House Price Prediction

Group Members:
Khawaja Abdullah **(20k-0385)**
Sarosh Jawed **(20k-0340)**
Anas Ali **(20k-0181)**

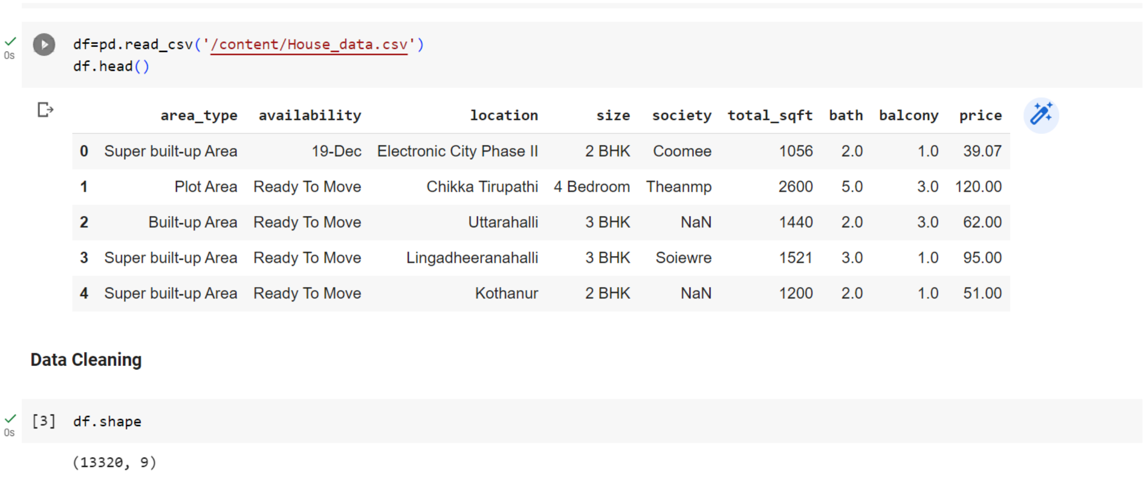
House price prediction project aims to predict house prices of a city based on previous labelled data using the features like number of bathrooms, total area in sqft,number of bedrooms and price. The project is made using python and libraries like numpy,pandas,sklearn,matplotlib and seaborn

Setting up the research goal

The project can be useful for real estate agents who can gather customer requirement of a house and then put it in the program to predict house prices of different areas . This will reduce their manual efforts of travelling and asking for prices of new houses. Some requirements were also gathered for the program like a house can not have bathrooms greater than number of rooms+2. Sqft area per room is about 250. Otherwise we will consider these data as outliers

Retrieving Data

Dataset was downloaded from kaggle. It had information like location, totalsize, bedrooms, bathrooms , price and etc. The dataset had 13320 rows initially.



```
df=pd.read_csv('/content/House_data.csv')
df.head()
```

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200	2.0	1.0	51.00

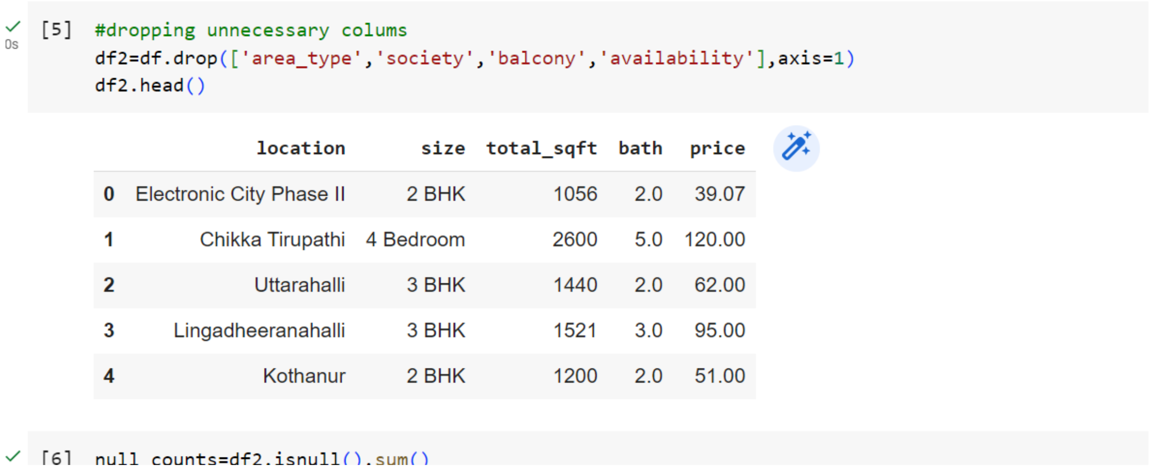
Data Cleaning

```
[3] df.shape
```

```
(13320, 9)
```

Data cleaning

We first remove the unnecessary columns

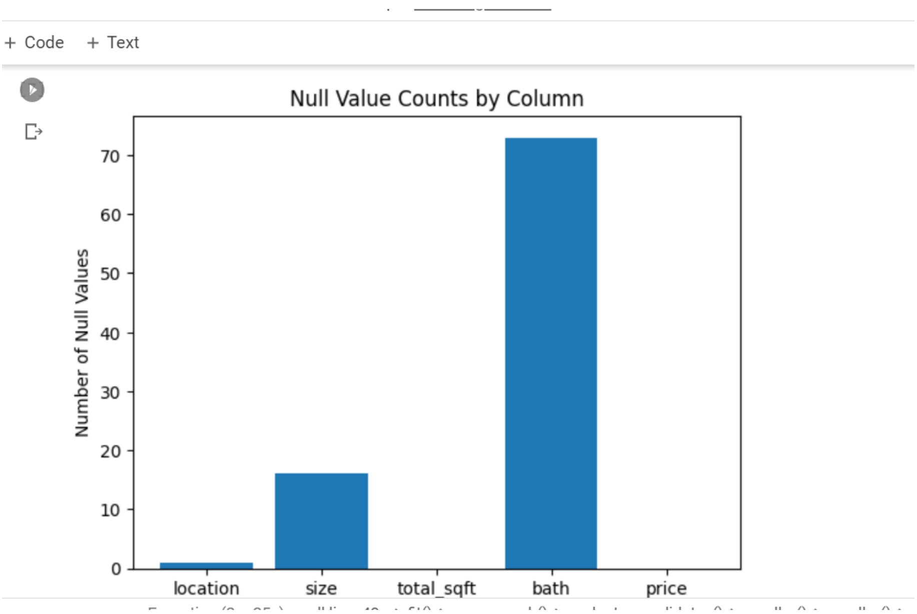


```
[5]  #dropping unnecessary colums
     df2=df.drop(['area_type','society','balcony','availability'],axis=1)
     df2.head()
```

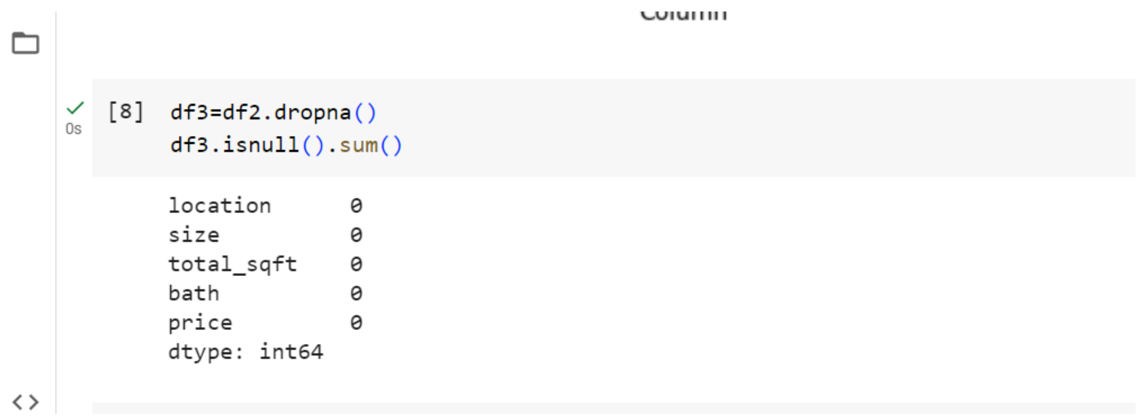
	location	size	total_sqft	bath	price
0	Electronic City Phase II	2 BHK	1056	2.0	39.07
1	Chikka Tirupathi	4 Bedroom	2600	5.0	120.00
2	Uttarahalli	3 BHK	1440	2.0	62.00
3	Lingadheeranahalli	3 BHK	1521	3.0	95.00
4	Kothanur	2 BHK	1200	2.0	51.00

```
[6]  null_counts=df2.isnull().sum()
```

We then check for null values and also visualize them.

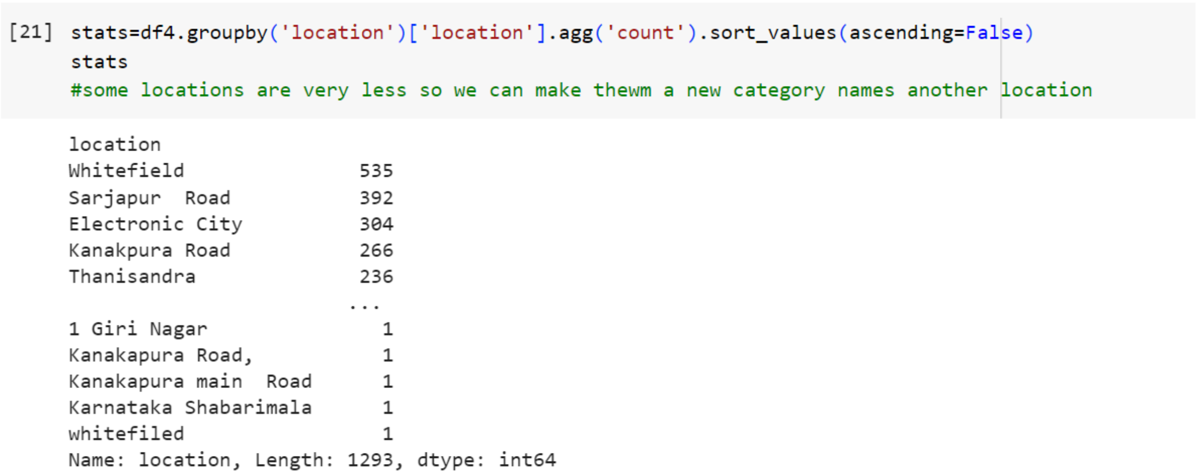


They were removed as they were not in very large number



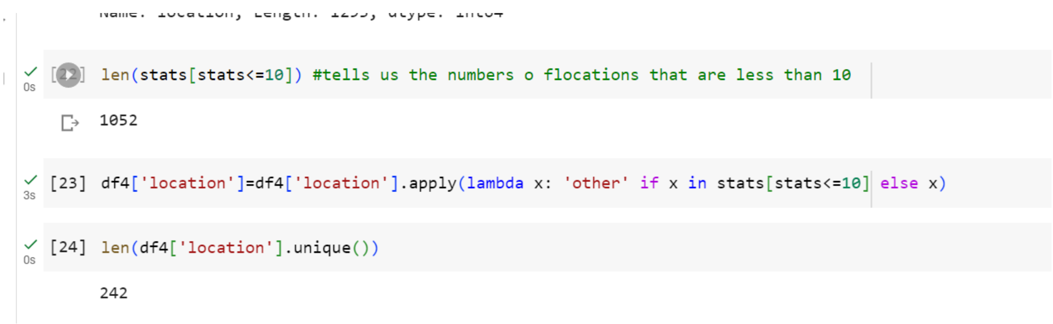
```
[8]  df3=df2.dropna()
     df3.isnull().sum()

     location      0
     size          0
     total_sqft    0
     bath          0
     price         0
     dtype: int64
```

We had losts of unique locations in location column. So we removed the ones that were in less frequency (less than 10)



```
[21] stats=df4.groupby('location')['location'].agg('count').sort_values(ascending=False)
     stats
     #some locations are very less so we can make thewm a new category names another location

     location
     Whitefield                 535
     Sarjapur  Road             392
     Electronic City            304
     Kanakpura Road             266
     Thanisandra                236
                               ...
     1 Giri Nagar                 1
     Kanakapura Road,             1
     Kanakapura main  Road        1
     Karnataka Shabarimala        1
     whitefiled                   1
     Name: location, Length: 1293, dtype: int64
```

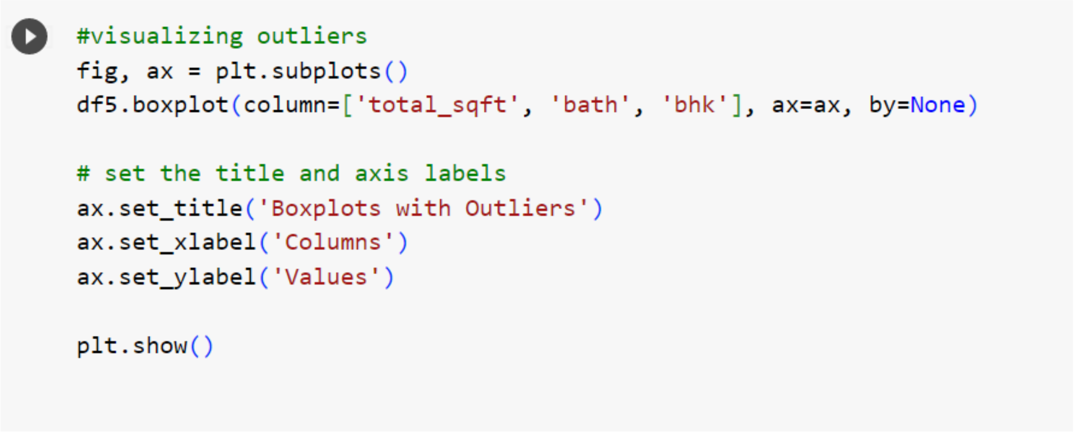
We changed them into a new category named ‘others’



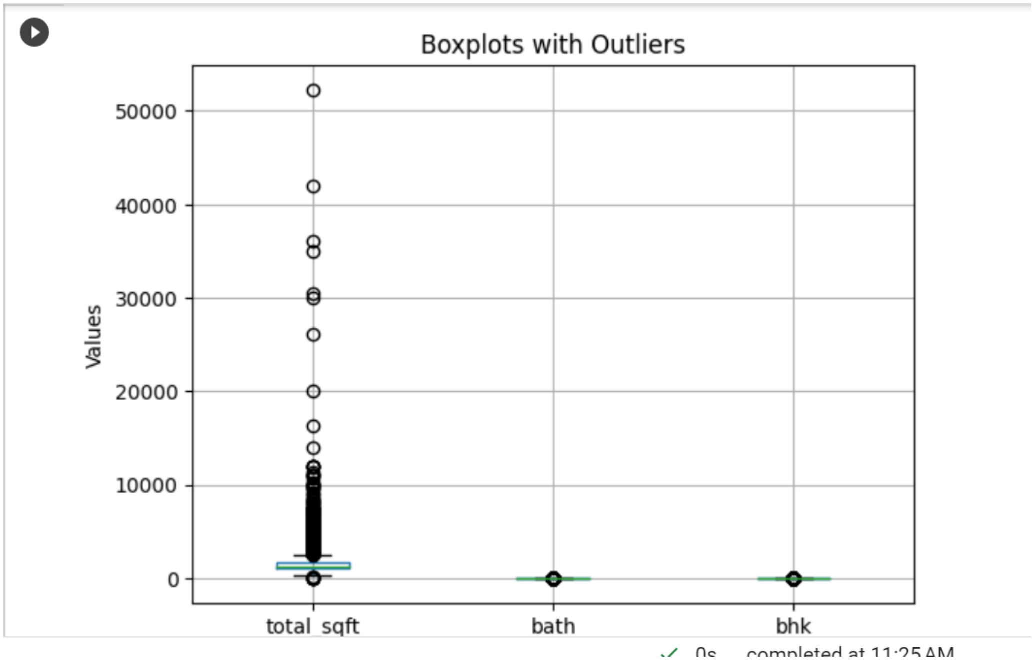
```
     len(stats[stats<=10]) #tells us the numbers o flocations that are less than 10

     1052

[23] df4['location']=df4['location'].apply(lambda x: 'other' if x in stats[stats<=10] else x)

[24] len(df4['location'].unique())

     242
```

Removing Outliers

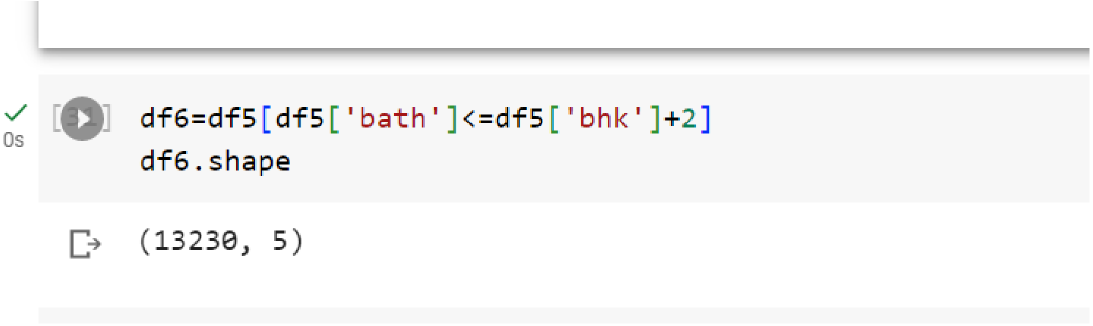
We visualize outliers using boxplot



```
#visualizing outliers
fig, ax = plt.subplots()
df5.boxplot(column=['total_sqft', 'bath', 'bhk'], ax=ax, by=None)

# set the title and axis labels
ax.set_title('Boxplots with Outliers')
ax.set_xlabel('Columns')
ax.set_ylabel('Values')

plt.show()
```

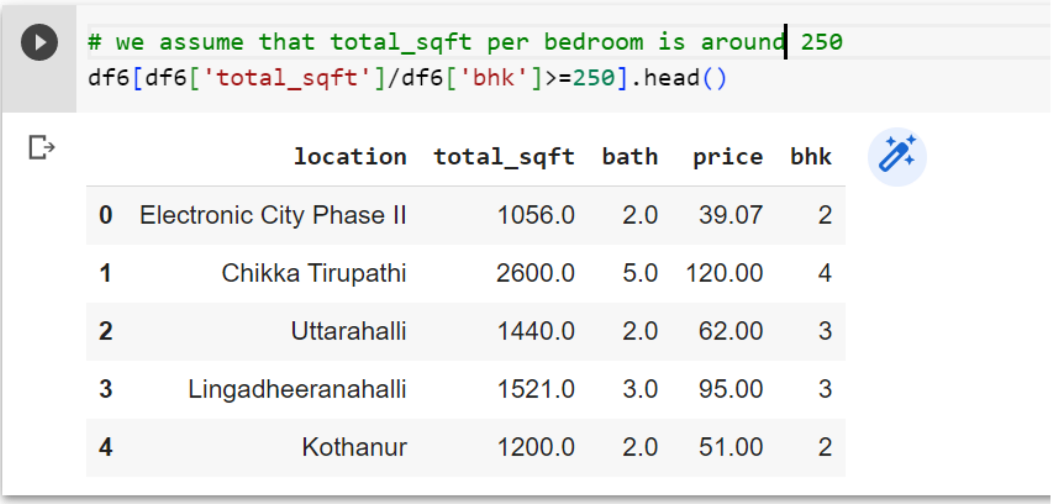


In requirement gathering part, we assumed that a house can not have bathrooms more than number of rooms+2 so we will count those as outlier which does not satisfy this requirement



```
df6=df5[df5['bath']<=df5['bhk']+2]
df6.shape
```

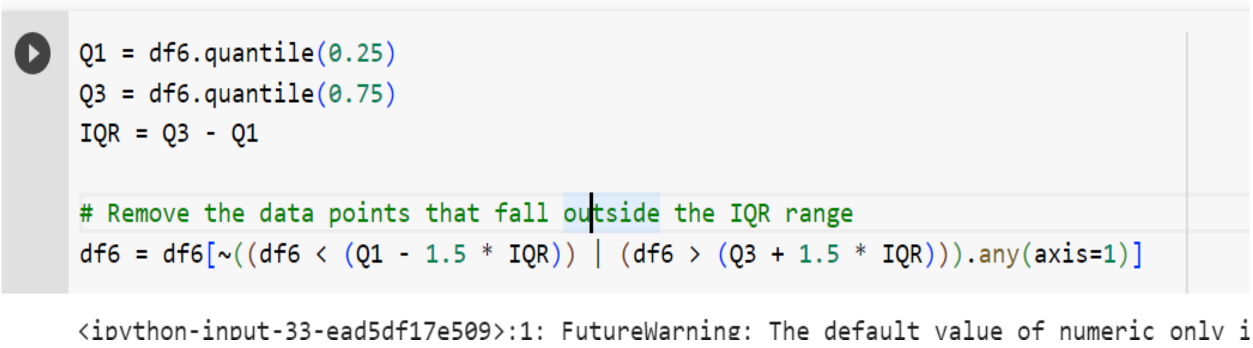
```
(13230, 5)
```



```
# we assume that total_sqft per bedroom is around 250
df6[df6['total_sqft']/df6['bhk']>=250].head()
```

	location	total_sqft	bath	price	bhk
0	Electronic City Phase II	1056.0	2.0	39.07	2
1	Chikka Tirupathi	2600.0	5.0	120.00	4
2	Uttarahalli	1440.0	2.0	62.00	3
3	Lingadheeranahalli	1521.0	3.0	95.00	3
4	Kothanur	1200.0	2.0	51.00	2

We also used Interquartile range to remove outliers in anyother column

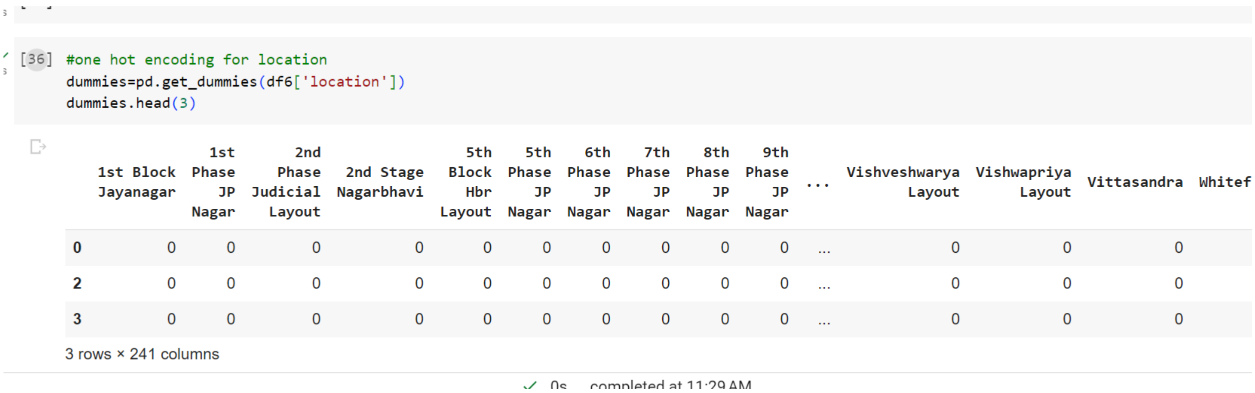


```
Q1 = df6.quantile(0.25)
Q3 = df6.quantile(0.75)
IQR = Q3 - Q1

# Remove the data points that fall outside the IQR range
df6 = df6[~((df6 < (Q1 - 1.5 * IQR)) | (df6 > (Q3 + 1.5 * IQR))).any(axis=1)]
```

```
<ipython-input-33-ead5df17e509>:1: FutureWarning: The default value of numeric only i
```

Model Building

Since we had locations column with string datatype, we did **one hot encoding**



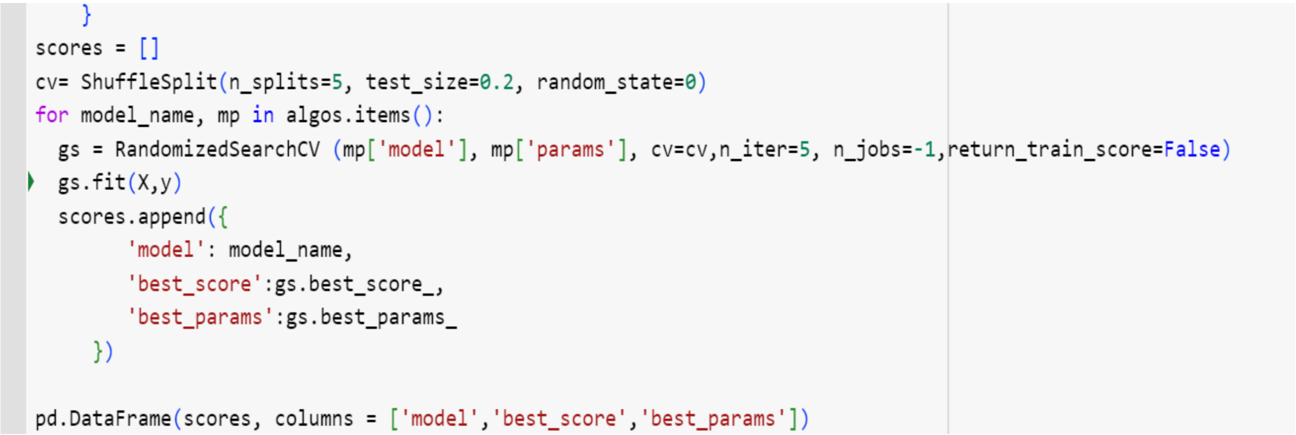
```
[36] #one hot encoding for location
     dummies=pd.get_dummies(df6['location'])
     dummies.head(3)
```

	1st Block Jayanagar	1st Phase JP Nagar	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	5th Block Hbr Layout	5th Phase JP Nagar	6th Phase JP Nagar	7th Phase JP Nagar	8th Phase JP Nagar	9th Phase JP Nagar	...	Vishveshwarya Layout	Vishwapriya Layout	Vittasandra	Whitef
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	

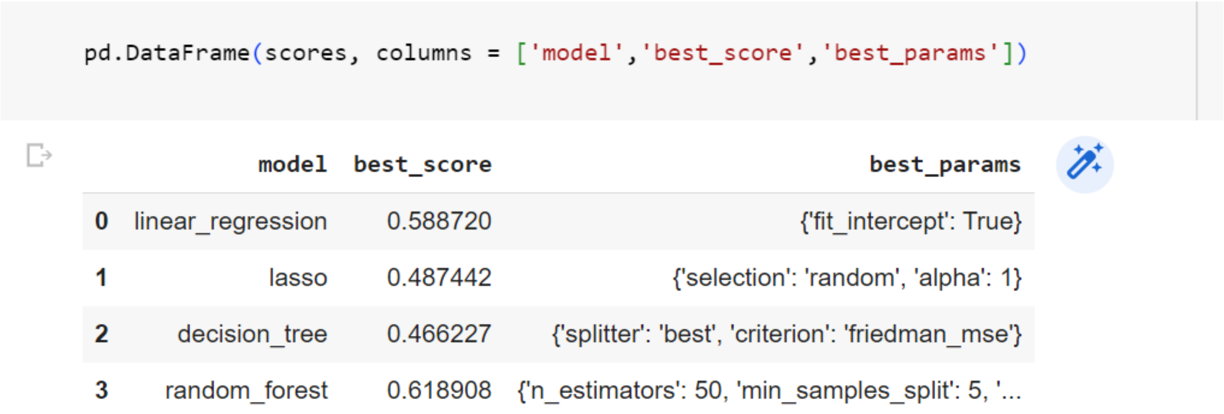
3 rows × 241 columns

✓ 0s completed at 11:29 AM

To check for best model, we did Randomized search cv on algorithms like Linear regression,Lasso regression,Decision Tree and random forest regression. The best model turned out to be Random forest regression

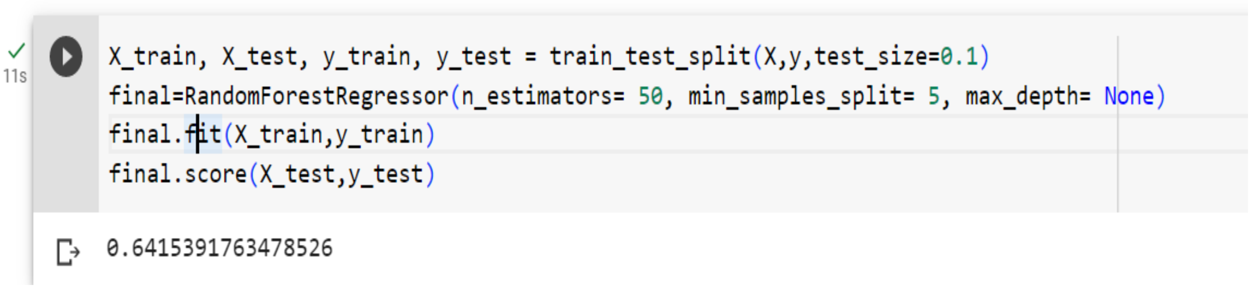


```
    }
scores = []
cv= ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)
for model_name, mp in algos.items():
  gs = RandomizedSearchCV (mp['model'], mp['params'], cv=cv,n_iter=5, n_jobs=-1,return_train_score=False)
  gs.fit(X,y)
  scores.append({
        'model': model_name,
        'best_score':gs.best_score_,
        'best_params':gs.best_params_
     })

pd.DataFrame(scores, columns = ['model','best_score','best_params'])
```



```
pd.DataFrame(scores, columns = ['model','best_score','best_params'])
```

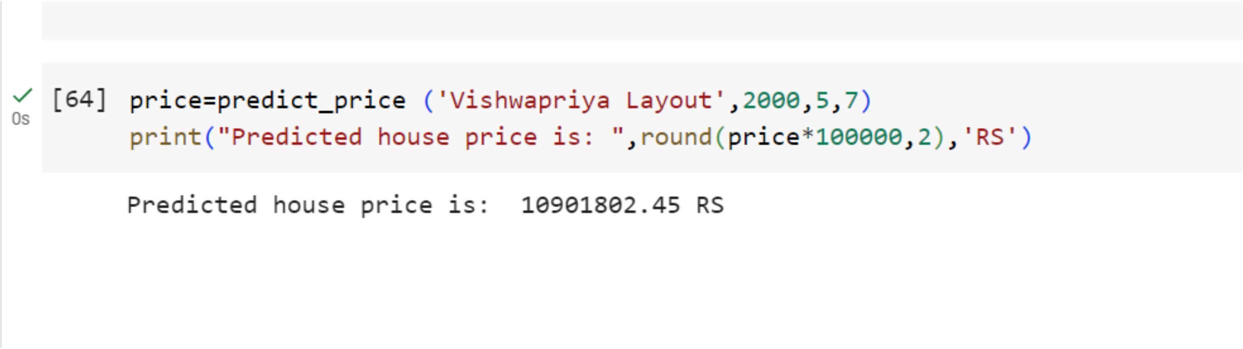
	model	best_score	best_params
0	linear_regression	0.588720	{'fit_intercept': True}
1	lasso	0.487442	{'selection': 'random', 'alpha': 1}
2	decision_tree	0.466227	{'splitter': 'best', 'criterion': 'friedman_mse'}
3	random_forest	0.618908	{'n_estimators': 50, 'min_samples_split': 5, '...



```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.1)
final=RandomForestRegressor(n_estimators= 50, min_samples_split= 5, max_depth= None)
final.fit(X_train,y_train)
final.score(X_test,y_test)
```

```
0.6415391763478526
```

Results



```
[64] price=predict_price ('Vishwapriya Layout',2000,5,7)
     print("Predicted house price is: ",round(price*100000,2),'RS')
```

```
Predicted house price is:  10901802.45 RS
```