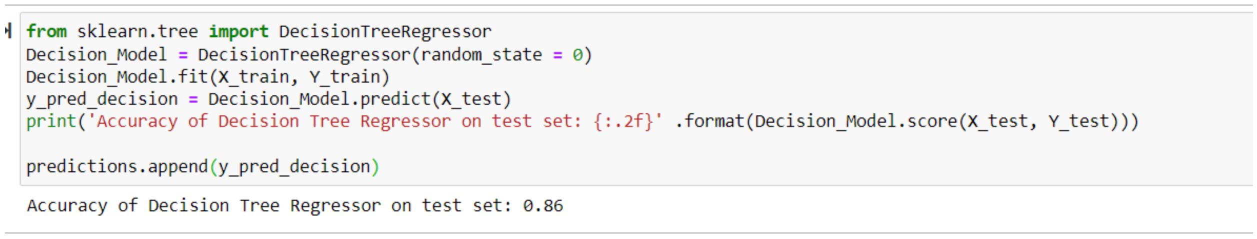
```
Accuracy of Linear regression on test set: 0.86
```



```
from sklearn.ensemble import RandomForestRegressor

Random_Model =RandomForestRegressor(n_estimators = 10, random_state = 0)
Random_Model.fit(X_train,Y_train)
y_pred_random = Random_Model.predict(X_test)
print('Accuracy of Random Forest Regressor on test set: {:.2f}' .format(Random_Model.score(X_test, Y_test)))

predictions.append(y_pred_random)
```

```
C:\Users\farhe\Anaconda3\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A column-vector y was passed when
a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  after removing the cwd from sys.path.
```

```
Accuracy of Random Forest Regressor on test set: 0.91
```

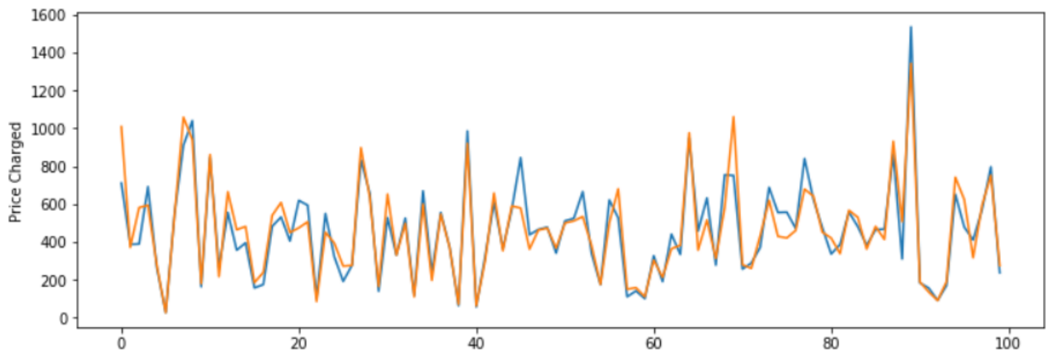


```
from sklearn.tree import DecisionTreeRegressor
Decision_Model = DecisionTreeRegressor(random_state = 0)
Decision_Model.fit(X_train, Y_train)
y_pred_decision = Decision_Model.predict(X_test)
print('Accuracy of Decision Tree Regressor on test set: {:.2f}' .format(Decision_Model.score(X_test, Y_test)))

predictions.append(y_pred_decision)
```

```
Accuracy of Decision Tree Regressor on test set: 0.86
```

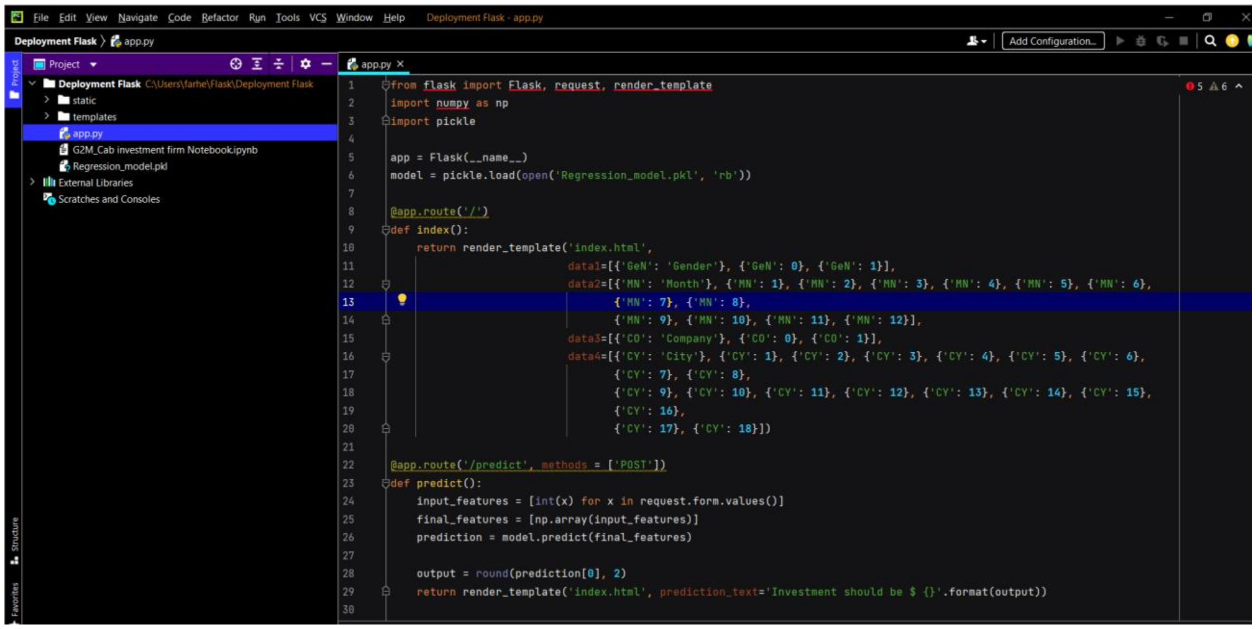
```
Y_test['predictions']
price_test = Y_test[:100]
fig, ax = plt.subplots()
fig.set_size_inches(12, 4)
plt.plot(price_test[['Price Charged', 'predictions']].values)
plt.ylabel('Price Charged', fontsize =10)
```

```
Text(0, 0.5, 'Price Charged')
```



```
pickle.dump(Random_Model, open('Regression_model.pkl', 'wb'))
pickle.load(open('Regression_model.pkl', 'rb'))
```

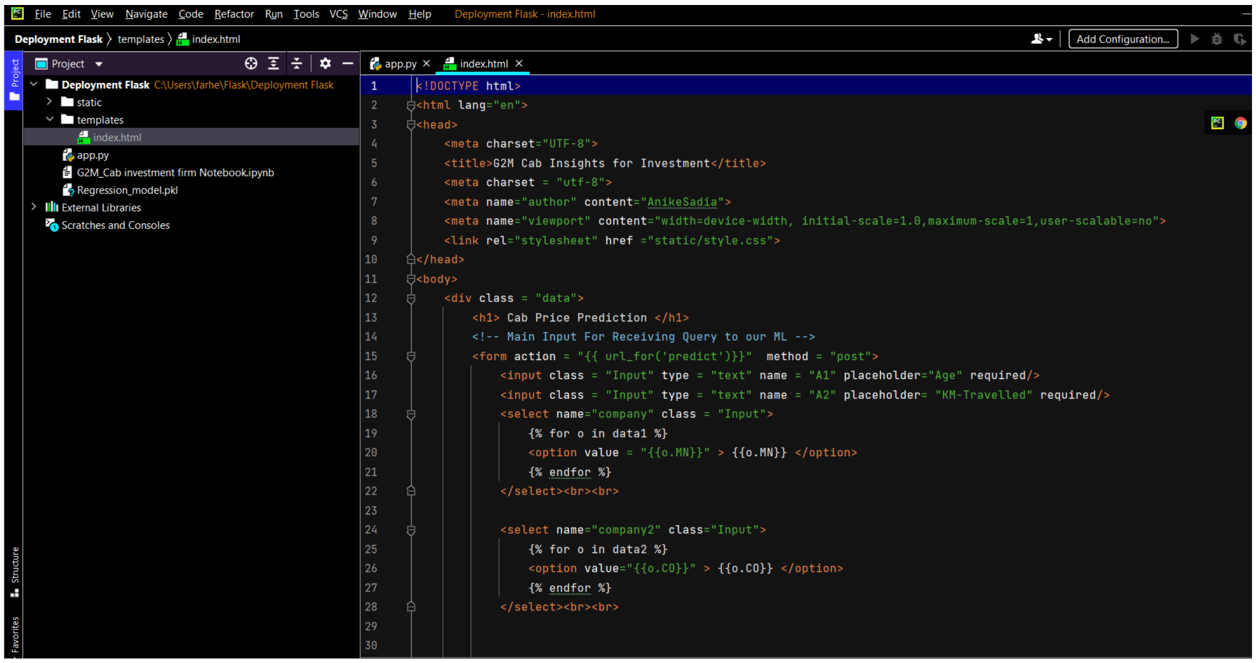
App.py:



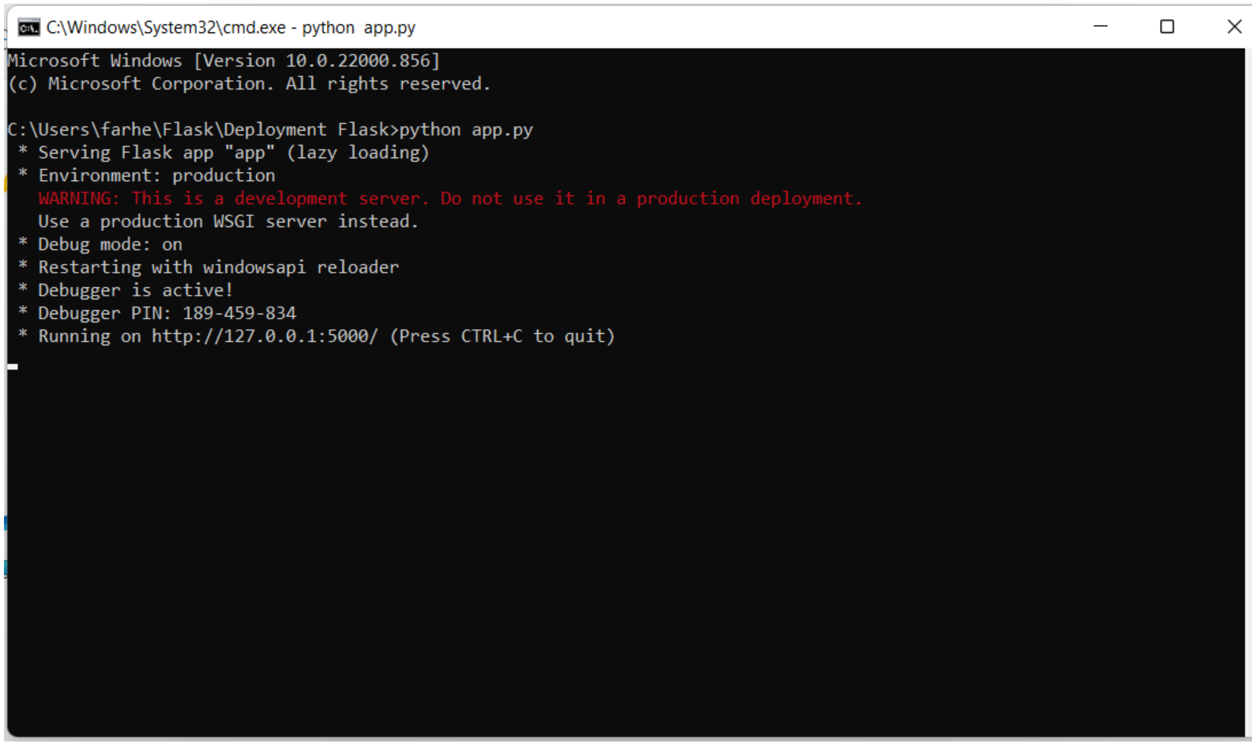
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Deployment Flask - app.py
Deployment Flask > app.py
Add Configuration...
Project
Deployment Flask C:\Users\farhe\Flask\Deployment Flask
static
templates
app.py
G2M_Cab investment firm Notebook.ipynb
Regression_model.pkl
External Libraries
Scratches and Consoles
app.py
5 6
Structure
Favorites

```
from flask import Flask, request, render_template
import numpy as np
import pickle

app = Flask(__name__)
model = pickle.load(open('Regression_model.pkl', 'rb'))

@app.route('/')
def index():
    return render_template('index.html',
                           data1=[{'GeN': 'Gender'}, {'GeN': 0}, {'GeN': 1}],
                           data2=[{'MN': 'Month'}, {'MN': 1}, {'MN': 2}, {'MN': 3}, {'MN': 4}, {'MN': 5}, {'MN': 6},
                                  {'MN': 7}, {'MN': 8},
                                  {'MN': 9}, {'MN': 10}, {'MN': 11}, {'MN': 12}],
                           data3=[{'CO': 'Company'}, {'CO': 0}, {'CO': 1}],
                           data4=[{'CY': 'City'}, {'CY': 1}, {'CY': 2}, {'CY': 3}, {'CY': 4}, {'CY': 5}, {'CY': 6},
                                  {'CY': 7}, {'CY': 8},
                                  {'CY': 9}, {'CY': 10}, {'CY': 11}, {'CY': 12}, {'CY': 13}, {'CY': 14}, {'CY': 15},
                                  {'CY': 16},
                                  {'CY': 17}, {'CY': 18}])

@app.route('/predict', methods = ['POST'])
def predict():
    input_features = [int(x) for x in request.form.values()]
    final_features = [np.array(input_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)
    return render_template('index.html', prediction_text='Investment should be $ {}'.format(output))

```

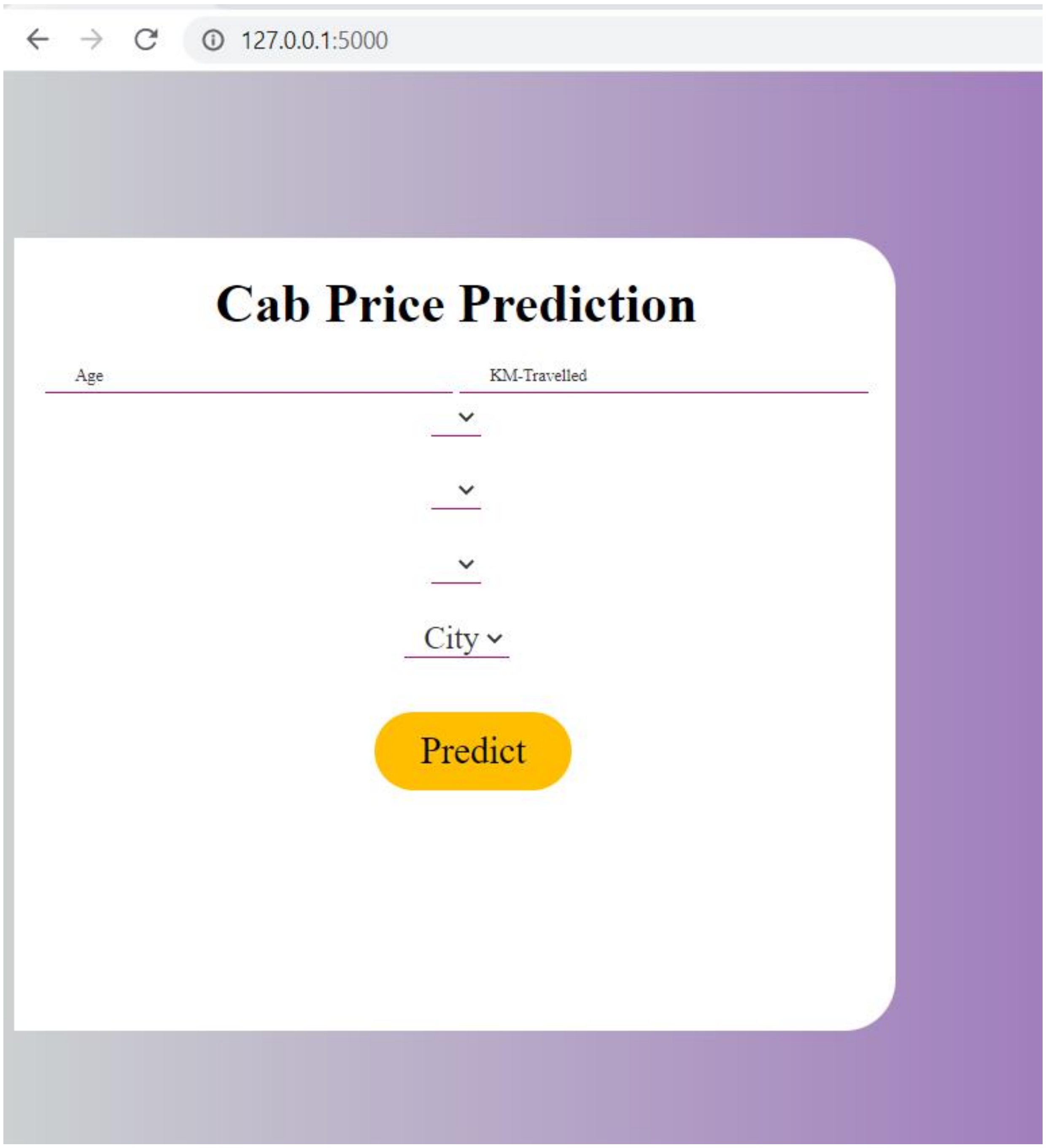
Index.html:

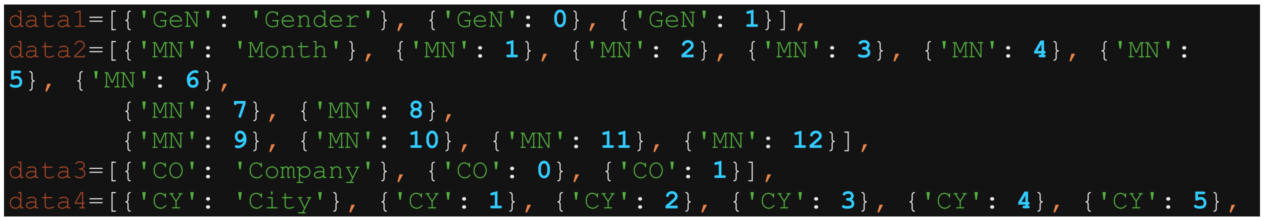

File Edit View Navigate Code Refactor Run Tools VCS Window Help Deployment Flask - index.html
Deployment Flask 〉 templates 〉 index.html
Add Configuration...
Project
Deployment Flask C:\Users\farhe\Flask\Deployment Flask
static
templates
index.html
app.py
G2M_Cab investment firm Notebook.ipynb
Regression_model.pkl
External Libraries
Scratches and Consoles
app.py
index.html
Structure
Favorites

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>G2M Cab Insights for Investment</title>
    <meta charset = "utf-8">
    <meta name="author" content="AnikeSadia">
    <meta name="viewport" content="width=device-width, initial-scale=1.0,maximum-scale=1,user-scalable=no">
    <link rel="stylesheet" href ="static/style.css">
</head>
<body>
    <div class = "data">
        <h1> Cab Price Prediction </h1>
        <!-- Main Input For Receiving Query to our ML -->
        <form action = "{{ url_for('predict')}}"  method = "post">
            <input class = "Input" type = "text" name = "A1" placeholder="Age" required/>
            <input class = "Input" type = "text" name = "A2" placeholder= "KM-Travelled" required/>
            <select name="company" class = "Input">
                {% for o in data1 %}
                <option value = "{{o.MN}}" > {{o.MN}} </option>
                {% endfor %}
            </select><br><br>

            <select name="company2" class="Input">
                {% for o in data2 %}
                <option value="{{o.CO}}" > {{o.CO}} </option>
                {% endfor %}
            </select><br><br>


```



```
Microsoft Windows [Version 10.0.22000.856]
(c) Microsoft Corporation. All rights reserved.

C:\Users\farhe\Flask\Deployment Flask>python 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
 * Restarting with windowsapi reloader
 * Debugger is active!
 * Debugger PIN: 189-459-834
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```





```
data1=[{'GeN': 'Gender'}, {'GeN': 0}, {'GeN': 1}],
data2=[{'MN': 'Month'}, {'MN': 1}, {'MN': 2}, {'MN': 3}, {'MN': 4}, {'MN':
5}, {'MN': 6},
       {'MN': 7}, {'MN': 8},
       {'MN': 9}, {'MN': 10}, {'MN': 11}, {'MN': 12}],
data3=[{'CO': 'Company'}, {'CO': 0}, {'CO': 1}],
data4=[{'CY': 'City'}, {'CY': 1}, {'CY': 2}, {'CY': 3}, {'CY': 4}, {'CY': 5},
{'CY': 6},
        {'CY': 7}, {'CY': 8},
        {'CY': 9}, {'CY': 10}, {'CY': 11}, {'CY': 12}, {'CY': 13}, {'CY': 14},
{'CY': 15},
        {'CY': 16},
        {'CY': 17}, {'CY': 18}])
```