This data Set contains House Price for metro cities in India. It comprises of 23 columns and 14620 Rows(No Null values). We aim to predict pricing bases on important features.

This data set was consists of below Attributes:

1.Id(int)

2.Date(int)

3.number of bedrooms(int)

4.number of bathrooms(float)-convert it into 'int' ,just to give you some practice!

5.living area(int)

6.lot area(int)

7.water front present(float)-convert it into 'int'

8.number of views(int)

and so on….

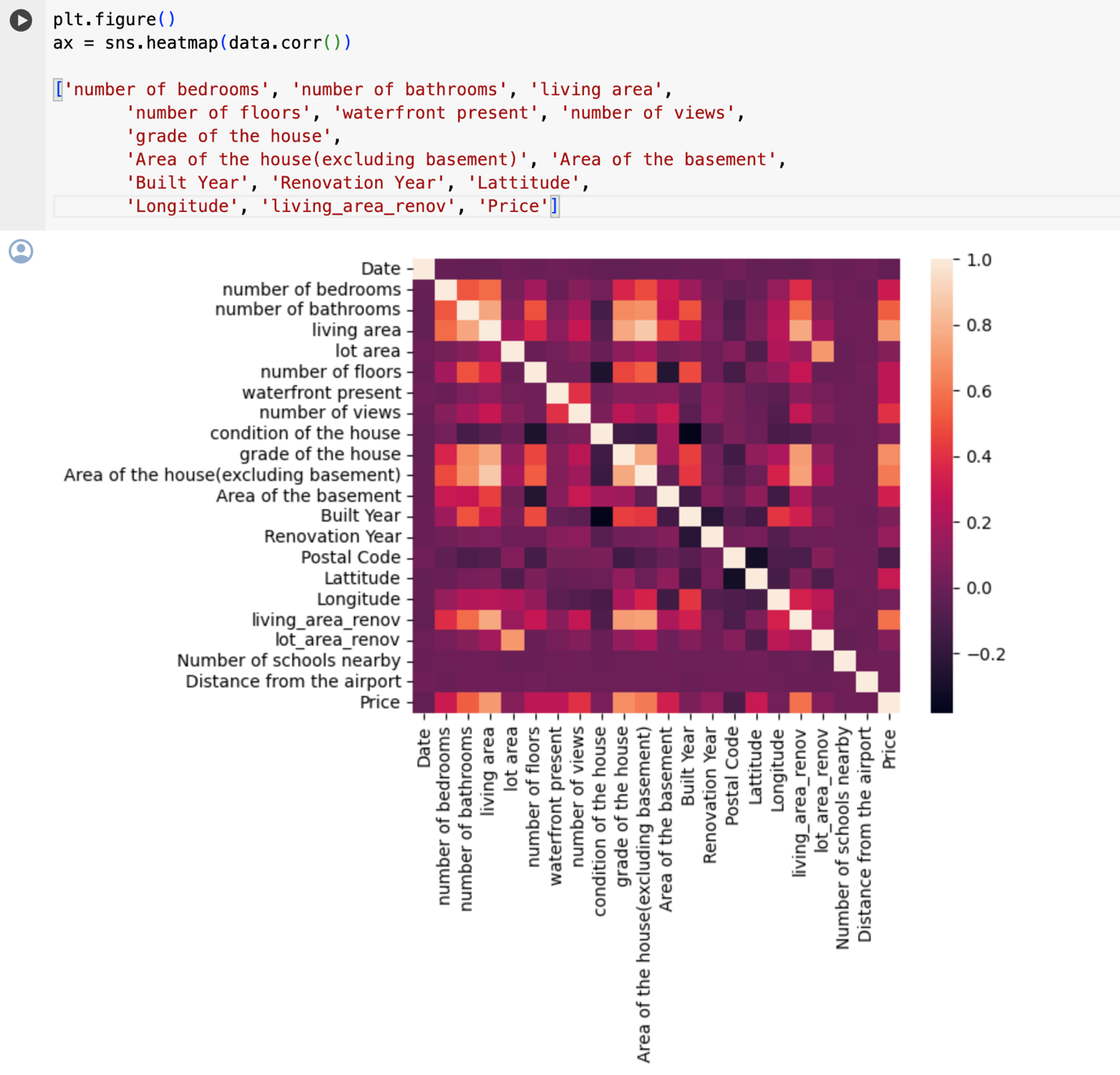


Fig 1.a Analysis of full dataset

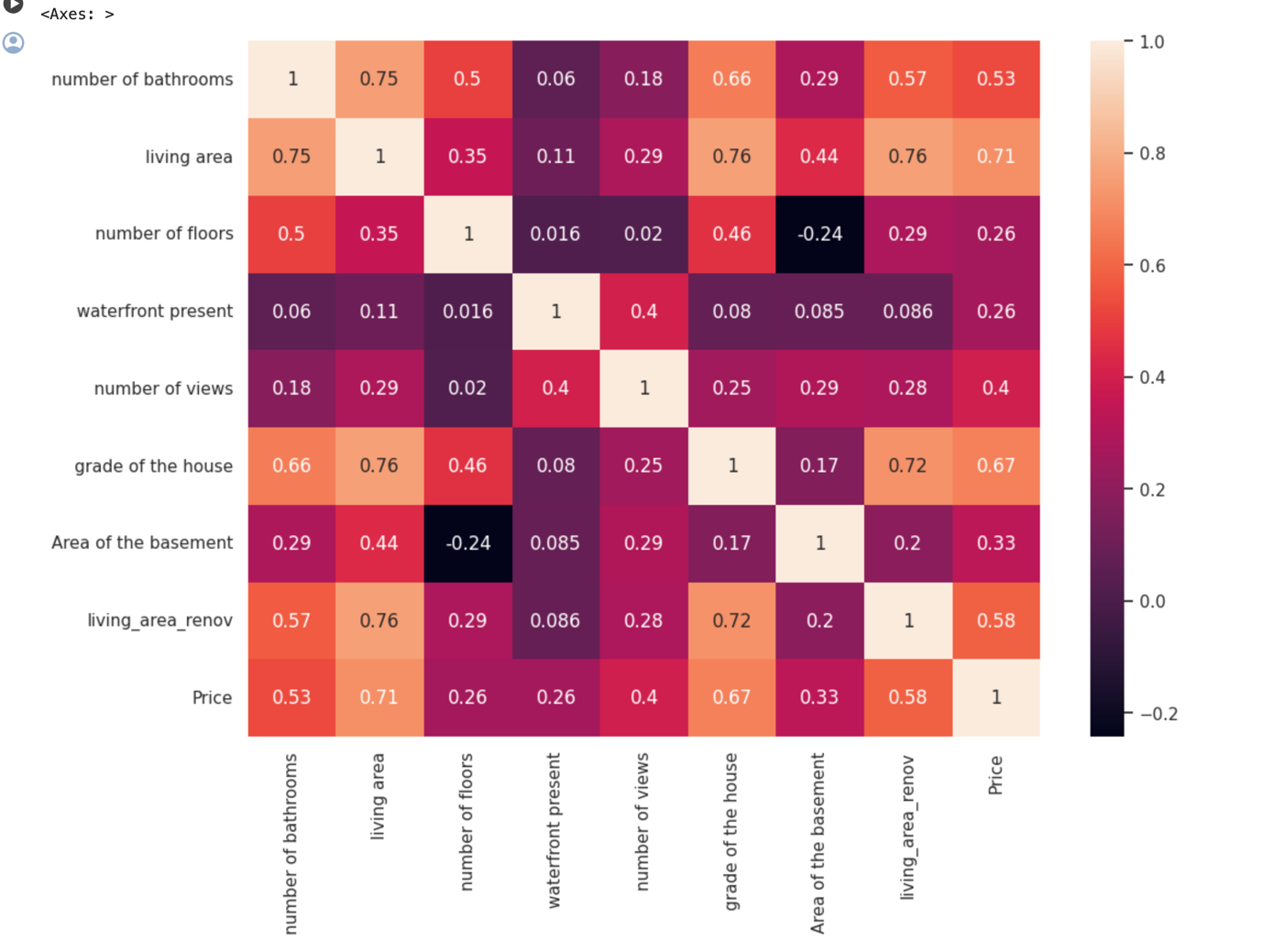


Fig 1.b HeatMap annotation after feature reduction

Important Points:

1. We used Z-Score for Normalization.
2. We used Linear Regression and Random Forest Regressor algorithm for Model building.

Below are scores metrics for Random Forest Regressor:

Mean Squared Error (MSE): 48559452291.03936

Root Mean Squared Error (RMSE): 220362.0935892545

Mean Absolute Error (MAE): 140907.11387195688

R-squared (R²): 0.6554086432159656

Below are scores metrics for Linear Regression:

Mean Squared Error (MSE): 52342062067.715996

Root Mean Squared Error (RMSE): 228783.87632811014

Mean Absolute Error (MAE): 152400.6186316584

R-squared (R²): 0.6285661939371446

AUTO ML

Best performing Algo : VotingEnsemble  
XGBoostRegressor

LightGBMRegressor

Normalisation Algo: "MaxAbsScaler"

R2 score: 0.67148

Median absolute error: 96252  
  
  
import urllib.request

import json

import os

import ssl

def allowSelfSignedHttps(allowed):

# bypass the server certificate verification on client side

if allowed and not os.environ.get('PYTHONHTTPSVERIFY', '') and getattr(ssl, '\_create\_unverified\_context', None):

ssl.\_create\_default\_https\_context = ssl.\_create\_unverified\_context

allowSelfSignedHttps(True) # this line is needed if you use self-signed certificate in your scoring service.

# Request data goes here

# The example below assumes JSON formatting which may be updated

# depending on the format your endpoint expects.

# More information can be found here:

# <https://docs.microsoft.com/azure/machine-learning/how-to-deploy-advanced-entry-script>

data = {

"Inputs": {

"data": [

{

"number of bedrooms": 5,

"number of bathrooms": 2.5,

"living area": 3650,

"number of floors": 2,

"waterfront present": 0,

"number of views": 4,

"grade of the house": 10,

"Area of the house(excluding basement)": 3370,

"Area of the basement": 280,

"living\_area\_renov": 2880

}

]

},

"GlobalParameters": 0.0

}

body = str.encode(json.dumps(data))

url = '<http://023631e1-ac74-4946-9b72-a0736f5b4f3d.eastus2.azurecontainer.io/score>'

headers = {'Content-Type':'application/json'}

req = urllib.request.Request(url, body, headers)

try:

response = urllib.request.urlopen(req)

result = response.read()

print(result)

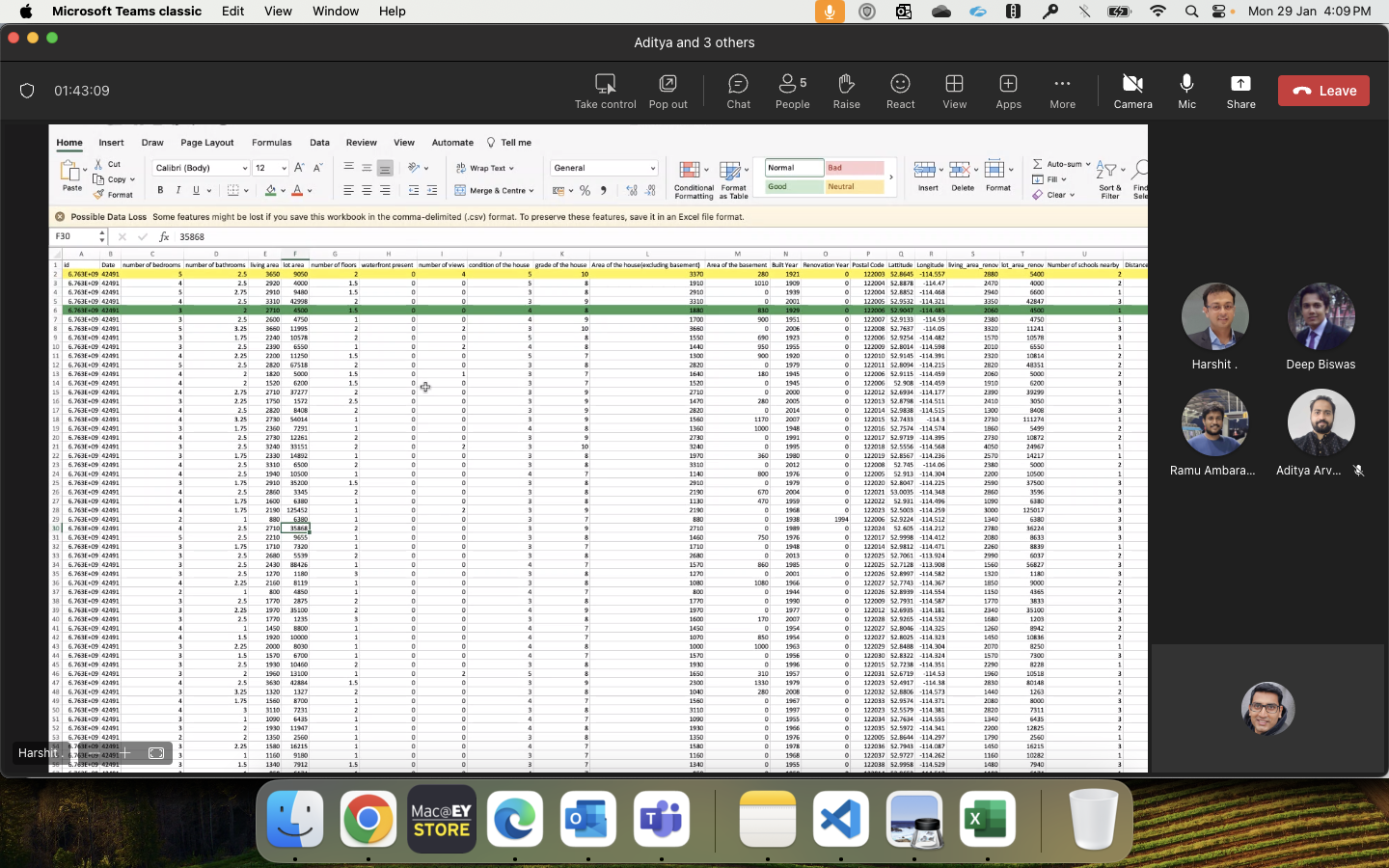
except urllib.error.HTTPError as error:

print("The request failed with status code: " + str(error.code))

# Print the headers - they include the requert ID and the timestamp, which are useful for debugging the failure

print(error.info())

print(error.read().decode("utf8", 'ignore'))

b'{"Results": [1997795.430789554]}'  
  
Data used for testing is highlighted in yellow(expected price was 2380000)  
  
  
  
  
  
GIT HUB LINK: [adi007me/ey-starks (github.com)](https://github.com/adi007me/ey-starks)