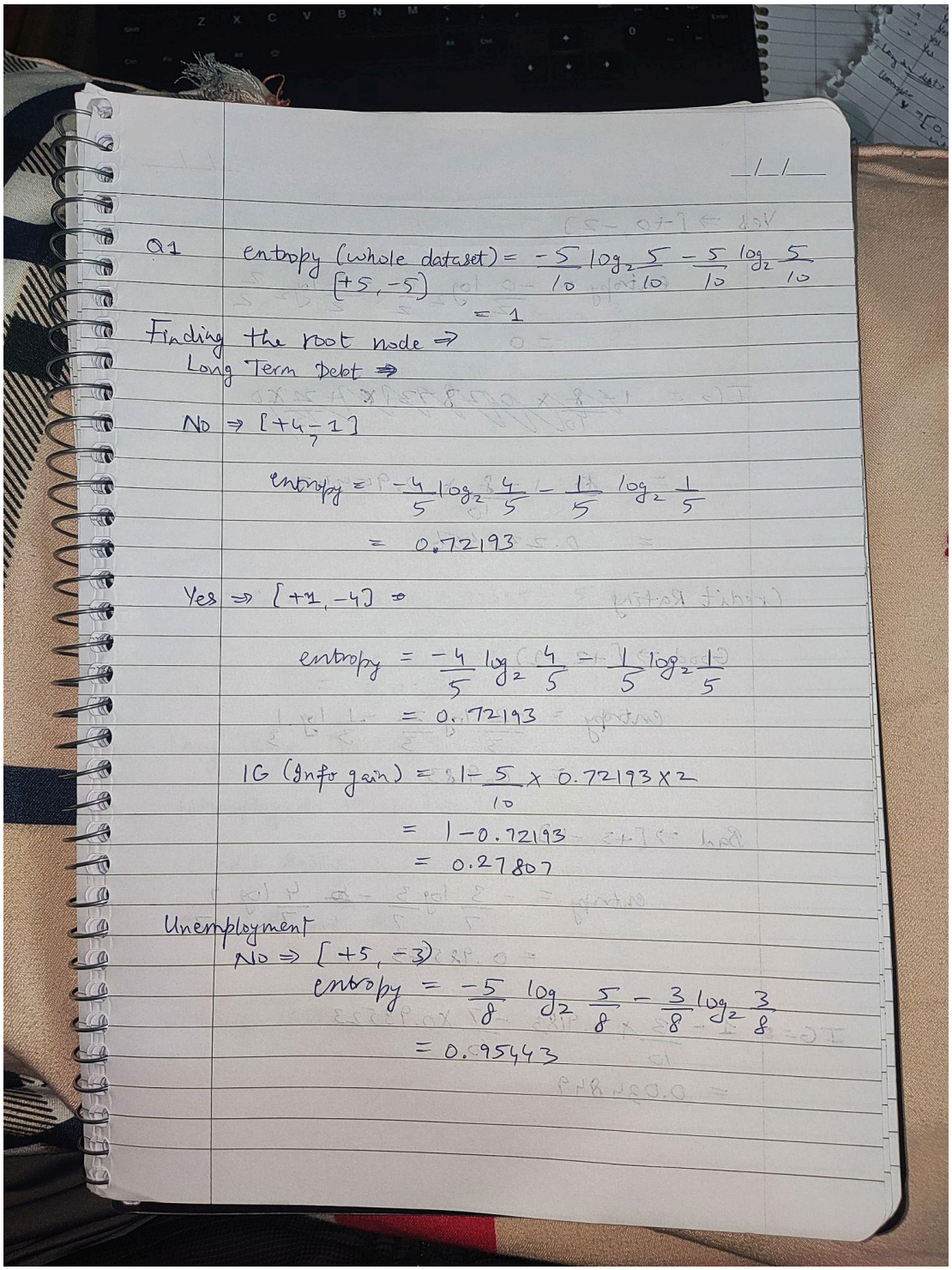
AI ASSIGNMENT 4 — Machine Learning

Saumil Lakra, 2021097

THEORY



Q1 entropy (whole dataset) = $\frac{-5}{10}\log_2\frac{5}{10} - \frac{5}{10}\log_2\frac{5}{10}$
[+5, -5]
= 1

Finding the root node ⇒
Long Term Debt ⇒

No ⇒ [+4, -1]

entropy = $\frac{-4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5}$
= 0.72193

Yes ⇒ [+1, -4] ∞

entropy = $\frac{-4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5}$
= 0.72193

IG (Info gain) = $1 - \frac{5}{10} \times 0.72193 \times 2$
= 1 − 0.72193
= 0.27807

Unemployment
No ⇒ [+5, -3]
entropy = $\frac{-5}{8}\log_2\frac{5}{8} - \frac{3}{8}\log_2\frac{3}{8}$
= 0.9543

Yes ⇒ [+0, −2]

$$\text{entropy} = -\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2}$$

$$= 0$$

IG = ~~$1 - \frac{8}{10} \times 0.95443 - \frac{2}{2} \times 0$~~

$$= 1 - \frac{8}{10} \times 0.95443$$

$$= 0.236456$$

Credit Rating

Good ⇒ [+2, −1]

$$\text{entropy} = -\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3}$$

$$= 0.9183$$

Bad ⇒ [+3, −4]

$$\text{entropy} = -\frac{3}{7}\log\frac{3}{7} - \frac{4}{7}\log\frac{4}{7}$$

$$= 0.98523$$

$$IG = 1 - \frac{3}{10} \times 0.9183 - \frac{7}{10} \times 0.98523$$

$$= 0.034849$$

Down Payment

Yes ⇒ [+2, -3]

$$\text{Entropy} = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}$$

$$= 0.97095$$

No ⇒ [+3, -2]

$$\text{entropy} = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}$$

$$= 0.97095$$

$$IG = 1 - \frac{5}{10} \times 0.97095 - \frac{5}{10} \times 0.97095$$

$$= 1 - 0.97095$$

$$= 0.02905$$

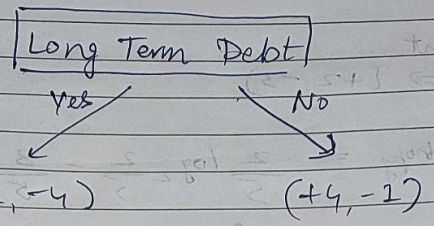
Long Term Debt ⇒ 0.27807

Unemployed ⇒ 0.236456

Credit Rating ⇒ 0.034849

Down Payment ⇒ 0.02905

⇒ Long Term Debt is the root Node.



For Long Term Debt = Yes.

Credit rating ⇒
Good ⇒ [+1, 0]
entropy = 0

Bad ⇒ [0, -4)
entropy = 0

IG = entropy (Long Term Debt = Yes) − 0 − 0

~~entropy (~~

⇒ entropy (Long Term Debt = Yes) =

(+1, -4)

$$\text{entropy} = -\frac{1}{5}\log_2\frac{1}{5} - \frac{4}{5}\log_2\frac{4}{5}$$

$$= 0.72193$$

$$\Rightarrow IG = 0.72193 - 0 - 0$$

$$= 0.72193$$

Unemployed

Y ⇒ [0, -1]

entropy = 0

N ⇒ [+1, -3]

$$\text{entropy} = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4}$$

$$= 0.81128$$

$$IG = 0.72193 - 0 - \frac{4}{5} \times 0.81128$$

$$= 0.072906.$$

Down Payment ⇒

Yes = [0, -2]

entropy = 0

No = [+1, -2]

$$\text{entropy} = -\frac{1}{3}\log\frac{1}{3} - \frac{2}{3}\log\frac{2}{3}$$

$$= 0.9183$$

$$IG = 0.72193 - \frac{3}{5} \times 0.9183$$

$$= 0.17095$$

__/__/__

⇒ With Long Term Debt as root node & equal to Yes.

Credit ration = 0.72193

Unemployed = 0.072906

Down Payment = 0.17095

⇒ credit ratio

⇒ Long Term Debt

Yes → Credit Ratio

No

For Long Term Debt = No

Entropy (Long Term Debt = No)

$$= -\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5}$$

$$= 0.72193$$

Unemployment

N = [74, 0]

entropy = 0

Y ⇒ [0, 1] entropy = 0

$IG = 0.72193 - 0 - 0$

$= 0.72193$

Credit rating

Good $\Rightarrow [+1, -1)$

entropy $= 1$

Bad $\Rightarrow [+3, 0)$

entropy $= 0$

$IG = 0.72193 - \frac{2}{5} \times 1$

$= 0.32193$

Down Payment

Y $\Rightarrow [+2, -1)$

entropy $= -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}$

$= 0.9183$

N $\Rightarrow [+2, 0)$

entropy $= 0$

$IG = 0.72193 - \frac{3}{5} \times 0.9183$

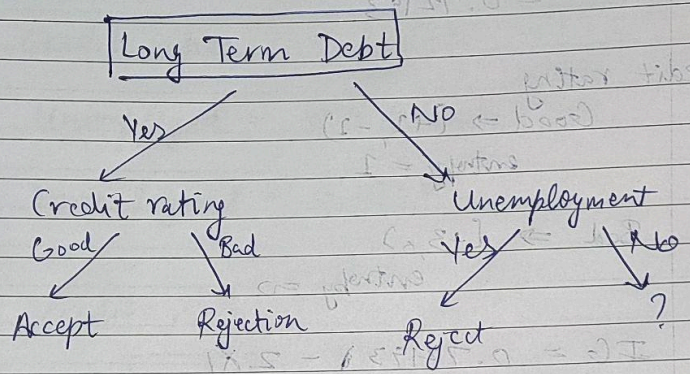
$= 0.17095$

unemployment $= 0.72193$

Credit Rating $= 0.32193$

Down Payment $= 0.17095$

⇒ Tree is



For LTD = No, Unemployment = No

~~Dependency~~ Down Payment

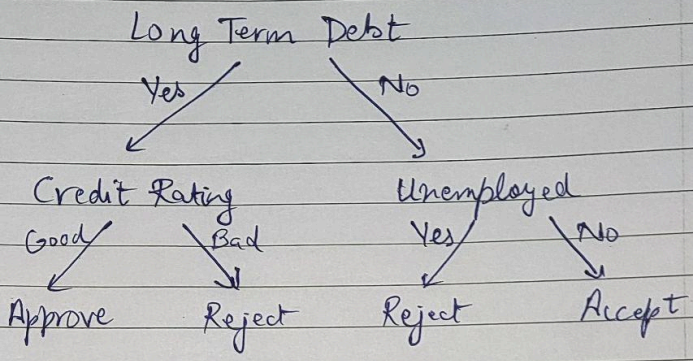
Y = [2,0) ⇒ entropy = 0

N ⇒ [3,0] ⇒ entropy = 0

Entropy (LTD = No, Unemployment = No) = [+4, -0)
⇒ entropy = 0

IG = 0 - 0 - 0
= 0

⇒ Final Tree



Training Error = $\frac{\text{Total misclassification}}{\text{Total instances}}$

$= \frac{0}{10} = 0$

2. Data Preprocessing and Exploratory Data Analysis

1. Task 1: Understanding the Dataset:

- Unique values in the train_test dataset

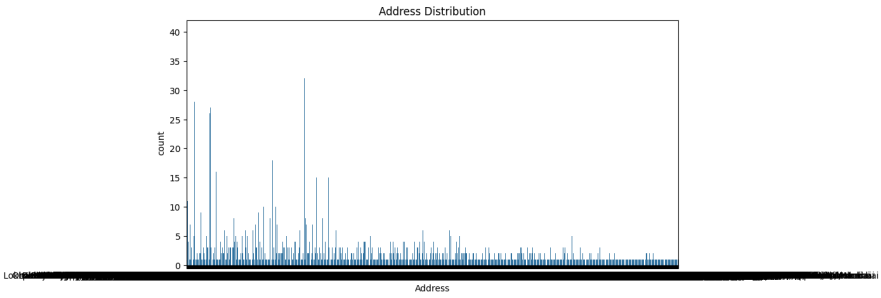
```
df_merge.nunique()
✓ 0.0s
```

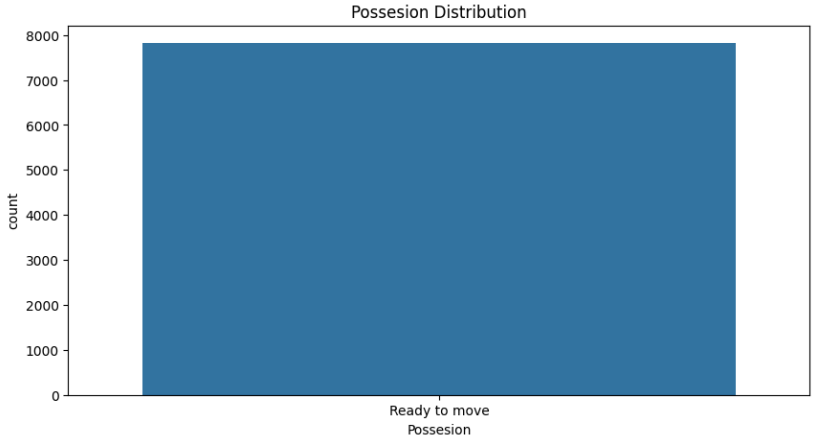
Address	3725
Possesion	1
Furnishing	3
Buildup_area	1038
Carpet_area	2893
Bathrooms	104
Property_age	46
Parking	10
Price	832
Brokerage	1785
Floor	132
Per_sqft_price	2801
BHK	9
Total_bedrooms	32

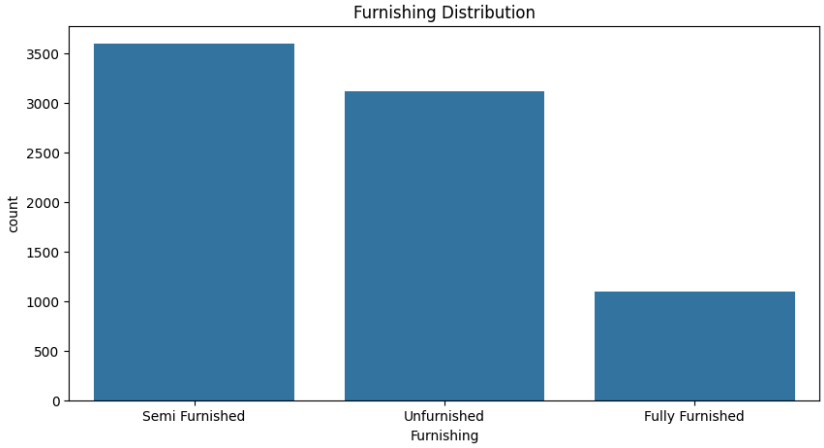
dtype: int64

Categorical Features: ['Address', 'Possesion', 'Furnishing']
Numerical Features: ['Buildup_area', 'Carpet_area', 'Bathrooms', 'Property_age', 'Parking', 'Price', 'Brokerage', 'Floor', 'Per_sqft_price', 'BHK', 'Total_bedrooms']

Count plots of categorical features:



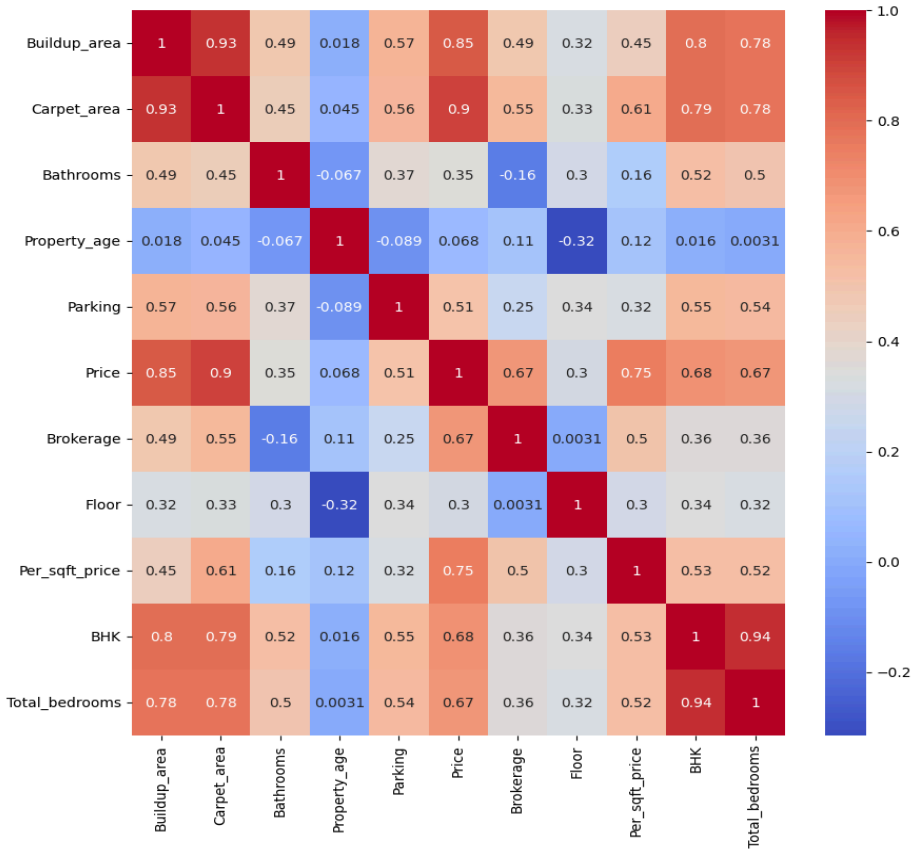




Statistical analysis of the numerical features:

	Buildup_area	Carpet_area	Bathrooms	Property_age	Parking	Price	Brokerage	Floor	Per_sqft_price	BHK	Total_bedrooms
count	7820.000000	7820.000000	7820.00000	7820.000000	7820.000000	7.820000e+03	7.820000e+03	7820.000000	7820.000000	7820.000000	7820.000000
mean	1116.095505	862.095282	1.97366	7.471611	1.303581	3.038559e+07	1.131909e+07	19.930279	23401.713972	2.154923	2.201048
std	722.222240	573.311089	0.90052	7.217703	0.797048	3.719014e+07	3.102861e+07	13.960965	13000.580530	0.999940	0.979875
min	180.000000	150.000000	1.00000	1.000000	0.000000	7.800000e+05	0.000000e+00	2.000000	1440.000000	1.000000	1.000000
25%	650.000000	473.881582	1.00000	2.000000	1.000000	1.050000e+07	9.999900e+04	10.000000	15600.000000	1.000000	1.000000
50%	943.500000	707.722575	2.00000	5.000000	1.000000	1.920000e+07	2.500000e+05	16.000000	21430.000000	2.000000	2.000000
75%	1322.000000	1050.000000	2.00000	10.000000	2.000000	3.500000e+07	1.070000e+07	23.000000	28850.000000	3.000000	3.000000
max	15000.000000	14000.000000	10.00000	99.000000	9.000000	5.000000e+08	5.000000e+08	99.000000	100000.000000	10.000000	10.000000

Correlation Heatmap:



2. Task 2: Drop Irrelevant Columns:

- Dropped column Possession because every entry is ‘Ready to move’.
- Also dropped Property_age because it doesn’t follow given correlation coefficient constraints.

3. Task 3: Encoding Categorical Features :

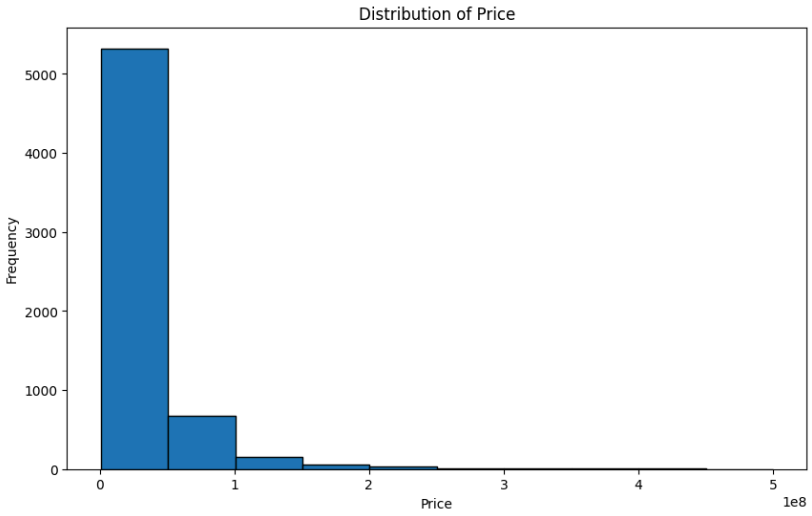
High cardinality in a dataset means that there are high count of unique values in the dataset. This can increase the computational complexity of the encoding because each unique entry will have its own unique labeling. More unique labels will be the dimension of the label. This can increase the computational complexity of the whole problem. This problem can be solved by grouping the related features that are related(in the same column) together. This can reduce the dimension of the labeling and further takes less computation.

4. Task 4: Feature Scaling:

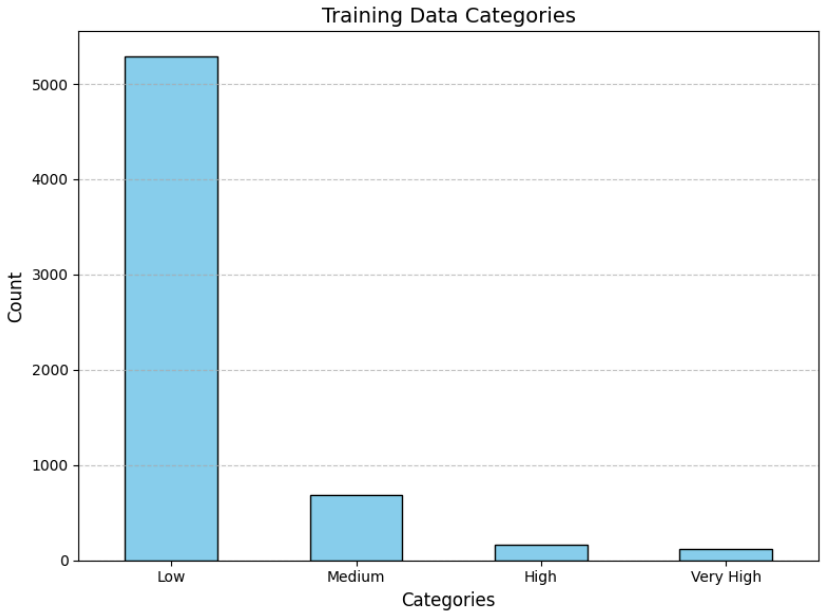
```
Decision Tree Regressor Results for Random Undersampling on scaled data
Mean Squared Error:  76416474593350.39
Mean Absolute Error:  4183704.603580563
R2 Score:  0.9345900212423094

Decision Tree Regressor Results for Random Oversampling on scaled data
Mean Squared Error:  8602279012787.724
Mean Absolute Error:  994694.3734015345
R2 Score:  0.9926367332373229

Decision Tree Regressor Results for Random Undersampling on unscaled data
Mean Squared Error:  45142072249360.62
Mean Absolute Error:  3854152.1739130435
R2 Score:  0.9613598768770504

Decision Tree Regressor Results for Random Oversampling on unscaled data
Mean Squared Error:  9433163941176.47
Mean Absolute Error:  1009392.5831202046
R2 Score:  0.9919255231768588
```

Scaled data perform marginally better for oversampling but the impact of scaling is not significant. For random undersampling, unscaled data performs better because the complexity of the dataset is very less

5. Task 5: Target Variable Imbalance Detection:




```
ranges = [0, 0.5e8, 1e8, 1.5e8, np.inf]
```



Prices that lie in the low category are highest and their range is [0,0.5e8]. The subsequent categories have far less count than the low segment. Medium

category is even less than 1000. High and very high categories are somewhat equal.

6. Task 6: Handling Imbalanced Data:

Data imbalance occurs when one class in a dataset has significantly more samples than the other classes. This can lead to the model being biased towards the major class and give poor results on the minor classes.

Random Undersampling:
Benefits: Reduces the dataset size since the majority class has been reduced. This can lead to a smaller dataset and thus can result in faster training times and low memory usage. The model can now be unbias while learning the patterns in the dataset.

Limitations:
Since the classes are being removed in this sampling method, it can also remove classes which could have been useful during the training. There is a loss of information. Smaller dataset can also lead to overfitting of the model on the train set and poor performance on the test set.

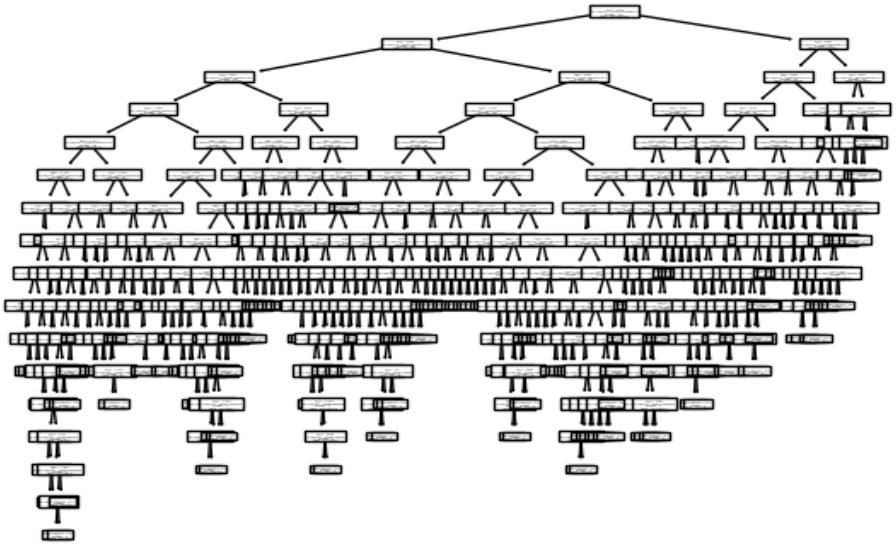
Random Oversampling:
Benefits: There is no loss of information since no data is being removed from the training data. Now since the dataset is larger, the model can learn better than it could have when dataset is smaller.

Limitations:
Replication of data can cause the model to overfit on the minority class and perform poorly on the test data. Since in this method there is an increase in the size of the dataset, there can be an increase in the computation and training time.

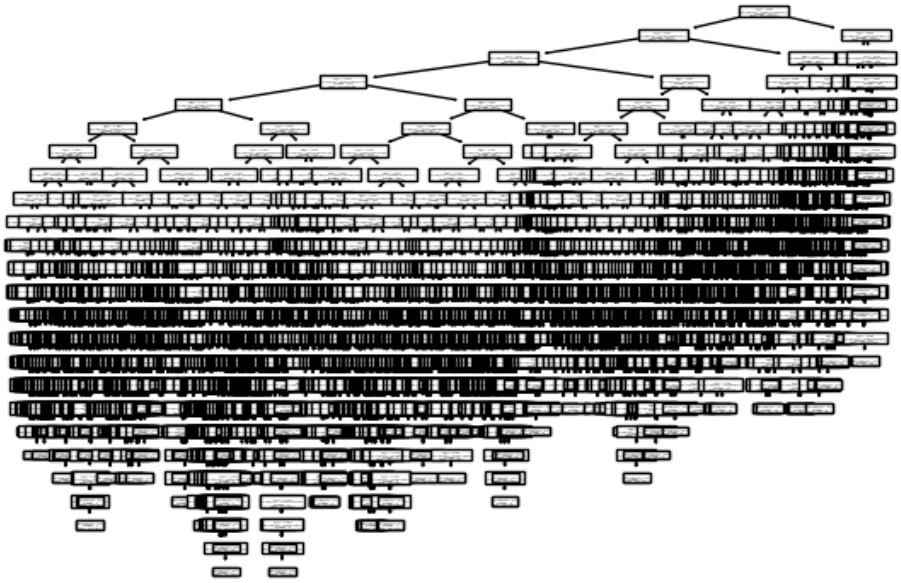
3. Building Decision Tree Model

1. Task 1: Model Training:

Scaled and undersampled data decision tree:



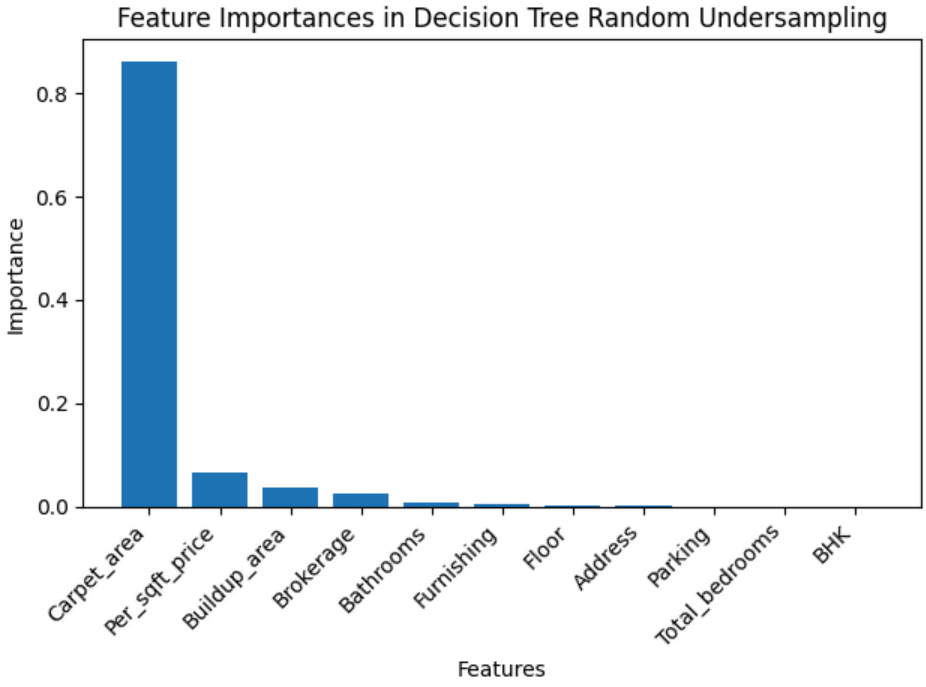
Scaled and oversampled data decision tree:

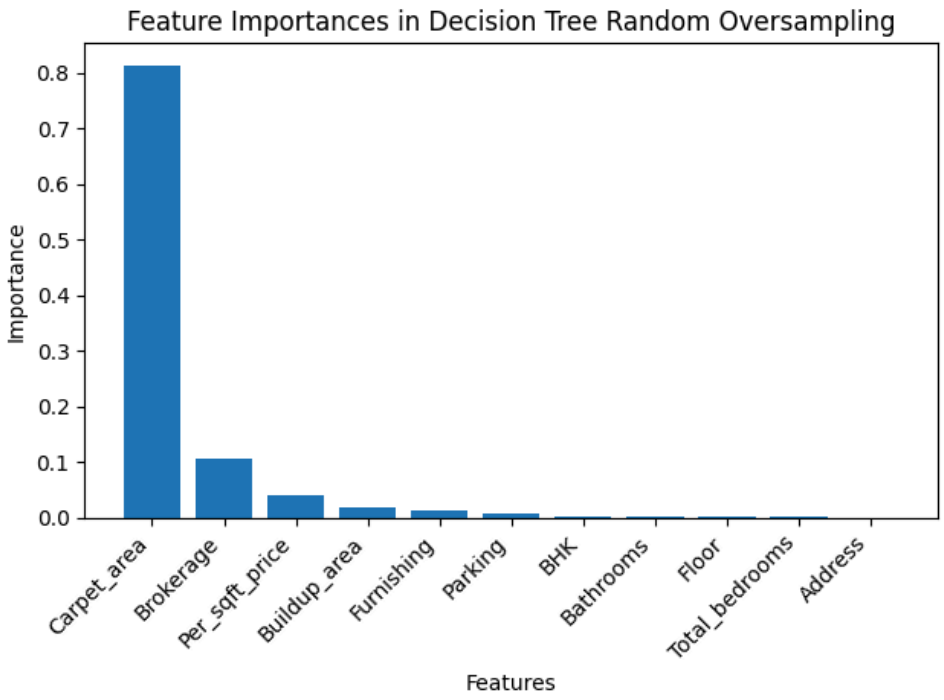


Since the oversampled data has a larger size, thus the model now has more options to make splits and thus it is more dense than the undersampled dataset. With more data, the tree is able to capture more information about the dataset. I have used the default argument of the decision tree regressor function.

2. Task 2: Feature Importance and Hyperparameter Tuning:

(a)





(b) Certain features are more important because they share high correlation with the target feature(Price). The variance in their values can most influence the target feature. Yes, it matches my expectations because the price of property largely depends on the carpet_area, brokerage and per_sqft_price.

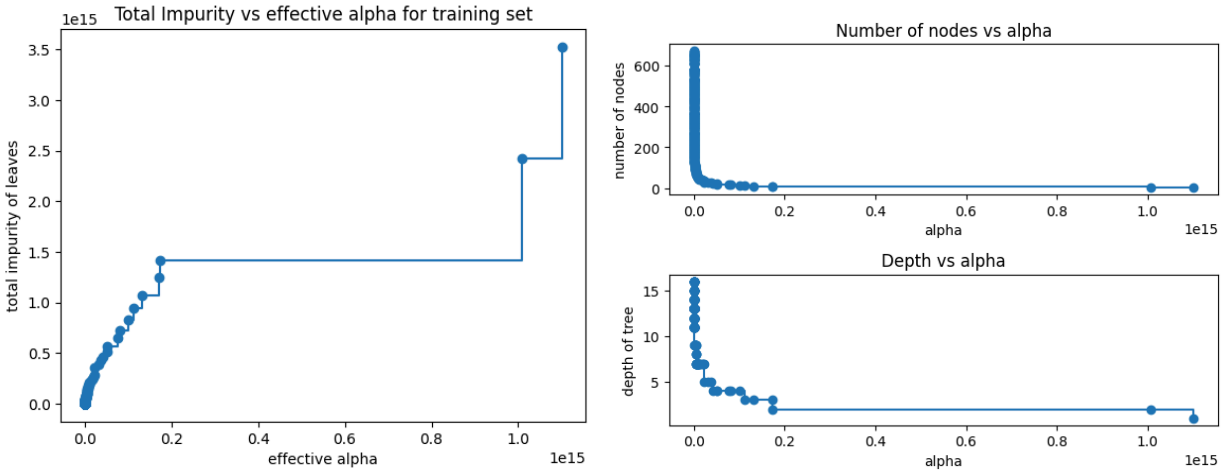
(c)

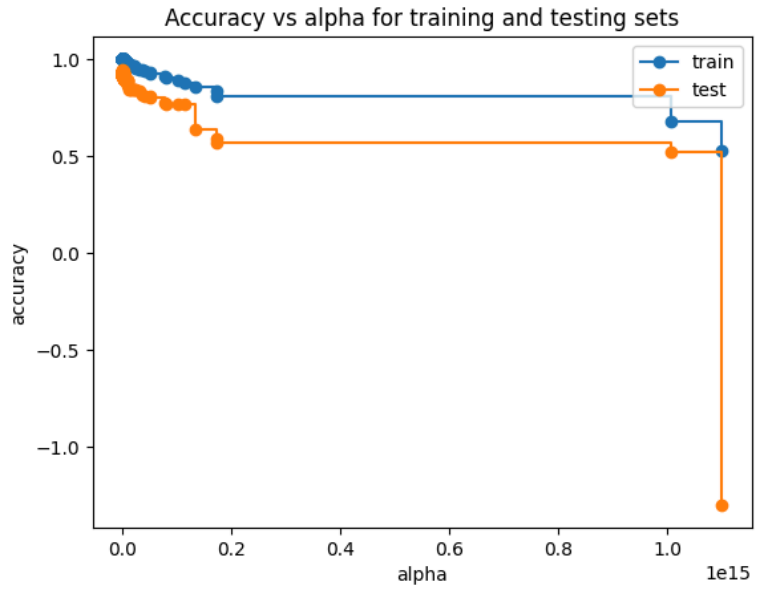
```
Decision Tree Regressor Results for Random Undersampling on scaled data
Mean Squared Error:  76416474593350.39
Mean Absolute Error:  4183704.603580563
R2 Score:  0.9345900212423094

Decision Tree Regressor Results for Random Oversampling on scaled data
Mean Squared Error:  8602279012787.724
Mean Absolute Error:  994694.3734015345
R2 Score:  0.9926367332373229

Grid Search Results for Random Undersampling on scaled data
Fitting 5 folds for each of 450 candidates, totalling 2250 fits
{'max_depth': None, 'max_features': None, 'min_samples_leaf': 2,
'min_samples_split': 4}
Mean Squared Error:  88296769737691.81
Mean Absolute Error:  4471875.639386189
R2 Score:  0.9244208809206479
```

```
Grid Search Results for Random Oversampling on scaled data
Fitting 5 folds for each of 450 candidates, totalling 2250 fits
{'max_depth': None, 'max_features': None, 'min_samples_leaf': 1,
'min_samples_split': 2}
Mean Squared Error:  8267053752557.545
Mean Absolute Error:  1017783.2480818414
R2 Score:  0.992923674989967

There is no improvement in the R2 Score after grid search.
```

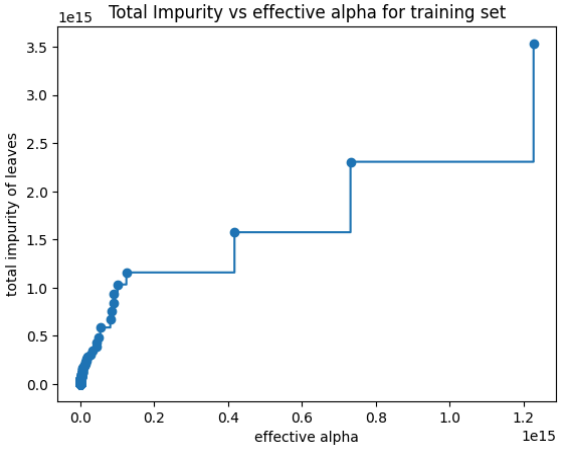
3. Task 3: Pruning Decision Tree:

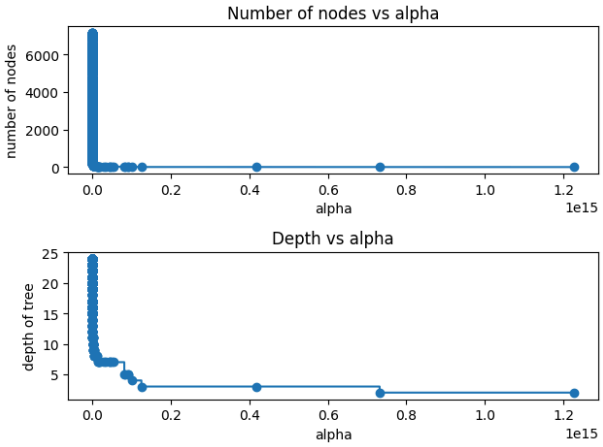
Scaled data plot (Undersampling):

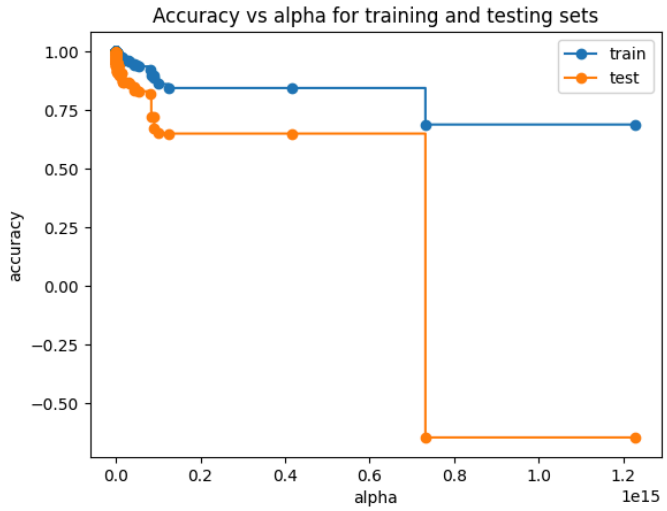




Scaled data plot (OverSampling):





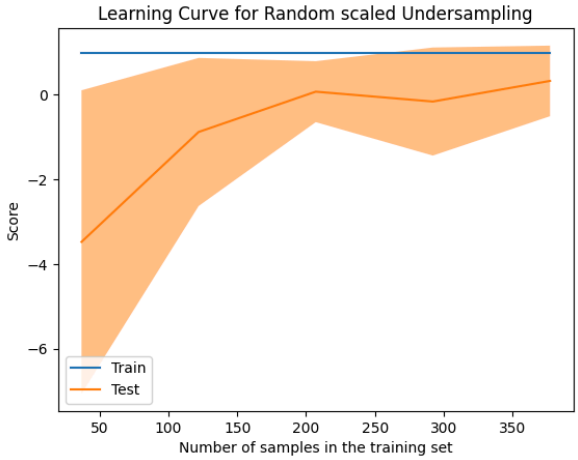


As alpha increases on the graphs, more of the tree is pruned. As alpha increases, more of the tree is pruned leading to better generalization on the test set.

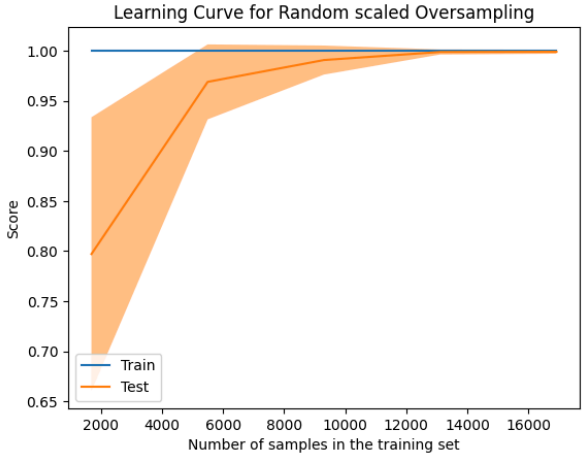
4. Task 4: Handling Overfitting:

Results on scaled data:

```
Cross Validation Results for Random Undersampling
Cross Validation RMSE:  [20437864.75612053 13264320.55037716
10349303.99756634 17895387.49558168
 67430115.123229  ]
Mean RMSE:  25875398.384574942
```

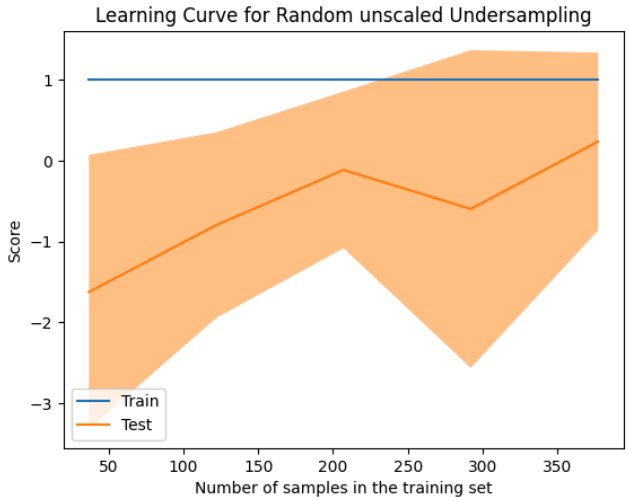



```
Cross Validation Results for Random Oversampling
Cross Validation RMSE:  [2498568.39738239 1268875.43862748       0.
0.
       0.         ]
Mean RMSE:  753488.7672019735
```

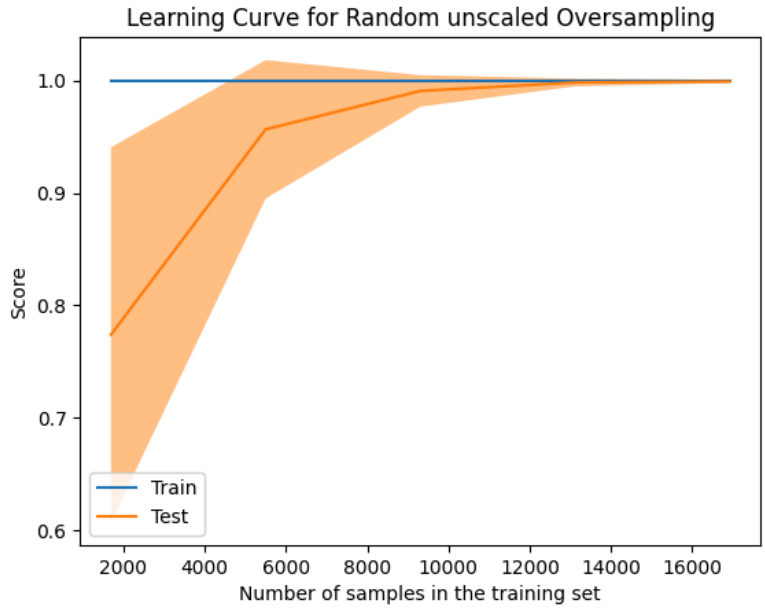


Results on unscaled data:

```
Cross Validation Results for Random Undersampling on unscaled data
Cross Validation RMSE:  [22844100.93149698  8573359.88611478
11915359.78376632 20118901.87773571
 58174239.38187396]
Mean RMSE:  24325192.37219755
```




```
Cross Validation Results for Random Oversampling on unscaled data
Cross Validation RMSE:  [2368167.31679192 1289143.49496194       0.
0.
       0.         ]
Mean RMSE:  731462.1623507725
```



4. Model Evaluation and Error Analysis

1. Task 1: Model Evaluation:

```
Fine tuned model evaluations on test data--------------------------------
Random Undersampling on scaled training data
Mean Squared Error:  116496637112288.14
Mean Absolute Error:  3239748.5875706216
```

```
R2 Score:  0.9843950836975179

Mean Squared Error:  88296769737691.81
Mean Absolute Error:  4471875.639386189
R2 Score:  0.9244208809206479

Random Oversampling on scaled training data
Mean Squared Error:  23.638426626323753
Mean Absolute Error:  0.0472768532526475
R2 Score:  0.9999999999999968
Mean Squared Error:  8267053752557.545
Mean Absolute Error:  1017783.2480818414
R2 Score:  0.992923674989967

Random Undersampling on unscaled training data
Mean Squared Error:  5009721319193.52
Mean Absolute Error:  946198.9818091511
R2 Score:  0.999319502986789
Mean Squared Error:  47702386913462.17
Mean Absolute Error:  3922881.8283166112
R2 Score:  0.9591683320736153

Random Oversampling on unscaled training data
Mean Squared Error:  117020375341.64311
Mean Absolute Error:  57478.71335149135
R2 Score:  0.9999840213337164
Mean Squared Error:  10207729407923.238
Mean Absolute Error:  1079494.5971867007
R2 Score:  0.9912625207157278
```

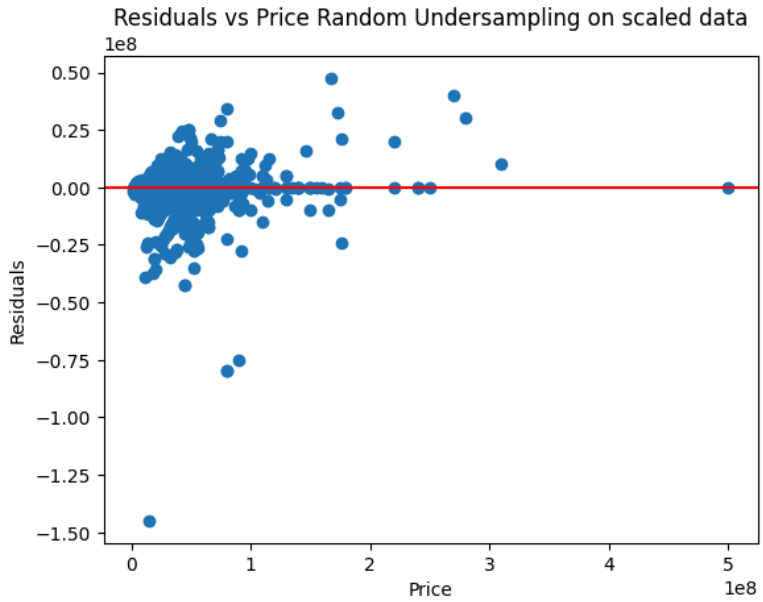
Oversampling on scaled data is giving a near perfect fit on the training data, with near-zero error values. On the other hand, in test data, even though R2 score is high, high error values suggest that the model is overfitting ont the training data.

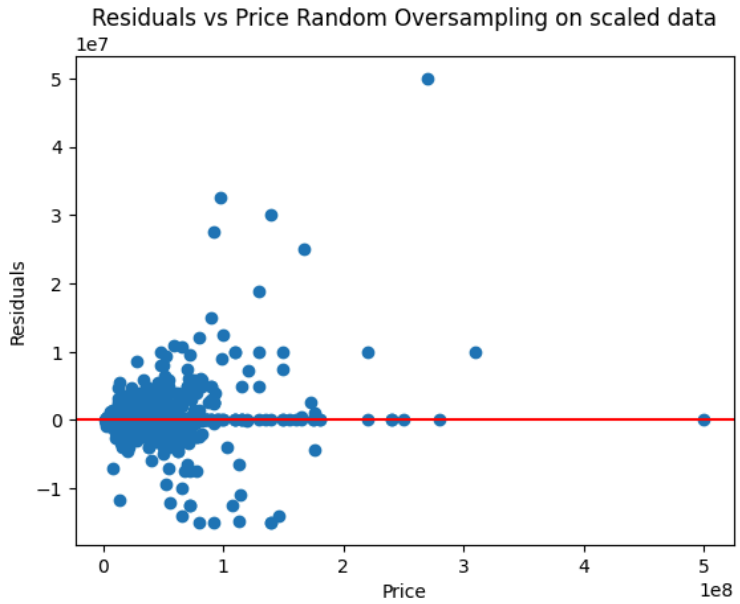
Undersampling on unscaled data gives a high R2 score on both training and test dataset.

Oversampling on unscaled data perform well on both dataset with low error values. There is a better generalization compared to the other strategies.

Overall, random sampling on scaled data has the best performance.

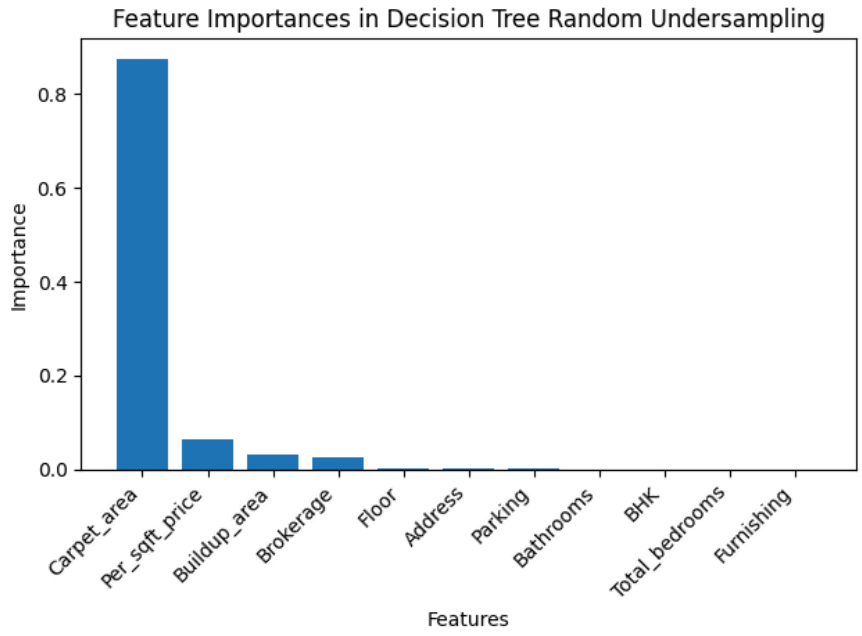
2. Task 2 : Residual and Error Analysis:

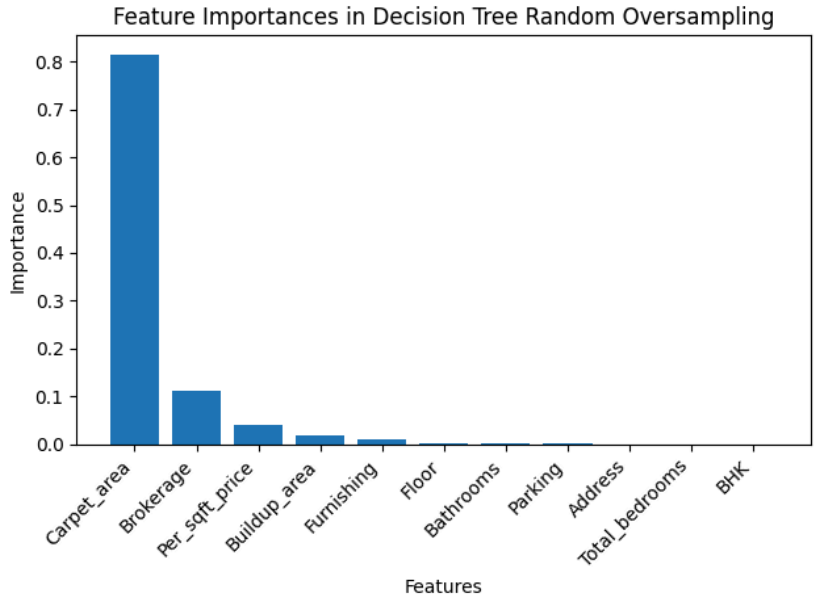




By observing the graph, the models is performing relatively poor on values that have high prices.

3. Task 3 : Feature Importance based analysis: (Both results on scaled dataset)





Carpet_area, brokerage, per_sqft_price are common important features.

RMSE for Random Undersampling: 9396636.08626469
RMSE for Random Oversampling: 2875248.4679689067
RMSE for Random Undersampling on unscaled data: 6906691.459263413
RMSE for Random Oversampling on unscaled data: 3194953.7411241555

First 2 results are on scaled data.

Bonus Challenge (6 Marks)

1. Task 1:Advanced Imbalance Handling

Decision Tree Regressor Results for SMOTE on scaled data
Mean Squared Error: 29786870062174.71
Mean Absolute Error: 1116405.4539641943
R2 Score: 0.9745034228758509

Decision Tree Regressor Results for ADASYN on scaled data
Mean Squared Error: 26578178900280.145
Mean Absolute Error: 1351326.2212276214
R2 Score: 0.9772499565501194

SMOTE has lower MSE as compared to ADASYN which means it shows lower variance from actual values.
MAE is smaller for SMOTE than ADASYN.
R2 score is higher for ADASYN means it captures variance in the data in a better way.

Task 2:Ensemble Learning: Random Forest
Decision Tree Regressor Results on scaled data
Mean Squared Error: 8301359166879.795
Mean Absolute Error: 900944.3734015345
R2 Score: 0.9928943106881717

Random Forest Regressor Results on scaled data
Mean Squared Error: 9392492711036.836
Mean Absolute Error: 685472.1334185848
R2 Score: 0.991960336406776

Both have comparable R2 scores with the decision tree performing slightly better.

Decision trees are prone to overfitting and may not be able to handle outliers. Random forest leads to a better generalization and can handle outliers in a better way.

There are more plots in the code file.

References:

- https://scikit-learn.org/dev/modules/generated/sklearn.tree.DecisionTreeRegressor.html
- https://scikit-learn.org/1.5/modules/generated/sklearn.tree.plot_tree.html
- https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.ADASYN.html
- https://scikit-learn.org/dev/modules/generated/sklearn.ensemble.RandomForestRegressor.html
- https://scikit-learn.org/1.5/auto_examples/model_selection/plot_learning_curve.html
- https://scikit-learn.org/1.5/modules/model_evaluation.html#scoring-parameter
- https://scikit-learn.org/1.5/modules/cross_validation.html
- https://scikit-learn.org/1.5/auto_examples/tree/plot_cost_complexity_pruning.html
- https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.RandomUnderSampler.html