Foundations of DataScience

Assignment 2

1	ANIRUDH A	2018B4A70936H
2	VISHAL KUMAR NK	2019A4PS0693H
3	SHIVANI THIRUNAGARI	2019A4PS0754H

Objectives of the Assignment:

1. To perform a thorough pre-processing of the dataset.
2. To find the subset of features that provide the optimal regression model using greedy forward feature selection.
3. To find the subset of features that provide the optimal regression model using greedy backward feature selection.

Data preprocessing:

Firstly, outliers were detected and removed from the dataset. For determining effective outliers, a 90% confidence interval (3*sigma) was considered. Then we normalized the data using min-max normalization. We preferred normalization over standardization so as to scale down the data and get lower values of error. Then, we tried to find if there were any missing values in the dataset. Those values were removed from the dataset. Finally, we shuffled the data and split it 70:30 for the training:testing data.

Description of the model:

We used the gradient descent algorithm to build the linear regression model. For this, we initialised parameters using a function called initialize_params(X, numberOfFeatures). We computed the sum of squares of errors using the function compute_cost(thetha, X, y). We performed gradient descent using the function gradientDescent(X, y, thetha, alpha, XT, yt). After this, we performed forward greedy and backward greedy feature subset selection.

We performed the gradient descent algorithm on the whole data without greedy feature subset selection and got the error value as `0.9003018775130878`.

Forward greedy feature subset selection:

Forward greedy algorithm starts with a single feature and determines which one feature subset gives the least RMSE value. Now, this is kept for the model, and the same is done by adding features one-by-one. Finally, the feature subset which gives the minimum RMSE is chosen as the best mode.

For the forward greedy algorithm, first-off we defined empty arrays modelRMSE that stores the least RMSE value for a given number of features, and another empty array called featuresSoFar and stored all the features in an

array called featuresRemaining. Then we iterated through each of the features and formed the gradient descent model using the features and chose the best feature for a given subset of features. The best feature combination for a given subset was chosen on the basis of the minimum value of RMSE. We appended this best feature for a given subset in the featuresSoFar array and removed it from the featuresRemaining array. Finally, we append all the least RMSE values for each of the number of subsets into modelRMSE array. We applied this to the testing data and found out the best model for the same.

Table: Min RMSE Values

No. of Features	Best subset	Minimum Testing Error	Minimum Training Error
1	[9]	0.24552178654835394	0.0023537454554710424
2	[9, 1]	0.2326631630796311	0.0032339588778994535
3	[9, 1, 13]	0.23260051864950118	0.010021474707442047
4	**[9, 1, 13, 6]**	**0.23260051864950102**	0.011413323782590929
5	[9, 1, 13, 6, 4]	0.23262246266948872	0.014514578806115374
6	[9, 1, 13, 6, 4, 7]	0.233265670234442	0.015432001909285493
7	[9, 1, 13, 6, 4, 7, 8]	0.23613737900960288	0.016332854499717647
8	[9, 1, 13, 6, 4, 7, 8, 11]	0.24780702880995392	0.016759164246494852
9	[9, 1, 13, 6, 4, 7, 8, 11, 2]	0.26098366858309385	0.014918373847895697
10	[9, 1, 13, 6, 4, 7, 8, 11, 2, 12]	0.27979187791357524	0.01867030291432725

11	[9, 1, 13, 6, 4, 7, 8, 11, 2, 12, 3]	0.2999351965123864	0.023130155767280232
12	[9, 1, 13, 6, 4, 7, 8, 11, 2, 12, 3, 5]	0.3195213596384551	0.029009255894417375
13	[9, 1, 13, 6, 4, 7, 8, 11, 2, 12, 3, 5, 10]	0.34212469799115	0.0335272040289555

Backward greedy feature subset selection:

The backward greedy algorithm is just the opposite of the forward greedy algorithm. Here, instead of considering the number of features from 1, we consider them as “n” or the total number of features and take out the combination which gives the least RMSE by removing features one-by-one.

We implemented this algorithm by first defining an empty array modelRMSE that stores the least RMSE value for a given number of features. Then we initialize an array called featuresSoFar which gives the features that have been considered so far. Here, instead of looping from 1 till 13, we loop from 13 till 1. Then we iterated through each of the features and formed the gradient descent model using the combination of features and chose the best combination for a given number of subsets. The best feature combination for a given subset was chosen on the basis of the minimum value of RMSE. Finally, we removed the featureChosen from the featuresSoFar array. Finally, we append all the least RMSE values for each of the number of subsets into modelRMSE array. We applied this to the testing data and found out the best model for the same.

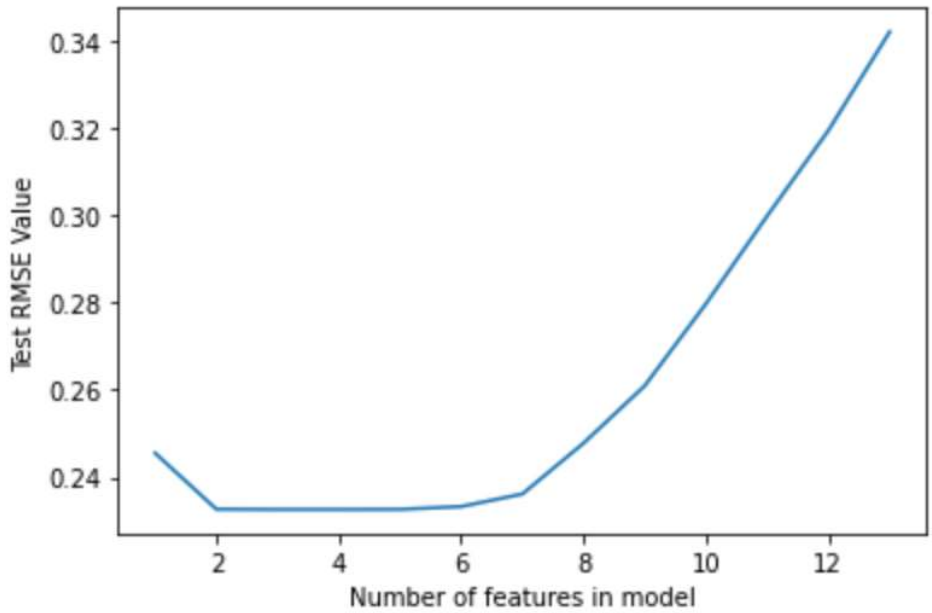
Table: Min RMSE Values

No. of Features	Best subset	Minimum Testing Error	Minimum Training Error
12	[1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13]	0.3195213596384551	0.029009255894417354
11	[1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13]	0.2971242594918011	0.023130155767280235
10	[1, 2, 4, 5, 6, 7, 8, 11, 12, 13]	0.27776935640096856	0.01867030291432726

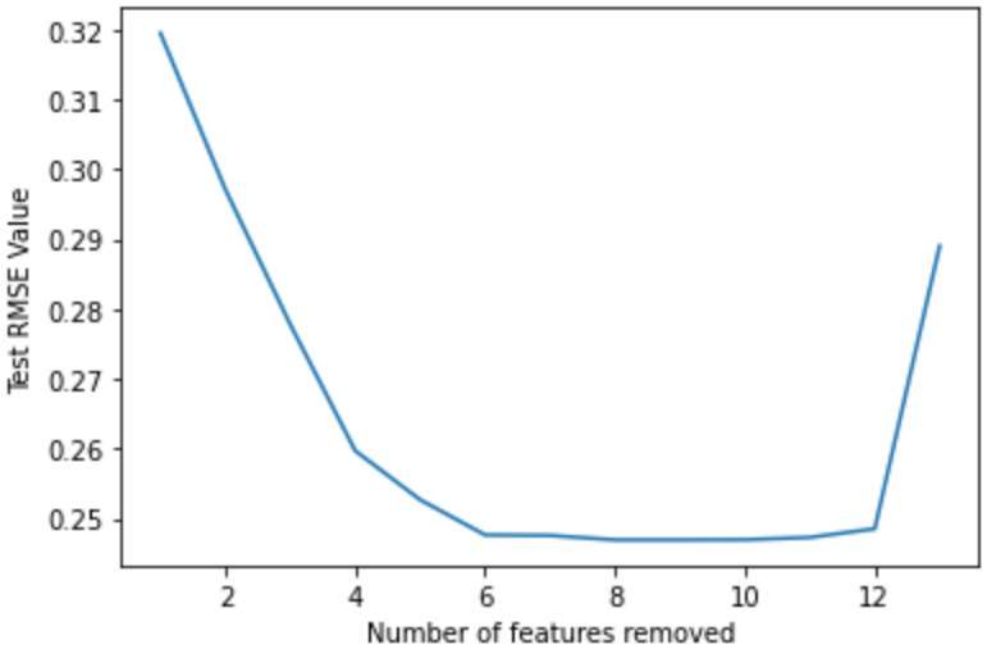
9	[1, 4, 5, 6, 7, 8, 11, 12, 13]	0.25968666128841994	0.014918373847895726
8	[1, 4, 5, 6, 7, 8, 11, 13]	0.25266298527149134	0.014651376557396133
7	[1, 4, 5, 6, 7, 8, 13]	0.2476229813267331	0.013890145797992812
6	**[1, 4, 6, 7, 8, 13]**	**0.24754488729076338**	0.013176565945993930
5	[1, 4, 6, 7, 13]	0.24791907599559926	0.013114229015425348
4	[1, 4, 7, 13]	0.24791907599559926	0.013071536466686504
3	[1, 7, 13]	0.2479439906068624	0.013046757380294302
2	[1, 7]	0.24826627225304512	0.015355388226472408
1	[1]	0.24850493607963095	0.015348216930709611

Plots for Testing RMSE v/s Number of Features:

1) **Forward-Greedy**



2) **Backward-Greedy**



Conclusion: A total of 6 models have been built - Based on 1)Forward Greedy, 2)Backward Greedy, 3)All Features (Each model was built once using number of iterations as the stopping criteria and once using convergence as the stopping criteria). The subset of features have been chosen based on the Testing RMSE values. Using forward greedy, an optimal of 4 features (Grade, Condition, Waterfront, SqftLot) was obtained and using backward greedy, an optimal of 6 features (Bedrooms, SqftLot, View, WaterFront, Condition, SqFtLot15) has been obtained. A U-shaped graph in both the cases indicates that testing error is high when very features or very high number of features are chosen (due to over-fitting and redundancy/correlation respectively). The testing error is also high when a model is built without performing any feature selection/pre-processing technique. In fact, an error of infinity can be obtained if Nan values are not removed before applying the model.