FODS ASSIGNMENT 2

**RUTHVIK REDDY 2018B4A70701H**

**KESHAV CHANDA 2018B1A70757H**

**DEEPANSHU MODY 2018B1A70949H**

**REPORT**

In this assignment, we have worked on predicting the house prices given 13 other attributes. In aiding us to predict the feature subset which gives us an optimal regression model, we performed greedy forward feature selection and greedy backward feature selection. But before that, we pre-processed the data by standardising, normalising, detecting outliers and also handled missing values.

**Model Description**

Gradient Descent is an iterative algorithm use in loss function to find the global minima. The loss can be any differential loss function. The different types of loss functions are linear loss, logistic loss, hinge loss, etc.

LinearRegression class uses Gradient descent and has the following methods:

\_\_init\_\_() – the constructor, contains the values for learning rate and the number of iterations, alongside the weights and bias (initially set to None). We’ll also create an empty list to track loss at each iteration.

\_mean\_squared\_error(y, y\_hat) – “private” method, used as our cost function.

fit(X, y) – iteratively optimizes weights and bias through gradient descent. After the calculation is done, the results are stored in the constructor. We’re also keeping track of loss here.

predict(X) – makes the prediction using the line equation.

**Algorithm Description**

A greedy feature selection is the one in which an algorithm will either select the best features one by one (forward selection) or removes worst feature one by one (backward selection).

**Forward Selection:** Forward selection is an iterative method in which we start with having no feature in the model. In each iteration, 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.

**Backward Selection:** In backward elimination, we start with all the features and removes the least significant feature at each iteration which improves the performance of the model. We repeat this until no improvement is observed on removal of features.

**Algorithm implementation**

**Forward feature selection**

First, the best single feature is selected using MSE

• Then, pairs of features are formed using one of the remaining features and this best feature, and the best pair is selected.

• Next, triplets of features are formed using one of the remaining features and these two best

features, and the best triplet is selected.

• This procedure continues until all are selected.

**Backward feature selection**

First, the MSE is computed for all n

features.

• Then, each feature is deleted one at a time, the MSE is computed for all subsets with n-1 features, and the worst feature is discarded.

• Next, each feature among the remaining n-1 is deleted one at a time, and the worst feature is discarded to form a subset with n-2 features.

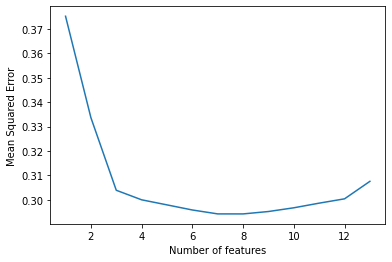
• This procedure continues until no features are left.

**Best features selected by Greedy Forward Feature Selection**

Selected features-

"Sqft\_living","view","grade","sqft\_basement","condition","bathrooms","sqft\_living15","floors"

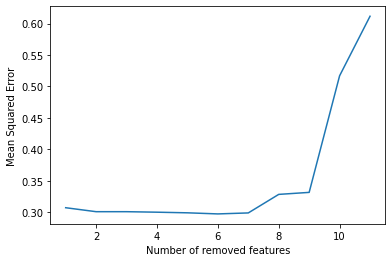
As shown in the graph, 8 features mentioned above were used to get the best results by the algorithm

****

**Best features selected by Greedy Backward Feature Selection**

Selected features- "Bedrooms","bathrooms","sqft\_living","view","grade","sqft\_basement","sqft\_living15"

As shown in the graph, by removing 6 features, namely sqft\_lot,sqft\_above,sqft\_lot15,floors, condition, waterfront the algorithm obtained the best results

****

**Table with the minimum training and testing errors**

|  | **Minimum Training error** | **Minimum Testing error** |
| --- | --- | --- |
| **Without pre-processing** | **0.59266** | **0.39636** |
| **Forward Feature Selection** | **0.43683** | **0.29415** |
| **Backward Feature Selectioin** | **0.43249** | **0.29473** |