

Name: Ritu Kukreja

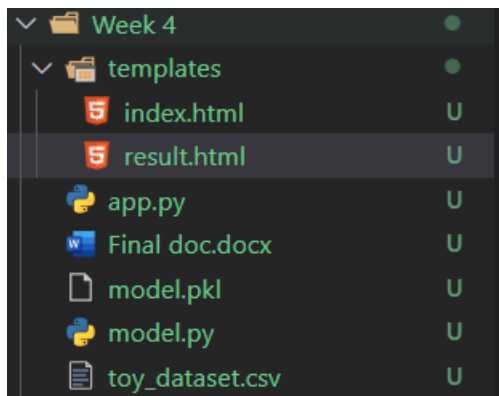
Batch Code: LISUM25

Submission date: 09/28/2023

Submitted to: Data Glacier

Following are the snapshot of each step of deployment:

1. This the folder of deployment



2. model.py – where I have created Linear Regression model

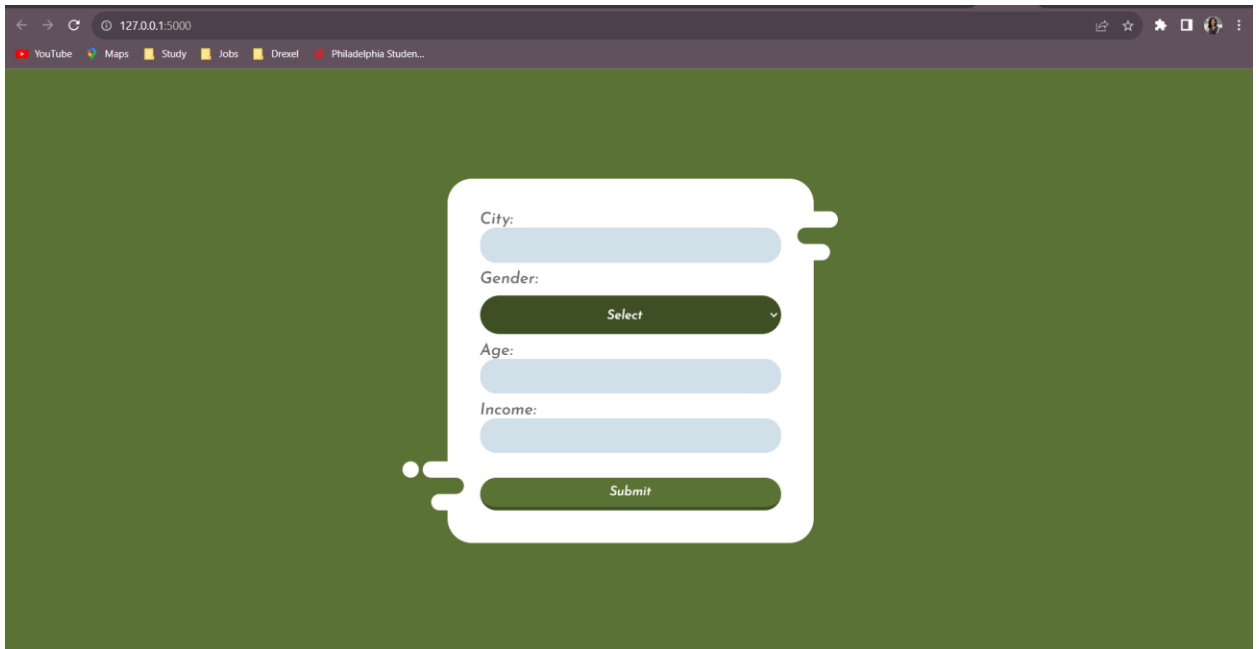
```
model.py x index.html U app.py U result.html U requestpy U
Week 4 > model.py > ...
1 # importing the libraries
2 import numpy as np
3 import pandas as pd
4 from sklearn.model_selection import train_test_split
5 from sklearn.linear_model import LinearRegression
6 from sklearn.preprocessing import OneHotEncoder
7 from sklearn.preprocessing import LabelEncoder
8 from sklearn.compose import ColumnTransformer
9 from sklearn.pipeline import Pipeline
10 from sklearn.impute import SimpleImputer
11 import pickle
12 import requests
13 import json
14
15 # Importing the dataset
16 dataset = pd.read_csv('toy_dataset.csv')
17 dataset.drop(['Number'], axis=1, inplace=True)
18 X = dataset.drop(columns=['Illness'])
19 y = dataset['Illness']
20
21 # Encode the target variable 'y'
22 label_encoder = LabelEncoder()
23 y = label_encoder.fit_transform(dataset.iloc[:, -1])
24
25 # Define categorical and numerical columns
26 categorical_columns = ['City', 'Gender']
27 numerical_columns = [col for col in X.columns if col not in categorical_columns]
28
29 # Create transformers for numerical and categorical columns
30 numerical_transformer = Pipeline(steps=[
31     ('num_imputer', SimpleImputer(strategy='mean')),
32 ])
33
34 categorical_transformer = Pipeline(steps=[
35     ('cat_imputer', SimpleImputer(strategy='most_frequent')),
36     ('onehot', OneHotEncoder(handle_unknown='ignore'))
37 ])
```

3. app.py – where I created a Flask app

```
model.py U index.html U app.py U result.html U request.py U
Week 4 > app.py > result
1 import numpy as np
2 import pandas as pd
3 from flask import Flask, request, render_template
4 import pickle
5
6 app = Flask(__name__)
7
8 # prediction function
9 def ValuePredictor(to_predict_list):
10     to_predict = np.array(to_predict_list).reshape(1, -1)
11     loaded_model = pickle.load(open("model.pkl", "rb"))
12     # Create a DataFrame for the new data point
13     new_data_df = pd.DataFrame([to_predict_list], columns=['City', 'Gender', 'Age', 'Income'])
14     result = loaded_model.predict(new_data_df)
15     return result[0]
16
17 @app.route('/')
18 def home():
19     return render_template("index.html")
20
21 @app.route('/result', methods=['POST'])
22 def result():
23     if request.method == 'POST':
24         to_predict_list = request.form.to_dict()
25         to_predict_list = list(to_predict_list.values())
26         result = ValuePredictor(to_predict_list)
27         if int(result) == 1:
28             prediction = 'Person is sick'
29         else:
30             prediction = 'Person is not sick'
31         return render_template("result.html", prediction = prediction)
32
33 if __name__ == '__main__':
34     app.run(port=5000, debug=True)
```

4. index.html – this is my home page where are asking for inputs

```
model.py U index.html U app.py U result.html U
Week 4 > templates > index.html > html > body > form > div.form-inner > select#gender
143 width: 60%;
144 }
145 }
146 </style>
147 </head>
148 <body>
149     <form action="/result" method="POST">
150         <div class="form-left-decoration"></div>
151         <div class="form-right-decoration"></div>
152         <div class="circle"></div>
153         <div class="form-inner">
154             <label for="city">City:</label>
155             <input type="text" id="city" name="city" required>
156
157             <label for="gender">Gender:</label>
158             <br/>
159             <select id="gender" name="gender" required style="margin: 10px 0;">
160                 <option value="0">Select</option>
161                 <option value="1">Female</option>
162                 <option value="2">Male</option>
163             </select>
164             <br/>
165             <label for="age">Age:</label>
166             <input type="number" id="age" name="age" required>
167
168             <label for="income">Income:</label>
169             <input type="number" id="income" name="income" required>
170
171             <button type="submit" value="Submit">Submit</button>
172         </div>
173     </form>
174 </body>
175 </html>
```



5. result.html – this is my result page to display the output

```
model.py U index.html U app.py U result.html U X
Week 4 > templates > result.html > html
126 background-color: #3e4f24;
127 color: white;
128 border-radius: 50px;
129 padding: 5px 10px;
130 border: none;
131 font-size: 16px;
132 font-family: 'Josefin Sans', sans-serif;
133 /* border-bottom: 4px solid #3e4f24; */
134 }
135 select{
136     margin: 10px 0;
137     width: 100%;
138     text-align: center;
139     padding: 15px;
140 }
141 @media (max-width: 568px) {
142     form {
143         width: 60%;
144     }
145 }
146 </style>
147 </head>
148 <body>
149     <div class="form-left-decoration"></div>
150     <div class="form-right-decoration"></div>
151     <div class="circle"></div>
152     <div class="form-inner">
153         <h1 style="color: black;">{{ prediction }}</h1>
154     </div>
155 </body>
156 </html>
```

