Flask Deployment

Housing Price Prediction Model Deployment

Sheldon Gordon
LISUM35
August 2, 2024

Agenda

- Executive Summary
- Approach
  - Model Development and Training
  - Deployment
- Model Deployment
  - Launch on Local Server
  - Model Deployed
- Results
- Appendix

Executive Summary

The main goal of this project is to create and deploy a machine learning model to predict housing prices using features such as the number of bedrooms, square footage, and the age of the house. This model is designed to deliver price predictions and can aid potential buyers and real estate professionals in making well-informed decisions.

Approach – Model Development and Training

Data: A dataset containing various attributes related to housing prices, including the number of bedrooms, square footage, year built, and price.

Model Development: The model was built using Python's scikit-learn linear regression model.

Model Training: The model was trained using housing dataset.

Model Saving: The trained model was saved as a pickle file for ease of deployment.

Approach – Deployment

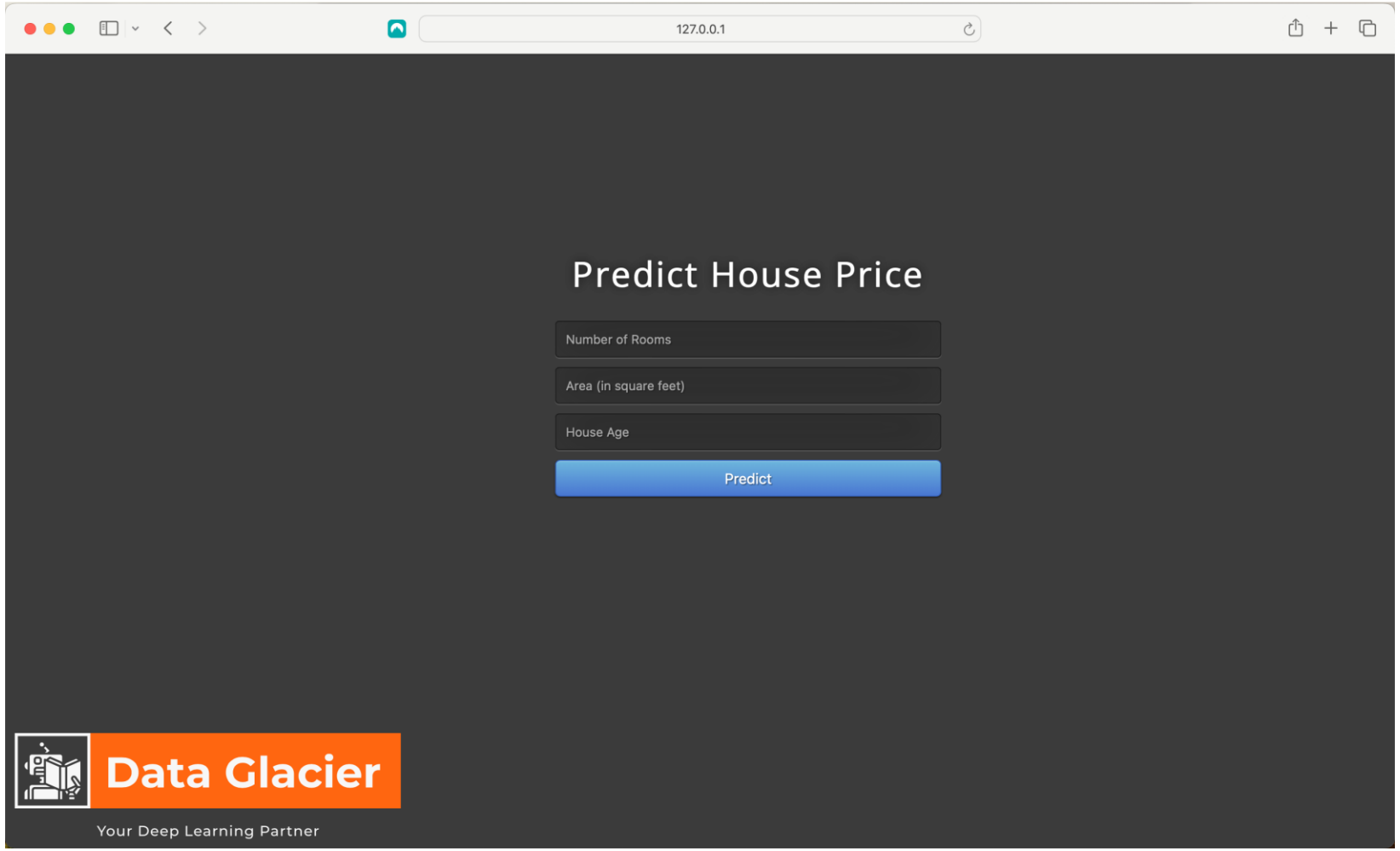
Web Application: The model was deployed using Flask, a lightweight WSGI web application framework in Python.

User Interface: A simple web interface was created to allow users to input housing attributes and receive price predictions.

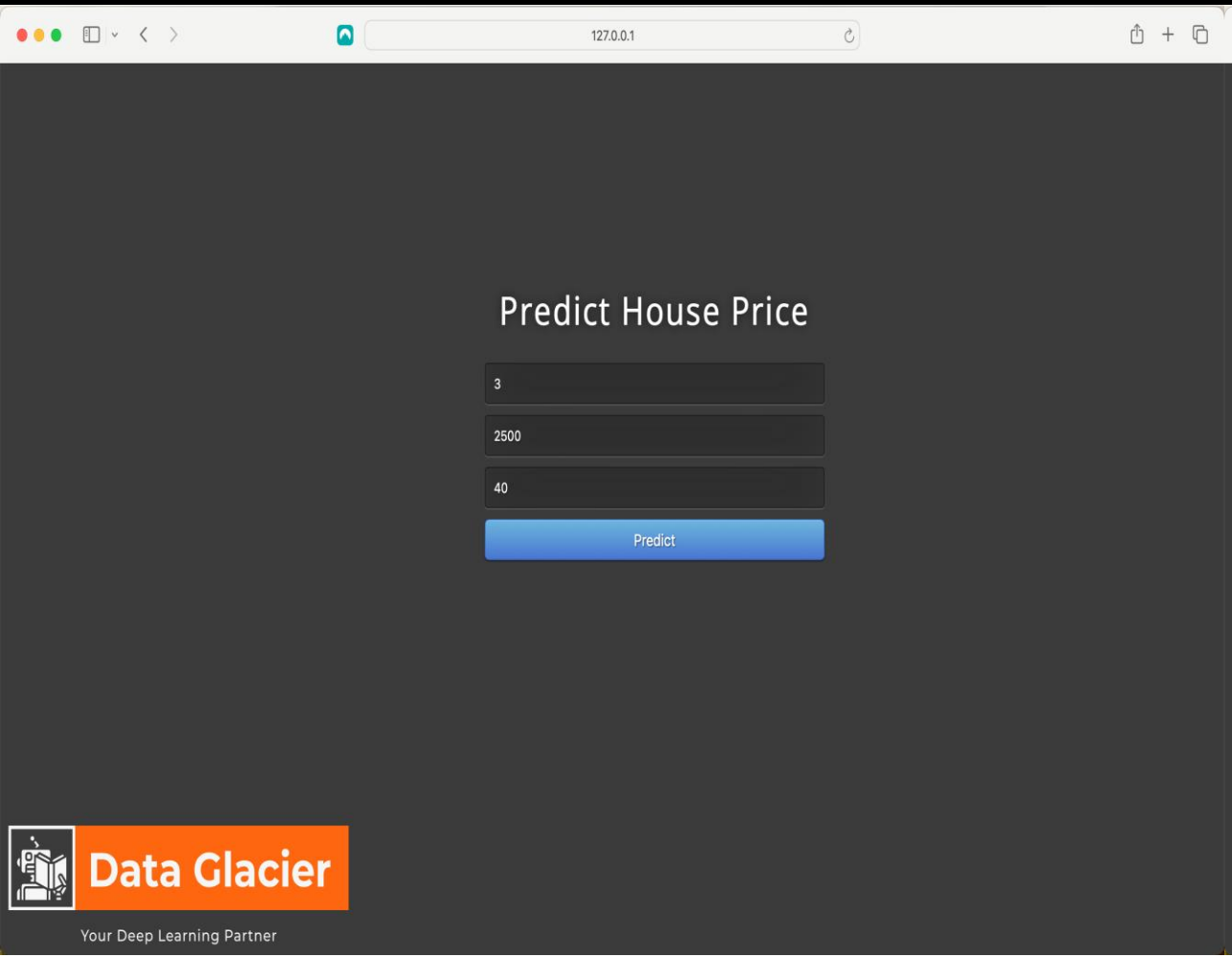
Server Setup: The Flask application was set up on a local server, with endpoints to handle predictions and display results.

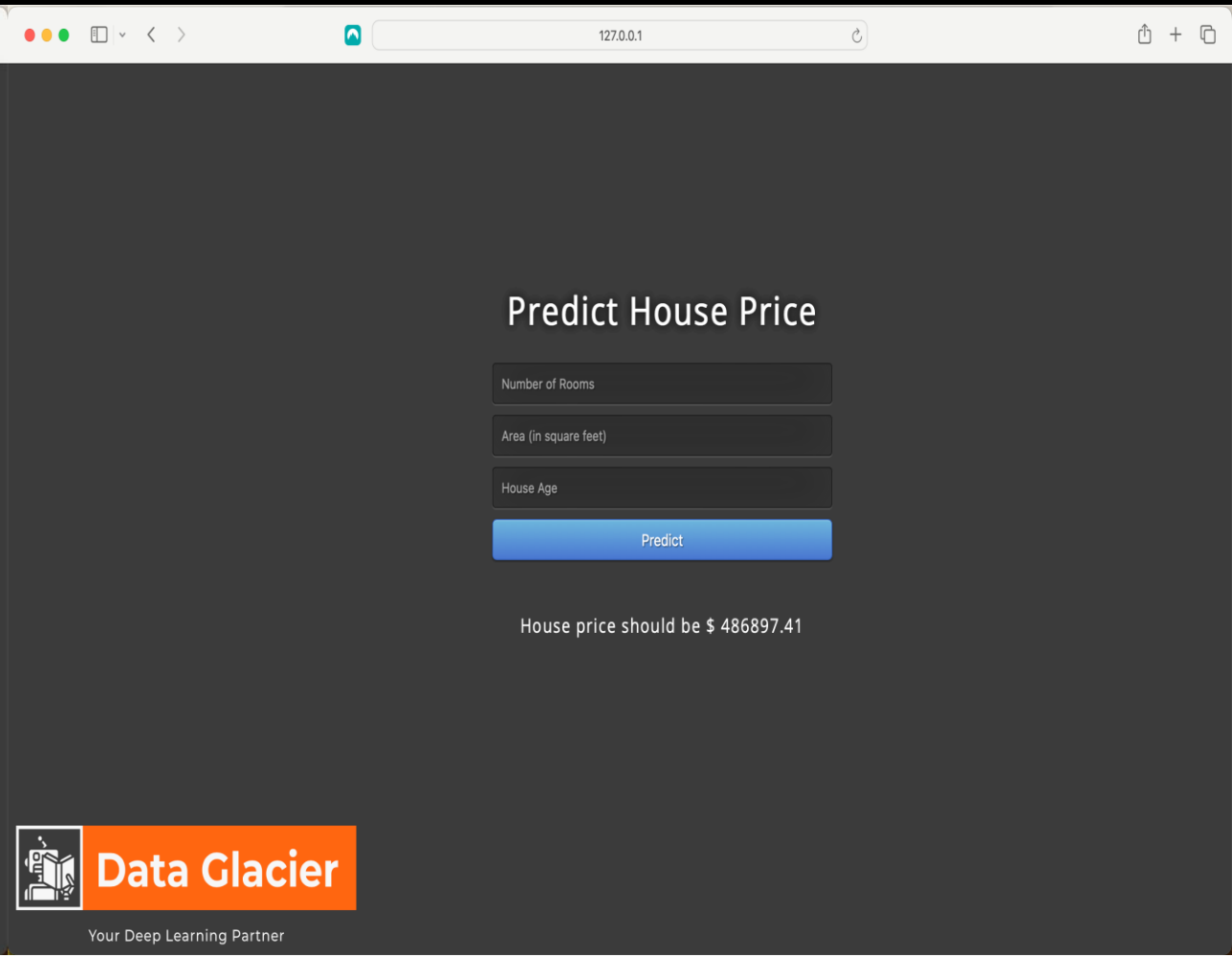
Deployment – Launch on Local Server

Deployment – Model Deployed



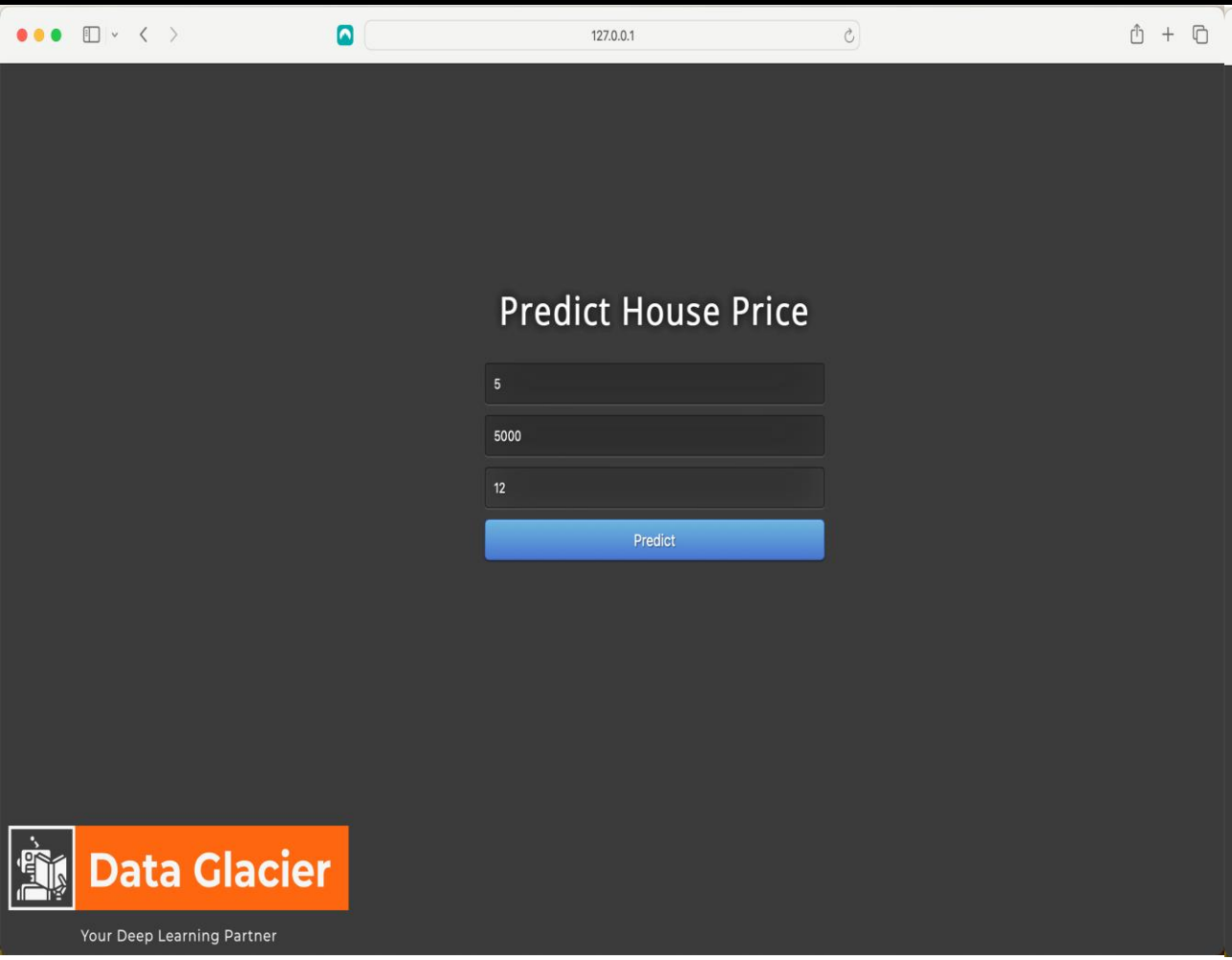
Result Example 1

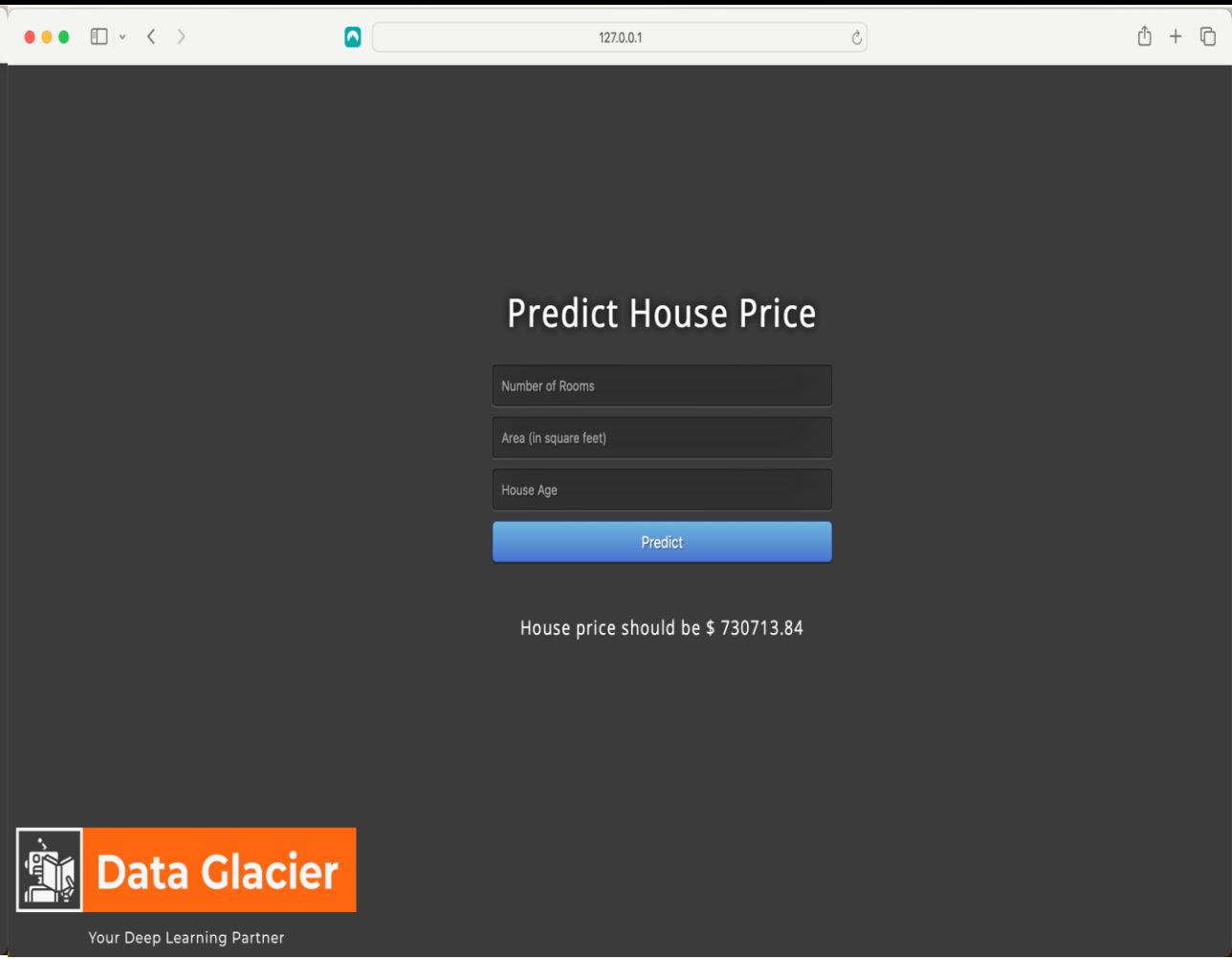




Data Glacier

Result Example 2





Data Glacier

Appendix – Code Snippet for Flask

app.py

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

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():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [int(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='House price should be $
{}'.format(output))

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

Appendix – Code Snippet for ML Model

model.py

```
# Importing the libraries
import numpy as np
import pandas as pd
import pickle

dataset = pd.read_csv('/Users/gordon/Documents/Studies/DS/Data Glacier/Flask Project/Flask-
Deployment2/price.csv')

X = dataset.iloc[:, :3]

y = dataset.iloc[:, -1]

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()

#Fitting model with training data
regressor.fit(X, y)

# Saving model to disk
pickle.dump(regressor, open('model.pkl','wb'))

# Loading model to compare the results
model = pickle.load(open('model.pkl','rb'))
print(model.predict([[2, 2200, 5]]))
```

Thank You