

Week 4

Flask Deployment

Uday Vikram Singh

- Name: Uday Vikram Singh
- Batch code: LISUM17
- Submission date: 28/01/2023
- Submitted to: GitHub link submitted on dashboard

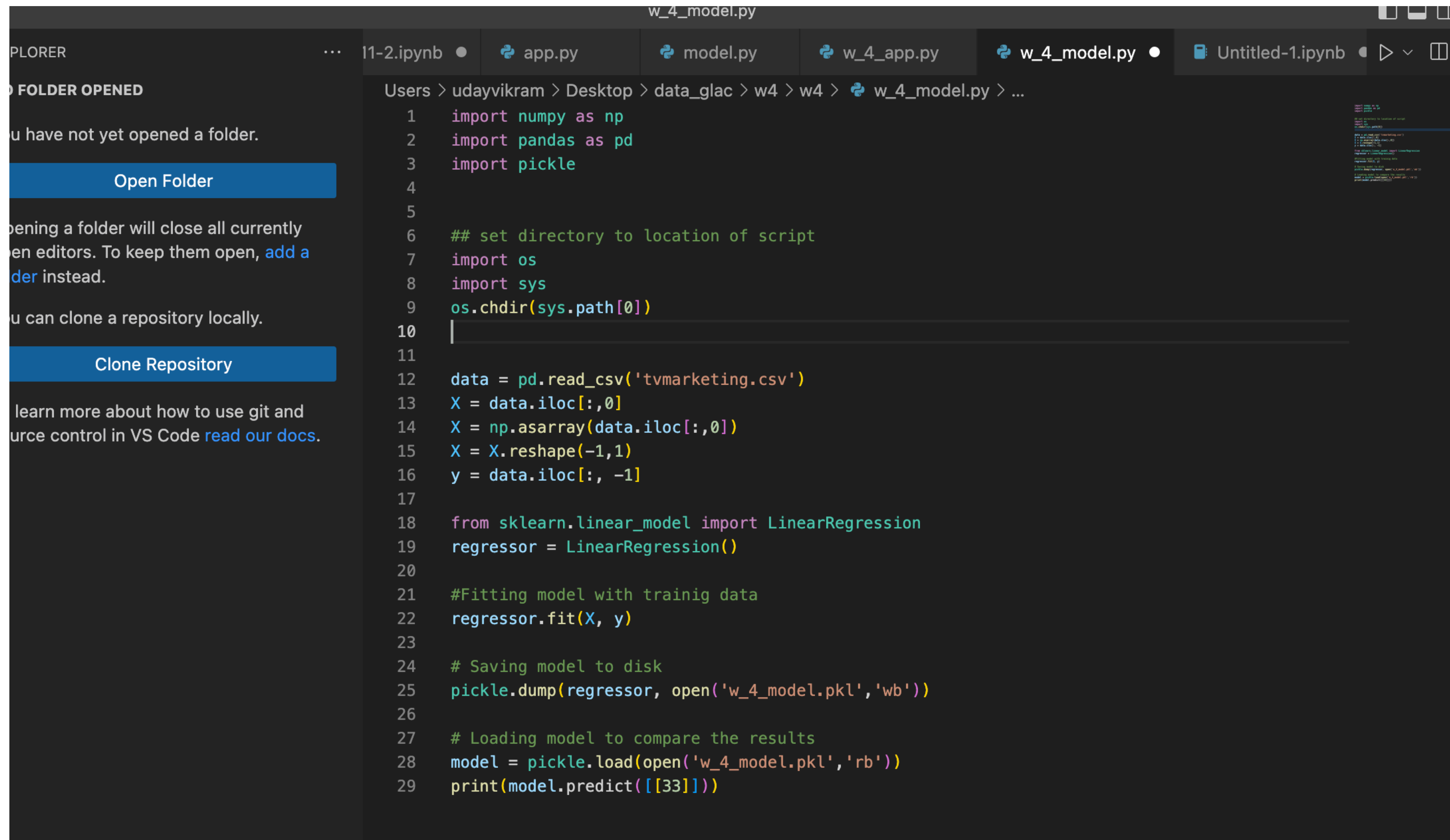
Notes: git repo has this document containing screenshots of deployment. It also contains the .py scripts and templates needed to deploy the model.

The model and app code is based on example provided in Flask Code section of Resources in dashboard.

Data

- Name: tvmarketing.csv
- Columns: TV (marketing budget), Sales
- Rows: 200
- Source: <https://www.kaggle.com/datasets/devzohaib/tvmarketingcsv>
- About: Data has only one independent variable (X). This makes it suitable for a toy model.

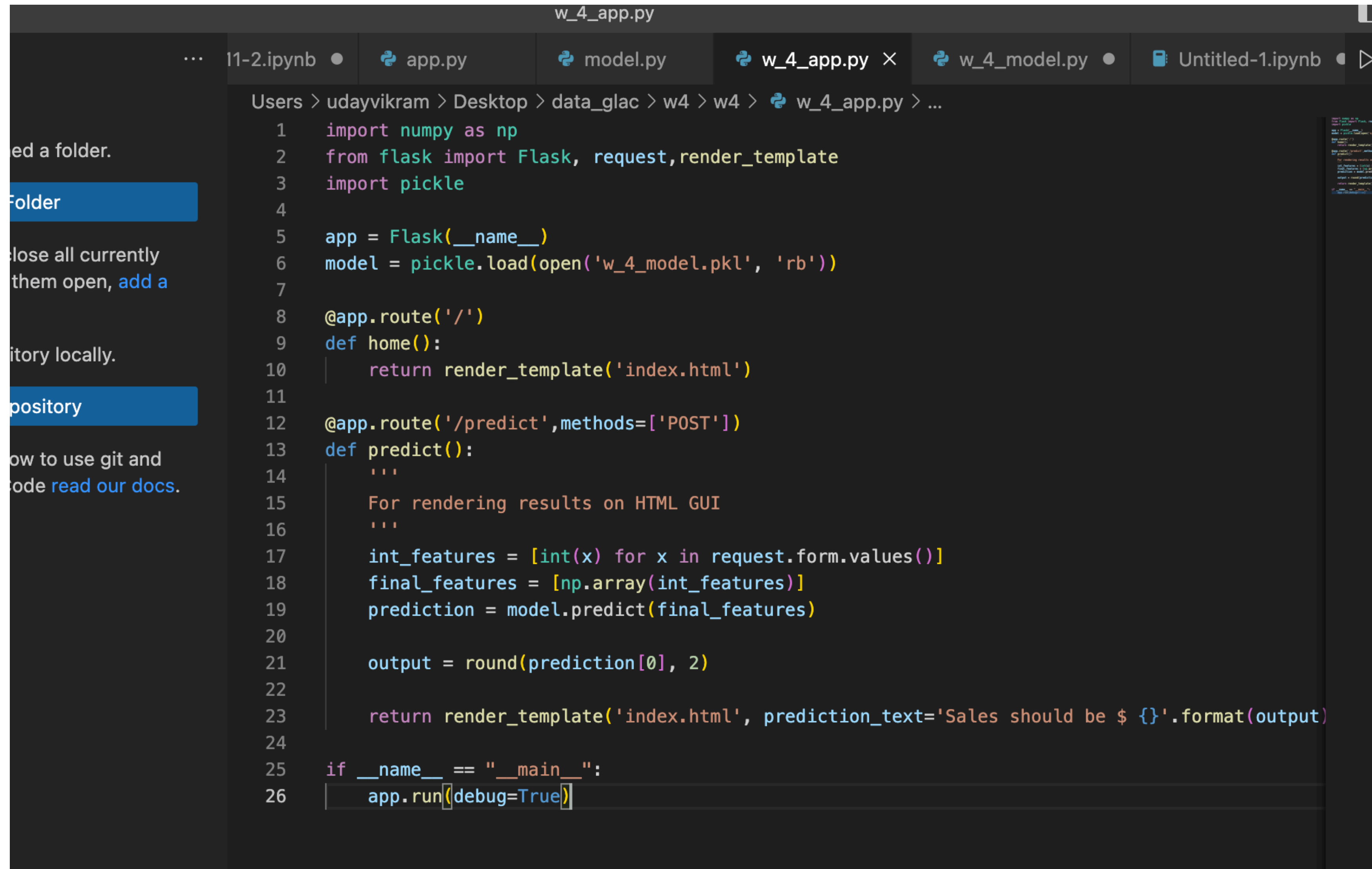
Load data and Build Model



The image shows a VS Code editor window with a dark theme. The Explorer sidebar on the left shows a file named 'w_4_model.py' and a folder named 'w_4_model.py'. The main editor area displays the following Python code:

```
w_4_model.py
1 import numpy as np
2 import pandas as pd
3 import pickle
4
5
6 ## set directory to location of script
7 import os
8 import sys
9 os.chdir(sys.path[0])
10
11
12 data = pd.read_csv('tvmarketing.csv')
13 X = data.iloc[:,0]
14 X = np.asarray(data.iloc[:,0])
15 X = X.reshape(-1,1)
16 y = data.iloc[:, -1]
17
18 from sklearn.linear_model import LinearRegression
19 regressor = LinearRegression()
20
21 #Fitting model with trainig data
22 regressor.fit(X, y)
23
24 # Saving model to disk
25 pickle.dump(regressor, open('w_4_model.pkl','wb'))
26
27 # Loading model to compare the results
28 model = pickle.load(open('w_4_model.pkl','rb'))
29 print(model.predict([[33]]))
```

Write app for deployment

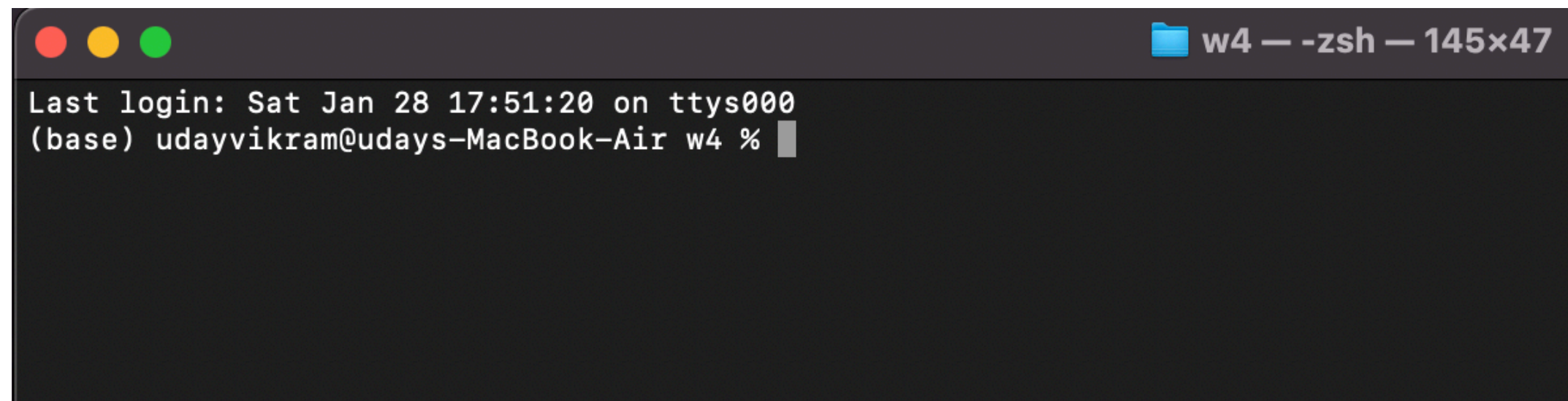


```
w_4_app.py
... 11-2.ipynb • app.py model.py w_4_app.py × w_4_model.py • Untitled-1.ipynb ▶
Users > udayvikram > Desktop > data_glac > w4 > w4 > w_4_app.py > ...
1 import numpy as np
2 from flask import Flask, request, render_template
3 import pickle
4
5 app = Flask(__name__)
6 model = pickle.load(open('w_4_model.pkl', 'rb'))
7
8 @app.route('/')
9 def home():
10     return render_template('index.html')
11
12 @app.route('/predict', methods=['POST'])
13 def predict():
14     '''
15     For rendering results on HTML GUI
16     '''
17     int_features = [int(x) for x in request.form.values()]
18     final_features = [np.array(int_features)]
19     prediction = model.predict(final_features)
20
21     output = round(prediction[0], 2)
22
23     return render_template('index.html', prediction_text='Sales should be $ {}'.format(output))
24
25 if __name__ == "__main__":
26     app.run(debug=True)
```

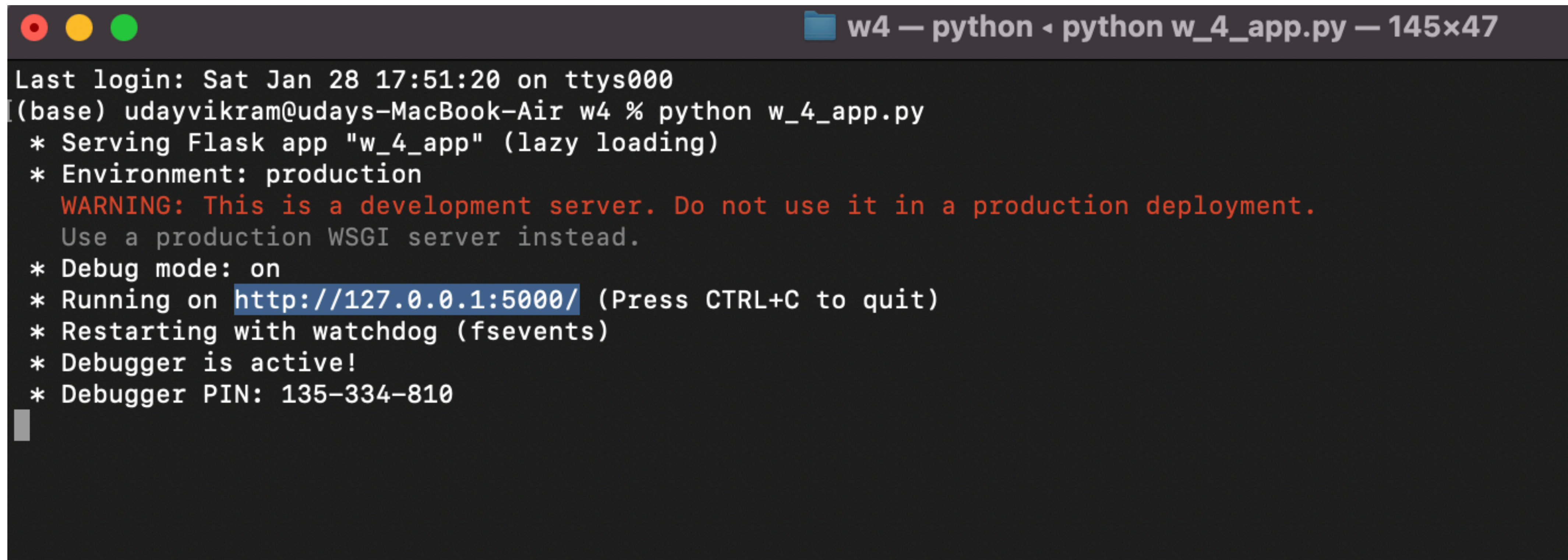
Edit HTML to ask correct prompt (based on dataset)

```
index.html
... 11-2.ipynb ●  app.py  model.py  w_4_app.py  <> index.html ×  w_4_model.py ●  [icon]
Users > udayvikram > Desktop > data_glac > w4 > w4 > templates > <> index.html > [icon] html > [icon] body > [icon] div.login > [icon]
1  <!DOCTYPE html>
2  <html >
3  <head>
4      <meta charset="UTF-8">
5      <title>ML API</title>
6      <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
7      <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
8      <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
9      <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
10     <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
11
12 </head>
13
14 <body>
15     <div class="login">
16         <h1>Predict Sales Based on Marketing Budget</h1>
17
18         <!-- Main Input For Receiving Query to our ML -->
19         <form action="{{ url_for('predict') }}" method="post">
20
21             <input type="text" name="TV" placeholder="Marketing Budget" required="required" />
22
23             <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
24         </form>
25
26         <br>
27         <br>
28         {{ prediction_text }}
29
30     </div>
31     
32
33 </body>
34 </html>
35
```


Open directory in command line and run the app

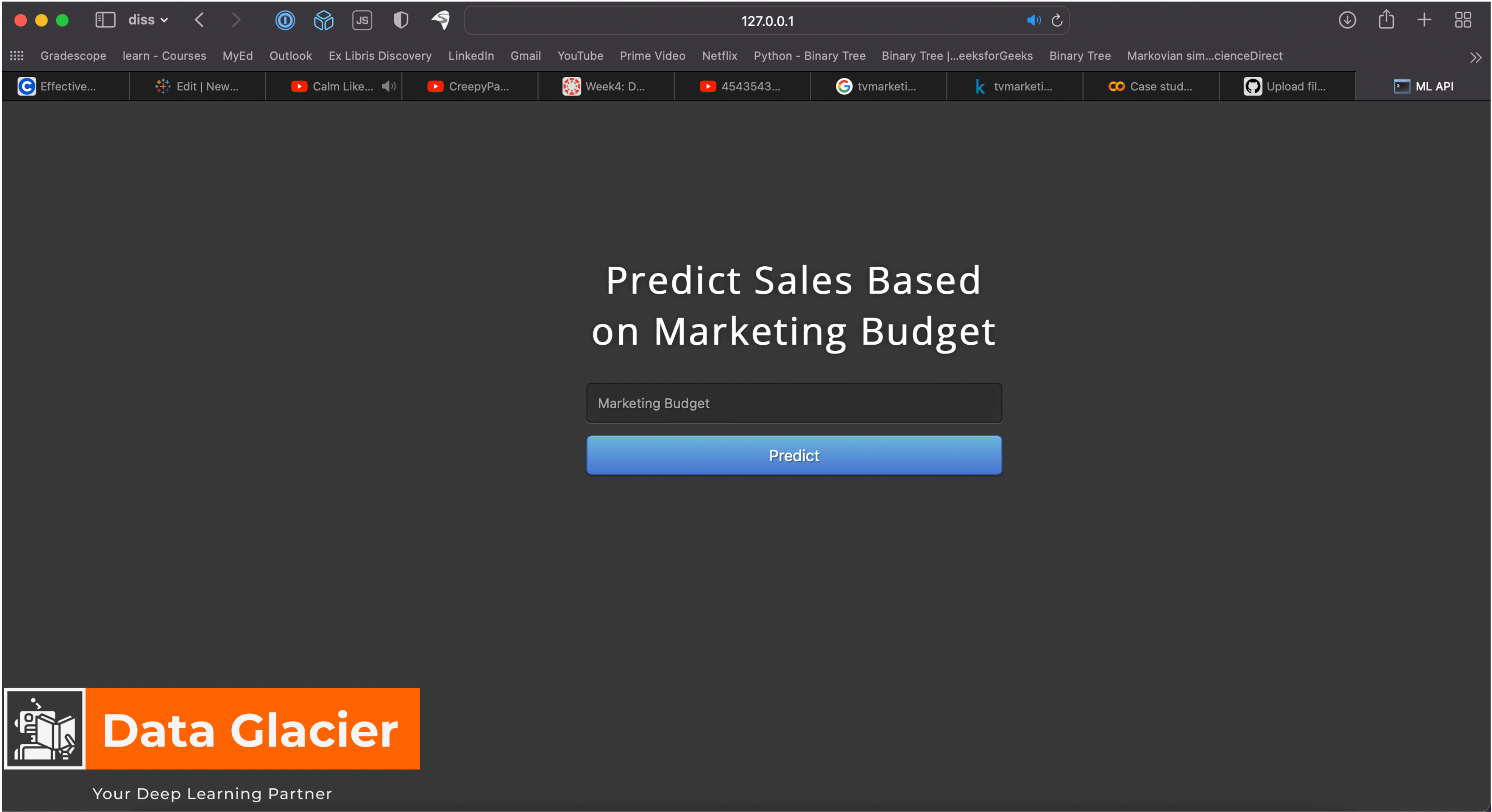


```
w4 — -zsh — 145x47
Last login: Sat Jan 28 17:51:20 on ttys000
(base) udayvikram@udays-MacBook-Air w4 %
```



```
w4 — python ◀ python w_4_app.py — 145x47
Last login: Sat Jan 28 17:51:20 on ttys000
(base) udayvikram@udays-MacBook-Air w4 % python w_4_app.py
* Serving Flask app "w_4_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
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with watchdog (fsevents)
* Debugger is active!
* Debugger PIN: 135-334-810
```


Open the URL where app is hosted



Try random value to test prediction

The image is a screenshot of a web browser window. The browser's address bar shows the URL '127.0.0.1'. The top of the browser window displays various icons and a search bar. Below the browser window, there is a dark gray application interface. The interface has a title 'Predict Sales Based on Marketing Budget' in white text. Below the title, there is a text input field labeled 'Marketing Budget'. Underneath the input field is a blue button with the text 'Predict'. Below the button, the text 'Sales should be \$ 8.65' is displayed. At the bottom left of the image, there is an orange banner with a white icon of a robot reading a book and the text 'Data Glacier' in white. Below the banner, the text 'Your Deep Learning Partner' is visible.

Try another random value

diss

127.0.0.1

Gradescope

learn - Courses

MyEd

Outlook

Ex Libris Discovery

LinkedIn

Gmail

YouTube

Prime Video

Netflix

Python - Binary Tree

Binary Tree [...eeksforGeeks

Binary Tree

Markovian sim...cienceDirect

Effective...

Edit | New...

Calm Like...

CreepyPa...

Week4: D...

4543543...

tvmarketi...

tvmarketi...

Case stud...

Upload fil...


Predict Sales Based
on Marketing Budget

45|

45

Predict

Sales should be \$ 9.17



Data Glacier

Your Deep Learning Partner