

## Week 4 – Model Deployment Flask

Name - Vishnu Priya Malchetti

Batch code - LISUM41: 30 Dec 2024 - 30 March 2025

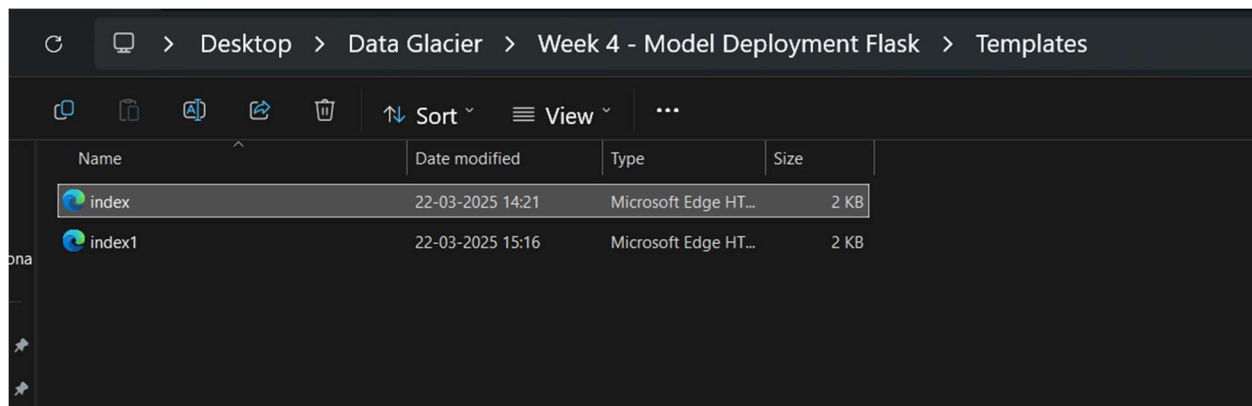
Submission date - 22/03/2025

Submitted to – Data Glacier

Install Spyder and create a python script to deploy the model

```
deploy_flask_app.py ×
1  # -*- coding: utf-8 -*-
2  """
3  Created on Mon Mar  3 21:51:29 2025
4
5  @author: Avinash
6  """
7
8  import numpy as np
9  from flask import Flask, request, render_template
10 import pickle
11
12 app = Flask(__name__)
13 model = pickle.load(open('toy_model.pkl', 'rb'))
14
15 @app.route('/')
16 def home():
17     return render_template('index.html')
18
19 @app.route('/predict', methods=['POST'])
20 def predict():
21     '''
22     For rendering results on HTML GUI
23     '''
24     int_features = [int(x) for x in request.form.values()]
25     final_features = [np.array(int_features)]
26     prediction = model.predict(final_features)
27     output = round(prediction[0], 2)
28     return render_template('index.html', prediction_text='Toy Popularity Score is {}'.format(ou
29
30 if __name__ == "__main__":
31     app.run(port=5000, debug=True)
32
```

Create required html files for the frontend in the Templates folder.



Using the Toy price and Advertising spend, will be determining the popularity score for the toy based on the input features.

```
index.html
File Edit View

<!DOCTYPE html>
<html>
<!-- From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href="https://fonts.googleapis.com/css?family=Pacifico" rel="stylesheet" type="text/css">
  <link href="https://fonts.googleapis.com/css?family=Arimo" rel="stylesheet" type="text/css">
  <link href="https://fonts.googleapis.com/css?family=Hind:300" rel="stylesheet" type="text/css">
  <link href="https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300" rel="stylesheet" type="text/css">
  <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
</head>

<body>
  <div class="login">
    <h1>Toy Popularity Prediction</h1>

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

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

    <br>
    <br>
    {{ prediction_text }}
  </div>

</body>
</html>
```

Creating pickle file

```
[ ]: Importing required libraries

[3]: import pandas as pd
    from sklearn.model_selection import train_test_split
    from sklearn.linear_model import LinearRegression
    from sklearn.metrics import mean_squared_error
    import pickle

[2]: pip install scikit-learn

Defaulting to user installation because normal site-packages is not writeableNote: you may need to restart the kernel to use updated packages.

[notice] A new release of pip is available: 24.3.1 -> 25.0.1
[notice] To update, run: C:\Users\Avinash\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.13_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip

Collecting scikit-learn
  Downloading scikit_learn-1.6.1-cp313-cp313-win_amd64.whl.metadata (15 kB)
Requirement already satisfied: numpy>=1.19.5 in c:\users\avinash\appdata\local\packages\pythonsoftwarefoundation.python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from scikit-learn) (2.2.1)
Collecting scipy>=1.6.0 (from scikit-learn)
```

## Import the Toy data and train the model

```
[4]: df = pd.read_csv("toydata2.csv")

[ ]: # Features and target variable

[10]: X = df[["Price", "Advertising_Spend"]]
    y = df["Popularity_Score"]

[ ]: # Train-test split

[11]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

[12]: model = LinearRegression()

[ ]: # Train the model

[13]: model.fit(X_train, y_train)

[13]: LinearRegression
    LinearRegression()
```

## Predict and save the model in.pkl file.

```
[ ]: # Predict on the test set

[14]: y_pred = model.predict(X_test)

[ ]: # Calculate and print the Mean Squared Error

[15]: mse = mean_squared_error(y_test, y_pred)
    print("Mean Squared Error:", mse)
    Mean Squared Error: 21.180542161654472

[ ]: # Save the model to a .pkl file

[16]: with open("toy_model.pkl", "wb") as file:
    pickle.dump(model, file)

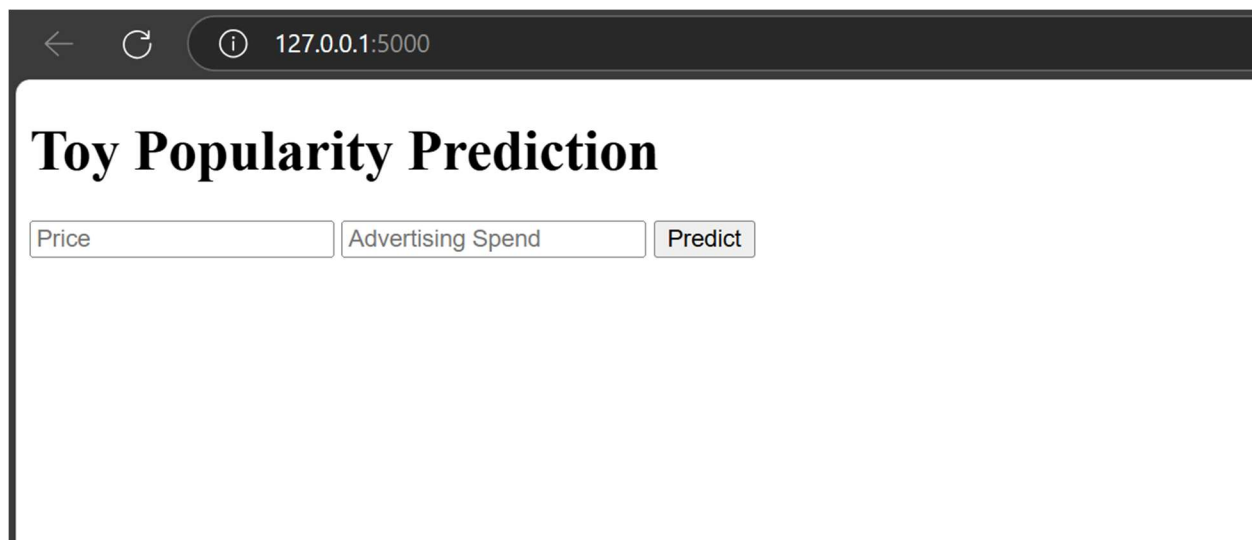
[ ]:
```

## Deploy the application

```
C:\Windows\System32 x + v
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Avinash\Desktop\Data Glacier\Week 4 - Model Deployment Flask>python deploy_flask_app.py
* Serving Flask app 'deploy_flask_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
```

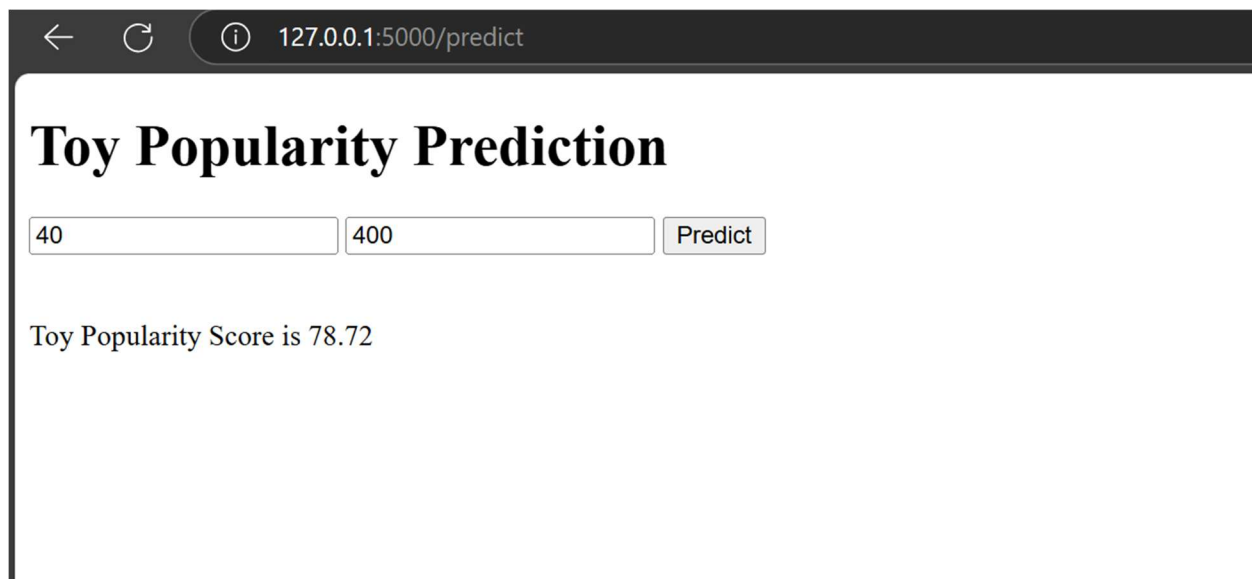
Application started



← ↻ ⓘ 127.0.0.1:5000

## Toy Popularity Prediction

Predict the popularity score



← ↻ ⓘ 127.0.0.1:5000/predict

## Toy Popularity Prediction

Toy Popularity Score is 78.72



127.0.0.1:5000/predict

# Toy Popularity Prediction

Toy Popularity Score is 87.14