

Name: Ameya Damle

Batch Code: LISUM23: 30

Submission Date: 28-July-2023

Submitted to: Data Glacier

## Deployment of Flask Web App

### Step 1:

Developing a model:

Predict the price of a house based on feature 'area' using Linear Regression Model.

```
Model_Deployment_on_Flask.ipynb > regressor = LinearRegression()
+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline ...
[1] ✓ 9.8s
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from flask import Flask, request, jsonify, render_template
import pickle
import json

[2] ✓ 3.7s
house_data = pd.read_excel('Housing.xlsx')
X = house_data.iloc[:, :-1].values
y = house_data.iloc[:, 1].values

[3] ✓ 0.0s
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state = 100)

[4] ✓ 0.1s
regressor = LinearRegression()
regressor.fit(X_train, y_train)
y_pred = regressor.predict(X_test)
print(y_pred)

... [10500307.49539785 10422298.85447766 10578316.13631805 10517469.3964003
10556473.7168604 10592669.72624737]
```

### Step 2:

Saving the trained model as pickle file in the root folder.

```
#saving the model in disk
pickle.dump(regressor, open('model.pkl','wb'))

model = pickle.load(open('model.pkl','rb'))
print(model.predict([[7420]]))

... [10544616.40344053]
```

Step 3:

Deployment of model using flask.

```
# Model Deployment
app = Flask(__name__)
model = pickle.load(open('model.pkl','rb'))

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

@app.route('/',methods=['POST'])
def predict():
    data = request.get_json(force=True)
    prediction = model.predict([[np.array(data['exp'])]])
    output = prediction[0]
    return jsonify(output)

if __name__ == '__main__':
    app.run(port=5000)

...
* Serving Flask app '__main__'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production W
* Running on http://127.0.0.1:5000
```

The script creates a Flask app and loads a machine learning model from a pickled file (model.pkl). The app has two routes defined: one for the home page ("/") that renders the "index.html" template, and another for handling POST requests sent to the root URL ("/") for making predictions. When the user submits the input data using the form on the home page, a POST request is sent to the server, and the predict() function is executed to make predictions using the model. The result of the prediction is then returned as a JSON response to the client.

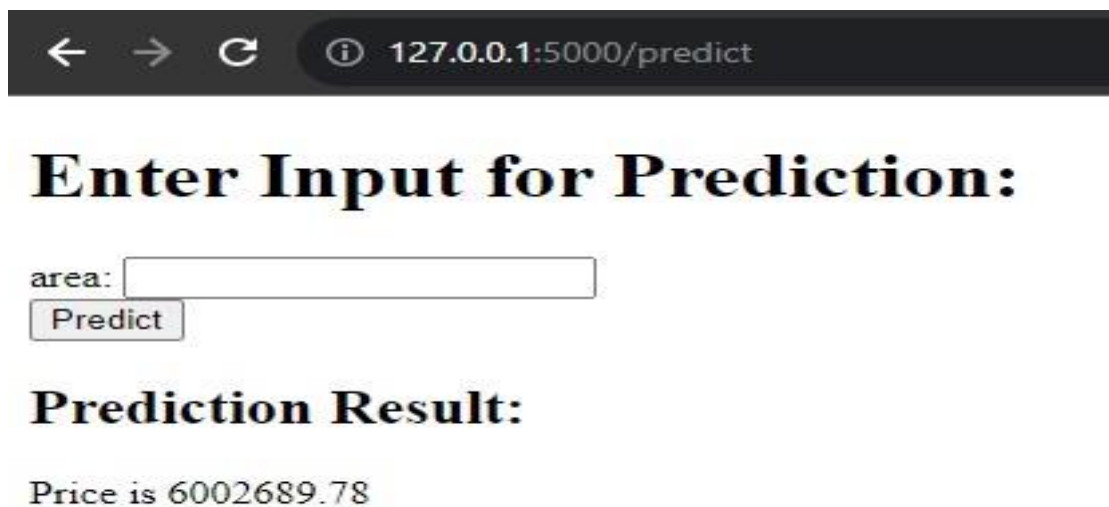
Step 4:

Checking the app.py file in cmd.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER COMMENTS python + ~
(base) D:\House-Price-Prediction>python app.py
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.0.87:5000
Press CTRL+C to quit
[]
```

Step 5:

Launching the webapp by typing address “127.0.0.1:5000” in any browser.



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/predict". The main content area has a heading "Enter Input for Prediction:" followed by a text input field labeled "area:". Below the input field is a button labeled "Predict". Underneath the button is another heading "Prediction Result:" followed by the text "Price is 6002689.78".

Step 6:


Creating a web app in Azure. While creating the Web application make sure you enable GitHub Actions in Deployment step and register your GitHub account as well.

## Create Web App ...

Resource Group \* ⓘ (New) HousePrice   
[Create new](#)

### Instance Details



Need a database? [Try the new Web + Database experience.](#) 

Name \* HousePriceAPI   
.azurewebsites.net

Publish \* ☒ Code ☐ Docker Container ☐ Static Web App

Runtime stack \* Python 3.8 


Operating System \* ☒ Linux ☐ Windows

Region \* East US   
 Not finding your App Service Plan? Try a different region or select your App Service Environment.

...

## Create Web App ...

Basics Deployment Networking Monitoring Tags Review + create

**Enable GitHub Actions to continuously deploy your app.** GitHub Actions is an automation framework that can build, test, and deploy your app whenever a new commit is made in your repository. If your code is in GitHub, choose your repository here and we will add a workflow file to automatically deploy your app to App Service. If your code is not in GitHub, go to the Deployment Center once the web app is created to set up your deployment. [Learn more](#) 


### GitHub Actions settings


Continuous deployment ☐ Disable ☒ Enable


### GitHub Actions details

Select your GitHub details, so Azure Web Apps can access your repository. You must have write access to your chosen repository to deploy with GitHub Actions.

GitHub account daameya [Change account](#) ⓘ

Organization \* daameya 

Repository \* House-Price-Prediction 

Branch \* main 

[Review + create](#)

[< Previous](#)

[Next : Networking >](#)

### Step 7:

Review and create the web application and it automatically activate a continuous pipeline with the repository in your GitHub profile.

## Deployment Center ☆ ...

<< Save Discard Browse Manage publish profile Sync Leave Feedback

Source \*

GitHub

Building with GitHub Actions. [Change provider.](#)

GitHub

App Service will place a GitHub Actions workflow in your chosen repository to build and deploy your app whenever there is a commit on the chosen branch. If you can't find an organization or repository, you may need to enable additional permissions on GitHub. You must have write access to your chosen GitHub repository to deploy with GitHub Actions. [Learn more](#)

Signed in as

daameya [Change Account](#) ⓘ

Organization \*

daameya

Repository \*


House-Price-Prediction

Branch \*

main

### Step 8:


Use the web address now available in the Actions section of the repository. After the deployment is successful you will be able to see a workflow folder in your repository which is created automatically.









 **House-Price-Prediction** Public

[Pin](#) [Unwatch](#) 1

main 1 branch 0 tags

[Go to file](#) [Add file](#) [Code](#)

 **daameya** Add or update the Azure App Service build and deployment workflow config 8de29bc 13 hours ago 10 commits

 .github/workflows	Add or update the Azure App Service build and deployment workflow c...	13 hours ago
 templates	updated model file	yesterday
 Deployment steps.pdf	deployment steps added	yesterday
 Housing.xlsx	updated model file	yesterday
 Model_Deployment_on_Flask.ipynb	updated model file	yesterday
 README.md	Update README.md	yesterday
 app.py	updated model file	yesterday
 model.pkl	updated model file	yesterday