

NAME : Snigdha Chigurupati

BATCH CODE: LISUM19

Submission Date: 3/29/2023

Building the machine learning model and deploying it on Heroku:

1. Importing libraries and loading the dataset

```
In [1]: 1 # Importing the libraries
        2 import numpy as np
        3 import pandas as pd
        4 import pickle
```

```
In [2]: 1 dataset = pd.read_csv('price.csv')
        2 dataset
```

```
Out[2]:
```

	bed_room	area	house_age	price
0	NaN	2300.0	9	50000
1	NaN	2152.0	6	45000
2	five	8520.0	7	60000
3	two	9025.0	10	65000
4	seven	9900.0	6	70000
5	three	8045.0	10	62000
6	ten	9564.0	7	72000
7	eleven	NaN	8	80000

2. Filling the null values of bed_room column with '0' and area with 'mean'. Then converting the words in the bed_room column to integer values for ease in building a ML algorithm

```
In [3]: 1 dataset['bed_room'].fillna(0, inplace=True)
        2
        3 dataset['area'].fillna(dataset['area'].mean(), inplace=True)
        4
        5 X = dataset.iloc[:, :3]
```

```
In [4]: 1 #Converting words to integer values
        2 def convert_to_int(word):
        3     word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6, 'seven':7, 'eight':8,
        4                 'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}
        5     return word_dict[word]
        6
        7 X['bed_room'] = X['bed_room'].apply(lambda x : convert_to_int(x))
        8
        9 y = dataset.iloc[:, -1]
```

3. Creating an instance/object of Linear Regression and training the model. Then saving the model to disk as a pickle file.

```
In [14]: 1 from sklearn.linear_model import LinearRegression
2 regressor = LinearRegression()
3
4 #Fitting model with trainig data
5 regressor.fit(X, y)
6
7 # Saving model to disk
8 pickle.dump(regressor, open('model.pkl','wb'))
9
10 # Loading model to compare the results
11 model = pickle.load(open('model.pkl','rb'))
12 print(model.predict([[5, 2150, 6]]))

[56050.91893625]
```

4. Creating a app.py file to import the pickle object created before and following below steps:
 - a. Using the index.html template to a format or interface to input the values.
 - b. Creating the predict function which takes the input values from the users and predicts values based on the backend Linear Regression model.
 - c. Outputting the value through the same index template which shows the result.

```
1 import numpy as np
2 from flask import Flask, request, render_template
3 import pickle
4 app = Flask(__name__)
5 model = pickle.load(open('model.pkl', 'rb'))
6 @app.route('/')
7 def home():
8     return render_template('index.html')
9
10 @app.route('/predict', methods=['POST'])
11 def predict():
12     int_features = [int(x) for x in request.form.values()]
13     final_features = [np.array(int_features)]
14     prediction = model.predict(final_features)
15     output = round(prediction[0], 2)
16     return render_template('index.html', prediction_text='House price should be $ {}'.format(output))
17
18 if __name__ == "__main__":
19     app.run(debug=True)
```

5. Pushing the all the files such as Procfile, requirements, pickle file etc. and code to github account.


templates	Create index.html
test	Update test.py
.gitignore	Initial commit
LICENSE	week 5
Procfile	Add files via upload
README.md	Update README.md
app.py	week 5
environment.yml	Update environment.yml
model.pkl	Update model.pkl
price.csv	week 5
requirements.txt	Add files via upload
response.json	Final Commit
runtime.txt	Update runtime.txt

- Now, creating a Heroku account and creating an app (with name 'heroku-dataglacier-snigdha') after downloading salesforce authenticator app from app store and providing the payment details. Now, under the 'deploy' tab, select deployment method as Github.

← → ↻ 🏠 dashboard.heroku.com/apps/heroku-dataglacier-snigdha/deploy/github

Favourites AWS Python

Salesforce Platform

 **HEROKU**

Jump to Favorites, Apps, Pipelines, Spaces...

Personal > heroku-dataglacier-snigdha

Overview Resources **Deploy** Metrics Activity Access Settings

Add this app to a pipeline


Create a new pipeline or choose an existing one and add this app to a stage in it.


Add this app to a stage in a pipeline to enable additional actions

Pipelines let you connect multiple apps together and **promote code** between them. [Learn more.](#)

Choose a pipeline


Deployment method


 Heroku Git
Use Heroku CLI


 GitHub
Connect to GitHub

7. Below, in the connect to github section, search for the repository that needs to be connected and click on connect.

Deployment method

 Heroku Git
Use Heroku CLI

 GitHub
Connect to GitHub


 Container Registry
Use Heroku CLI

Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.


Search for a repository to connect to


Missing a GitHub organization? [Ensure Heroku Dashboard has team access.](#)


 Snigdha-Chigurupati/Data-Glacier

8. It is now connected to the repository.

Deployment method



 Heroku Git
Use Heroku CLI

 GitHub
Connected

 Container Registry
Use Heroku CLI

App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to  [Snigdha-Chigurupati/Data-Glacier](#) by  [Snigdha-Chigurupati](#)

Releases in the [activity feed](#) link to GitHub to view commit diffs

9. In the manual deployment, select the branch which has the files and code and click on `deploy branch`.


Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more](#).

Choose a branch to deploy

 DataGlacier_Snigdha_main

[Deploy Branch](#)

Receive code from GitHub

Build DataGlacier_Snigdha_main 2f6ead91

```
-----> Building on the Heroku-22 stack
-----> Using buildpack: heroku/python
-----> Python app detected
-----> Using Python version specified in runtime.txt
-----> No change in requirements detected, installing from cache
-----> Using cached install of python-3.9.16
-----> Installing pip 23.0.1, setuptools 63.4.3 and wheel 0.38.4
```



☒ Autoscroll with output


[View build log](#)

Release phase

Deploy to Heroku

10. Click on view build log to get the link to the deployment server and check the logs.

 Personal >  heroku-dataglacier-snigdha

 Snigdha-Chigurupati/Data-Glacier

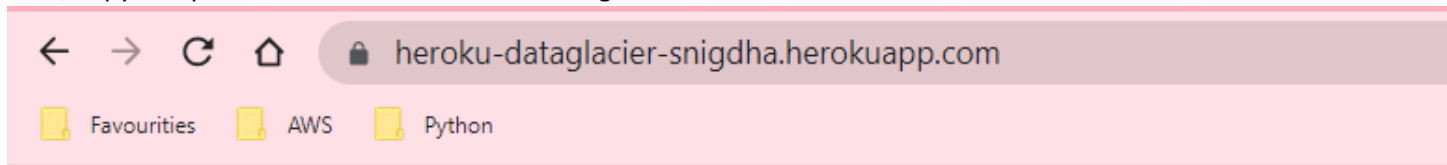
[Overview](#) [Resources](#) [Deploy](#) [Metrics](#) [Activity](#) [Access](#) [Settings](#)

[Activity Feed](#) > [Build Log](#)

```
-----> Building on the Heroku-22 stack
-----> Using buildpack: heroku/python
-----> Python app detected
-----> Using Python version specified in runtime.txt
-----> No change in requirements detected, installing from cache
-----> Using cached install of python-3.9.16
-----> Installing pip 23.0.1, setuptools 63.4.3 and wheel 0.38.4
-----> Installing SQLite3
-----> Installing requirements with pip
-----> Discovering process types
Procfile declares types -> web
-----> Compressing...
Done: 132.1M
-----> Launching...
Released v24
https://heroku-dataglacier-snigdha.herokuapp.com/ deployed to Heroku
```

Build finished

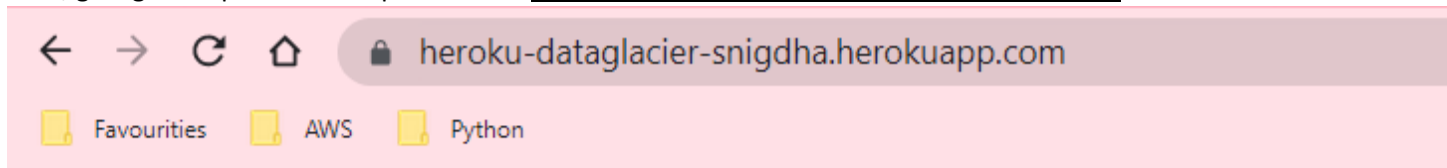
11. Now, copy and paste the link in a new window to get the below:



Predict House Price

Number of Rooms	Area (in square feet)	House Age	Predict
-----------------	-----------------------	-----------	---------

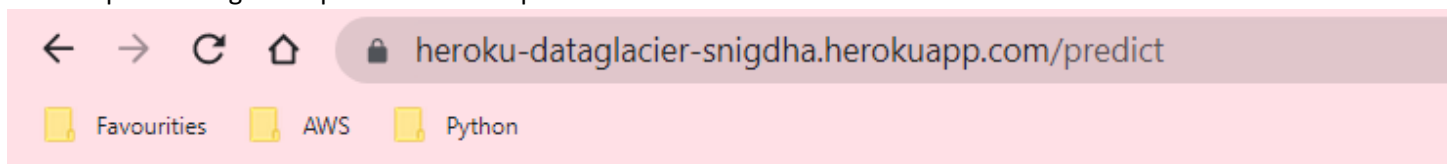
12. Now, giving the input values :Input values – Number of Rooms – 5, Area – 2150, House Age -6



Predict House Price

5	2150	6	Predict
---	------	---	---------

13. Click on predict to get the predicted house price:



Predict House Price

Number of Rooms	Area (in square feet)	House Age	Predict
-----------------	-----------------------	-----------	---------

House price should be \$ 56050.92