Report for Assignment-2

-Navam Shrivastav 2019A7PS0092H
-Aayush Keval Shah 2019A7PS0137H
-Aditya Choudhari 2019AAPS0309H

In this assignment we are required to develop **Greedy forward and backward feature selection algorithm** from scratch and use it on a given house prices dataset. Feature selection is an important step as it tells us which features are actually the governing features in the given set of input features as some features may not be related to the target feature. The irrelevant features will increase computational cost and may also lead to overfitting, which makes their removal necessary. For this we are supposed to build linear regression models and find out which are the most important features by analysing the RMS Errors obtained from these models. **This assignment is developed in python taking help from NumPy, Pandas and Matplotlib frameworks.**

Following are the key steps we performed to develop the algorithm:

1. Preprocessing:

The given dataset has 13 input features which are used to predict the house prices. The dataset has around 1188 rows of data. The dataset was first converted into an array with the help of NumPy and Pandas and also shuffled with the help of NumPy's built-in function of random shuffling. Next, all the input and the target features were normalized by subtracting the mean of that column from the entry and then dividing this by the standard deviation of the column. This shuffled and normalised dataset was split into 70% training data and 30% testing data. Some of the entries in the given data were missing. To handle such entries we removed the rows which had one or more missing values. Also we had to detect the outliers and remove them. For this, after normalising we know that 97% of the population lies within 3*standard deviation from the mean of the data. In our case the mean was 0 and the standard deviation was 1. So whichever input entry had an absolute value greater than 3, we removed such rows. These were the steps we used for preprocessing the data i.e. **shuffling**

the data, splitting the data into training and testing data, normalisation, handling missing values and detecting outliers.

2. Greedy forward feature selection:

Suppose the given number of input features are n. In greedy forward feature selection we first consider n models of 1 feature each. Whichever 1 feature model gives the least RMS Error we use that model in subsequent steps. Then we combine remaining features with the previous best one and whichever gives the least RMS Error we use them in the next steps. In this case there would be n-1 models of 2 features. We continue in this way till all of the n features are considered in our final model. There would be only 1 model of all the n features. While doing so we keep a track of which features were selected for a particular number of features and also of the training and testing errors in each case.

3. Greedy backward feature selection:

Suppose the given number of input features are n. In greedy backward feature selection we first consider 1 model of given n features. Then we remove 1 feature from the n features one at a time and find out the least RMS Error. Thus this step would require us to build n models of n-1 features each. In the next step we again remove 1 feature one at a time from the previous best set of n-1 features and whichever model gives us the least RMS Error we remove that feature. There would be n-1 models of n-2 features each in this step. We continue in this way till all of the n features are removed in our final model. There would be only 1 model of 0 features. While doing so we keep a track of which features were selected for a particular number of features and also of the training and testing errors in each case.

4. Implementation of both the algorithms:

In **Greedy forward** we first started with the 0 features model and subsequently kept on adding the best input features one by one. For keeping track of the features added we kept a boolean array where true implied that the feature was already used. Initially the whole array was filled with false as the boolean value. The main code was the 2 nested for loop where the outer loop was for keeping the track of number of features added and the inner loop iterated over the

features not already included in the best model and whose value was false in the boolean array. We looped till all the features were not included in the final model.

In **Greedy backward** we first started with all the 13 features in the model and subsequently kept on removing the input features one by one. For keeping track of the features removed we kept a boolean array where false implied that the feature was already removed. Initially the whole array was filled with true as the boolean value. The main code was the 2 nested for loop where the outer loop was for keeping track of number of features removed and the inner loop iterated over the features already included in the best model and whose value was true in the boolean array. We looped till all the features were removed in the final model.

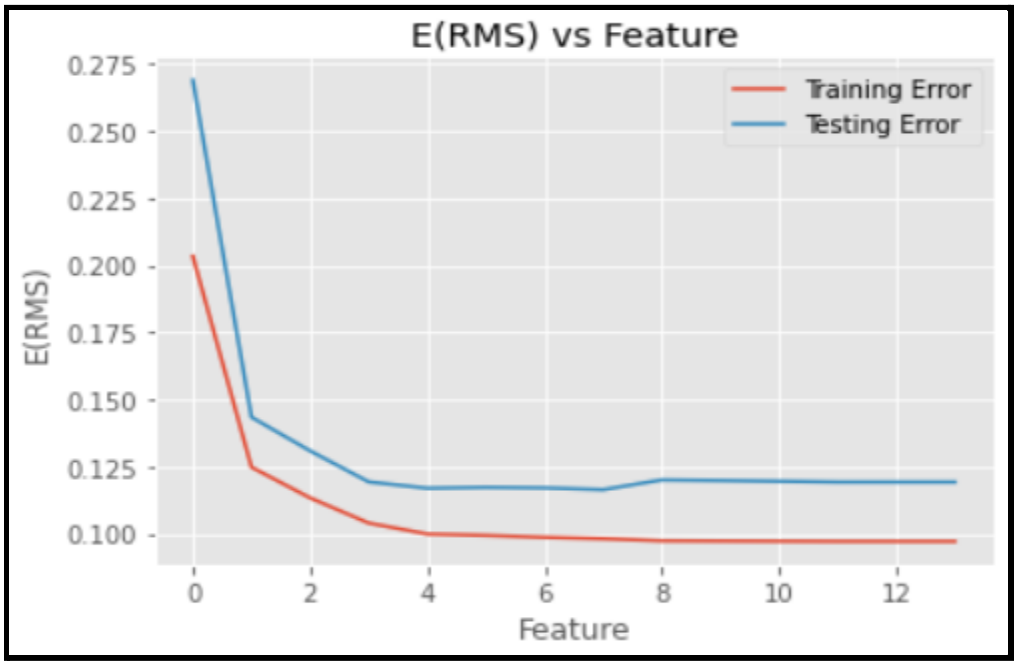
For training both the above algorithms we used the **Gradient descent algorithm** which was developed by us in Assignment-1 of this course to find the weights in the linear regression model.

5. Results:

a) Greedy forward feature selection:

Number of features	Training Error	Testing Error
0	0.20328645	0.26886215
1	0.12475663	0.14337581
2	0.11332473	0.13084047
3	0.10395085	0.11927415
4	0.09987658	0.11693813
5	0.09935406	0.11726231
6	0.09865620	0.11706397
7	0.09811940	0.11639888
8	0.09735066	0.12002655
9	0.09726691	0.11980214

10	0.09719869	0.11952496
11	0.09713366	0.11922008
12	0.09713366	0.11922008
13	0.09713366	0.11922008



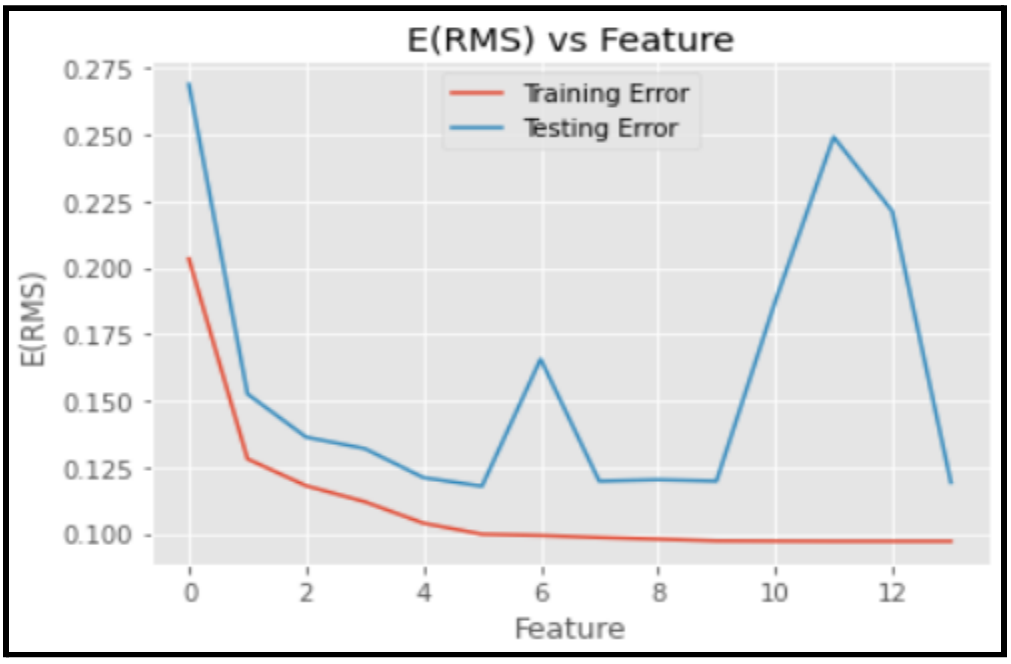
E(RMS) vs Feature for Greedy forward

Order of adding features is **[3, 9, 7, 8, 12, 13, 2, 5, 1, 10, 4, 6, 11]** where i denotes the input feature i with 1 <= i <=13. The **best Greedy forward model is the one having 7 features** which are **[3, 9, 7, 8, 12, 13, 2]** i.e. **sqft_living, grade, view, condition, sqft_living15, sqft_lot15, bathrooms.**

b) Greedy backward feature selection:

Number of features	Training Error	Testing Error
0	0.20328645	0.26886215
1	0.12807549	0.15248833

2	0.11798042	0.13626233
3	0.11198796	0.13195968
4	0.10394030	0.12105383
5	0.09978198	0.11783173
6	0.09931440	0.16559185
7	0.09855823	0.11974238
8	0.09793367	0.12025351
9	0.09728194	0.11975157
10	0.09719869	0.18706657
11	0.09713366	0.24903056
12	0.09713366	0.22108416
13	0.09713366	0.11922008



E(RMS) vs Feature for Greedy backward

Order of removing features is **[3, 6, 4, 1, 13, 12, 5, 2, 8, 11, 10, 7, 9]** where i denotes the input feature i with 1 <= i <=13.The **best Greedy backward model is the one having 5 features** which are **[8, 11, 10, 7, 9]** i.e. **condition, sqft_basement, sqft_above, view, grade.**

c) Linear regression model without any pre-processing and feature selection:

Training Error	Testing Error
31930580018.849915	48070025414.30089

This **error is quite high as there is no normalisation** of the data given as no preprocessing was required to be done on the data before using it for the linear regression model.

d) Comparison among the best results obtained from the above three methods:

Method	Number of features	Training Error	Testing Error
Greedy forward	7	0.09811940	0.11639888
Greedy backward	5	0.09978198	0.11783173
Linear regression	NA	31930580018.849915	48070025414.30089

Thus from the optimal models obtained from Greedy forward and backward which are tabulated in the table above, we can see that both the **training and testing errors are quite similar** for them. But the number of features are **7** [**sqft_living, grade, view, condition, sqft_living15, sqft_lot15, bathrooms]** for Greedy forward and **5** [**condition, sqft_basement, sqft_above, view, grade**] for Greedy backward models.