

Name: Prajesh Tejani

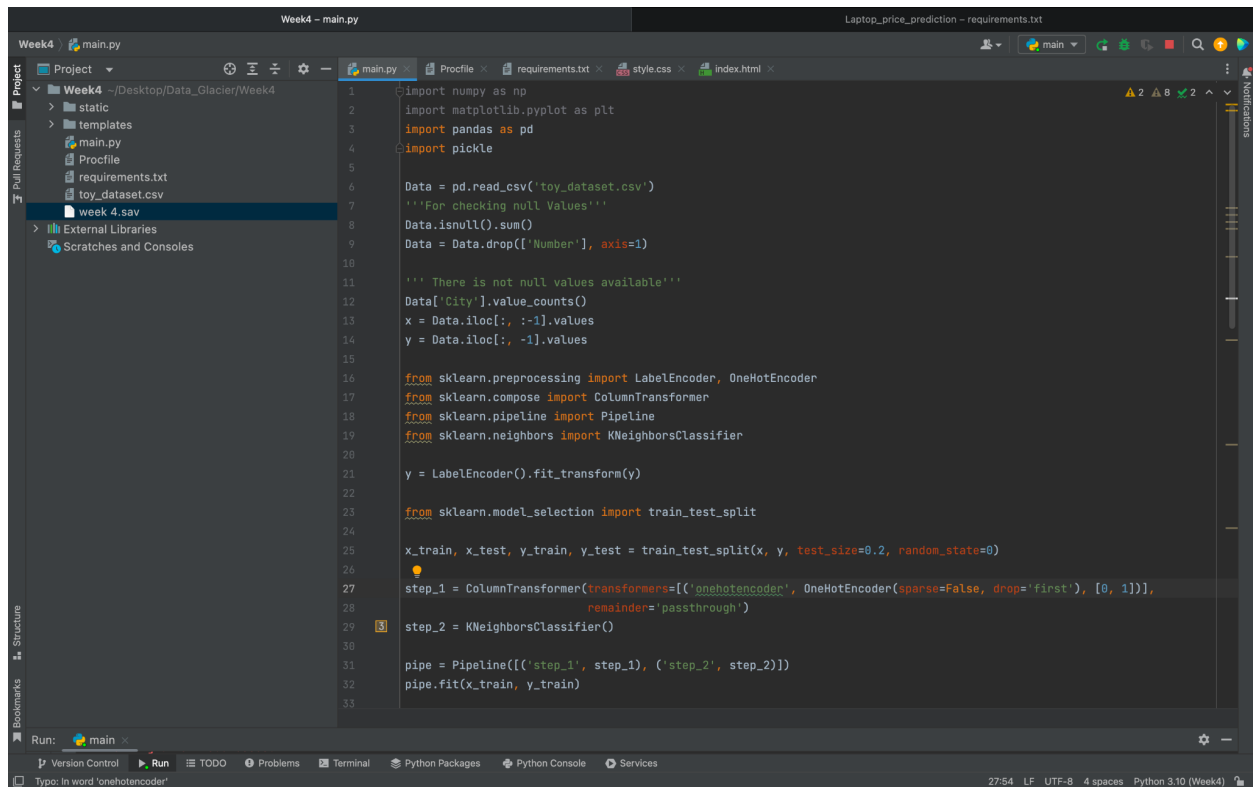
Batch Code: LISUM10: 30

Submission date : 22nd June 2022

Submitted to:

Snapshots of Deployment:

1st: Create Model



```
Week4 - main.py
Project
Week4 - ~/Desktop/Data_Glacier/Week4
static
templates
main.py
Procfie
requirements.txt
toy_dataset.csv
week 4.sav
External Libraries
Scratches and Consoles

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4 import pickle
5
6 Data = pd.read_csv('toy_dataset.csv')
7 '''For checking null Values'''
8 Data.isnull().sum()
9 Data = Data.drop(['Number'], axis=1)
10
11 ''' There is not null values available'''
12 Data['City'].value_counts()
13 x = Data.iloc[:, :-1].values
14 y = Data.iloc[:, -1].values
15
16 from sklearn.preprocessing import LabelEncoder, OneHotEncoder
17 from sklearn.compose import ColumnTransformer
18 from sklearn.pipeline import Pipeline
19 from sklearn.neighbors import KNeighborsClassifier
20
21 y = LabelEncoder().fit_transform(y)
22
23 from sklearn.model_selection import train_test_split
24
25 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
26
27 step_1 = ColumnTransformer(transformers=[('onehotencoder', OneHotEncoder(sparse=False, drop='first'), [0, 1])],
28                               remainder='passthrough')
29 step_2 = KNeighborsClassifier()
30
31 pipe = Pipeline([('step_1', step_1), ('step_2', step_2)])
32 pipe.fit(x_train, y_train)
33
```

```
# confusion_metrics
from sklearn.metrics import confusion_matrix, accuracy_score

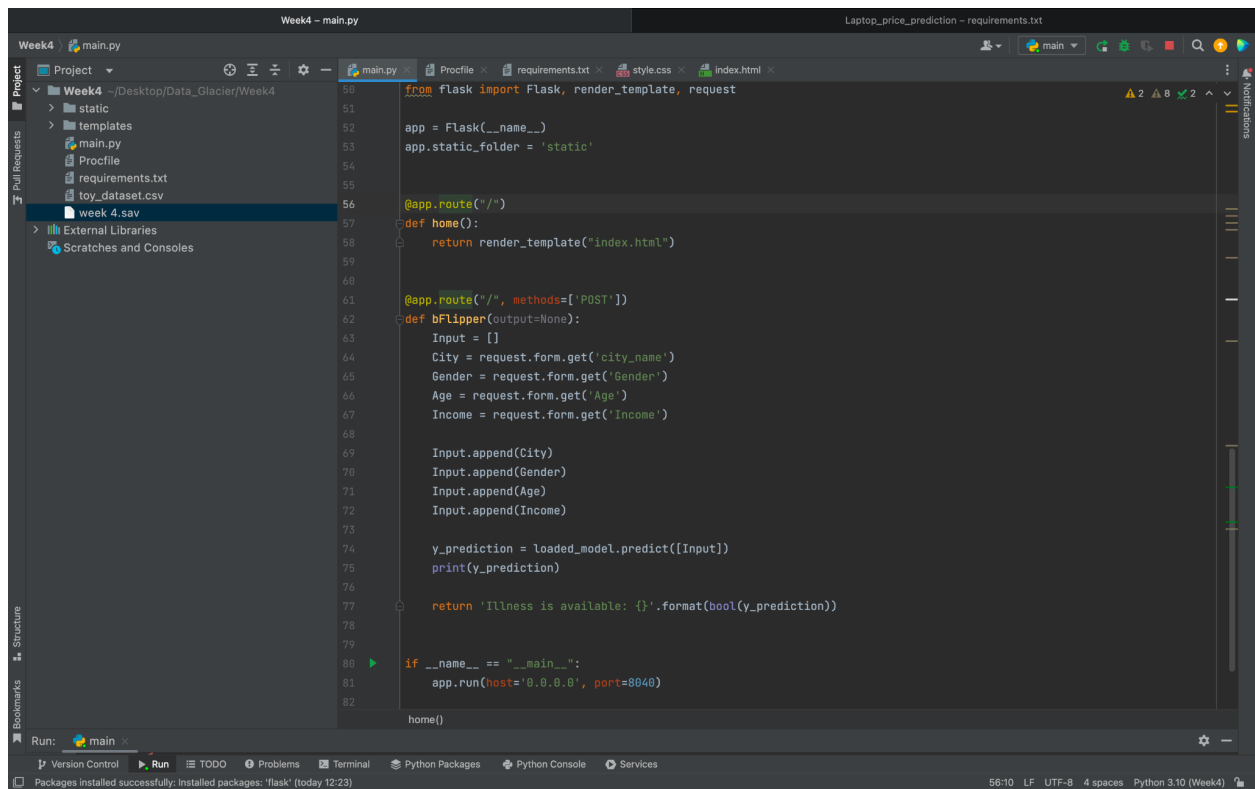
y_pred = pipe.predict(x_test)
cm = confusion_matrix(y_test, y_pred)
print('Accuracy_score =', accuracy_score(y_test, y_pred))
```

2nd: Save the model

```
# save the model to disk
filename = 'week 4.sav'
pickle.dump(pipe, open(filename, 'wb'))

# load the model from disk
loaded_model = pickle.load(open(filename, 'rb'))
result = loaded_model.score(x_test, y_test)
print(result)
```

3rd: Deploy on local Host



The screenshot shows a code editor with a project named 'Week4' and a file named 'main.py'. The code is a Flask application that loads a pre-trained model and provides a web interface for predictions. The application has two routes: a home page and a prediction endpoint. The prediction endpoint takes city, gender, age, and income as input and returns a prediction result.

```
from flask import Flask, render_template, request

app = Flask(__name__)
app.static_folder = 'static'

@app.route("/")
def home():
    return render_template("index.html")

@app.route("/", methods=['POST'])
def bFlipper(output=None):
    Input = []
    City = request.form.get('city_name')
    Gender = request.form.get('Gender')
    Age = request.form.get('Age')
    Income = request.form.get('Income')

    Input.append(City)
    Input.append(Gender)
    Input.append(Age)
    Input.append(Income)

    y_prediction = loaded_model.predict([Input])
    print(y_prediction)

    return 'Illness is available: {}'.format(bool(y_prediction))

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=8040)
```

4th: Create form and give it style for input...(templates/index.html)

```
Week4 - main.py
Laptop_price_prediction - requirements.txt

Project
  Week4
    static
    templates
    main.py
    Profile
    requirements.txt
    toy_dataset.csv
    week 4.sav
  External Libraries
  Scratches and Consoles

Run: main
Version Control Run TODO Problems Terminal Python Packages Python Console Services
Packages installed successfully: installed packages: 'flask' (today 12:23) 56:10 LF UTF-8 4 spaces Python 3.10 (Week4)
```

```
50 from flask import Flask, render_template, request
51
52 app = Flask(__name__)
53 app.static_folder = 'static'
54
55
56 @app.route("/")
57 def home():
58     return render_template("index.html")
59
60
61 @app.route("/", methods=['POST'])
62 def bFlipper(output=None):
63     Input = []
64     City = request.form.get('city_name')
65     Gender = request.form.get('Gender')
66     Age = request.form.get('Age')
67     Income = request.form.get('Income')
68
69     Input.append(City)
70     Input.append(Gender)
71     Input.append(Age)
72     Input.append(Income)
73
74     y_prediction = loaded_model.predict([Input])
75     print(y_prediction)
76
77     return 'Illness is available: {}'.format(bool(y_prediction))
78
79
80 if __name__ == "__main__":
81     app.run(host='0.0.0.0', port=8040)
82
83 home()
```

5th: static/style.css

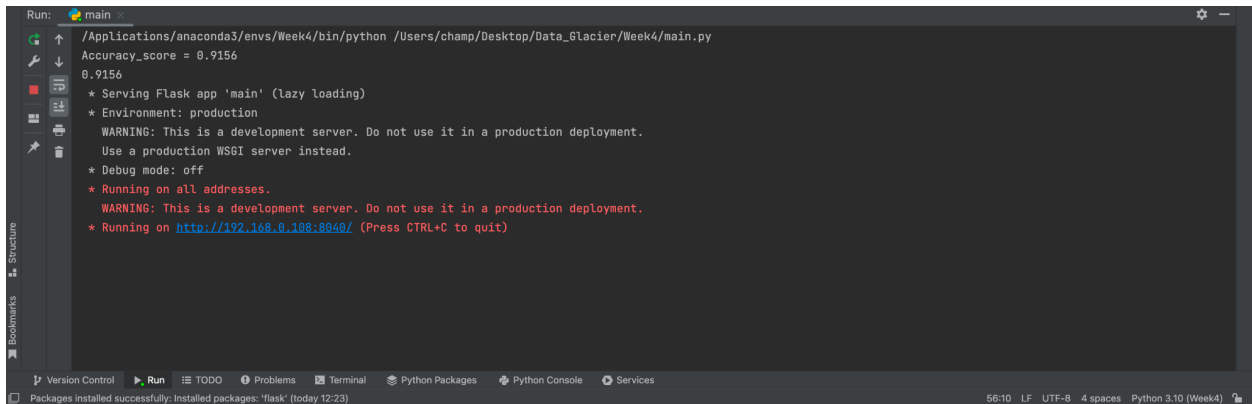
```
Week4 - style.css
Laptop_price_prediction - requirements.txt

Project
  Week4
    static
    templates
    index.html
    main.py
    Profile
    requirements.txt
    toy_dataset.csv
    week 4.sav
  External Libraries
  Scratches and Consoles

Run: main
Version Control Run TODO Problems Terminal Python Packages Python Console Services
Packages installed successfully: installed packages: 'flask' (today 12:23) 17:11 LF UTF-8 4 spaces Python 3.10 (Week4)
```

```
1 form-form-example {
2     display: table;
3 }
4
5 div.form-example {
6     display: table-row;
7 }
8
9 label, input {
10    display: table-cell;
11    margin-bottom: 10px;
12 }
13
14 label {
15     padding-right: 10px;
16 }
17 |
```

Output on Flask:

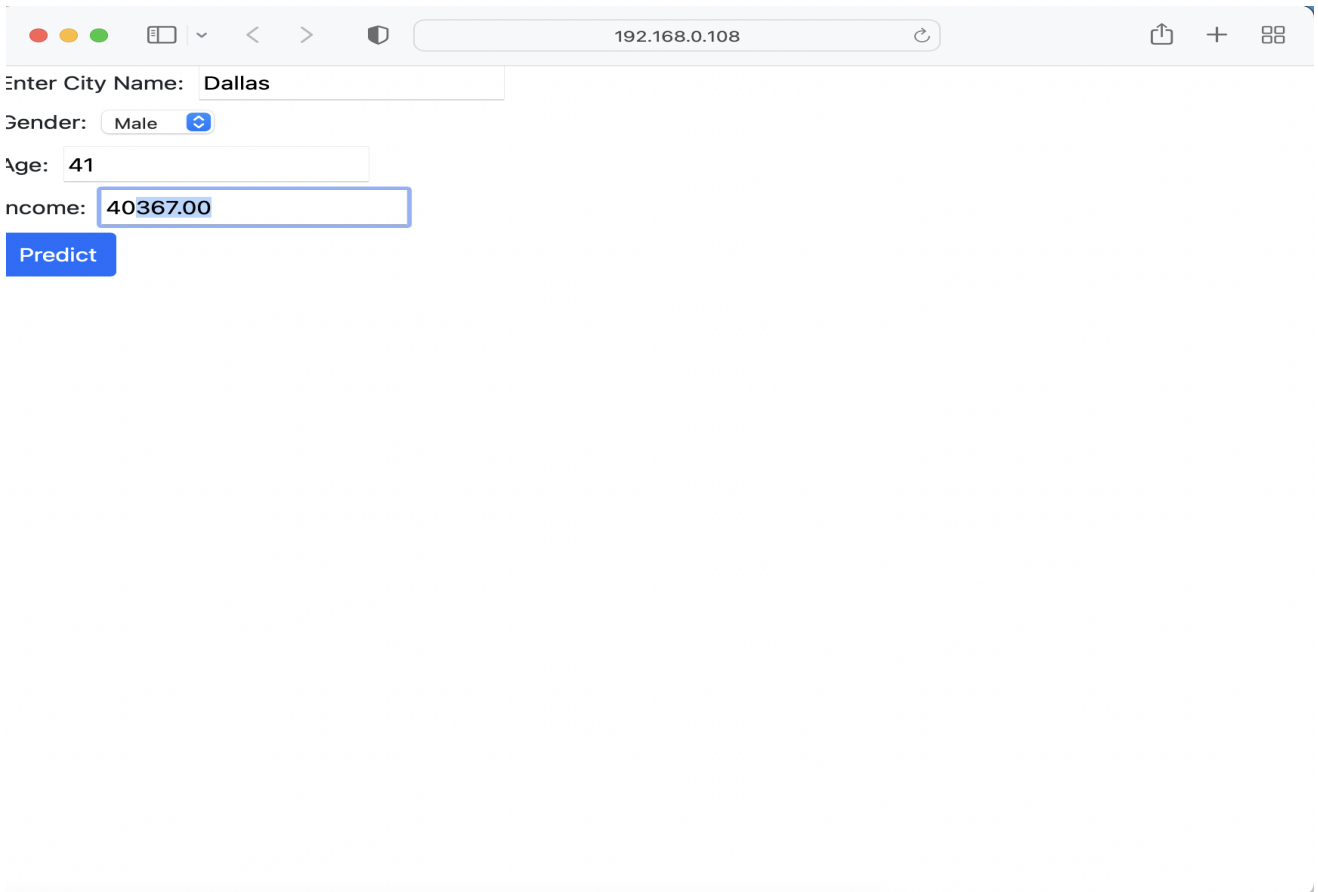


The screenshot shows a terminal window with the following output:

```
Run: main
/Users/champ/Desktop/Data_Glacier/Week4/main.py
Accuracy_score = 0.9156
0.9156
* 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: off
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://192.168.0.108:8949/ (Press CTRL+C to quit)
```

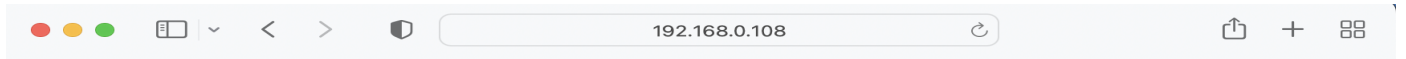
The terminal window includes a sidebar with 'Structure' and 'Bookmarks' tabs, and a bottom status bar showing 'Packages installed successfully: Installed packages: 'flask' (today 12:23)'.

Output on Local host:



The screenshot shows a web browser window with the URL `192.168.0.108`. The form contains the following elements:

- Enter City Name:
- Gender: (dropdown menu)
- Age:
- Income:
-



Illness is available: False