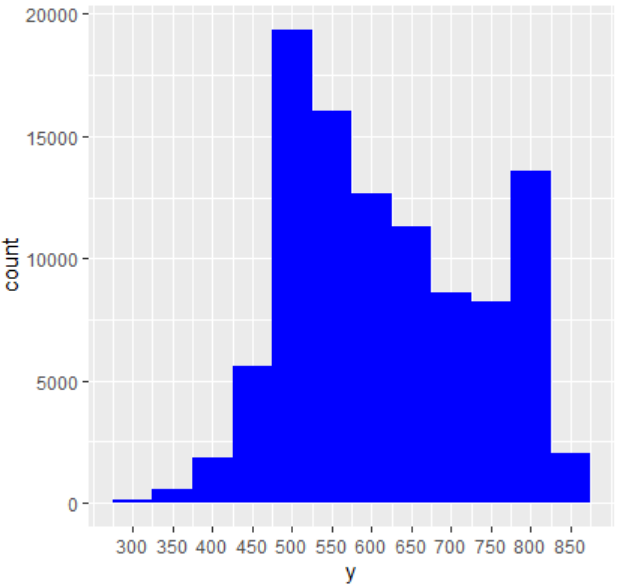
**Test Solution - Report**

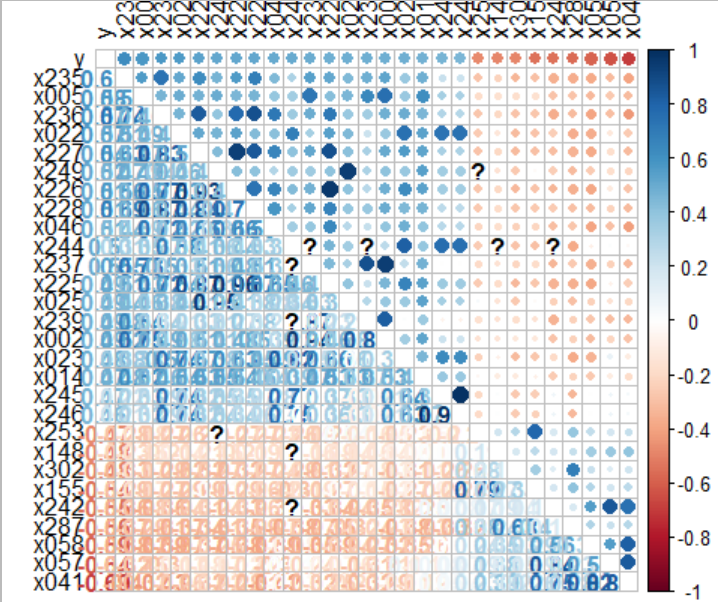
High level overview of the methodology used is as follows:

1. Simple analysis of the data provided => 304 variables and 1 target variable. Visualized the target variable to see its reach and pattern.

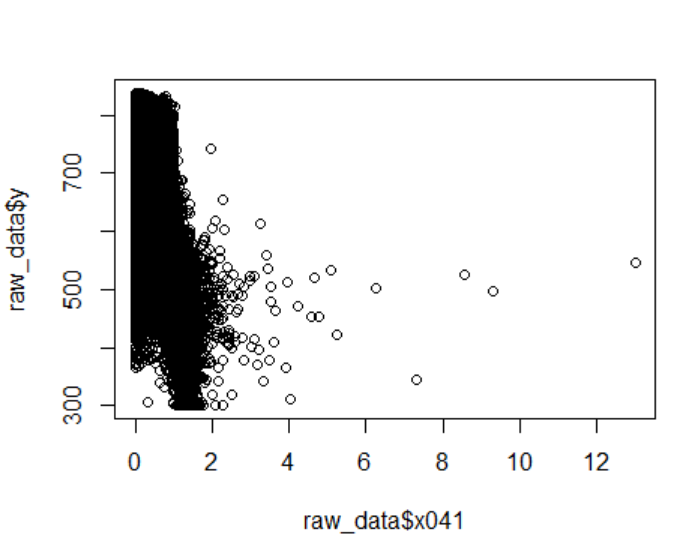
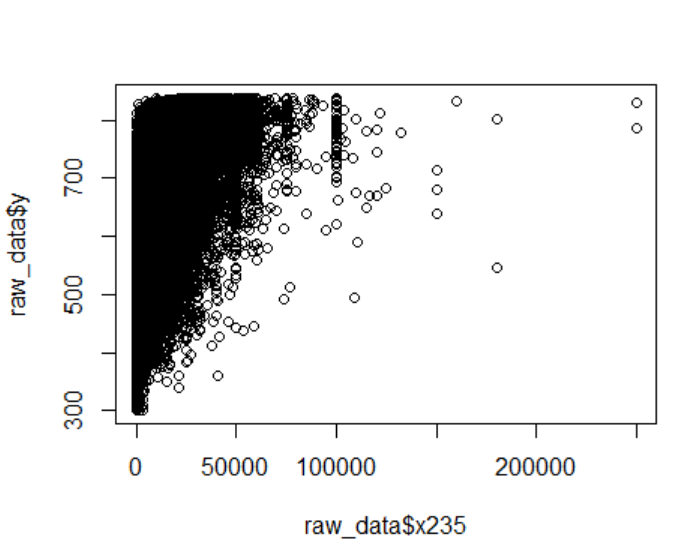


2. Many variables are of no value by looking at it by naked eye. E.g. some variables have same value for all rows. Need to remove these variables. Correlation analysis was applied on the dataset to see its correlation with target variable.

3. As correlation coefficient value ranges from -1 to 1, generally, the ones with absolute value >=0.5 are considered as highly correlated. But looking at this spectrum, ones with absolute value >=0.45 are considered for next stage.



4. For each of the selected variable, checked the plot with target variable.



Similarly, for other variables.

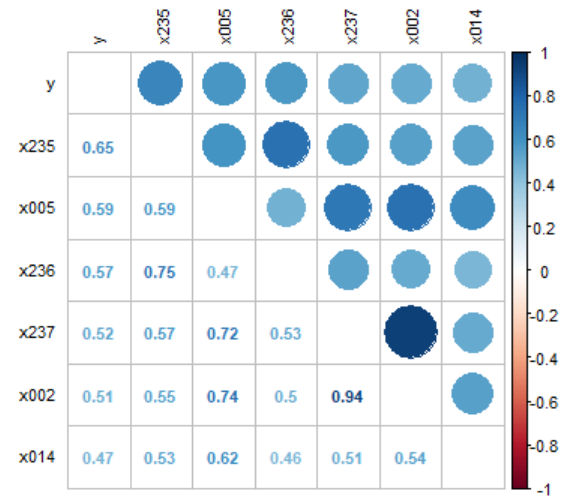
5. Checked for skewness and normality of the target variable. No skewness is found, so no change.

6. For missing values, MICE package was used. The mice package in R, helps imputing missing values with plausible data values. These plausible values are drawn from a distribution specifically designed for each missing datapoint.

7. For the columns which has numeric data but only has few values, they are converted into factors. For other columns, they are revalued by creating new bins because of low count of values.

8. After separating out numeric and factor columns, multicollinearity is identified from variables present. For numeric variables, it is identified from correlation analysis and for factor variables, it is identified from visual plot and Chi Square Tests. If multicollinearity is observed, then the ones with less importance with target variable is removed.

For Numeric Variables, threshold of 0.75 is used. If more than 0.75 correlation coefficient, then those are correlated. Hence, x002 and x237 are correlated and x237 is more correlated with y. So, x002 was removed.



9. For many columns, skewness was found. As there were many negative and zero values. Hence, log and other transformations were not possible. Cube root transformation was applied on that to remove skewness.

10. Some columns have high values. Hence, they were normalized.

11. For Factors, they are converted into many columns with one hot encoding.

12. Many columns are generated because of the above step. Hence, only important variables need to be selected for prediction. Boruta, a wrapper based on the Random Forest, is used to select important features.

13. Dataset is split into training data and test data in 80:20 ratio.

14. 3 different algorithms (Lasso, XGBoost and Random Forest) are applied on the training data.

15. 2 models are built based on the results of these algorithms. 1st model does the average predictions of all 3 algorithms and the 2nd model average of top 2 performing algorithms i.e. XGBoost and Random Forest.

Assumptions made:

1. Test data set contains 305 columns. x001 to x304 columns along with target column “y”.

2. Each of the columns in test set follow the same distribution as per the training data set.

3. There would be no missing values for “y”.

Algorithms and Techniques:

Data Imputation: MICE

Skewness Removal: Cube Root transformation

Feature Selection: Boruta

Algorithms for Model: Lasso, XGBoost, Random Forest

Other: One hot encoding

Tools and Frameworks used:

i) R – 3.6.1 ii) RStudio iii) Docker Desktop for Windows iv) Windows 10 Machine

Results and Evaluations:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No. | Model | MAE | RMSE | MAPE |
| 1 | Lasso | 47.10 | 61.90 | 8.06% |
| 2 | XG Boost | 44.47 | 58.98 | 7.62% |
| 3 | Random Forest | 44.32 | 58.46 | 7.62% |
| 4 | Model 1 (Average of prediction results of Lasso, XG Boost and Random Forest) | 44.41 | 58.76 | 7.62% |
| 5 | Model 2 (Average of prediction results of XG Boost and Random Forest i.e. top 2 performing models) | 44.07 | 58.31 | 7.56% |

Out of the 3 algorithms used, Random Forest gives the best results compared to others in all measures.

Model 2 improves the result (MAE, RMSE, MAPE) and model 1 gives the best accuracy %.

As accuracy is calculated by considering absolute error <=3 as correct and others being incorrect, it is not a good measure in this case. e.g. if target is 300 and error is 3, then expected prediction is less than 1%. But if target is 800 and error is 3, then expected prediction is 0.3%.

More useful evaluation metric would be RMSE, which punishes big errors than small ones. MAPE and MAE could be more linear. At the end of the day, best metric would be based on the user’s requirement and use case being involved.