Housing Data Predictions Linear Regression and KNN

Bilal Ullah

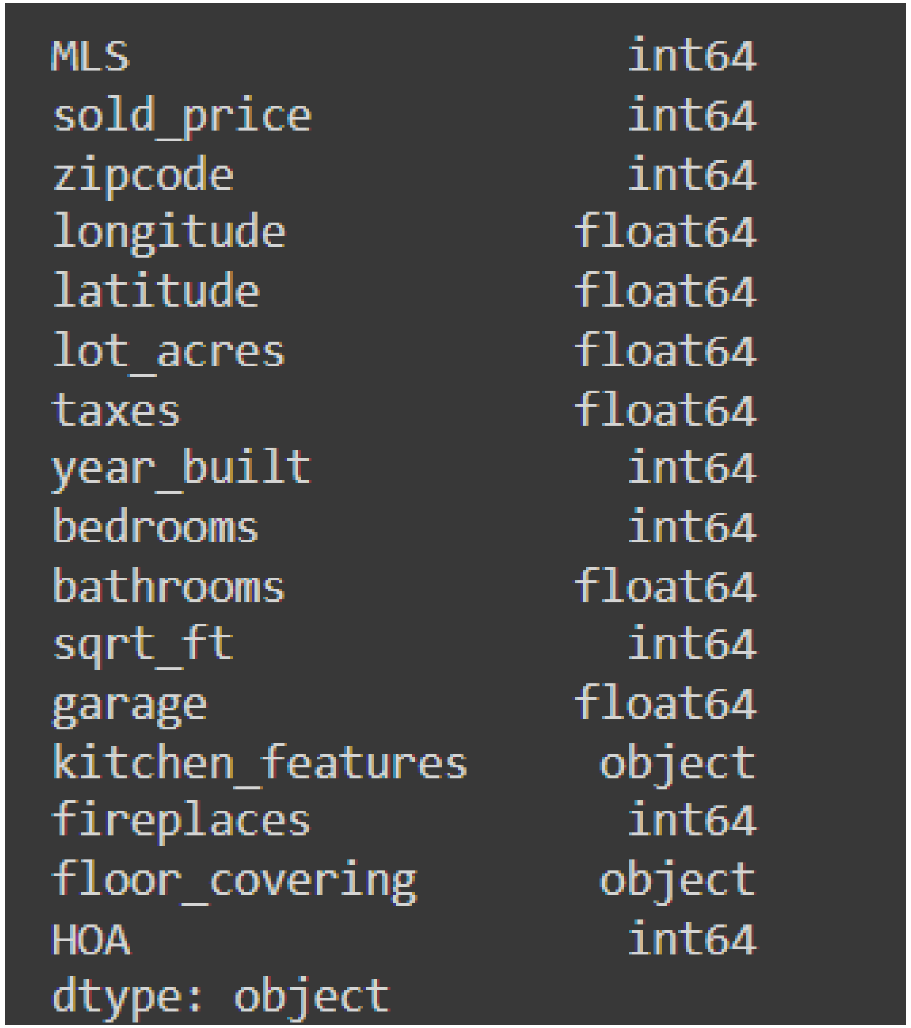
Use Case

- Property developers in Arizona are having trouble evaluating house prices
- I have been tasked by investors to accurately predict house prices so they can determine whether they should invest in that area or not

Dataset

Quick look at the features provided with the dataset

Decision to be made on what is useful to the KNN and Regression models

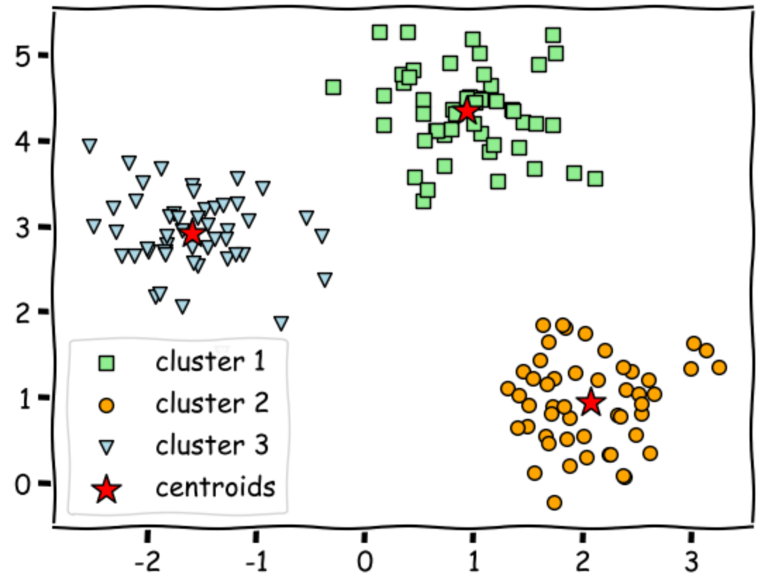


```
MLS                    int64
sold_price             int64
zipcode                int64
longitude            float64
latitude             float64
lot_acres            float64
taxes                float64
year_built             int64
bedrooms               int64
bathrooms            float64
sqrt_ft                int64
garage               float64
kitchen_features      object
fireplaces             int64
floor_covering        object
HOA                    int64
dtype: object
```

KNN Classification

Model Reasoning:

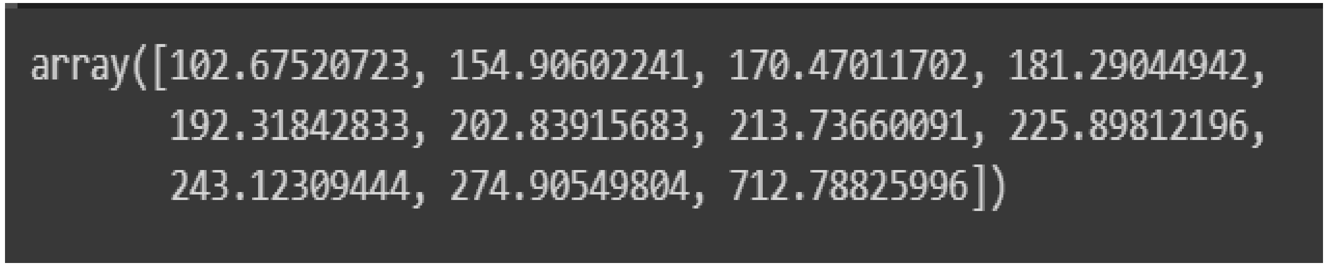
- Dealing with locations
- Best to split location data into categories to deal with continuous values of longitude and latitude
- This grouping will allow for higher accuracy when combined with a regression model



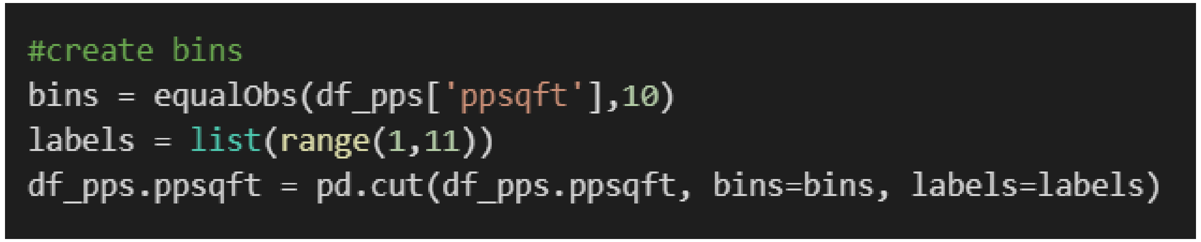
Creating bins

Bin Selection:

- Creating bins based on number of values



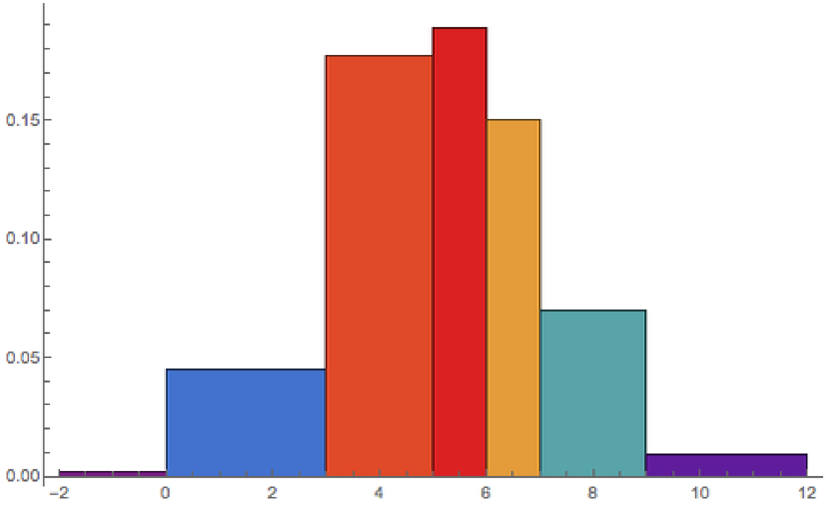
```
array([102.67520723, 154.90602241, 170.47011702, 181.29044942,
       192.31842833, 202.83915683, 213.73660091, 225.89812196,
       243.12309444, 274.90549804, 712.78825996])
```



```
#create bins
bins = equalObs(df_pps['ppsqft'],10)
labels = list(range(1,11))
df_pps.ppsqft = pd.cut(df_pps.ppsqft, bins=bins, labels=labels)
```

Bin Values

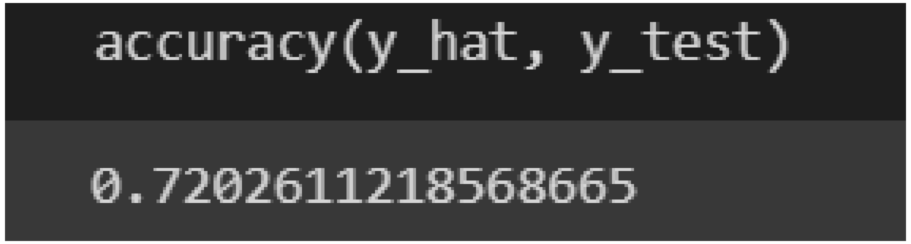
I decided to go with n(obv) over equal bin lengths as the clustering would be more closely related to the location



KNN Classifier Results

72% of the results were accurate

Important to not there are 10 bins were many could have been only 1 bin off

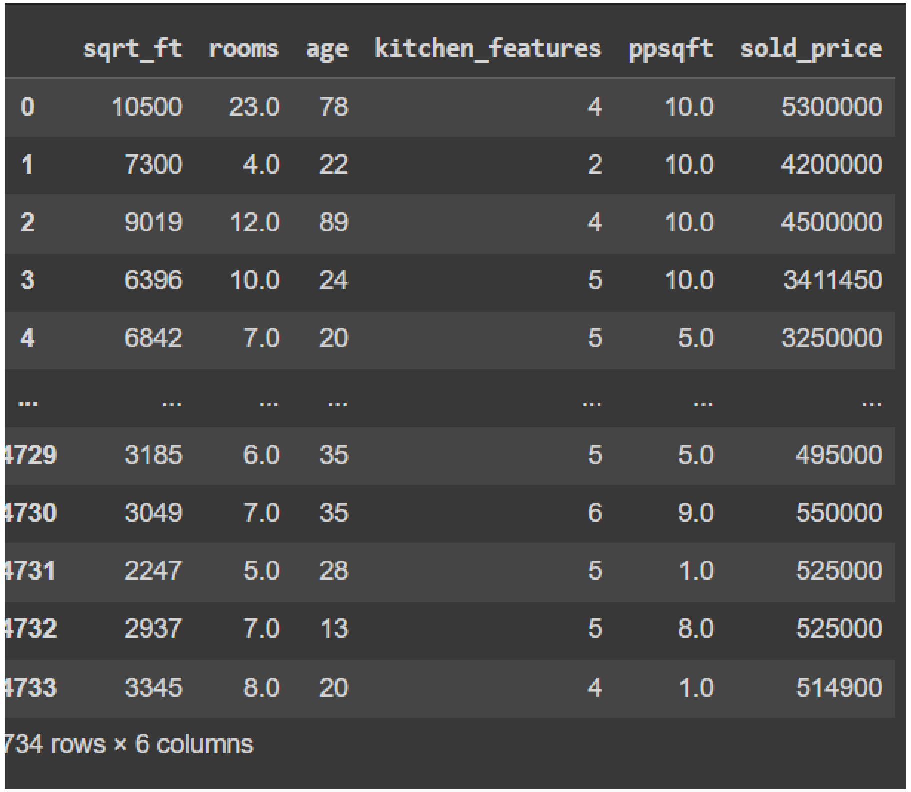


```
accuracy(y_hat, y_test)

0.7202611218568665
```

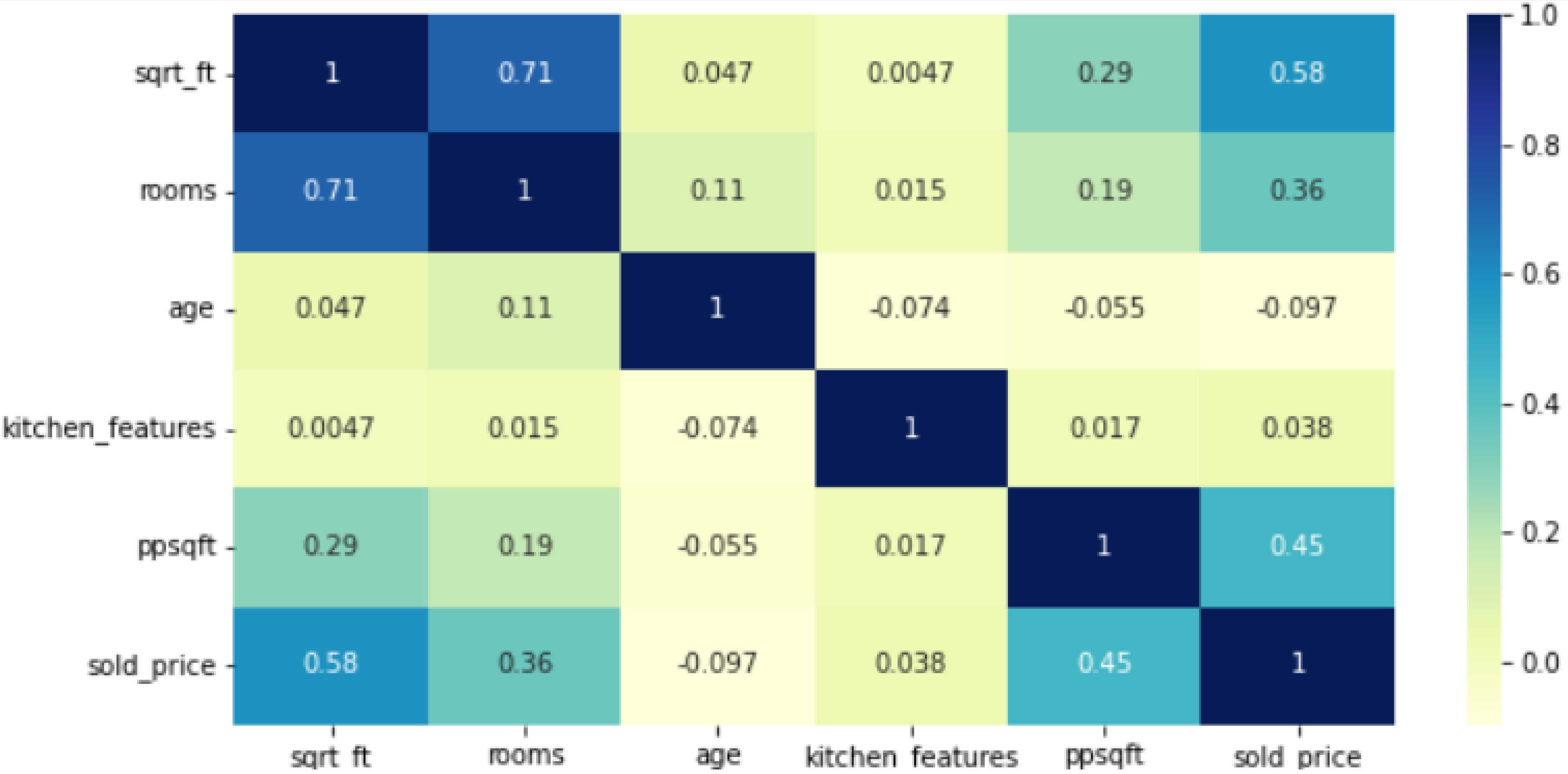
Feature Selection & Regression model

Features selected based on relationship to use case and correlation



	sqrt_ft	rooms	age	kitchen_features	ppsqft	sold_price
0	10500	23.0	78	4	10.0	5300000
1	7300	4.0	22	2	10.0	4200000
2	9019	12.0	89	4	10.0	4500000
3	6396	10.0	24	5	10.0	3411450
4	6842	7.0	20	5	5.0	3250000
...	...	...	...	...	...	...
4729	3185	6.0	35	5	5.0	495000
4730	3049	7.0	35	6	9.0	550000
4731	2247	5.0	28	5	1.0	525000
4732	2937	7.0	13	5	8.0	525000
4733	3345	8.0	20	4	1.0	514900

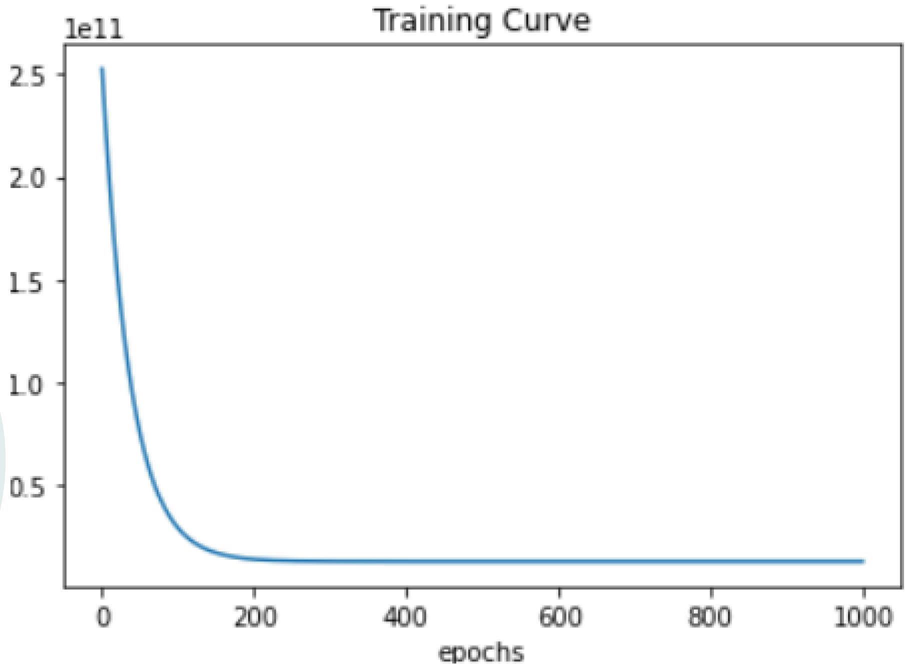
734 rows × 6 columns


sqrt_ft
rooms
age
kitchen_features
ppsqft
sold_price
1
0.71
0.047
0.0047
0.29
0.58
0.71
1
0.11
0.015
0.19
0.36
0.047
0.11
1
-0.074
-0.055
-0.097
0.0047
0.015
-0.074
1
0.017
0.038
0.29
0.19
-0.055
0.017
1
0.45
0.58
0.36
-0.097
0.038
0.45
1
sqrt ft
rooms
age
kitchen features
ppsqft
sold price
1.0
0.8
0.6
0.4
0.2
0.0

MultiVariate Regression

The learning curve shows epochs on the x-axis and learning or improvement on the y-axis

We ca see the gradual improvement and as the curve tends towards 0

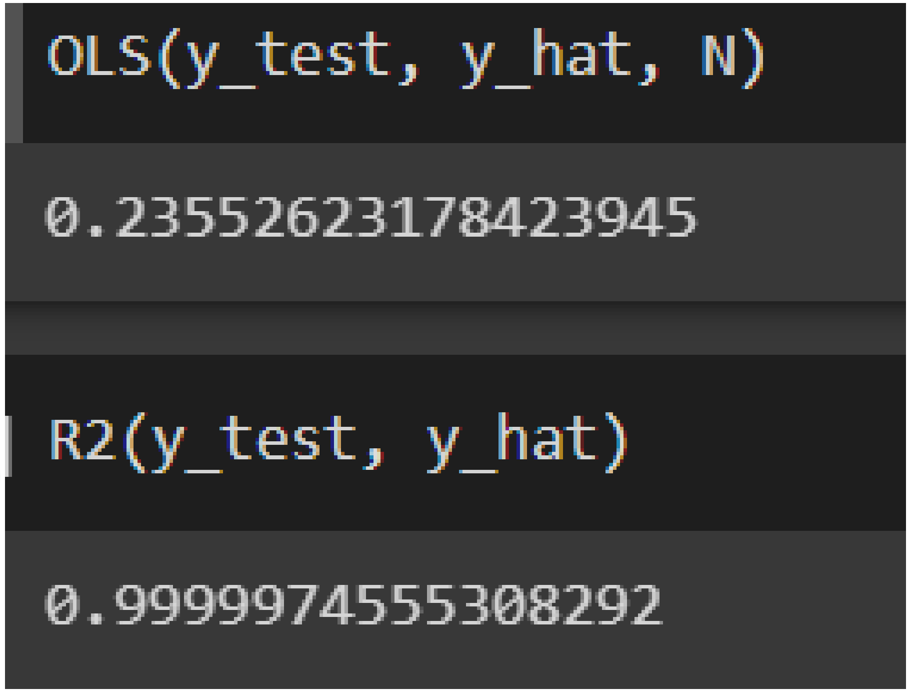


Model Results

Evaluation indicate positive results from the model

And error in regression seems to be low from looking at the OLS

And the R2 shows the data is very strongly related to the results



```
OLS(y_test, y_hat, N)
```

```
0.23552623178423945
```

```
R2(y_test, y_hat)
```

```
0.9999974555308292
```

Summary

- Model is able to successfully classify into bins based on sqft and price sold using KNN
- These predictions are then applied in the MV Regression
  - Feature selection
  - Trained --> Tested
  - Positive Results
- Now the model can be applied for the investors and their use case, accurate predictions of price should be able to be made now based on a few features
- LIVE DEMO!!