

Week 4: Deployment on Flask

Name: ZYAD HUSSEIN

Batch Code: <http://127.0.0.1:9000/>

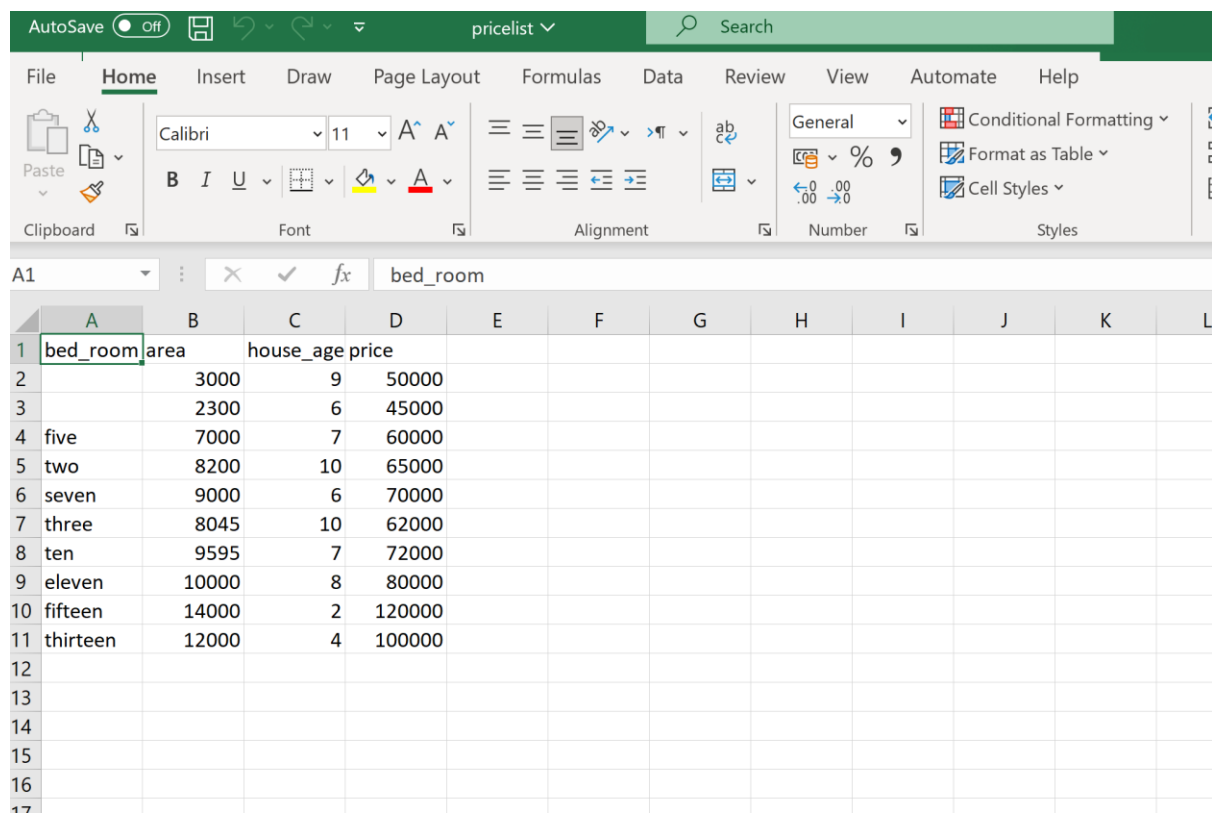
Submission Date: 28/06/2023

Submitted to: Data Glacier

In this assignment, the same dataset file but changed the data inside of it as illustrated in figure 1. The flask implementation was understood using flask deployment videos provided. the index.html file attached to the google drive was also used to apply the web app layout.

The model.py file provided utilizes the dataset to train a regression model model.pkl. it was used to train/test the model with the pricelist dataset. In myapp.py (the python app file containing the flask app) two functions were implemented, home function and predict function. Final features were acquired from initial features to be used in model predictions. The output variable rounds the model result to two decimal points and the predict function returns the result on the web application.

Figure 1 demonstrates the implemented code to build the application. As requested, simple small dataset was used. However, if the project was to be implemented again, cab data from week 3 can be used as it is a viable data for this project. Flask library was downloaded using terminal command: pip install flask.



	A	B	C	D	E	F	G	H	I	J	K	L
1	bed_room	area	house_age	price								
2		3000	9	50000								
3		2300	6	45000								
4	five	7000	7	60000								
5	two	8200	10	65000								
6	seven	9000	6	70000								
7	three	8045	10	62000								
8	ten	9595	7	72000								
9	eleven	10000	8	80000								
10	fifteen	14000	2	120000								
11	thirteen	12000	4	100000								
12												
13												
14												
15												
16												
17												

Fig 1: dataset file preview.

```

C:\Users\20101\Desktop\week4\myapp.py
y - week5 X v-rep-bug0.py - assignment X forwardkinematics.py X myapp.py X
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Jun 26 19:36:02 2023
4
5 @author: zyad hussein
6 """
7 import numpy as np
8 from flask import Flask, request, render_template
9 import pickle
10
11 app = Flask(__name__)
12 model = pickle.load(open('model.pkl', 'rb'))
13
14 @app.route('/')
15 #def home():
16 #    return 'hello world'
17 def home():
18     return render_template('index.html')
19 @app.route('/predict', methods=['POST'])
20
21
22
23 def predict():
24     ifeatures = [int(x) for x in request.form.values()]
25     ffeatures = np.array(ifeatures)
26     pred = model.predict(ffeatures)
27     output = round(pred[0], 2)
28     return render_template('index.html', prediction_text='price')
29 app.run(port=9000)

```

model linear_model_base.LinearRegress

Help Variable Explorer Plots Files

Console 1/A X

```

quit)
127.0.0.1 - - [28/Jun/2023
20:19:16] "POST /predict HTTP/
1.1" 200 -
127.0.0.1 - - [28/Jun/2023
20:19:16] "GET /static/css/
style.css HTTP/1.1" 404 -
127.0.0.1 - - [28/Jun/2023
20:19:16] "GET /static/images/
Original.svg HTTP/1.1" 404 -
127.0.0.1 - - [28/Jun/2023
20:19:27] "POST /predict HTTP/
1.1" 200 -
127.0.0.1 - - [28/Jun/2023
20:19:27] "GET /static/css/
style.css HTTP/1.1" 404 -
127.0.0.1 - - [28/Jun/2023
20:19:27] "GET /static/images/
Original.svg HTTP/1.1" 404 -

```

IPython Console History

LSP Python: ready conda: base (Python 3.9.13) Line 28, Col 57 UTF-8 CRLF RW Mem 44%

Fig 2: flask code implementation.

Figure 3 demonstrates the modifications of model.py file to aid in the training/testing of the regression model used.

```

model.py
# importing the libraries
import numpy as np
import pandas as pd
import pickle

dataset = pd.read_csv('pricelist.csv')

dataset['bed_room'].fillna(0, inplace=True)
dataset['area'].fillna(dataset['area'].mean(), inplace=True)

X = dataset.iloc[:, :3]

#Converting words to integer values
def convert_to_int(word):
    word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6, 'seven':7, 'eight':8,
                  'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}
    return word_dict[word]

X['bed_room'] = X['bed_room'].apply(lambda x : convert_to_int(x))

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]]))

```

Fig 3: model.py modified.

Figure 4 illustrates the output of the implemented flask code to build the web application without the style.css file.

Predict House Price

price is 50266.97

Company Logo

Fig 4: web application implementation without css styling file.

Figure 5 reveals the web application implementation alongside the styling css code.

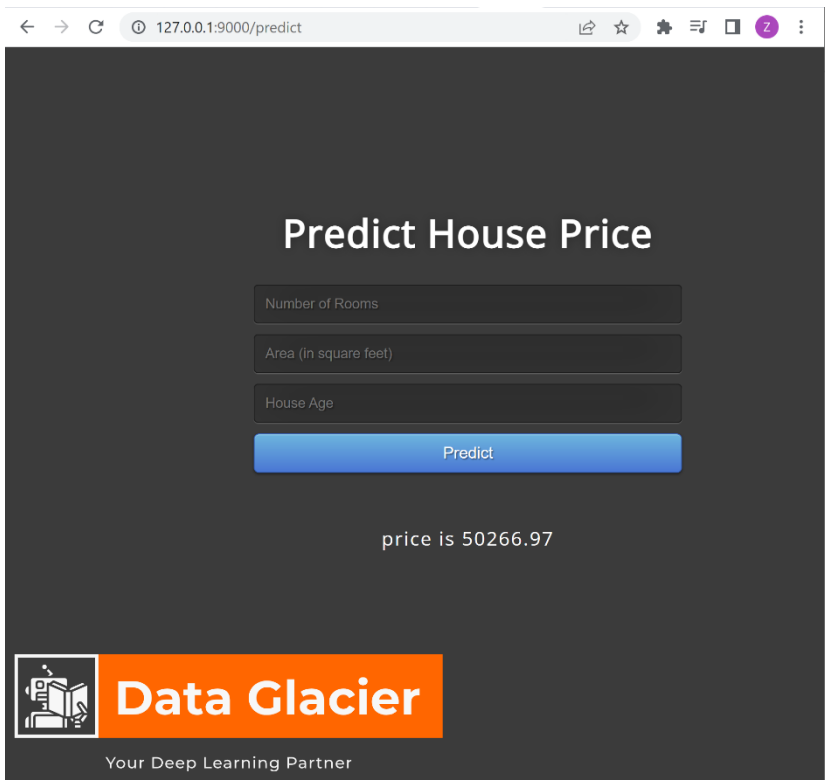


Fig 5: the built web application with css styling code.