**MidTerm Project: Predicting SalePrice**

**Team Members:**

**Harika Addagada (hxa150930)**

**Ravali Nallamasu(rxn152730)**

**Aman Miryala (axm151830)**

**Himanshu Parashar(hxp151330)**

**Purpose:**

The purpose of this project is to find a model that best predicts the SalePrice of a house by comparing various models built using techniques such as linear regression, lasso, ridge etc

**Dataset description:**

We used “Ames Housing” dataset. This dataset has 82 attributes and 2930 observations.

**Data Cleaning/ Data Pre-processing:**

* Initially, we had a dataset of 82 variables with 2930 rows.
* We removed the fields Order and PID right away as they don’t add value to the data in predicting.
* We then summarized the data using summary() command.
* We noticed that there are few attributes which have character values. So, we converted them into factors.
* From the summary, we observed that few variables such as Fence, Pool.QC, Alley etc have a lot of missing values(more than 80% of the values are NA’s). So we decided to omit these attributes.
* Next, we removed the attributes that have do not have good variation. For instance we removed attributes like Street which have more than 90 percent of the values are in one category and the rest distributed among other categories.
* We plotted various graphs such as histograms, dot and line graphs to see the distribution of the variables and eliminated few attributes that are not showing much variation such as Land.Contour. By doing so we were finally left with 37 attributes. The attributes are :

“MS.SubClass”,”MS.Zoning”,”Lot.Area”,”Lot.Shape”,”Lot.Config”,”Land.Slope”,”Neighborhood”,”Condition.1”,”Overall.Qual”,”Overall.Cond”,”Year.Built”,”Year.Remod.Add”,”Roof.Style”,”Mas.Vnr.Type”,”Mas.Vnr.Area”,”Exter.Qual”,”Exter.Cond”,”Foundation”,

“Bsmt.Qual”,”Bsmt.Exposure”,”BsmtFin.Type.1”,”BsmtFin.SF.1”,”BsmtFin.Type.2”,”BsmtFin.SF.2”,”Bsmt.Unf.SF”,”Heating”,”X1st.Flr.SF”,”X2nd.Flr.SF”,”Low.Qual.Fin.SF”,”Full.Bath” ,”Half.Bath”,”Bedroom.AbvGr”,”Kitchen.AbvGr”,”Kitchen.Qual”,”Functional”,”Garage.Type”,”Garage.Finish”,”Garage.Cars”,”Garage.Area”,”Wood.Deck.SF”,”Screen.Porch”,”Sale.Type”,”Sale.Condition”,”SalePrice”

* Next, we removed the outliers from Gr.Liv.Area which have a sqft of more than 4000.
* And finally, removed the Na’s in the left 37 variables.

**Models:**

Lasso:

* First, we built a matrix using model.matrix() method and the stored all the values of shares in a variable
* Next we split the data into train and test data using the sample() method. We took 80% of the data i.e., 2140 rows as train data and the rest as test data.
* Then we created a lasso model using the glmnet() function by passing the train data, SalePrice field of the train data, alpha as 1 and lambda as a sequence of hundred numbers( lambda=10^seq(10,-2,length=100))
* Next we plotted the coefficients versus L1Norm plot for the lasso model obtained inorder to get an idea about how the coefficients are varying with L1 Norm(See fig 1.1)
* Next we performed cross validation by using cv.glmnet() function to determine the best lambda value.
* We then plotted the output of the cv.glmnet()(see fig 1.2)
* From the plot, we found the minimum value of lambda as 198.8304
* Next, we predicted the output on test data using the predict() method with the lasso model built in the previous step and the minimum value of lambda (198.8306)as parameters.
* Next, we calculated the Mean squared error(MSE) using mean() function and we determined the value as 421309091.The square root of the MSE (which is equal to 20,525.81) when compared with the mean of the SalePrice(which is equal to 186000 ) indicates that lasso performs a very good job in predicting the Sale Price

Forward and Backward:

* Applied the regsubsets() function by specifying the argument method as “forward” and taking nvmax as 25 , we got the best one-variable model containing “Overall Qual” and the best two-variable model additionally includes “Gr.Liv.Area”.
* Based on the adjR2 plot (see fig 2.1) 23 predictors seem to be the best with high R2 and min Cp values.
* The 23 best predictor model variables are MS.Subclass, Lot.Area, Neighborhood, Overall Qual, Year.Built, Year.Remod.Add, Bsmt.Exposure, BsmtFin.Type.1, Heating, Kitchen, TotRms.AbvGrd, Exter.1stPreCast
* However, the best 6 model-variable with low Cp, BIC and high R2 (see fig 2.2)are Neighborhood, Overall Qual, Year.Remod.Add, Heating, X1st.Flr.SF, Kitchen.

The lowest BIC values obtained for the above values are

NeighborhoodNridgHt - 3.240248e+04 Overall.Qual - 3.032227e+04 Year.Remod.Add - 4.985969e+02 Heating.QCPo - 6.816420e+03 X1st.Flr.SF - 7.080724e+01 Kitchen.QualPo - 8.775761e+03

* When applied the backward method, we got the best one-variable model contains Overall Qual and the best two-variable model additionally includes Gr.Liv.Area. They are the same as forward selection methods.
* **Cross- validation approach:**  On applying the cross – validation approach, with 10 folds on the whole data set we found the lowest MSE as 1116094417 for 23 predictors.

Using Ridge regression:

* First, we used model.matrix() method to convert the data in matrix form and stored it in the variable.
* Then we created sequence of lambdas for testing i.e **grid=10^seq(10,-2,length=100)** , doing ridge regression on each lambda by setting alpha=0. We now have scaled data and coefficient matrix (with row as predictor and column as lambda value). Dimension for our coefficient matrix came out to be “148 100”. for 50 we get lambda value 11497.57.
* The square root of sum of squares of coefficients for 50 = 162881.6. For 60 it is 195799.4.
* Plot of mse vs log(lambda) – fig2.1
* Now we calculate the coefficients using predict and for lambda=50
* We have 2675 records and we divided the records 50% each for training and test data.
* Now using glmnet and our test and train data we form a model with lambda as the grid we created and giving threshold as 1e-12.
* Then using predict and taking our model, we calculated mean square for different values of lambda. Like for s=4 our mean square error is 522123970. For s=0 using “exact=T” it comes 522169576.
* Then after using lm to fit our linear model (regression in this case) with our train data, we feed them in glmnet method with alpha = 0.
* Then we did simple 10 fold cross validation by using cv.glmnet on test and training data and we got plot between Mean squared error and Log of Lambda (shown in Fig2.1)
* Then we saw the best lambda we got i.e minimum lambda. It came out to be 7576.474.
* The Mean squared error we got finally using best lambda is – **547505516**. Which is less than the earlier mse value we got using best lambda and using other attributes.

**Using Linear Models and Generalized Linear Models:**

* We constructed linear regression model using lm() command and using all the variables. Then we summarized the resulting model using summary() command and noted that only “MS.SubClass”,”Lot.Area”,”Lot.Config”,”Neighborhood”,”House.Style” ”Overall.Qual”,”Overall.Cond”,”Year.Built”,”Year.Remod.Add”,”Roof.Style”,”Mas.Vnr.Type”,”Exter.Qual”,”Foundation”,”Bsmt.Qual”,”Bsmt.Exposure”,”BsmtFin.Type.1”, “BsmtFin.SF.1”,”Bsmt.Unf.SF”,”Total.Bsmt.SF”,”Heating.QC”,”X1st.Flr.SF”,”Gr.Liv.Area”,”Bedroom.AbvGr”,”Kitchen.Qual”,”Garage.Type”,”Garage.Finish”,”Garage.Cars”,”Garage.Area”,”Yr.Sold” attributes have significance values greater than or equal to 0.05. So, we decided to consider these attributes and eliminate the rest of the variables.
* Next we constructed a new data frame using the above attributes and converted it using model.matrix() method.
* We next constructed linear regression model on using lm() method taking only 2140 rows(80% of total data) as training data and the above dataframe.
* Next, we summarized the model in the above step and noted that the multiple r- squared error is 0.8902 . We also calculated the mean squared error using the mean() function and passing the residual squares obtained from the model as arguments. We found that the mean-squared error(MSE) is 683872834.
* Next, we calculated mean-squared error(MSE) of the predicted values of the model on the test data. We found this value as 592993380. We noted that there is a lot of difference between the MSE on test and training data.
* Next, we constructed regression model using glm(). We then calculated the cross validation error by using cv.glm(). The delta value(delta[1]) obtained is 683067783

Graphs:

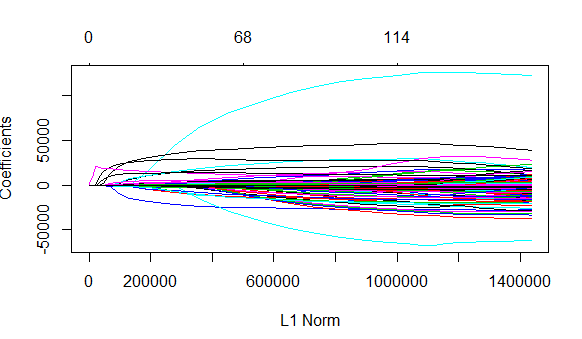
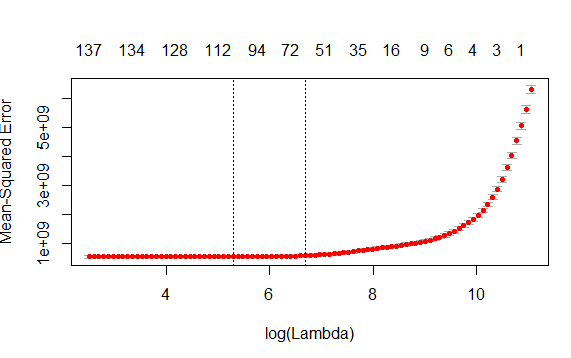
 

Fig:1.1 Coefficients Versus L1 Norm for Lasso Fig: 1.2 Mean-Squared Error versus Log(Lambda) for Lasso

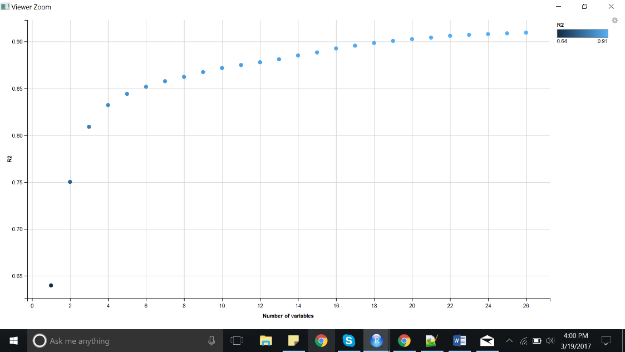
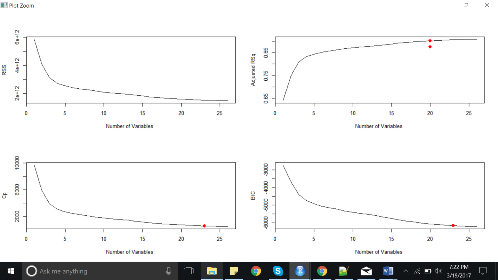
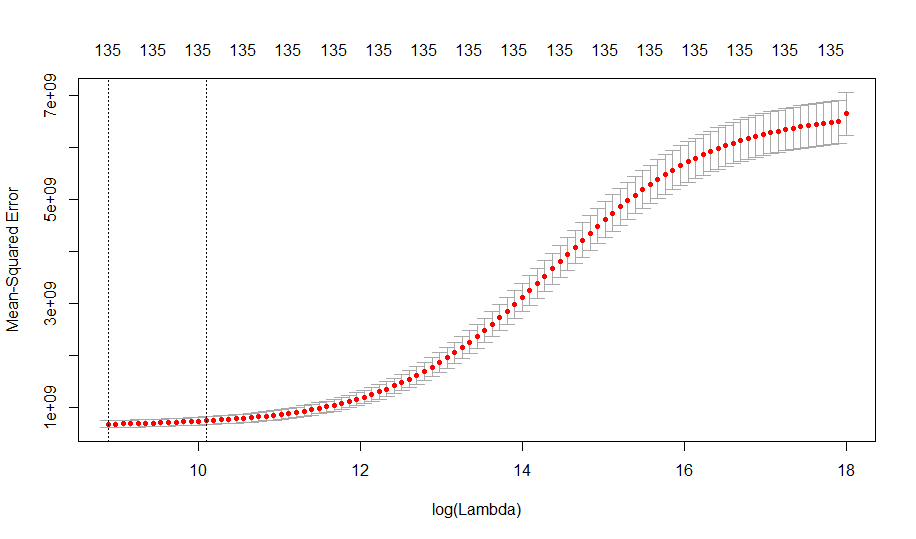
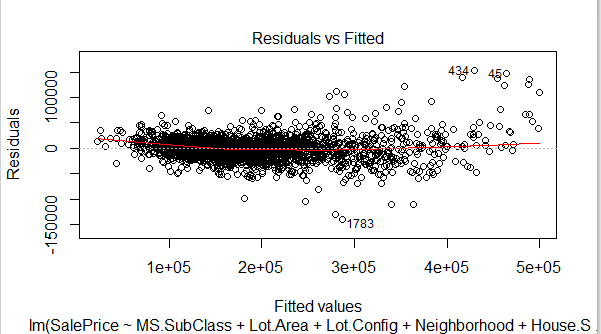
 

Fig:2.1 R2 Vs no. of Variables Fig 2.2 forward and backward susbset selection

  Fig:3.1 Mean Squared Error versus Log(Lambda) for ridge regression Fig:4.1 Residuals Vs Fitted using lm()

**Summary:**

In the Ames housing dataset, the number of attributes is very large. So, a lot of preprocessing had to be done inorder to eliminate the unnecessary attributes and leave only those attributes that add value to the models. We removed various attributes that had a lot NA values and also those which had very little variation. After all the preprocessing, we were left with a dataset containing 37 attributes. We then built models to predict the SalePrice.

For lasso model, we obtained the MSE value as 421309091. The square root of the MSE (which is equal to 20,525.81) when compared with the mean of the SalePrice(which is equal to 186000 ) indicates that lasso performs a very good job in predicting the Sale Price.

Using the cross-validation approach for forward/backward subset selection we got the lowest MSE as 1116094417 for 23 predictors.

When we used the ridge model we obtained the best MSE value as 547505516. Given the data and its attributes we can conclude that we can actually form a good regression model/predictor for this data if we carefully preprocess(Cleanup) the data first and try to form a predictor by selecting attributes which considerably contributes to the model

For model constructed using lm() method, the MSE value is 592993380 and for model constructed using glm() method, the delta value is 683067783 indicating that these methods tend to perform poorly when compared to the rest.

All the models constructed gave nearly similar MSE values. However, out of all the models, the model built using Lasso gave lower MSE value and the model built using forward and backward subset selection gave the highest MSE.