FoDS Assignment

Om Agarwal-2019A7PS0052H

Saransh Dwivedi- 2019A7PS0173H

Ojashvi Tarunabh-2019A7PS0025H

Preprocessing

Normalisation-

Given that the scale of all the variables was different, we decided to normalised our data-frame before training any models to bring all the data values in the range 0 to 1. We used the formula

$$X_{normalized} = \frac{X - min(X_i)}{max(X_i) - min(X_i)}$$

We then shuffled the dataset rows randomly using built in functions and selected a 70-30 train-test split to run our models on.

Detecting Outliers-

In statistics, an outlier is a data point that differs significantly from other observations. Here we attempted to remove the outlier using z -values which is calculated by the formula

$$z = \frac{x - \mu}{\sigma}$$

The dataframe which had values greater than three standard deviations from the mean were removed from the dataset as the bracket of three standard deviations includes almost 99% of the available dataframe.

Handling Missing Values-

The given dataset had some missing values in three (out of 13) of the given input features which were handled by replacing each of them by the mean of the rest of the values in that column.
The column ‘Water Front’ had all the values 0 except one row which had a value of 1. This feature essentially seemed to be redundant and hence in order to prevent wrong learning by the algorithm we removed this column.

Linear Regression with Multiple Input Features

The cost function of linear regression is as follows:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(x_i^T\theta - y_i)^2$$

The basic idea behind it is that we start at a random point on the loss function curve and take small steps towards the minimum based on the current errors.

$$\theta_j := \theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}(\hat{y}^i - y^i)x_j^i$$

The size of our steps is determined by a hyperparameter “learning rate”. A higher learning rate means we take bigger steps towards the minimum, but also leaves room for the possibility of overshooting and oscillating around the minimum.

Feature Selection-

Greedy Forward Feature Selection-

Greedy Forward Selection is an iterative algorithm in which we try to select the best possible feature iteratively from the available set of features with the criteria of selection being the minimum L2 norm of the error. We keep adding the feature which best improves our model till an addition of a new variable does not improve the performance of the model.

Greedy Backward Feature Selection-

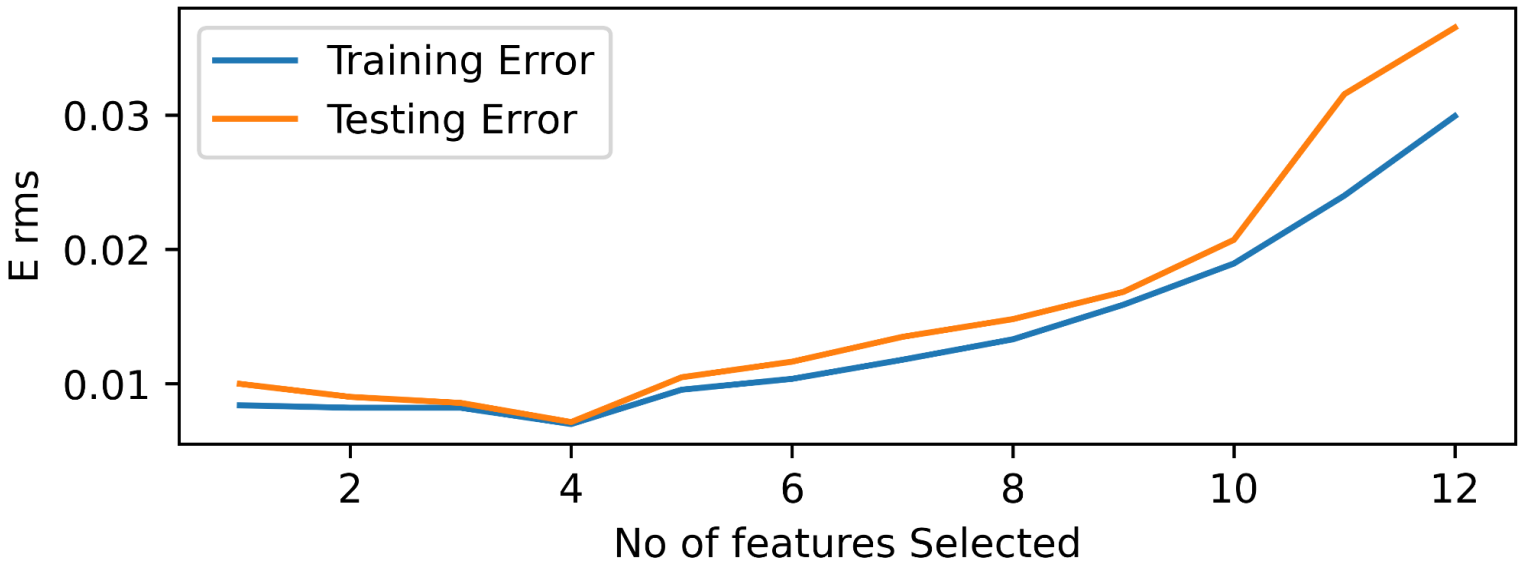
Greedy Backward Selection starts with all features contained in the dataset**.** It then runs the model and tries to improvise its prediction by removing out the less contributing input features iteratively.
In every iteration it removes a single feature which has maximum error in output prediction. It is similar to greedy forward features but may give different results.

Runs of models

1.Greedy Forward Feature Selection

Here, we ran the linear regression model for various numbers of variables from 1 to 12 and tabulated the train and test errors at each number of variables. These results can be seen below –

```
With 1 feature, The minimum training error is 0.007403975681377384 and minimum testing error is 0.00769974709909612
With 2 features, The minimum training error is 0.0072206139334508035 and minimum testing error is 0.007930726741487154
With 3 features, The minimum training error is 0.00722001586713538 and minimum testing error is 0.00778158520085538
With 4 feature, The minimum training error is 0.00700945197832408 and minimum testing error is 0.007156512676563692
With 5 feature, The minimum training error is 0.009562265561466153 and minimum testing error is 0.01048795129211011
With 6 features, The minimum training error is 0.010371397976318877 and minimum testing error is 0.0116479623893838
With 7 features, The minimum training error is 0.011792814197946954 and minimum testing error is 0.0134961686980836
With 8 features, The minimum training error is 0.013313371617933546 and minimum testing error is 0.0148145520543942
With 9 features, The minimum training error is 0.0158846654273624 and minimum testing error is 0.016838901559214095
With 10 features, The minimum training error is 0.01894886449818896 and minimum testing error is 0.0207108971189239
With 11 features, The minimum training error is 0.023995876440320918 and minimum testing error is 0.031544180787258
With 12 features, The minimum training error is 0.029911920258078085 and minimum testing error is 0.036493957148581
```



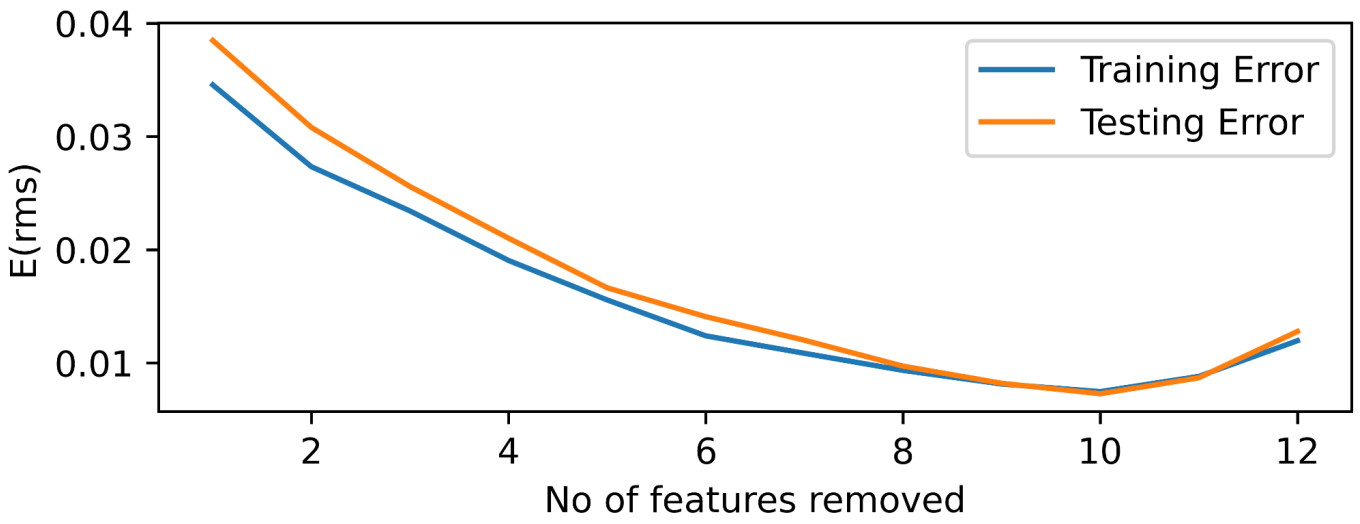
We get an optimum model for four features.
Features selected:-

- sqft_living
- condition
- bedrooms
- grade

2.Greedy Backward Feature Selection

Here, we ran the linear regression model for various numbers of variables from 1 to 12 and tabulated the train and test errors at each number of variables. These results can be seen below –

```
With 1 feature removed, The testing error is 0.03846213303489379 and the training error is 0.03454277174352706
With 2 features removed, The testing error is 0.030788877770133474 and the training error is 0.02731990955218315
With 3 features removed, The testing error is 0.02557028487331666 and the training error is 0.023413299238855257
With 4 features removed, The testing error is 0.02100705842176033 and the training error is  0.019043196917208186
With 5 features removed, The testing error is 0.016628899240003728 and the training error is 0.015557890343068871
With 6 features removed, The testing error is 0.014083374874216655 and the training error is0.012388764742972597
With 7 features removed, The testing error is 0.01200551362256512 and the training error is 0.010848499623349682
With 8 features removed, The testing error is 0.009713802625442788 and the training error is 0.00934179653610245
With 9 features removed, The testing error is 0.008202067789678448 and the training error is 0.008140694813301054
With 10 features removed, The testing error is 0.00727818386927288 and the training error is 0.0074662090728484344
With 11 features removed, The testing error is 0.008686882672472009and the training error is 0.008808299919723997
With 12 features removed, The testing error is 0.012772996130361489 and the training error is 0.011958776650066203
```



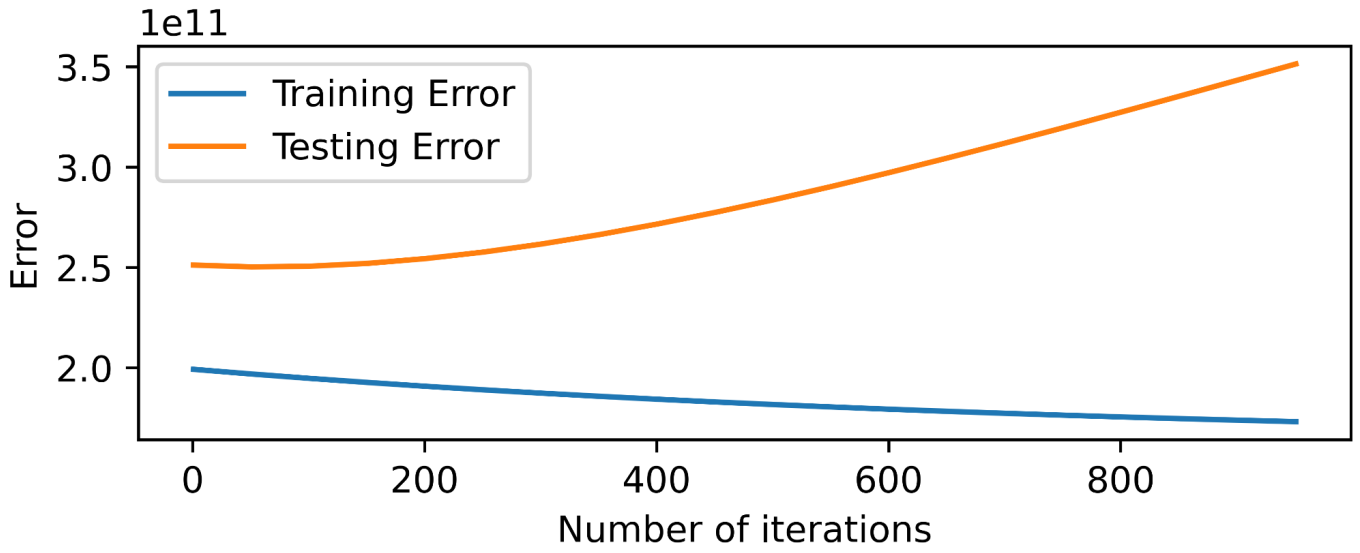
Here, we get an optimum model when 10 features are removed. Therefore, only 2 features are selected.
Features selected:-

- sqft_living
- grade

3. Linear regression model without any pre-processing and feature selection

Here, we ran the linear regression without any pre-processing and features selection for 1000 iterations and tabulated the testing error and training error after every 50 iterations:-

```
The training error is 199202561123.79272 and the testing error is 251164921409.34122
The training error is 196872445033.23196 and the testing error is 250271379072.3427
The training error is 194695061450.93127 and the testing error is 250595988222.89374
The training error is 192659818182.04724 and the testing error is 252011058599.9467
The training error is 190756858222.32474 and the testing error is 254399310944.5613
The training error is 188977008728.86395 and the testing error is 257653098883.73105
The training error is 187311733532.8028 and the testing error is 261673686812.04666
The training error is 185753088948.0722 and the testing error is 266370579813.1137
The training error is 184293682647.44562 and the testing error is 271660901939.92392
The training error is 182926635392.98297 and the testing error is 277468819431.32855
The training error is 181645545422.74704 and the testing error is 283725005681.71356
The training error is 180444455309.42148 and the testing error is 290366145004.19434
The training error is 179317821119.25705 and the testing error is 297334472435.2821
The training error is 178260483711.681 and the testing error is 304577347022.1264
The training error is 177267642030.9877 and the testing error is 312046856213.09796
The training error is 176334828251.8409 and the testing error is 319699449139.6026
The training error is 175457884649.91458 and the testing error is 327495596732.46027
The training error is 174632942077.9329 and the testing error is 335399476760.784
The training error is 173856399935.6778 and the testing error is 343378682015.7743
The training error is 173124907530.26926 and the testing error is 351403949986.94305
The final training error is 172448746126.58942 and the final testing error is 359287977741.31323
```



Conclusion

- For greedy forward feature selection we get a minimum error for four features (sqft_living, condition, bedrooms and grade).
- For greedy backward we get a minimum error when 10 features are dropped. Hence, only two features(sqft_living, grade) got selected.
- We observe that without any pre-processing and feature selection we get very large error values. This is due to the reason for not normalizing the dataset. As the range of various columns vary significantly from each other the gradient function changes abruptly hence with respect to the price prediction the values differ greatly which should not be an ideal case for a regression problem. This may give erroneous results due to random changes in weight matrix.