Week 4 Deployment on Flask

Name: Bindu Musham

Batch code: LISUM20

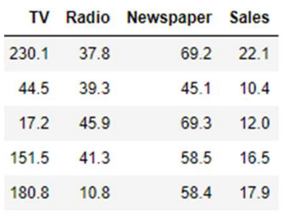
Submission date: 28th April 2023

Submission to: Data Glacier

Data:

Dataset is taken from Kaggle https://www.kaggle.com/code/ashydv/sales-prediction-simple-linear-regression/input

The dataset has sales data based on the money spent on 3 different modes of marketing, TV, Radio, and Newspaper.



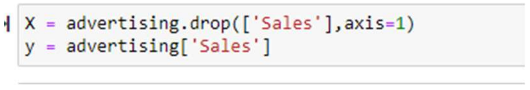
TV	Radio	Newspaper	Sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	12.0
151.5	41.3	58.5	16.5
180.8	10.8	58.4	17.9

Problem Statement:

Build a model to predict sales on the money spent on these modes of advertising predictor variables.

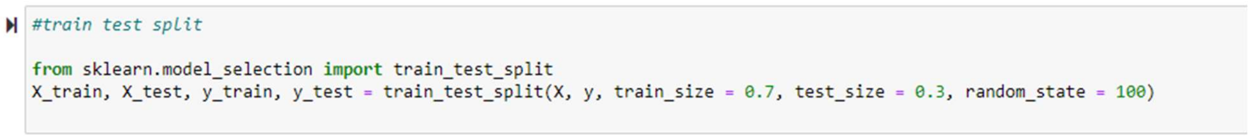
Build a model:

i) Create dependent (y) and independent variables (X)



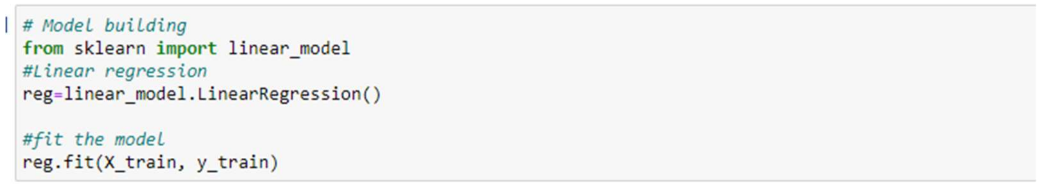
```
X = advertising.drop(['Sales'],axis=1)
y = advertising['Sales']
```

ii) The train test split with 70% train set and 30% test set.



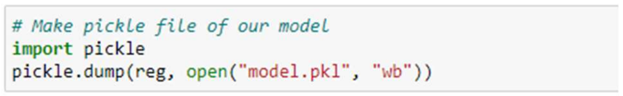
```
#train test split

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.7, test_size = 0.3, random_state = 100)
```

iii) Perform linear regression and fit the train set to predict sales based on advertising costs.



```
# Model building
from sklearn import linear_model
#Linear regression
reg=linear_model.LinearRegression()

#fit the model
reg.fit(X_train, y_train)
```

iv) Saved the model using pickle.



```
# Make pickle file of our model
import pickle
pickle.dump(reg, open("model.pkl", "wb"))
```

Creating a Flask web application:

1. Create a folder named **Week 4 deployment on flask** for this project.

   app.py

   templates/index.html

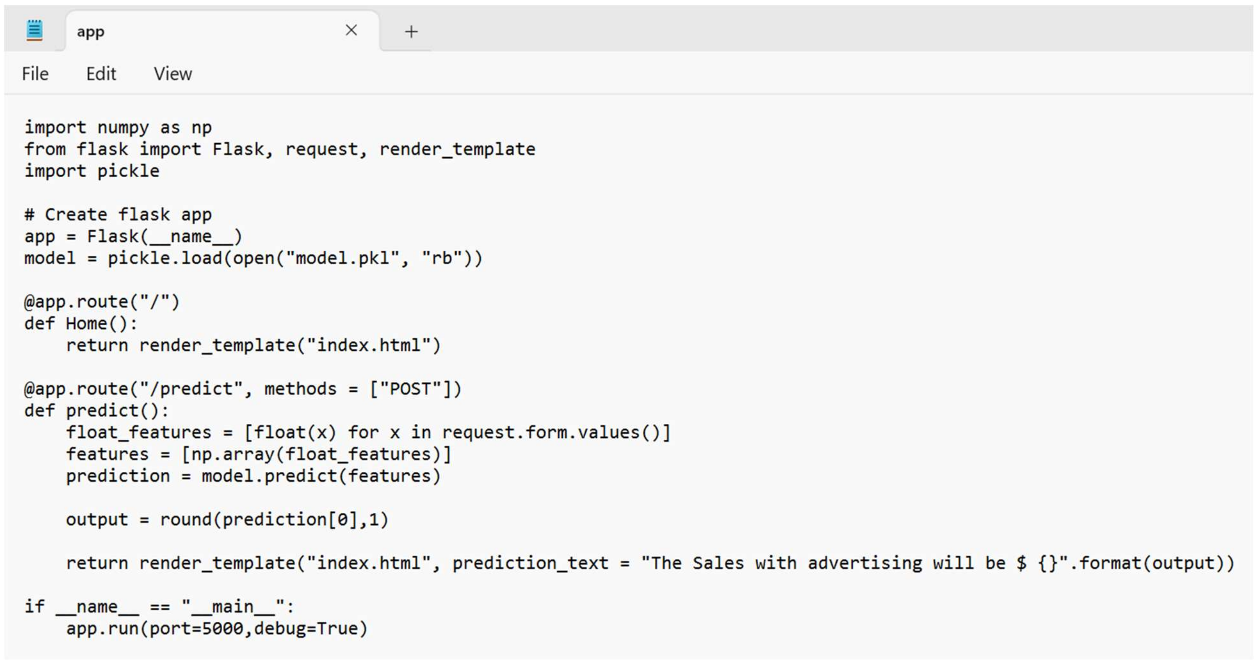
   static/style.css

   advertising.csv

   Flask_sales_prediction.ipynb

2. When we ran above python jupyter notebook automatically the model.pkl file is created using pickle.

3. Create a main app.py file which has the main code with model file.



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

# Create flask app
app = Flask(__name__)
model = pickle.load(open("model.pkl", "rb"))

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

@app.route("/predict", methods = ["POST"])
def predict():
    float_features = [float(x) for x in request.form.values()]
    features = [np.array(float_features)]
    prediction = model.predict(features)

    output = round(prediction[0],1)

    return render_template("index.html", prediction_text = "The Sales with advertising will be $ {}".format(output))

if __name__ == "__main__":
    app.run(port=5000,debug=True)
```

Steps in app.py:

i) Flask instance is created with a unique name **app.**

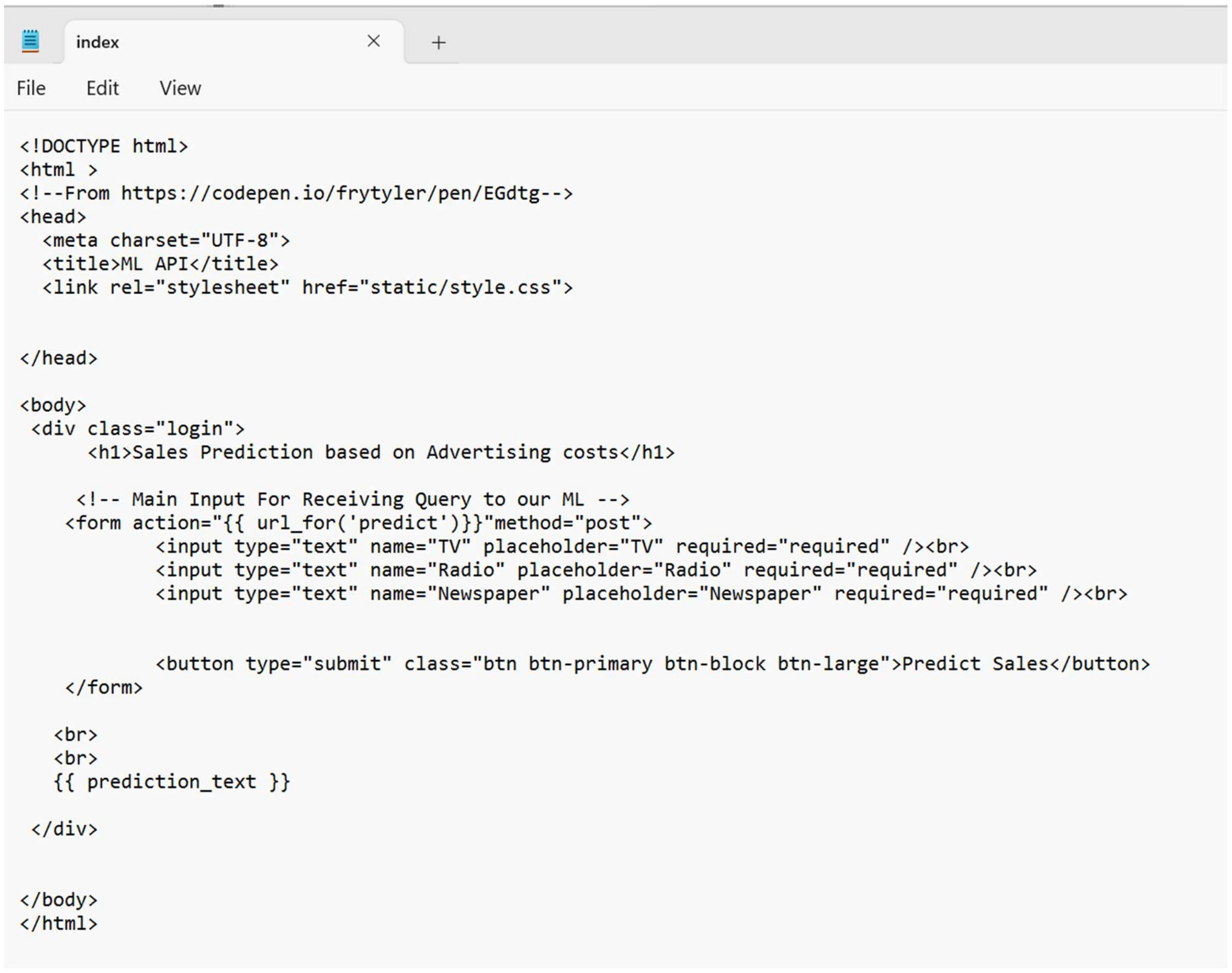
ii) Load the **model.pkl** file.

iii) The route path provides the home page details. Whenever anyone hits the server. It will display the **index.html** page.

iv) In the index.html code, it takes the advertising costs for TV, Radio and Newspaper as inputs and predicts the sales value.

v) The index.html is directly taken templates folder.

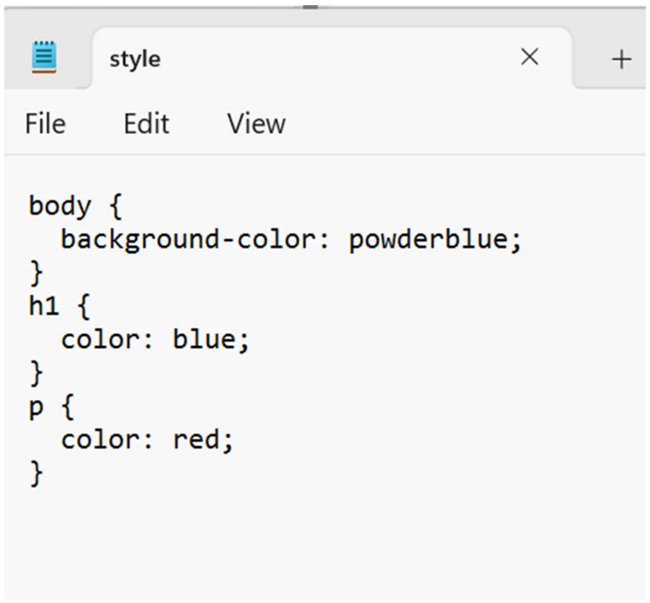
Index.html:



```
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link rel="stylesheet" href="static/style.css">


</head>

<body>
 <div class="login">
      <h1>Sales Prediction based on Advertising costs</h1>

     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
            <input type="text" name="TV" placeholder="TV" required="required" /><br>
            <input type="text" name="Radio" placeholder="Radio" required="required" /><br>
            <input type="text" name="Newspaper" placeholder="Newspaper" required="required" /><br>


            <button type="submit" class="btn btn-primary btn-block btn-large">Predict Sales</button>
    </form>

   <br>
   <br>
   {{ prediction_text }}

 </div>


</body>
</html>
```

Style.css:

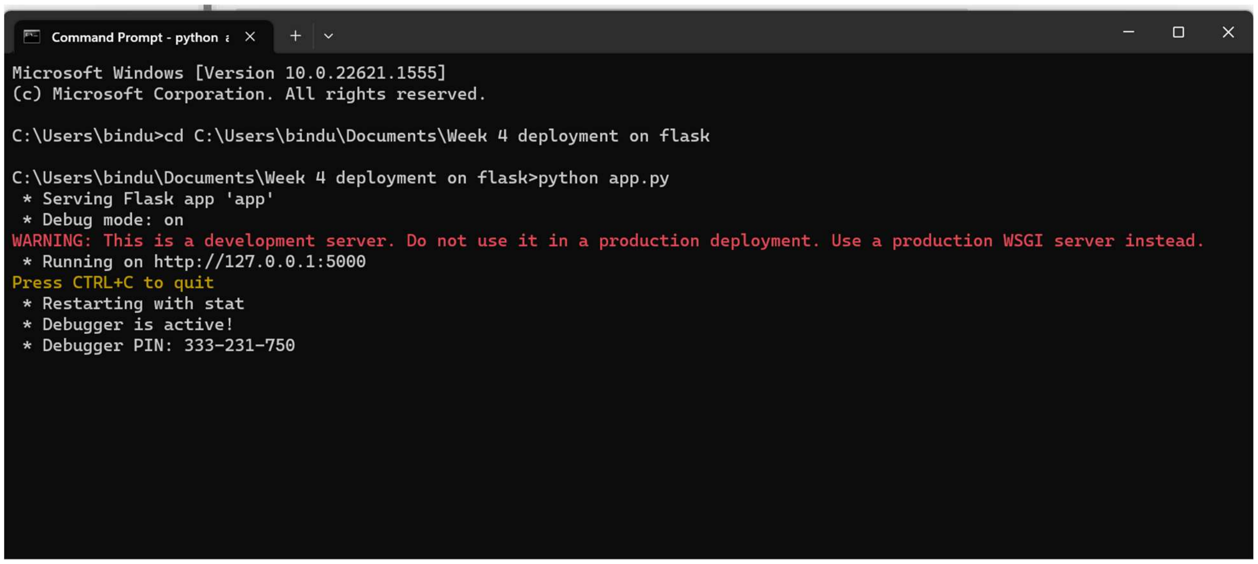
Created a style.css for colorful homepage.



```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

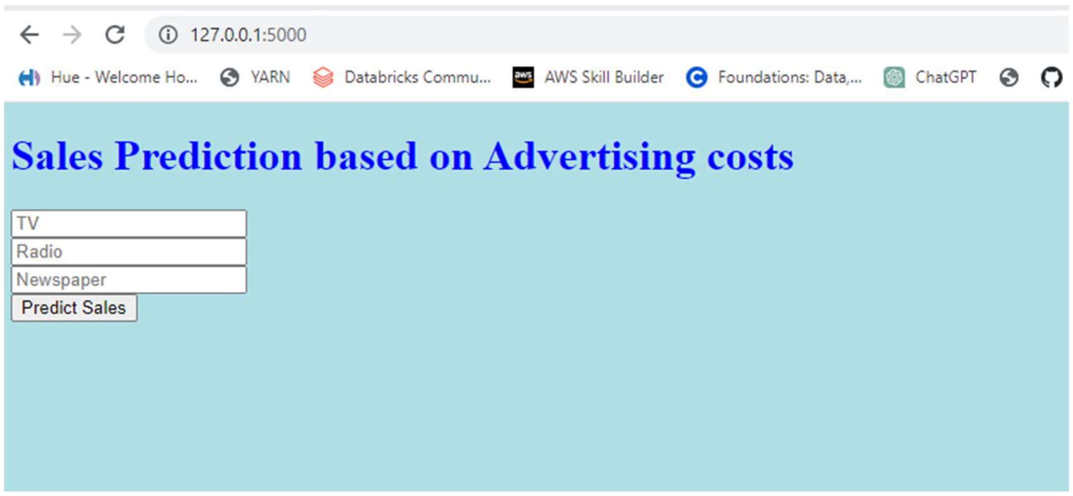
Launch web application:

i) Run app.py as below.



```
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bindu>cd C:\Users\bindu\Documents\Week 4 deployment on flask

C:\Users\bindu\Documents\Week 4 deployment on flask>python 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 instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 333-231-750
```

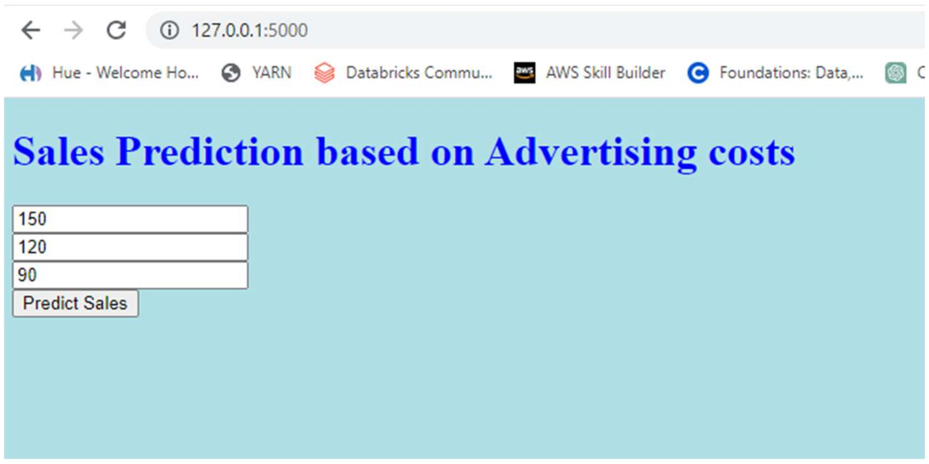
ii) Open the URL http://127.0.0.1:5000 in browser.



Sales Prediction based on Advertising costs

TV
Radio
Newspaper
Predict Sales

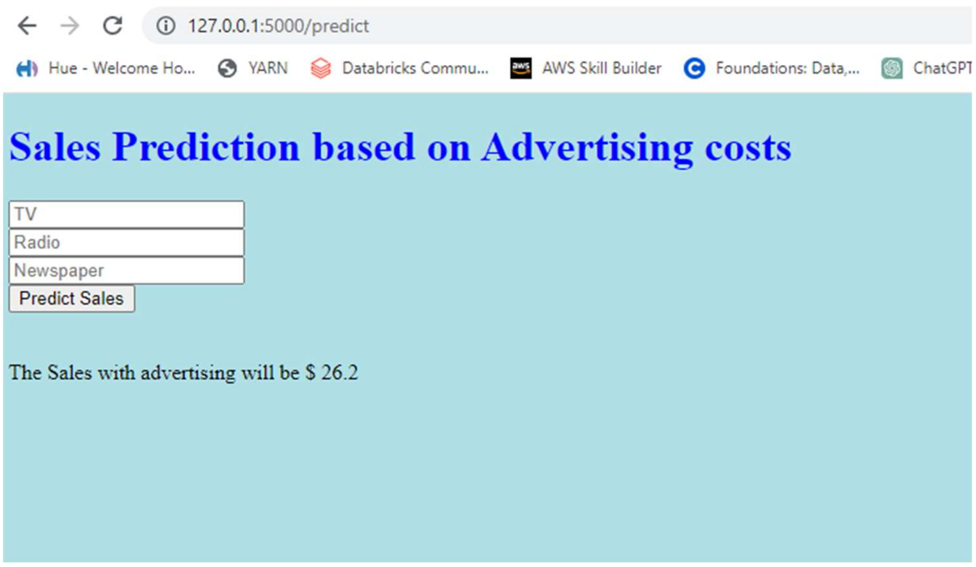
iii) Give the input advertising costs for TV, Radio, and Newspaper.



Sales Prediction based on Advertising costs

150
120
90
Predict Sales

iv) Click on predict sales to get sales based on above costs.



Sales Prediction based on Advertising costs

The Sales with advertising will be $ 26.2