Santander Customer Transaction Prediction

Date : 30-10-2019

Author : Pritam Sonawane

Introduction

Problem Statement

Data

Methodology

Pre Processing

Model Development

Decision tree for regression

Random Forest for regression

Logistic Regression classification

Naive bayes classification

Model Evaluation

Introduction

Problem Statement:

In this challenge, we need to identify which customers will make a specific transaction in the future, irrespective of the amount of money transacted.

Data-set :

There are 200,000 observations in the training and testing dataset. In addition to this there is also the target column in training data which has two boolean values, 0 or 1. and 200 independent variables in both training and testing dataset.

Training dataset:

```
In [12]: head(train_df)
```

A data.frame: 6 × 202

ID_code	target	var_0	var_1	var_2	var_3	var_4	var_5	var_6	var_7	⋯	var_190	var_191	var_192	var_193	var_194	var_195	var_196	va
<chr>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	⋯	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
train_0	0	8.9255	-6.7863	11.9081	5.0930	11.4607	-9.2834	5.1187	18.6266	⋯	4.4354	3.9642	3.1364	1.6910	18.5227	-2.3978	7.8784	
train_1	0	11.5006	-4.1473	13.8588	5.3890	12.3622	7.0433	5.6208	16.5338	⋯	7.6421	7.7214	2.5837	10.9516	15.4305	2.0339	8.1267	
train_2	0	8.6093	-2.7457	12.0805	7.8928	10.5825	-9.0837	6.9427	14.6155	⋯	2.9057	9.7905	1.6704	1.6858	21.6042	3.1417	-6.5213	
train_3	0	11.0604	-2.1518	8.9522	7.1957	12.5846	-1.8361	5.8428	14.9250	⋯	4.4666	4.7433	0.7178	1.4214	23.0347	-1.2706	-2.9275	1(
train_4	0	9.8369	-1.4834	12.8746	6.6375	12.2772	2.4486	5.9405	19.2514	⋯	-1.4905	9.5214	-0.1508	9.1942	13.2876	-1.5121	3.9267	
train_5	0	11.4763	-2.3182	12.6080	8.6264	10.9621	3.5609	4.5322	15.2255	⋯	-6.3068	6.6025	5.2912	0.4403	14.9452	1.0314	-3.6241	

Testing dataset

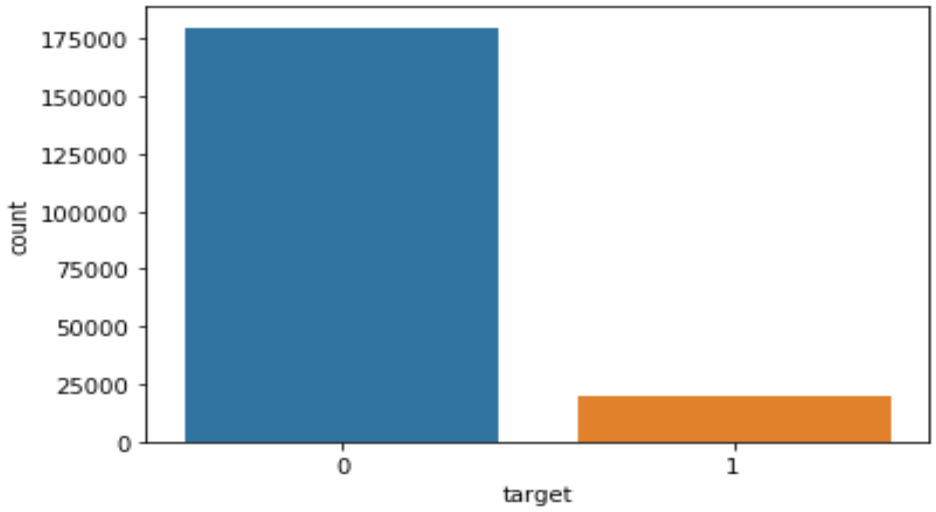
```
head(test_df)
```

A data.frame: 6 × 201

ID_code	var_0	var_1	var_2	var_3	var_4	var_5	var_6	var_7	var_8	...	var_190	var_191	var_192	var_193	var_194	var_195	var_196
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	...	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
test_0	11.0656	7.7798	12.9536	9.4292	11.4327	-2.3805	5.8493	18.2675	2.1337	...	-2.1556	11.8495	-1.4300	2.4508	13.7112	2.4669	4.3654
test_1	8.5304	1.2543	11.3047	5.1858	9.1974	-4.0117	6.0196	18.6316	-4.4131	...	10.6165	8.8349	0.9403	10.1282	15.5765	0.4773	-1.4852
test_2	5.4827	-10.3581	10.1407	7.0479	10.2628	9.8052	4.8950	20.2537	1.5233	...	-0.7484	10.9935	1.9803	2.1800	12.9813	2.1281	-7.1086
test_3	8.5374	-1.3222	12.0220	6.5749	8.8458	3.1744	4.9397	20.5660	3.3755	...	9.5702	9.0766	1.6580	3.5813	15.1874	3.1656	3.9567
test_4	11.7058	-0.1327	14.1295	7.7506	9.1035	-8.5848	6.8595	10.6048	2.9890	...	4.2259	9.1723	1.2835	3.3778	19.5542	-0.2860	-5.1612
test_5	5.9862	-2.2913	8.6058	7.0685	14.2465	-8.6761	4.2467	14.7632	1.8790	...	-2.1115	7.1178	-0.4249	8.8781	14.9438	-2.2151	-6.0233

In this training dataset there is a target variable having values in the form of 0 and 1.

The data distribution is as follows:



There is around 90% values are 0 and 10 % values are 1. Data is quite imbalanced , imbalanced data set can be handled by data shuffling or data sampling.

Methodology

Pre Processing

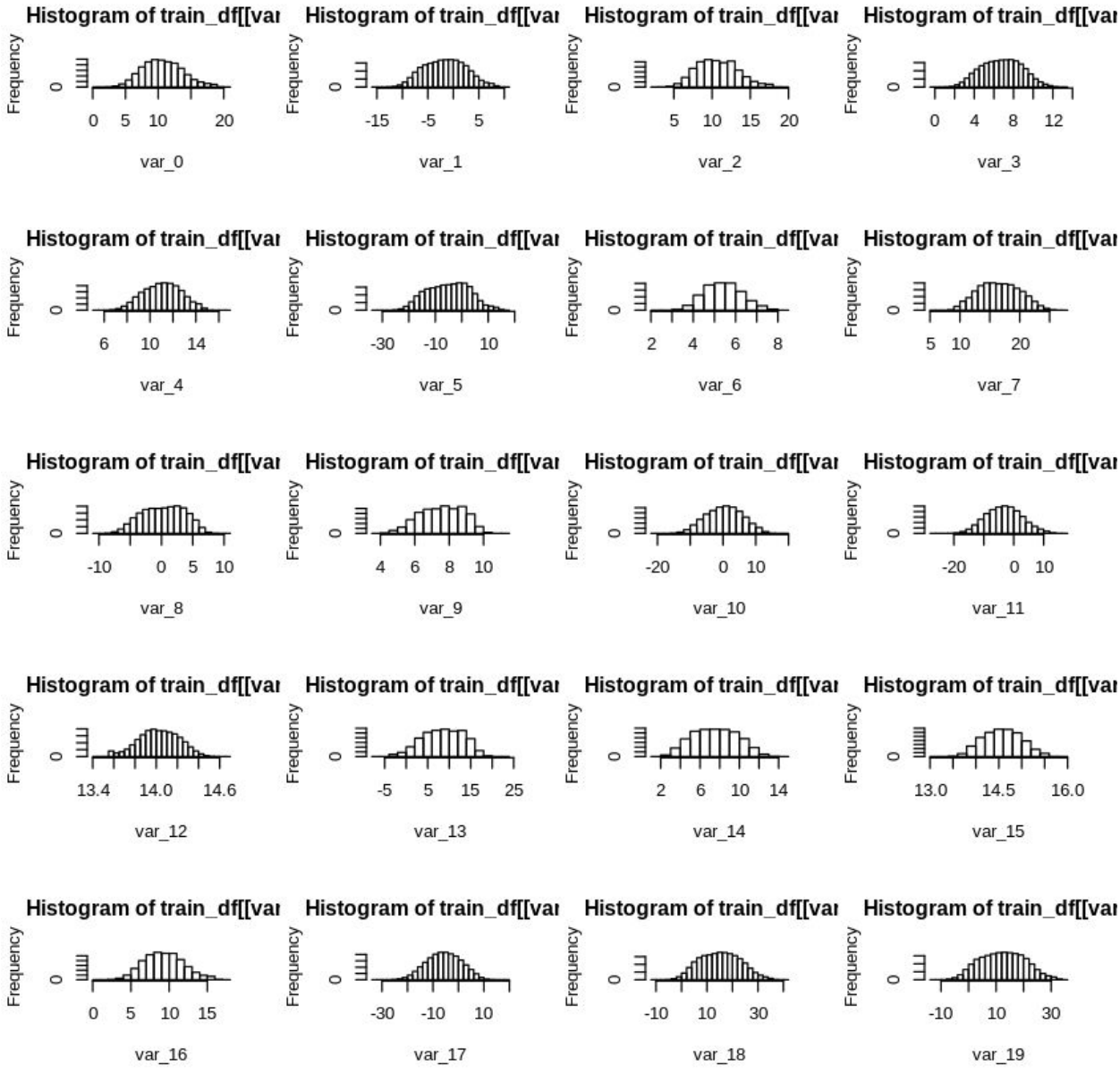
Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format.
Steps Involved in Data Preprocessing:

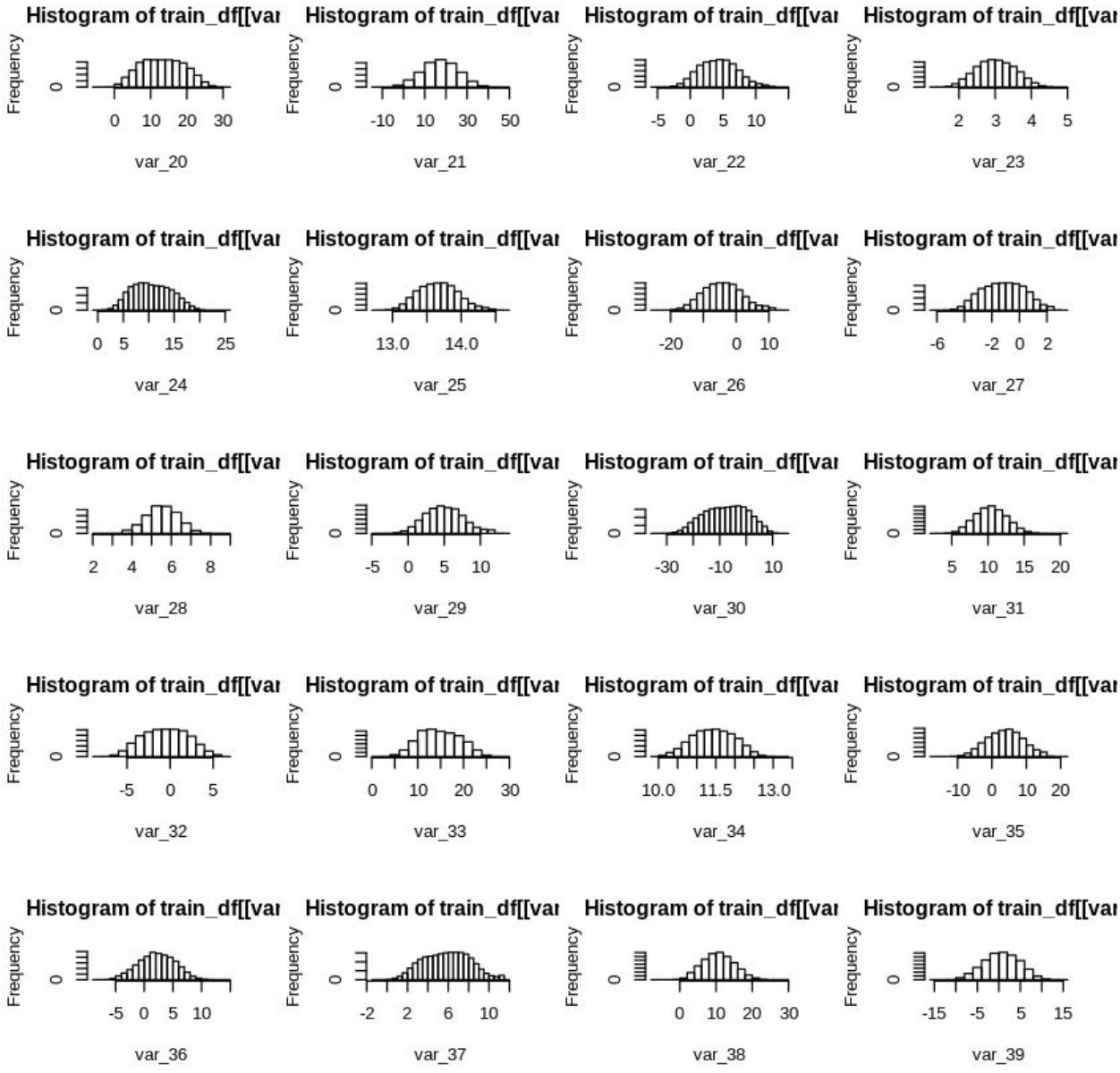
1. Data Cleaning:

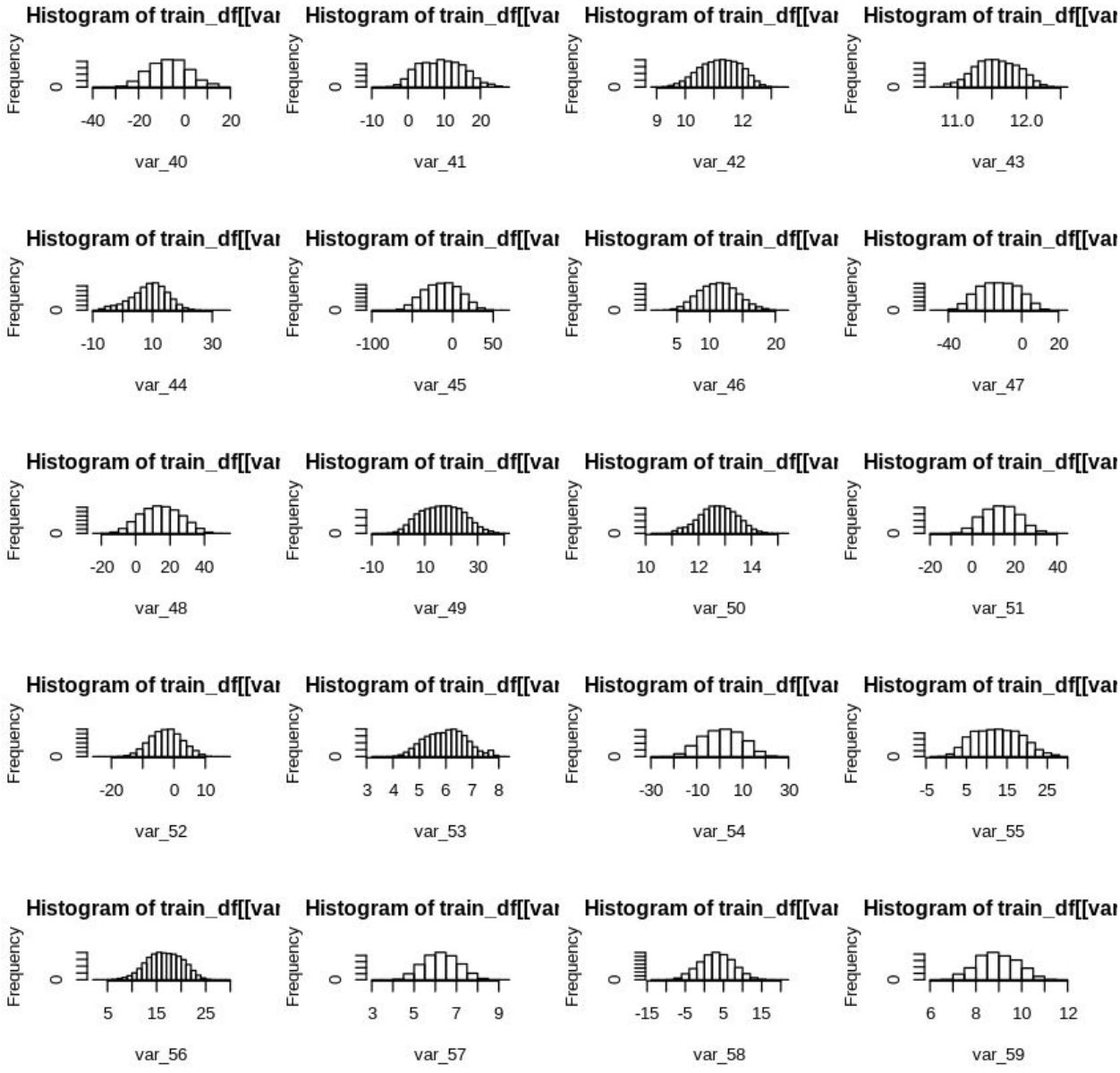
This step is important because in most situations data provided by the customer has a bad quality or just cannot be directly fed to some kind of ML model. It includes data type conversion, data validation, handling dates, handling nominal and categorical variables. But in this case dataset variable data type is as follows:

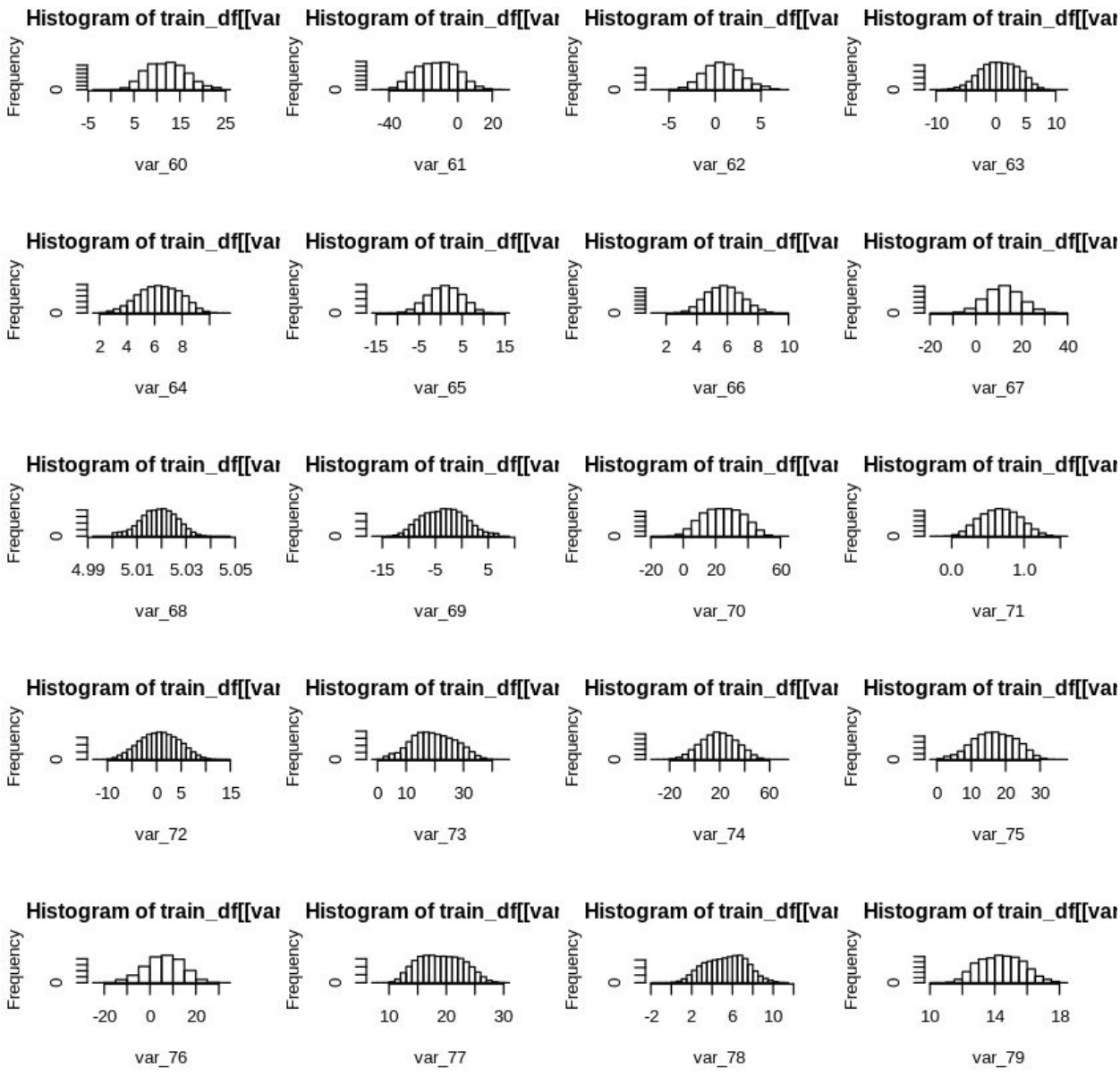
- Here we need to convert target variable which is in numeric form into factor because we are developing a classification model if we keep it as it is model will treat it as regression model.

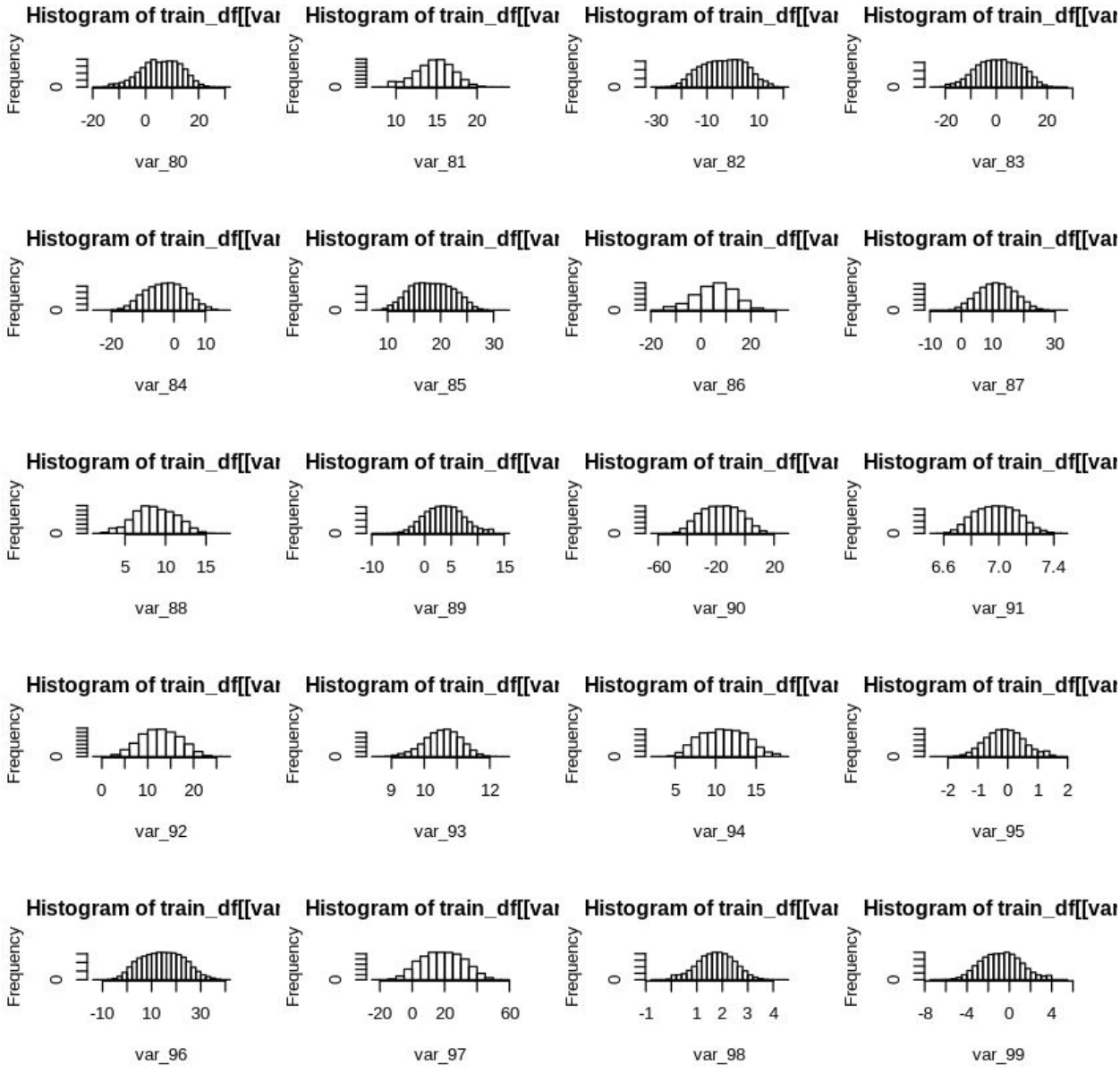
Following Histogram displays the distribution of all numerical features per each class

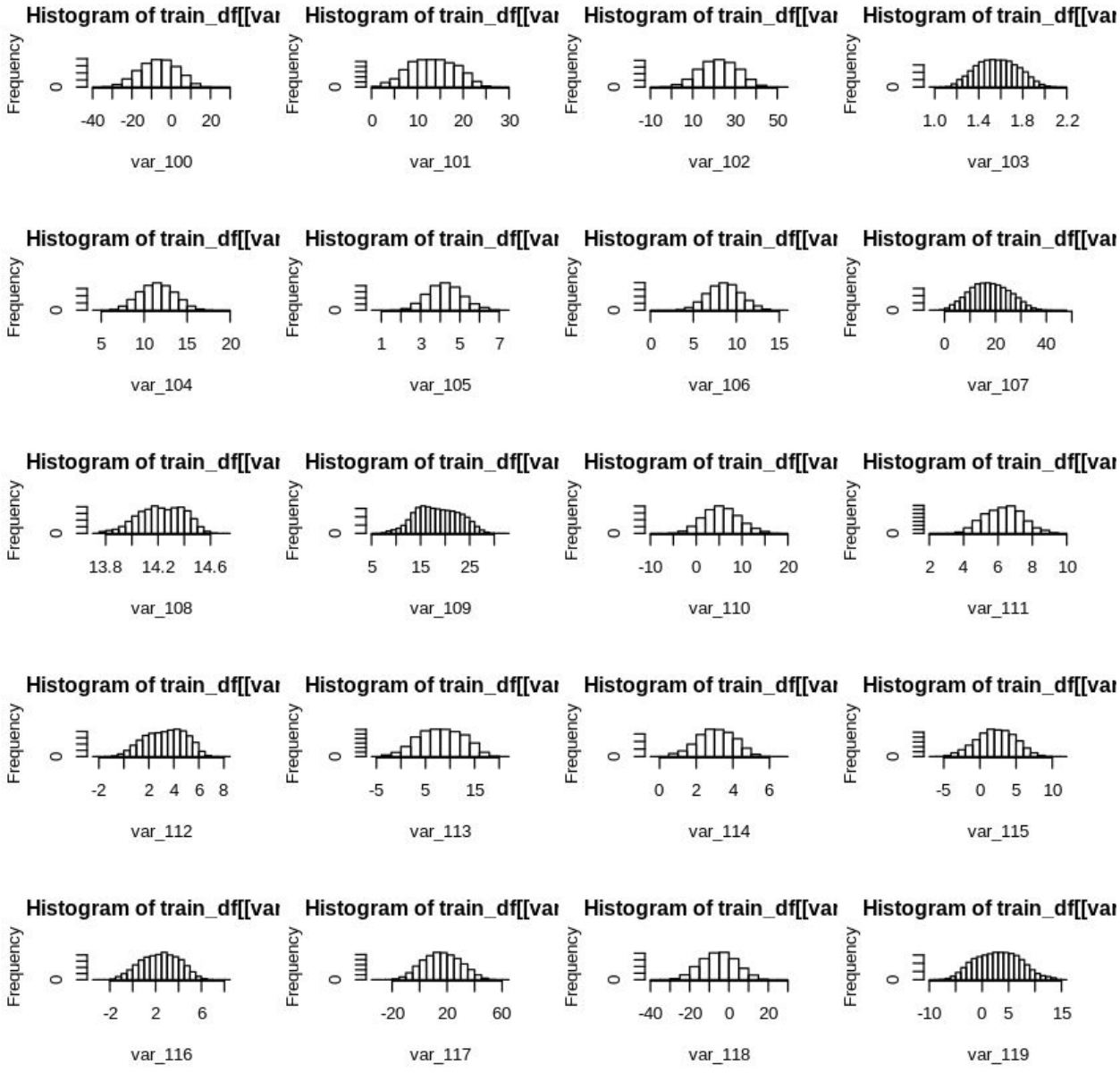

Histogram of train_df[[var
Frequency
var_0
var_1
var_2
var_3
var_4
var_5
var_6
var_7
var_8
var_9
var_10
var_11
var_12
var_13
var_14
var_15
var_16
var_17
var_18
var_19

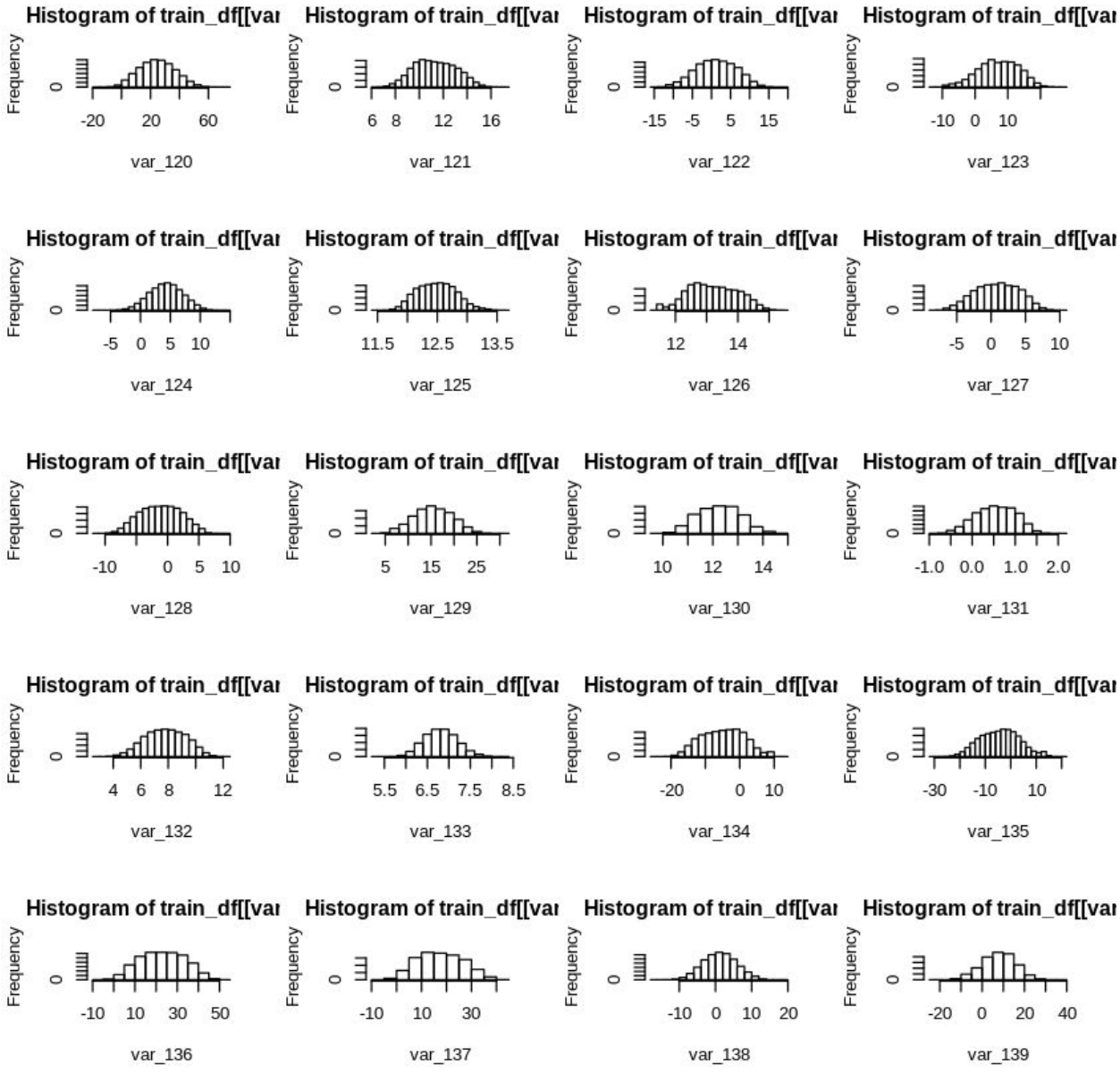

Histogram of train_df[[var
Frequency
var_20
var_21
var_22
var_23
var_24
var_25
var_26
var_27
var_28
var_29
var_30
var_31
var_32
var_33
var_34
var_35
var_36
var_37
var_38
var_39

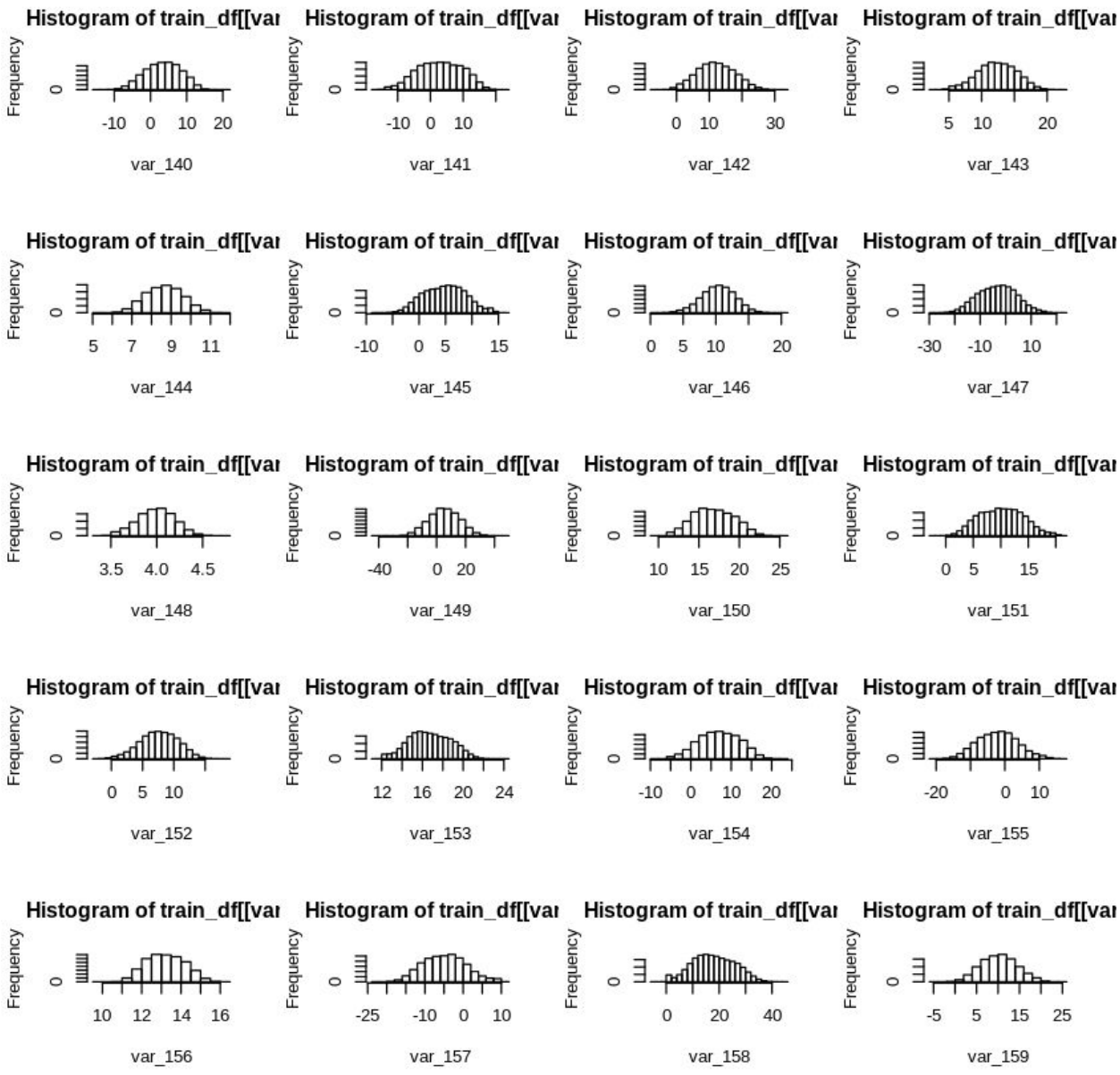

Histogram of train_df[[var
Frequency
var_40
var_41
var_42
var_43
var_44
var_45
var_46
var_47
var_48
var_49
var_50
var_51
var_52
var_53
var_54
var_55
var_56
var_57
var_58
var_59

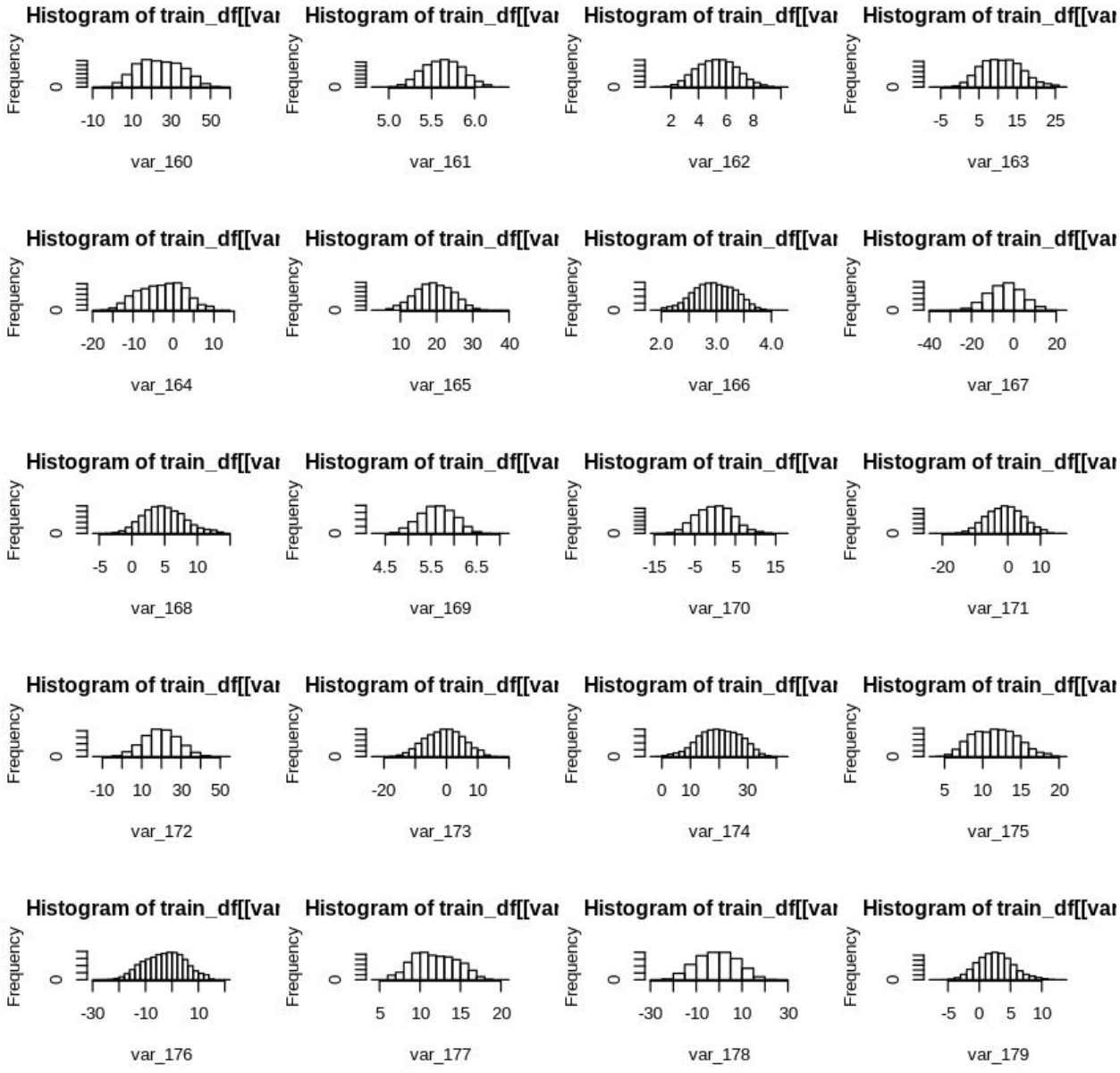

Histogram of train_df[[var
Frequency
0
-5 5 15 25
var_60
Histogram of train_df[[var
Frequency
0
-40 0 20
var_61
Histogram of train_df[[var
Frequency
0
-5 0 5
var_62
Histogram of train_df[[var
Frequency
0
-10 0 5 10
var_63
Histogram of train_df[[var
Frequency
0
2 4 6 8
var_64
Histogram of train_df[[var
Frequency
0
-15 -5 5 15
var_65
Histogram of train_df[[var
Frequency
0
2 4 6 8 10
var_66
Histogram of train_df[[var
Frequency
0
-20 0 20 40
var_67
Histogram of train_df[[var
Frequency
0
4.99 5.01 5.03 5.05
var_68
Histogram of train_df[[var
Frequency
0
-15 -5 5
var_69
Histogram of train_df[[var
Frequency
0
-20 0 20 60
var_70
Histogram of train_df[[var
Frequency
0
0.0 1.0
var_71
Histogram of train_df[[var
Frequency
0
-10 0 5 15
var_72
Histogram of train_df[[var
Frequency
0
0 10 30
var_73
Histogram of train_df[[var
Frequency
0
-20 20 60
var_74
Histogram of train_df[[var
Frequency
0
0 10 20 30
var_75
Histogram of train_df[[var
Frequency
0
-20 0 20
var_76
Histogram of train_df[[var
Frequency
0
10 20 30
var_77
Histogram of train_df[[var
Frequency
0
-2 2 6 10
var_78
Histogram of train_df[[var
Frequency
0
10 14 18
var_79

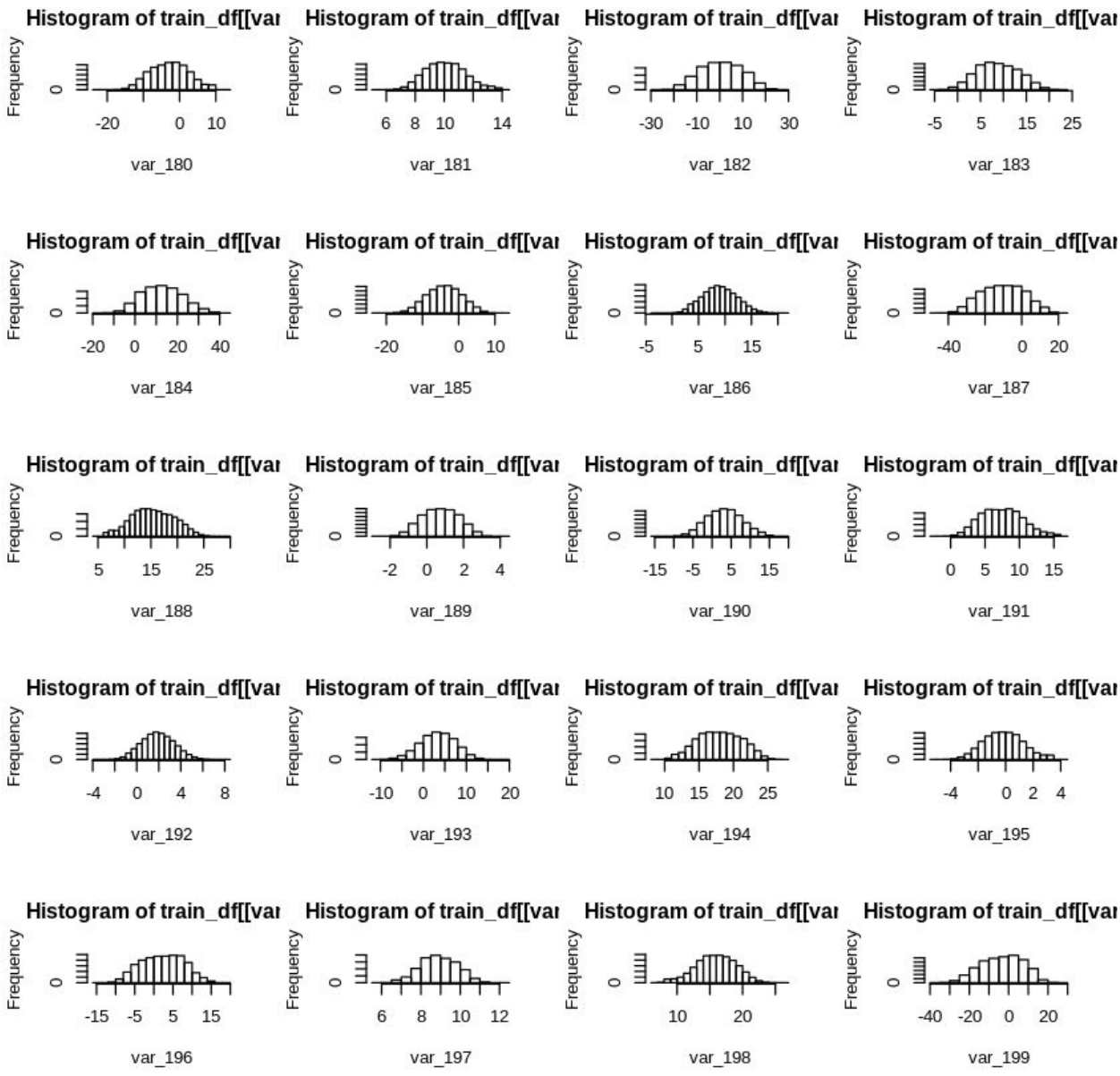

Histogram of train_df[[var
Frequency
var_80
var_81
var_82
var_83
var_84
var_85
var_86
var_87
var_88
var_89
var_90
var_91
var_92
var_93
var_94
var_95
var_96
var_97
var_98
var_99


Histogram of train_df[[var
Frequency
var_100
var_101
var_102
var_103
var_104
var_105
var_106
var_107
var_108
var_109
var_110
var_111
var_112
var_113
var_114
var_115
var_116
var_117
var_118
var_119


Histogram of train_df[[var
Frequency
var_120
var_121
var_122
var_123
var_124
var_125
var_126
var_127
var_128
var_129
var_130
var_131
var_132
var_133
var_134
var_135
var_136
var_137
var_138
var_139


Histogram of train_df[[var
Frequency
var_140
var_141
var_142
var_143
var_144
var_145
var_146
var_147
var_148
var_149
var_150
var_151
var_152
var_153
var_154
var_155
var_156
var_157
var_158
var_159


Histogram of train_df[[var
Frequency
var_160
var_161
var_162
var_163
var_164
var_165
var_166
var_167
var_168
var_169
var_170
var_171
var_172
var_173
var_174
var_175
var_176
var_177
var_178
var_179



From above histograms we can say that almost all features follows normal distribution.

The standard deviation controls the spread of the distribution. A smaller standard deviation indicates that the data is tightly clustered around the mean; the normal distribution will be taller. A larger standard deviation indicates that the data is spread out around the mean; the normal distribution will be flatter and wider.

2. Missing Values Analysis

The concept of missing values is important to understand in order to successfully manage data. If the missing values are not handled properly then we may end up drawing an inaccurate inference about the data.

We can impute missing values by mean ,median of variable or for categorical variable we use mode.

Here there is no missing value found in the data set .

3. Outlier analysis:

An outlier is an element of a data set that distinctly stands out from the rest of the data. It can affect the overall observation made from the data series.

Outliers are very important because they affect the mean and median which in turn affects the error (absolute and mean) in any data set. When you plot the error you might get big deviations if outliers are in the data set.

In this dataset I have imputed outliers by replacing values with mean value.

4. Correlation analysis :

A Correlation matrix represent the relationship between two variables. Explore the relationship between scatterplots and correlations

correlations have two properties: strength and direction. The **strength** of a correlation is determined by its numerical value. The **direction** of the correlation is determined by whether the correlation is positive or negative.

Positive correlation: Both variables move in the same direction. In other words, as one variable increases, the other variable also increases. As one variable decreases, the other variable also decreases.

Negative correlation: The variables move in opposite directions. As one variable increases, the other variable decreases. As one variable decreases, the other variable increases.

```
In [20]: train_df$target<-as.numeric(train_df$target)
         train_correlations <- cor(train_df[,c(2:202)])
         train_correlations
```

A matrix: 201 × 201 of type dbl

	target	var_0	var_1	var_2	var_3	var_4	var_5	var_6	var_7	var
target	1.0000000000	5.253472e-02	5.036960e-02	5.585854e-02	1.110990e-02	1.092479e-02	3.096018e-02	6.686773e-02	-0.0030501129	1.957457e-
var_0	0.0525347238	1.000000e+00	-5.176865e-04	6.458643e-03	3.499149e-03	1.336080e-03	3.415320e-03	7.420855e-03	0.0023259584	5.072480e-
var_1	0.0503696032	-5.176865e-04	1.000000e+00	4.060420e-03	2.418031e-05	1.323930e-04	-9.072044e-04	3.261139e-03	0.0014414408	4.160315e-
var_2	0.0558585358	6.458643e-03	4.060420e-03	1.000000e+00	1.007775e-03	6.206326e-04	1.754570e-03	9.283694e-04	-0.0009706660	2.765832e-
var_3	0.0111098982	3.499149e-03	2.418031e-05	1.007775e-03	1.000000e+00	-3.010952e-04	3.174832e-03	-5.693194e-04	0.0025155960	3.637596e-
var_4	0.0109247890	1.336080e-03	1.323930e-04	6.206326e-04	-3.010952e-04	1.000000e+00	-1.226677e-03	-1.214738e-05	0.0044176245	1.104992e-

From above calculation We can observed that the correlation between the training and testing attributes is very small.

There are 200 features that are mostly un-correlated between them.

5. Normalization

Normalization is the process of reducing unwanted variation either within or between variables. Normalization bring all of the variables into proportion with one another on common scale.

To bring data points to same range we used the following formula :

$$\text{New\_value} = \frac{Values - minValues}{maxValues - minValues}$$

Model Development

Model Selection :

From the earlier analysis on dataset we know that our target variable contains continuous values ,so will use regression machine learning model.

Divided data into train and test

Divided the data into 80% training and 20% testing dataset.

Random Forest for regression

A random forest allows us to determine the most important predictors across the explanatory variables by generating many decision trees and then ranking the variables by importance.

```
In [33]: rf_2=randomForest(target ~ . , data = training_data, ntree=50)
```

```
Call:
 randomForest(formula = target ~ ., data = training_data, ntree = 50)
               Type of random forest: classification
                     Number of trees: 50
No. of variables tried at each split: 14

        OOB estimate of  error rate: 10%
Confusion matrix:
       0   1  class.error
0 143842  80 0.0005558566
1  15916 163 0.9898625536
```

Number of variables randomly sampled as candidates at each split are 14

Number of trees = 50

Error rate = 10 %

Performance of classification model :

```
In [39]: ##Evaluate the performance of classification model
         ConfMatrix_RF = table(testing_data$target, pred)
         confusionMatrix(ConfMatrix_RF)

         Confusion Matrix and Statistics

            pred
                 0     1
           0 35979     1
           1  4015     4

                        Accuracy : 0.8996
                          95% CI : (0.8966, 0.9025)
             No Information Rate : 0.9999
             P-Value [Acc > NIR] : 1

                           Kappa : 0.0017

          Mcnemar's Test P-Value : <2e-16

                     Sensitivity : 0.8996099
                     Specificity : 0.8000000
                  Pos Pred Value : 0.9999722
                  Neg Pred Value : 0.0009953
                      Prevalence : 0.9998750
                  Detection Rate : 0.8994975
            Detection Prevalence : 0.8995225
               Balanced Accuracy : 0.8498050

                'Positive' Class : 0
```

From above confusion matrix **accuracy is 89.96 %.**

Logistic Regression

The training dataset the dependent variable have values in the form of 0 and 1. The dependent variable isn't normally distributed. Logistic Regression belongs to the family of generalized linear models. It is a binary classification algorithm used when the response variable is dichotomous (1 or 0). Inherently, it returns the set of probabilities of target class. Following are the assumptions made by Logistic Regression:

1. The response variable must follow a binomial distribution.
2. Logistic Regression assumes a linear relationship between the independent variables and the link function (logit).
3. The dependent variable should have mutually exclusive and exhaustive categories.

In R, you can implement Logistic Regression using the glm function. Now, let's understand and interpret the crucial aspects of summary:

1. The glm function internally encodes categorical variables into n - 1 distinct levels.
2. Estimate represents the regression coefficients value. Here, the regression coefficients explain the change in log(odds) of the response variable for one unit change in the predictor variable.
3. Std. Error represents the standard error associated with the regression coefficients.
4. z value is analogous to t-statistics in multiple regression output. z value > 2 implies the corresponding variable is significant.
5. p value determines the probability of significance of predictor variables. With 95% confidence level, a variable having p < 0.05 is considered an important predictor. The same can be inferred by observing stars against p value.

```
Confusion Matrix and Statistics

   logit_Predictions
       0     1
  0 35510   470
  1  2963  1056

               Accuracy : 0.9142
                 95% CI : (0.9114, 0.9169)
    No Information Rate : 0.9618
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3446

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.9230
            Specificity : 0.6920
         Pos Pred Value : 0.9869
         Neg Pred Value : 0.2628
             Prevalence : 0.9618
         Detection Rate : 0.8878
   Detection Prevalence : 0.8995
      Balanced Accuracy : 0.8075

       'Positive' Class : 0
```

From above confusion matrix for Logistic regression we can say that the accuracy of the model is 91.42 %.

Accuracy - It determines the overall predicted accuracy of the model. It is calculated as Accuracy = (True Positives + True Negatives)/(True Positives + True Negatives + False Positives + False Negatives)

Naive bayes Classification

Naive Bayes algorithm is the algorithm that learns the probability of an object with certain features belonging to a particular group/class. In short, it is a probabilistic classifier.

The probability of an event based on previous knowledge available on the events. More formally, Bayes' Theorem is stated as the following equation:

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)}$$

P(A/B) : Probability (conditional probability) of occurrence of event A given the event B is true.

P(A) and P(B) : Probabilities of the occurrence of event A and B respectively.

P(B/A) : Probability (conditional probability) of occurrence of event B given the event A is true

Advantages :

- It is a relatively easy algorithm to build and understand.
- It is faster to predict classes using this algorithm than many other classification algorithms.

```
#Look at confusion matrix
Conf_matrix = table(observed = testing_data$target, predicted = NB_Predictions)
confusionMatrix(Conf_matrix)
```

```
Confusion Matrix and Statistics

        predicted
observed     0     1
       0 35420   560
       1  2554  1465

               Accuracy : 0.9221
                 95% CI : (0.9195, 0.9248)
    No Information Rate : 0.9494
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4476

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.9327
            Specificity : 0.7235
         Pos Pred Value : 0.9844
         Neg Pred Value : 0.3645
             Prevalence : 0.9494
         Detection Rate : 0.8855
   Detection Prevalence : 0.8995
      Balanced Accuracy : 0.8281

       'Positive' Class : 0
```

From above confusion matrix the accuracy of the naive bayes classification is 92.21%.