

Data Intake Report

Name: Pranav Walia

Report date: July 5

Internship Batch: LISUM 10: 30

Version:<1.0>

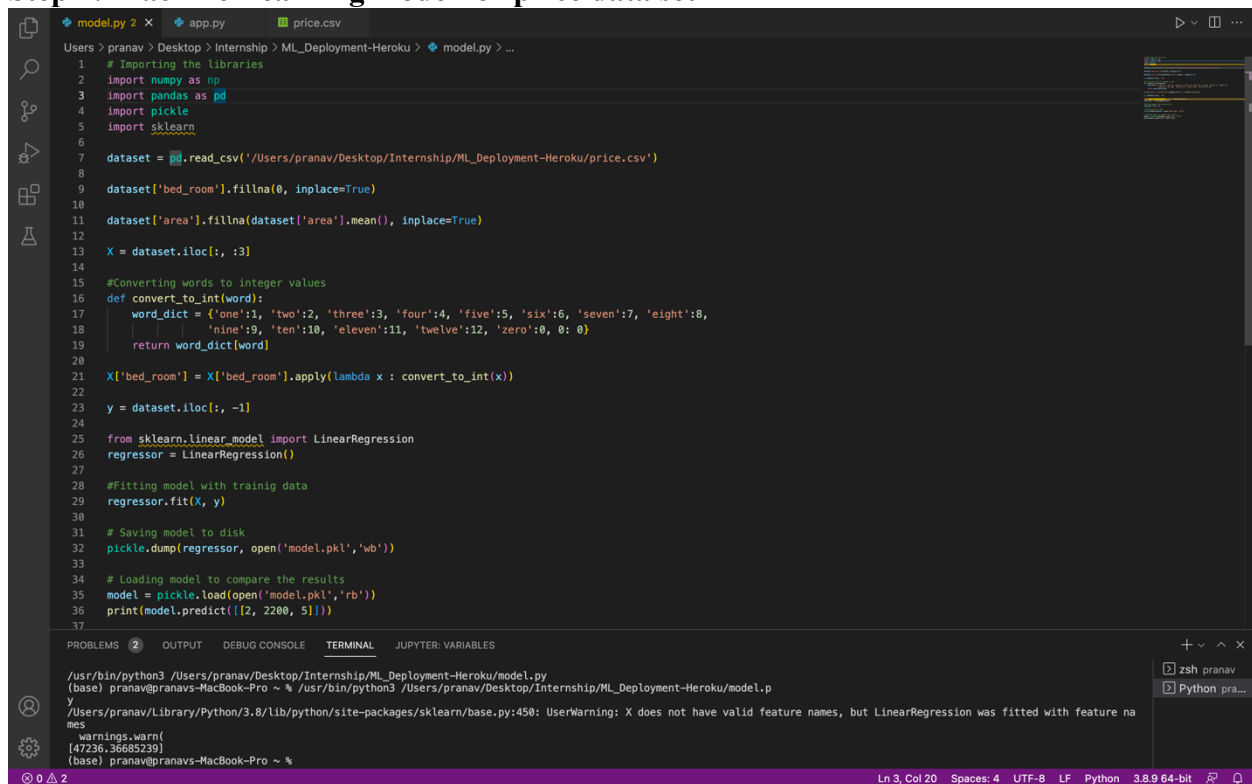
Data intake by: Pranav Walia

Data intake reviewer: Pranav Walia

Data storage location: github

Cloud and API deployment

Step 1: Machine Learning model for price data set



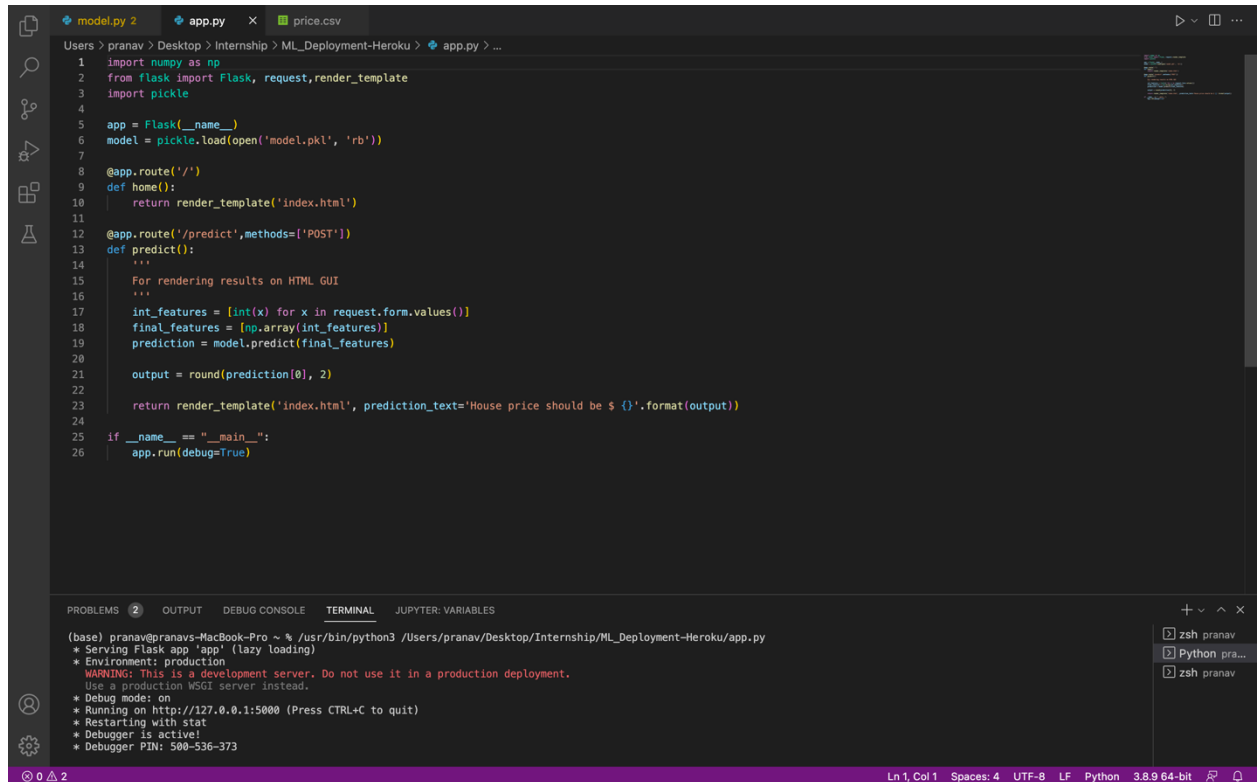
```
1 # Importing the libraries
2 import numpy as np
3 import pandas as pd
4 import pickle
5 import sklearn
6
7 dataset = pd.read_csv('/Users/pranav/Desktop/Internship/ML_Deployment-Heroku/price.csv')
8 dataset['bed_room'].fillna(0, inplace=True)
9 dataset['area'].fillna(dataset['area'].mean(), inplace=True)
10
11 X = dataset.iloc[:, :3]
12
13 #Converting words to integer values
14 def convert_to_int(word):
15     word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6, 'seven':7, 'eight':8,
16                 'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}
17     return word_dict[word]
18
19 X['bed_room'] = X['bed_room'].apply(lambda x : convert_to_int(x))
20
21 y = dataset.iloc[:, -1]
22
23 from sklearn.linear_model import LinearRegression
24 regressor = LinearRegression()
25
26 #Fitting model with training data
27 regressor.fit(X, y)
28
29 # Saving model to disk
30 pickle.dump(regressor, open('model.pkl','wb'))
31
32 # Loading model to compare the results
33 model = pickle.load(open('model.pkl','rb'))
34 print(model.predict([12, 2200, 5]))
35
36
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER VARIABLES

```
/usr/bin/python3 /Users/pranav/Desktop/Internship/ML_Deployment-Heroku/model.py
(base) pranav@pranavs-MacBook-Pro ~ % /usr/bin/python3 /Users/pranav/Desktop/Internship/ML_Deployment-Heroku/model.p
y
/Users/pranav/Library/Python/3.8/lib/python/site-packages/sklearn/base.py:458: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature na
mes
warnings.warn(
[47236.36685239]
(base) pranav@pranavs-MacBook-Pro ~ %
```

Ln 3, Col 20 Spaces: 4 UTF-8 LF Python 3.8.9 64-bit

Step 2: Creating the web app

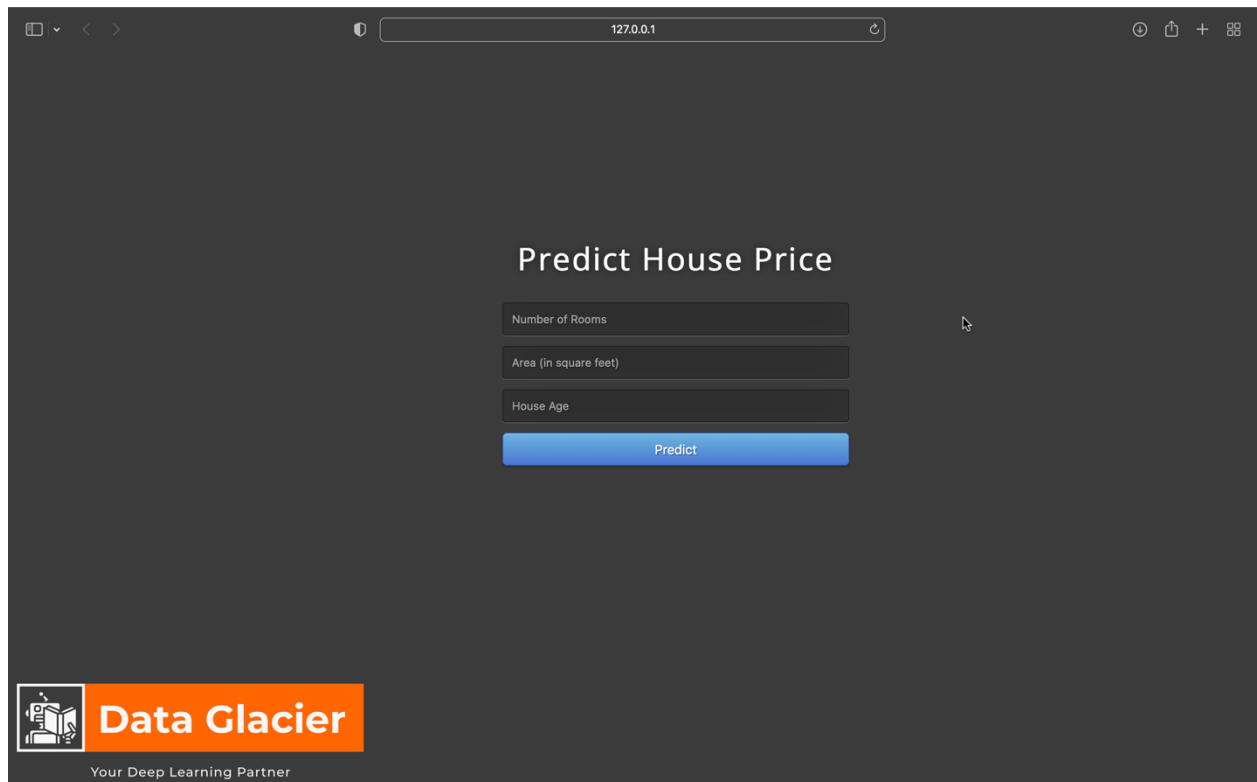


The screenshot shows a JupyterLab interface with a code editor and a terminal. The code editor displays the `app.py` file, which is a Flask web application. The terminal shows the output of running the application, including a warning about the development server and the URL `http://127.0.0.1:5000`.

```
1 import numpy as np
2 from flask import Flask, request, render_template
3 import pickle
4
5 app = Flask(__name__)
6 model = pickle.load(open('model.pkl', 'rb'))
7
8 @app.route('/')
9 def home():
10     return render_template('index.html')
11
12 @app.route('/predict', methods=['POST'])
13 def predict():
14     """
15     For rendering results on HTML GUI
16     """
17     int_features = [int(x) for x in request.form.values()]
18     final_features = np.array(int_features)
19     prediction = model.predict(final_features)
20
21     output = round(prediction[0], 2)
22
23     return render_template('index.html', prediction_text='House price should be $ {}'.format(output))
24
25 if __name__ == "__main__":
26     app.run(debug=True)
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER VARIABLES

```
(base) pranav@pranavs-MacBook-Pro ~ % /usr/bin/python3 /Users/pranav/Desktop/Internship/ML_Deployment-Heroku/app.py
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000 (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PID: 500-536-373
```



The screenshot shows a web browser window with the URL `127.0.0.1`. The page displays a form titled "Predict House Price" with three input fields: "Number of Rooms", "Area (in square feet)", and "House Age". A blue "Predict" button is located below the input fields. At the bottom of the page, there is a logo for "Data Glacier" and the text "Your Deep Learning Partner".


Predict House Price

Number of Rooms

Area (in square feet)

House Age

Predict

 **Data Glacier**

Your Deep Learning Partner

Step 3: Committing code in online repo

GitHub- <https://github.com/pranav611/Heroku-deployment-.git>

The screenshot shows a GitHub repository page for 'pranav611/Heroku-deployment-'. The repository is public and has 1 branch (main) and 0 tags. The file list shows several files and folders, including 'static', 'templates', 'LICENSE', 'Procfile', 'README.md', 'app.py', 'model.pkl', 'model.py', 'price.csv', and 'requirements.txt'. The 'README.md' file is selected, showing its content: 'ML Model Deployment on Heroku'. The right sidebar contains information about the repository, including the 'About' section (no description, website, or topics provided), 'Releases' (no releases published), 'Packages' (no packages published), and 'Languages' (Python 100%).

File/Folder	Commit Message	Commit Time
static	Add files via upload	35 minutes ago
templates	Add files via upload	35 minutes ago
LICENSE	Initial commit	42 minutes ago
Procfile	Add files via upload	35 minutes ago
README.md	Add files via upload	35 minutes ago
app.py	Add files via upload	35 minutes ago
model.pkl	Add files via upload	35 minutes ago
model.py	Add files via upload	35 minutes ago
price.csv	Add files via upload	35 minutes ago
requirements.txt	Add files via upload	35 minutes ago

README.md

ML Model Deployment on Heroku

Step 4: Heroku deployment

The screenshot shows the Heroku dashboard for an application named 'heroku-deployment-main'. The browser's address bar displays 'dashboard.heroku.com/apps/heroku-deployment-main/deploy/github'. The page features a search bar at the top with the text 'Jump to Favorites, Apps, Pipelines, Spaces...'. Below this, there's a dropdown menu showing 'main' and a checkbox for 'Wait for CI to pass before deploy' with a note: 'Only enable this option if you have a Continuous Integration service configured on your repo.' A button labeled 'Enable Automatic Deploys' is also present.

The main content area is divided into two sections. On the left, under 'Manual deploy', it says 'Deploy the current state of a branch to this app.' On the right, under 'Deploy a GitHub branch', it says 'This will deploy the current state of the branch you specify below. [Learn more.](#)' Below this, there's a section 'Choose a branch to deploy' with a dropdown menu showing 'main' and a 'Deploy Branch' button.

Below the deployment options, there's a progress bar showing the deployment steps: 'Receive code from GitHub', 'Build main 21096705', 'Release phase', and 'Deploy to Heroku'. Each step has a green checkmark indicating it was successful.

At the bottom of the progress bar, a message states 'Your app was successfully deployed.' with a 'View' button.

The footer of the page includes links for 'heroku.com', 'Blogs', 'Careers', 'Documentation', and 'Support'. It also contains 'Terms of Service', 'Privacy', 'Cookies', and a copyright notice '© 2022 Salesforce.com'.