Data Mining and Predictive Modelling

CT119-3-2-DMPM

Group Assignment

APD2F2011CS(DA), APU2F2011CS(DA)

Name	TP Number
Law Pei Lin	TP051691
MD Nowshad UL Alam	TP057903
Tarun Aitha	TP058015

Table of Contents

List of Figures iv
List of Tables vii
Introduction 1
Process Flow Diagram 2
1.0 Sample (Data Selection) 3
2.0 Explore (Exploratory Data Analysis) 7
2.1 Statistical Analysis (Class Variable): 8
2.2 Statistical Analysis (Interval Variable) 13
2.3 Variable Worth 15
2.4 Chi-Square Plot 15
2.5 Graph Analysis 17
3.0 Modify (Data Pre-processing): 34
4.0 Model 47
4.1 NEURAL NETWROK (Nowshad) 47
Neural Network (before any Pruning) 47
Neural Network (after Pruning) 49
4.2 HP TREE (Nowshad) 55
HP TREE (Before Pruning) 55
HP TREE (After Pruning) 57
4.3 Decision Tree (Tarun Aitha TP058015) 64
Decision tree(Before Pruning) 64
Decision tree (After pruning) 66
4.4 Stepwise Regression (Tarun Aitha TP058015) 71
Stepwise Regression (Before Pruning) 71
4.5 Forward Logistic Regression – Law Pei Lin (TP051691) 76
Model Construction 76
Model Validation 77
Experiment by using Default Filter 79
4.6 HP Forest (Random Forest) – Law Pei Lin (TP051691) 86
Model Construction 86
Model Validation and Evaluation 88
5.0 Assess (Model Comparison) 93
6.0 Conclusion 95
7.0 References: 96

List of Figures

Figure 1: Process Flow Diagram 2
Figure 2: Sample->File Import 3
Figure 3: File import Node 3
Figure 4: Properties 4
Figure 5: Importing the file 4
Figure 6: View the data set 5
Figure 7: Rejecting variables 5
Figure 8: Target variables 6
Figure 9: Merge Node 6
Figure 10: Modification 6
Figure 11: Merging of the data set 6
Figure 12: Explore Process FLow Diagram 7
Figure 13: Variable Worth 15
Figure 14: Chi-Square Plot 15
Figure 15: Status Group Bar Chart 17
Figure 16: Quantity 18
Figure 17: Quantity Group 18
Figure 18: Waterpoint Type 20
Figure 19: Waterpoint type group 20
Figure 20: Extraction Type 21
Figure 21: Extraction Type Class 22
Figure 22: Extraction Type Group 22
Figure 23: Region 23
Figure 24: Payment Type 24
Figure 25: Payment 24
Figure 26: Source 25
Figure 27: Source Class 26
Figure 28: Source Type 26
Figure 29: Water Quality 27
Figure 30: Quality Group 27
Figure 31: Management Bar Chart 28
Figure 32: Management Group 29
Figure 33: Waterpoint Type Group 29
Figure 34: amount_tsh 30
Figure 35: Construction Year 31
Figure 36: Quantity Vs. Status Group 32
Figure 37: Waterpoint Vs. Status Group 32
Figure 38: Extraction Type Vs. Status Group 33
Figure 39: Region Vs. Status Group 34
Figure 40: Data Pre-Processing Process Flow Diagram 35
Figure 41: Variable selection 36
Figure 42: Interval Variables 37
Figure 43: Longtiude 37
Figure 44: gps_height 38
Figure 45: Properties 38
Figure 46: Properties 39

Figure 47: Imputation Summary 39
Figure 48: Public_Meeting 40
Figure 49: Public_Meeting(Imputed) 40
Figure 50: Scheme_Managemt 40
Figure 51: Scheme_management(Imputed) 41
Figure 52: Train Varibales 41
Figure 53: Replacement 42
Figure 54: After Replacement 42
Figure 55: Properties 43
Figure 56: Data 43
Figure 57: District_code 44
Figure 58: District_Code(Transformed) 44
Figure 59: Population 45
Figure 60: Population(Transformed) 45
Figure 61: Construction Year 46
Figure 62: Construction Year (Transformed) 46
Figure 63: Data Allocations 47
Figure 64: Neural Network 48
Figure 65: Validation 48
Figure 66: Measurement level 49
Figure 67: Iteration Plot 49
Figure 68: Forward Iteration 50
Figure 69: Backward Iteration 50
Figure 70: StepWise Iteration 50
Figure 71: 20 Units 51
Figure 72: 30 Units 51
Figure 73: 40 Units 51
Figure 74: iteration Plot 52
Figure 75: iteration Plot 52
Figure 76 iteration Plot 53
Figure 77: Missclassification Rate 53
Figure 78: Fit Statisitics 54
Figure 79: Target Variable Validation 54
Figure 80: Outcomes 54
Figure 81: Validation (Score Method) 56
Figure 82: HP Tree 56
Figure 83: Splitting Rule 56
Figure 84: Missclassifaction Rate 57
Figure 85: Model Events 57
Figure 86: Variable Summary 57
Figure 87: Sub-Tree 58
Figure 88: Number of leaves 58
Figure 89: iteration Plot 59
Figure 90: iteration Plot 59
Figure 91: iteration Plot 59
Figure 92: iteration Plot 60
Figure 93: iteration Plot 60
Figure 94: iteration Plot 60

Figure 95: iteration Plot 61
Figure 96: iteration Plot 61
Figure 97: iteration Plot 61
Figure 98: Replacament Graph 62
Figure 99: Bar Chat for Status_Group (Functional) 62
Figure 100: Bar Chat for Status_Group (Functional needs repair) 63
Figure 101: Bar Chat for Status_Group (non-Functional) 63
Figure 102: Score Implemtation 64
Figure 103: Score Method 64
Figure 104: Decision Tree 65
Figure 105: Model Events 65
Figure 106: Splitting Rule 65
Figure 107: Decsion Tree 66
Figure 108: Misclassification Rate 66
Figure 109: Model Modification 67
Figure 110: Decision Tree(Before Pruning) 67
Figure 111: Misclassification Rate 68
Figure 112: Model Modification 68
Figure 113: Decision Tree (After Pruning) 69
Figure 114: Misclassification Rate 69
Figure 115: Classification Table 70
Figure 116: StepWise Regression 72
Figure 117: Model Modification 72
Figure 118: MIsclassification Rate 73
Figure 119: Iteration Plot 73
Figure 120: Iteration Plot 74
Figure 121: Iteration Plot 74
Figure 122: Iteration Plot 75
Figure 123: Iteration Plot 75
Figure 124: Score 76
Figure 125: Validation 76
Figure 126: Logit Link function (MacKenzie et al., 2018) 77
Figure 127: Regression Node 77
Figure 128: Properties of Forward Logistic Regression Node 77
Figure 129: Iteration Plot of Forward Logistic Regression (Before Filtering) 78
Figure 130: Summary of Forward Logistic Regression 78
Figure 131: Analysis of Effect 79
Figure 132: Odds Ratio Estimates of Forward Logistic Regression 79
Figure 133: Odds Ratio Estimates of Forward Logistic Regression (2) 80
Figure 134: Event Classification Table of Forward Logistic Regression 80
Figure 135: Properties of Filter Node 81
Figure 136: Iteration plot of Forward Logistic Regression (After Filtering) 81
Figure 137: Summary of Forward Logistic Regression (After Filtering) 82
Figure 138: Analysis of Effects For Forward Logistic Regression (After Filtering) 82
Figure 139: Odds Ratio Estimation Forward Logistic Regression (After Filtering) 83
Figure 140: Odds Ratio Estimation Forward Logistic Regression (After Filtering) (2) 83
Figure 141: Classification Table of Validate Model 84

Figure 142: Confusion Matrix / Event Classification of Forward Logistic Regression (After Filtering) 84
Figure 143: Iteration Plot (1) 86
Figure 144: Iteration Plot (2 86
Figure 145: Iteration Plot (3) 86
Figure 146: HP Forest Node (1) 87
Figure 147: HP Forest Node (2) 87
Figure 148: Default Properties of HP Forest Node 87
Figure 149: Iteration Plot (1) 88
Figure 150: Iteration Plot (2) 88
Figure 151: Iteration Plot (3) 88
Figure 152: Iteration Plot (4) 88
Figure 153: Input Variables and Output Variables used by HP Forest 89
Figure 154: Iteration Plot (HP Forest) 90
Figure 155: Iteration History (Optimal Model) 90
Figure 156: Iteration History (Total Number of Trees Created) 90
Figure 157: Loss Reduction Variable Importance 91
Figure 158: Classification Table of HPForest 91
Figure 159: Event Classification / Confusion Matrix of HP Forest 92
Figure 160: Model Comparison Node (2) 94
Figure 161: Model Comparison Node (1) 94
Figure 162: Fit Statistics 94
Figure 163: Comparison of Event Classification Table 94

List of Tables

Table 1: DataSet Analysis 8
Table 2: Statistical Analysis (Class Vairbale) 13
Table 3: Statistical analysis (Interval Variable) 14
Table 4: Variable Worth and Chi-Square Plot Analysis 16
Table 5: iteration 50
Table 6: Iteration 51
Table 7: Iteration 53
Table 8: Confusion Matrix 55
Table 9: Confusion Matrix (Result) 55
Table 10: Itration table 59
Table 11: Iteration table 60
Table 12: Iteration Table 61
Table 13: Summary of the iterations 70
Table 14: Result Matrix 71
Table 15: Final Outcome 71
Table 16: Variation in data partiation 74
Table 17: Using variations is data partion 75
Table 18: 3x3 confusion matrix of Forward Logistic Regression 85
Table 19: Multi-class Classification Events of Forward Logistic Regression. 85
Table 20: Precision of each target prediction in the logistic regression model 85
Table 21: Experiment with different data partition 86
Table 22: Experiment with different data partitions. 88
Table 23: 3x3 Confusion Matrix of Validate Data (HP Forest) 92

Table 24: Multi-class Classification Events (HP Forest)..... 92
Table 25: Precision of each target prediction in (HP Forest). 93
Table 26: Summary of Models 95

Introduction

Data mining is the process of extracting data from large data sets and finding patterns in them. Predictive modelling is used in statistics to predict the outcomes. Data mining and predictive modelling uses many algorithms to find out and discover knowledge to find the best solution. We as a team will be using data mining and predictive modelling in this project for the data set from Tarrifa and the Tanzania ministry of water which was published in kaggle.com and applying the predictive modelling to the data set and giving the best model for the data set in this report.

In order to do the modelling, we will use a software called SAS Enterprise Miner. SAS Enterprise Miner is an advanced analytics tools which helps the users easily and quickly create predictive and descriptive models with the help of streamlined data mining.

In this project first we will understand the data, study the data, and fix the outcome of it by selecting the target variable. The first thing before modelling is import the data set into the Enterprise miner. We will be applying sampling to the data set and then rejecting the variables which we will not be using and fixing the target variable. As the downloaded file contains 2 data sets, we will be merging them with the help of *merge* node. Then we use stat explore and graph explore the variables in the data set and performing the data exploration. As for the data pre-processing, we will be using *impute* node to impute the variables, *drop* node to drop some of the variables. There are many predictive models such as decision tree, regression, neural networks, HP forest and many more. As a group we have responsibility to design each 2 different predictive models and combine them and make a report.

After the model creation and best model is chosen by model comparison and the report will be concluded by giving the best predictive model for the data set.

Process Flow Diagram

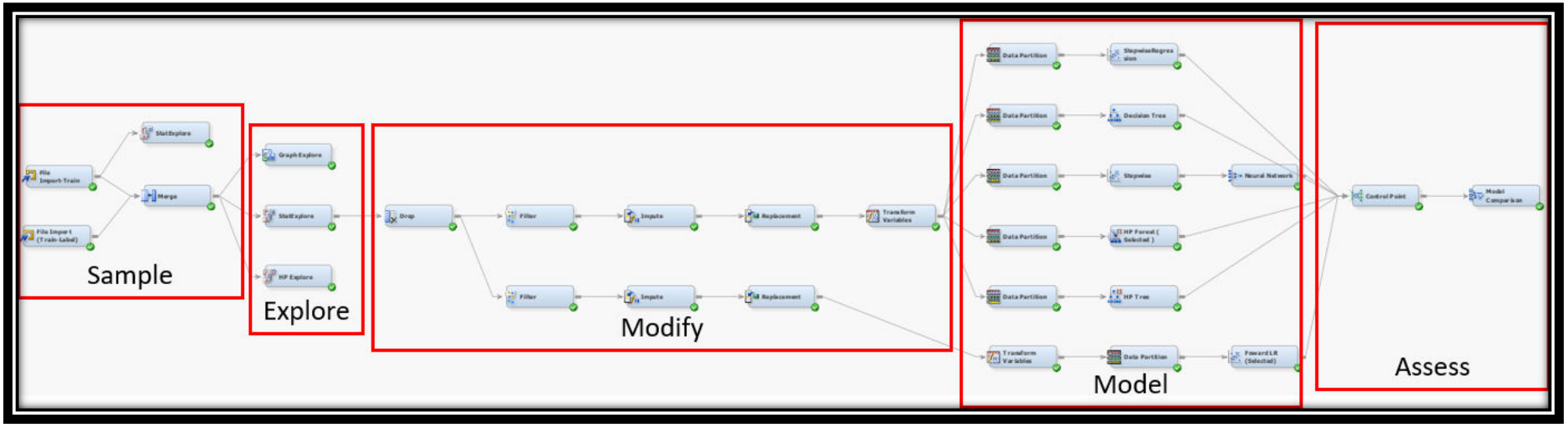


Figure 1: Process Flow Diagram

Figure 1 shows the process flow of the data mining which follow SEMMA methodology. The details of each stage will discussed in this assignment.

1.0 Sample (Data Selection)

Data sampling is a statistical analysis technique that involves selecting, manipulating, and analysing a representative selection of data points to uncover patterns and trends in a larger data collection. It allows data scientists, predictive modellers, and other data analysts to deal with a limited, manageable amount of data on a statistical population to swiftly build and test analytical models while still generating reliable results (Biscobing, 2018).

In our project we will be using only some variables which are needed and the other we will be not using. So, we can reject the variables which we don't use.

First, we import the file to the SAS-E-Miner. We have 1 dataset with two files which are train and train-label, therefore we will import both files by using file import node. To find the import function we will have to go to the sample function and click the file import.

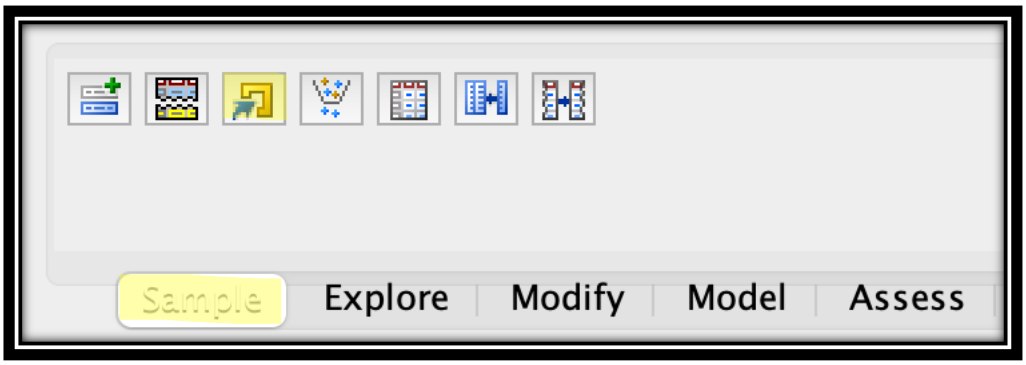


Figure 2: Sample->File Import

When click and drag the import to the diagram we can see it happens like the below diagram.

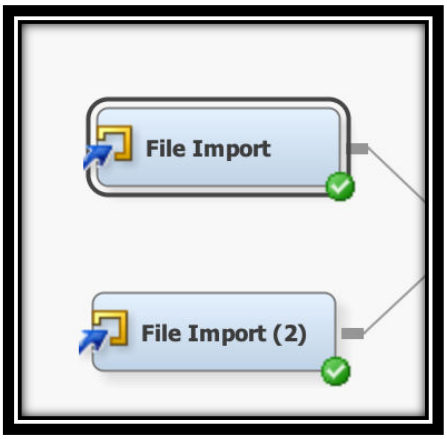


Figure 3: File import Node

Then when we click the import function at property box, we need to click on the three doted button to import the file.

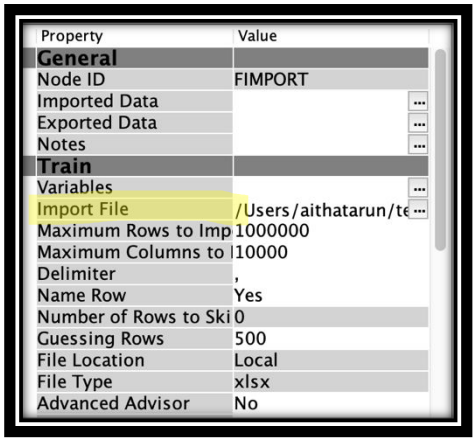


Figure 4: Properties

When we click on the three dotted line and then a dialog box will appear and then we choose the location of the data set and then import it.

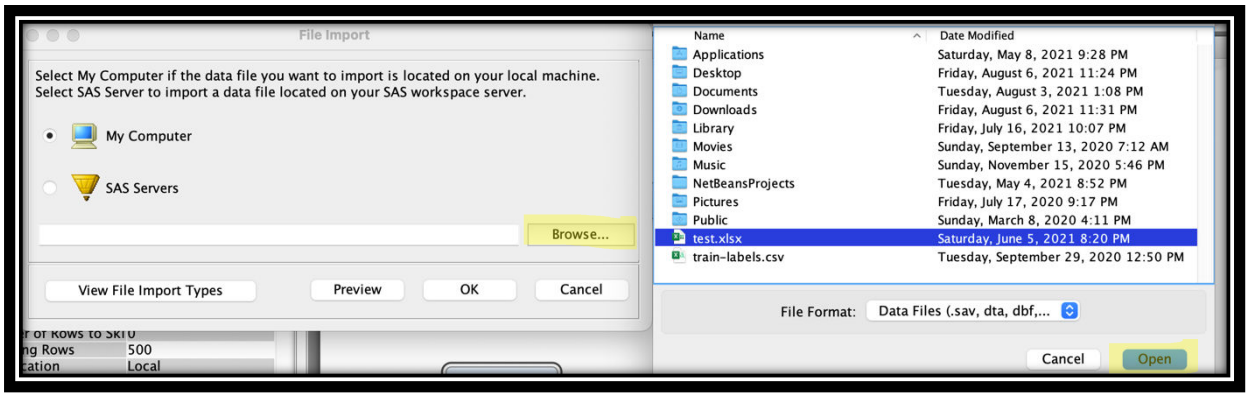
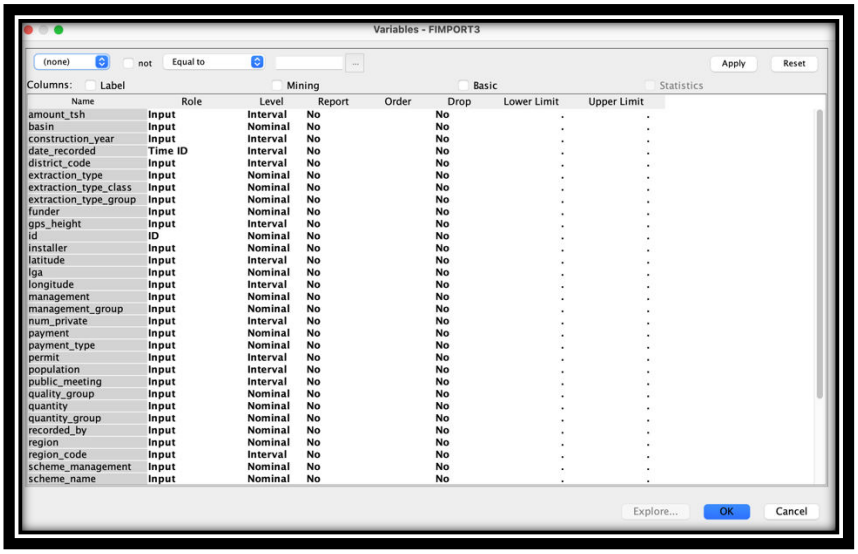


Figure 5: Importing the file

After importing the file, we have to run it so when we run it the enterprise miner will run and import the file.

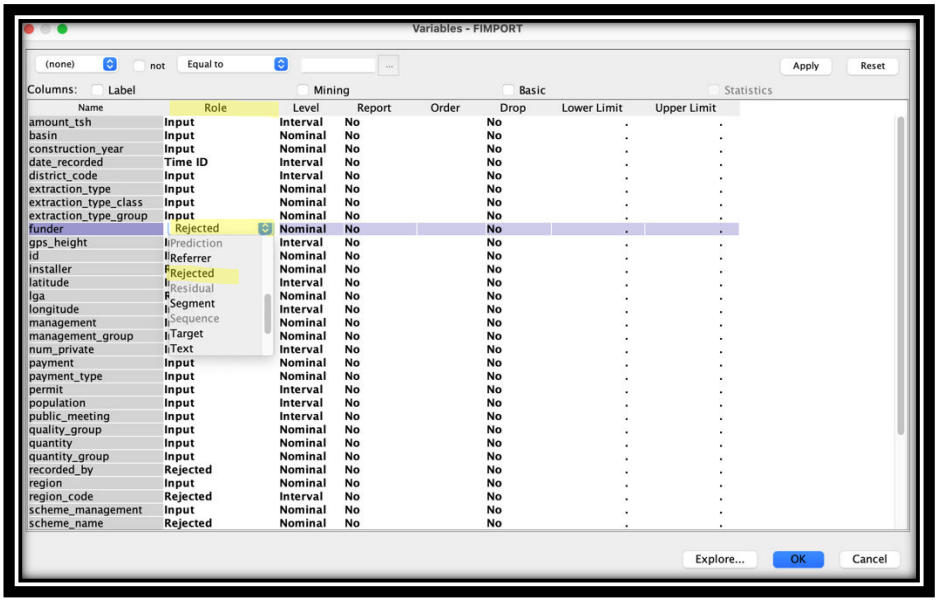
To view the data set we need to click on the tree doted button which lies besides the variables in the section of the train. When we click it, we can see all the variables in the data set and their roles.



Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
amount_tsh	Input	Interval	No		No	.	.
basin	Input	Nominal	No		No	.	.
construction_year	Input	Interval	No		No	.	.
date_recorded	Time ID	Interval	No		No	.	.
district_code	Input	Interval	No		No	.	.
extraction_type	Input	Nominal	No		No	.	.
extraction_type_class	Input	Nominal	No		No	.	.
extraction_type_group	Input	Nominal	No		No	.	.
funder	Input	Nominal	No		No	.	.
gps_height	Input	Interval	No		No	.	.
id	ID	Nominal	No		No	.	.
installer	Input	Nominal	No		No	.	.
latitude	Input	Interval	No		No	.	.
lga	Input	Nominal	No		No	.	.
longitude	Input	Interval	No		No	.	.
management	Input	Nominal	No		No	.	.
management_group	Input	Nominal	No		No	.	.
num_private	Input	Interval	No		No	.	.
payment	Input	Nominal	No		No	.	.
payment_type	Input	Nominal	No		No	.	.
permit	Input	Interval	No		No	.	.
population	Input	Interval	No		No	.	.
public_meeting	Input	Interval	No		No	.	.
quality_group	Input	Nominal	No		No	.	.
quantity	Input	Nominal	No		No	.	.
quantity_group	Input	Nominal	No		No	.	.
recorded_by	Input	Nominal	No		No	.	.
region	Input	Nominal	No		No	.	.
region_code	Input	Interval	No		No	.	.
scheme_management	Input	Nominal	No		No	.	.
scheme_name	Input	Nominal	No		No	.	.

Figure 6: View the data set

Some of the variables will be rejected due to number of levels is too high which is not supported by SAS Enterprise Miner. We can set it by clicking on the role of variables and then changing it.



Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
amount_tsh	Input	Interval	No		No	.	.
basin	Input	Nominal	No		No	.	.
construction_year	Input	Nominal	No		No	.	.
date_recorded	Time ID	Interval	No		No	.	.
district_code	Input	Interval	No		No	.	.
extraction_type	Input	Nominal	No		No	.	.
extraction_type_class	Input	Nominal	No		No	.	.
extraction_type_group	Input	Nominal	No		No	.	.
funder	Rejected	Nominal	No		No	.	.
gps_height	Prediction	Interval	No		No	.	.
id	Referrer	Nominal	No		No	.	.
installer	Rejected	Nominal	No		No	.	.
latitude	Residual	Interval	No		No	.	.
lga	Segment	Nominal	No		No	.	.
longitude	Sequence	Interval	No		No	.	.
management	Target	Nominal	No		No	.	.
management_group	Text	Nominal	No		No	.	.
num_private	Input	Interval	No		No	.	.
payment	Input	Nominal	No		No	.	.
payment_type	Input	Nominal	No		No	.	.
permit	Input	Interval	No		No	.	.
population	Input	Interval	No		No	.	.
public_meeting	Input	Interval	No		No	.	.
quality_group	Input	Nominal	No		No	.	.
quantity	Input	Nominal	No		No	.	.
quantity_group	Input	Nominal	No		No	.	.
recorded_by	Rejected	Nominal	No		No	.	.
region	Input	Nominal	No		No	.	.
region_code	Rejected	Interval	No		No	.	.
scheme_management	Input	Nominal	No		No	.	.
scheme_name	Rejected	Nominal	No		No	.	.

Figure 7: Rejecting variables

Here the rejected variables are *funder, installer, lga, recorderd_by, region_code, scheme_name, subvillage, ward, wpt_name.*

The target variable should be only one in the modelling so the target variable will be the *status_group.*

Name	Role
id	ID
status_group	Target

Figure 8: Target variables

This is set as target variables because the status_group is refering to the status of the pump which match to our project objective, predict the water pump is functional, non-functional or functional but needs repair.

As for now we have imported and sampled the data set but there are two data sets in the diagram, in order to combine them we need to merge the data sets, the enterprise miner has a feature to merge multiple data sets and make them into one combined data set. To use the function, we can find the merge function in the sample area.

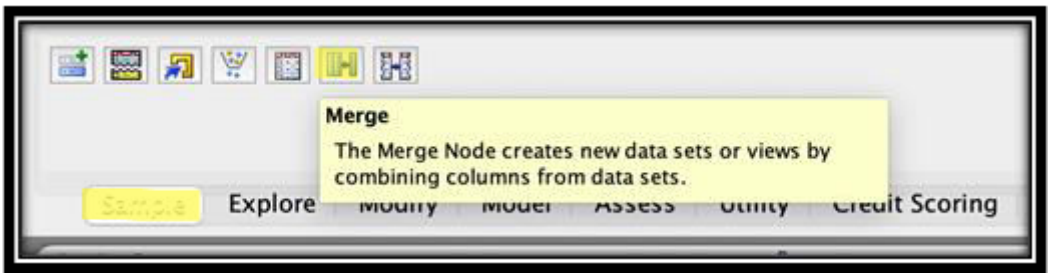


Figure 9: Merge Node

We can just drag it and combine the data files into the merge function and run it we will find the data files combined.

gps_height	none
id	By

Figure 10: Modification

The data files is merge by id as the id is the unique value in the dataset.

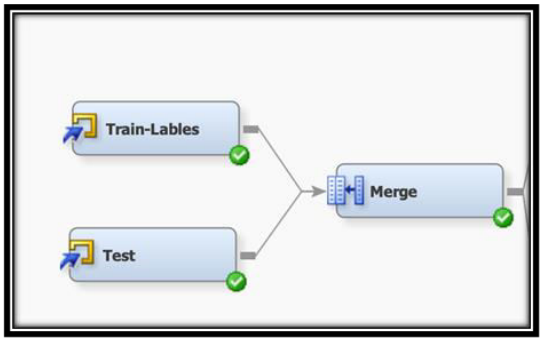


Figure 11: Merging of the data set

2.0 Explore (Exploratory Data Analysis)

The exploratory data analysis is done after data selection. The exploratory data analysis is carried out using the Stat Explore node, HP Stat Explore node, and Graph Explore node. In this phase, the dataset will be analysed and the noisy data in the dataset will be identified.

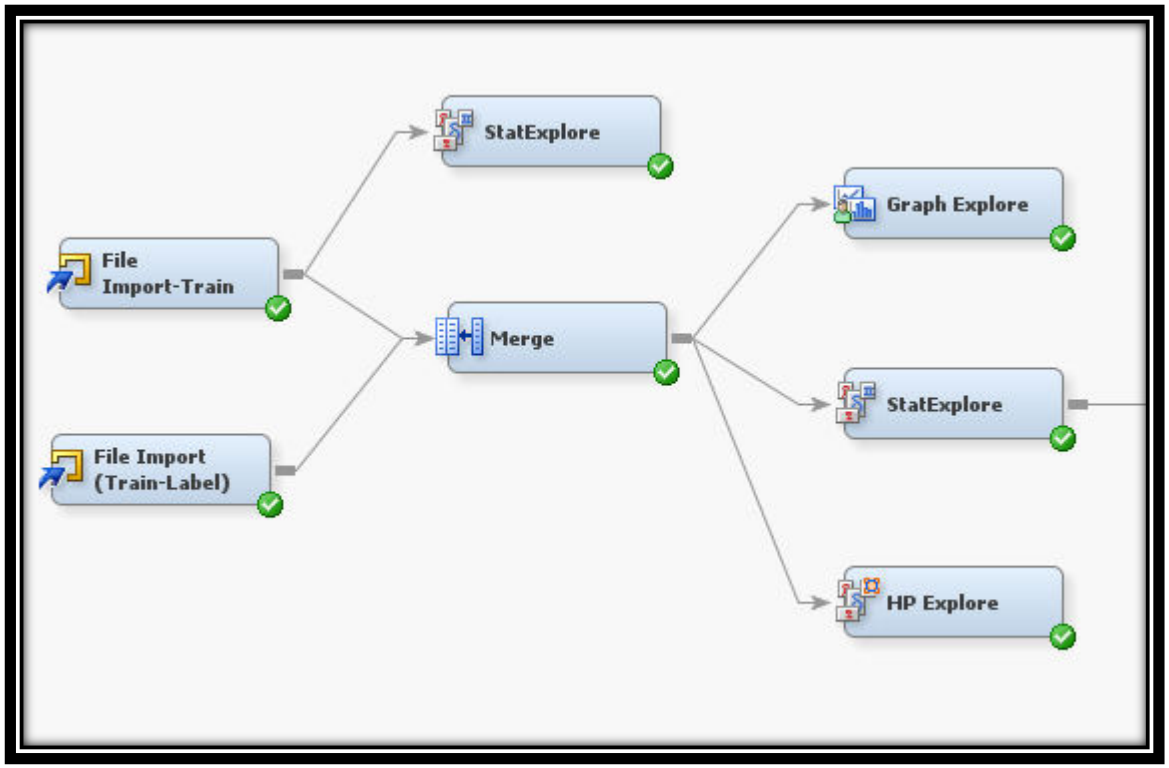


Figure 12: Explore Process FLow Diagram

Before merging two files, we encounter an error in merging, therefore, the StatExplore node takes place to find out the reason that unable merge data files. From the StatExplore, we found that there are some variables are having a high number of levels which is not acceptable by SAS Enterprise Miner. The variable with the number of levels more than 500 is rejected as shown in figures 6, 7, and 8.

Dataset Analysis:

Number of Observation	59400

Table 1: DataSet Analysis

2.1 Statistical Analysis (Class Variable):

The tables below show the analysis of overall analysis mainly from statistical analysis. The data pre-processing is noted in the table so that the data pre-processing can be easily done by identifying what are the noisy data in the dataset.

Attribute Categorical	**Description of Attribute**	**Findings**	**Pre-processing /Notes**
basin	Geographic water basin	No missing Values Number of levels = 9 Mode 1: Lake Victoria (17.25%) Mode 2: Pangani (15.05%)	No Pre-processing Needed
extraction_type	The kind of extraction the waterpoint uses	No missing Values Number of levels = 18 Mode 1: gravity (45.08%) Mode 2: nira/tanira (15.05%)	No Pre-processing Needed
extraction_type_class	The kind of extraction the waterpoint uses	No missing Values Number of levels = 7 Mode 1: Gravity (45.08%) Mode 2: handpump (27.70%)	Drop because is the same kind of information with extraction_type
extraction_type_group	The kind of extraction the	No missing Values	Drop because is repeated with

	waterpoint uses	Number of levels = 13 Mode 1: gravity (45.08%) Mode 2: nira/tanira (13.73%)	extraction_type
Funder	Who funded the well	Missing Value = 222 Number of levels = 513	Rejected because the number of levels is too high and not accepted by SAS Enterprise Miner.
Installer	Organization that installed the well	Missing Value = 196 Number of levels = 513	Rejected because the number of levels is too high and not accepted by SAS Enterprise Miner.
lga	Geographic Location	No missing Values Number of levels = 513	Rejected because the number of levels is too high and not accepted by SAS Enterprise Miner.
Management	How the waterpoint is managed	No missing Values Number of levels = 12 Mode 1: VWC (68.19%) Mode 2: WUG (10.97%)	“Unknown” replace with “other”
Management_group	How the waterpoint is managed	No missing Values Number of levels = 5 Mode 1: user-group (88.37%) Mode 2: commercial (6.12%)	Drop because low variable worth
Payment	What the water costs	No missing Values	Drop because repeated with

		Number of levels = 7 Mode 1: never pay (42.67%) Mode 2: pay per bucket (15.13%)	payment_type
Payment_type	What the water costs	No missing Values Number of levels = 7 Mode 1: never pay (42.67%) Mode 2: per bucket (15.13%)	Replace “Unknown” with “never pay”
permit	If the waterpoint is permitted	3056 missing values. Number of levels = 3 Mode 1: True (65.41%) Mode 2: False (29.45%)	Drop because lowest variable worth
public_meeting	True/false	3334 missing values. Number of levels = 3 Mode 1: True (85.88%) Mode 2: False (8.51%)	Drop because low variable worth
Quality_group	The quality of the water	Number of levels = 6 Mode 1: good (85.55%) Mode 2: salty (8.75%)	Drop because the variable in the Chi-Square plot is low and variable worth is low.
quantity	The quantity of water	Number of levels = 5 Mode 1: enough (55.87%) Mode 2: insufficient (25.47%)	Replace “Unknown” with “enough”

Quantity_group	The quantity of water	Number of levels = 5 Mode 1: enough (55.87%) Mode 2: insufficient (25.47%)	Drop because repeated with quantity
Recorded_by	Group entering this row of data	Number of levels = 1	Rejected Due to not needed in Data modeling
region	Geographic location	Number of levels = 21 Mode 1: Iringa (8.91%) Mode 2: Shinyanga (8.39%)	No Data Pre-processing needed.
scheme_management	Who operates the waterpoint	3877 missing values. Number of levels = 13 Mode 1: VWC (61.94%) Mode 2: WUG (8.76%)	Replace with mode: VWC Replace “Unknown” with “VWC”
scheme_name	Who operates the waterpoint	834 missing values. Number of levels = 513	Rejected due to Number of Levels is too high
source	The source of water	Number of levels = 10 Mode 1: Spring (28.65%) Mode 2: Shallow Well (28.32%)	Replace “Unknown” with “spring”
Source_class	The source of water	Number of levels = 3 Mode 1: groundwater (77.09%) Mode 2: surface (22.44%)	MODE: Unknown to Groundwater Replace “Unknown” with “groundwater”
Source_type	The source of water	Number of levels = 7	Replace “Unknown” with “spring”

		Mode 1: spring (28.65%) Mode 2: shallow well (28.32%)	
subvillage	Geographic Location	4 missing values. Number of levels = 513	Rejected due to Number of Levels is too high and not accepted by SAS Enterprise Miner
Ward	Geographic Location	Number of levels = 513	Rejected due to Number of Levels is too high and not accepted by SAS Enterprise Miner
Water_quality	The quality of water	Number of levels = 8 Mode 1: soft (85.55%) Mode 2: salty (8.18%)	Replace “Unknown” with “Soft”
Waterpoint_type	The kind of water point	Number of levels = 7 Mode 1: communal standpipe (48.02) Mode 2: hand pump (29.44)	Replace “Unknown” with “communal standpipe”
Waterpoint_type_group	The kind of water point	Number of levels = 6 Mode 1: communal standpipe (58.29) Mode 2: hand pump (29.44)	Drop because is repeated with Waterpoint_type
Wpt_name	Name of waterpoint if there is one	Number of levels = 513	Rejected due to Number of Levels is too high
Status_group	Distribution of label	No missing value Number of levels = 3	This is target Variable

		Mode 1: functional (54.31%) Mode 2: non-functional (38.42%)	

Table 2: Statistical Analysis (Class Vairbale)

2.2 Statistical Analysis (Interval Variable)

Attribute	**Description of Attribute**	**Findings**	**Pre-processing /Notes**
Construction Year	The Year of the waterpoint was constructed	Missing Value = 698	Transform
amount_tsh	Total static head (amount water available to waterpoint)	High Positively skewed (outlier), Right Tail, Skewness = 57.8078	Transform
Population	Population around the well	High Positively skewed (outlier), Right Tail, Skewness = 12.66071	Transform
latitude	GPS coordinate	Negatively skewed Skewness = -0.15204	Transform
Distinct_code	Geographic location (coded)	High Positively Skewed Skewness = 3.962045	Transform
Gps_height	Alitude of the well	Fairly Skewness Skewness = 0.462402	Replacement: Lower limit: 0
Longitude	GPS coordinate	High Negatively skewed, Left tail Skewness = -4.19105	Filter: Drop the missing value. Lower limit: 2

Num_private	-	High Positively Skewed, Right Tail Skewness = 91.93375	Drop because not required in data modeling
Region_code	Geographic location (coded)	High Positively Skewed, Right Tail Skewness =3.17818	Drop because low variable worth

Table 3: Statistical analysis (Interval Variable)

2.3 Variable Worth

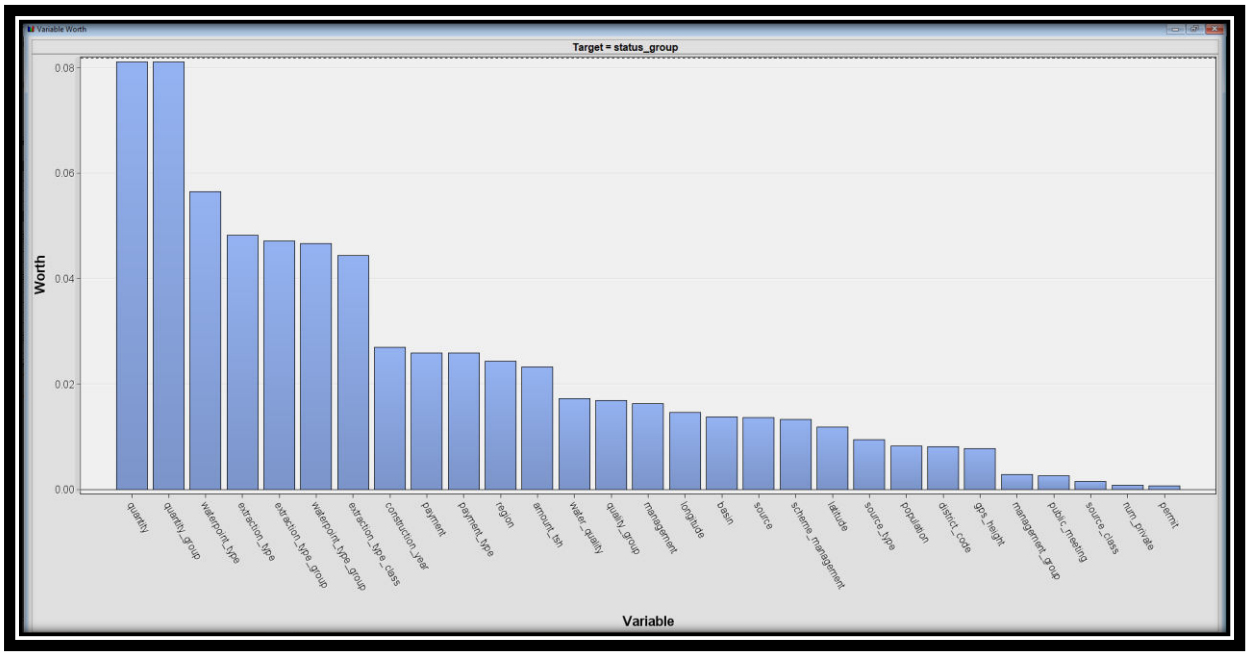


Figure 13: Variable Worth

2.4 Chi-Square Plot

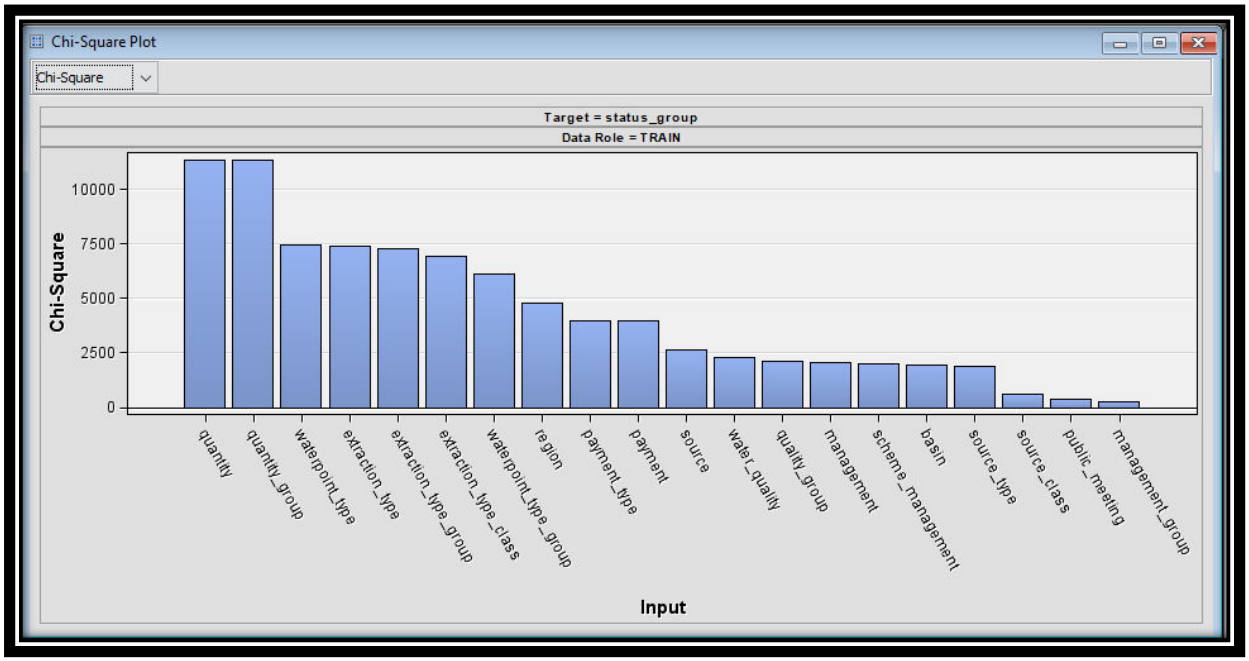


Figure 14: Chi-Square Plot

Attribute	**Findings**	**Pre-processing**
Variable Worth	**High Variable Worth:** Quantity, quantity_group, waterpoint_type, extraction_type, extraction_ type_group, waterpoint_type_group, extraction_type _class, construction_year, payment, payment_type, region, amount_tsh, water_quality, quality_group, management, longitude, basin, source, scheme_management, latitude, source_type, population, distinct_code, gps_height **Low Variable Worth:** Management_group, public_meeting, source_class, num_private, permit	Drop low variable worth
Chi-square Plot	**Variable suitable to use in prediction:** Quantity, quantity_group, waterpoint_type, extraction_type, extraction_type_group, extraction_type_class, waterpoint_type_group, region, payment_type, payment, source, water_quality, quality_group, management, Scheme_management, basin, source_type, source_class **Variable not suitable to use in prediction:** Public_meeting, management_group	Drop variable not suitable to use in prediction

Table 4: Variable Worth and Chi-Square Plot Analysis

2.5 Graph Analysis

Status Group

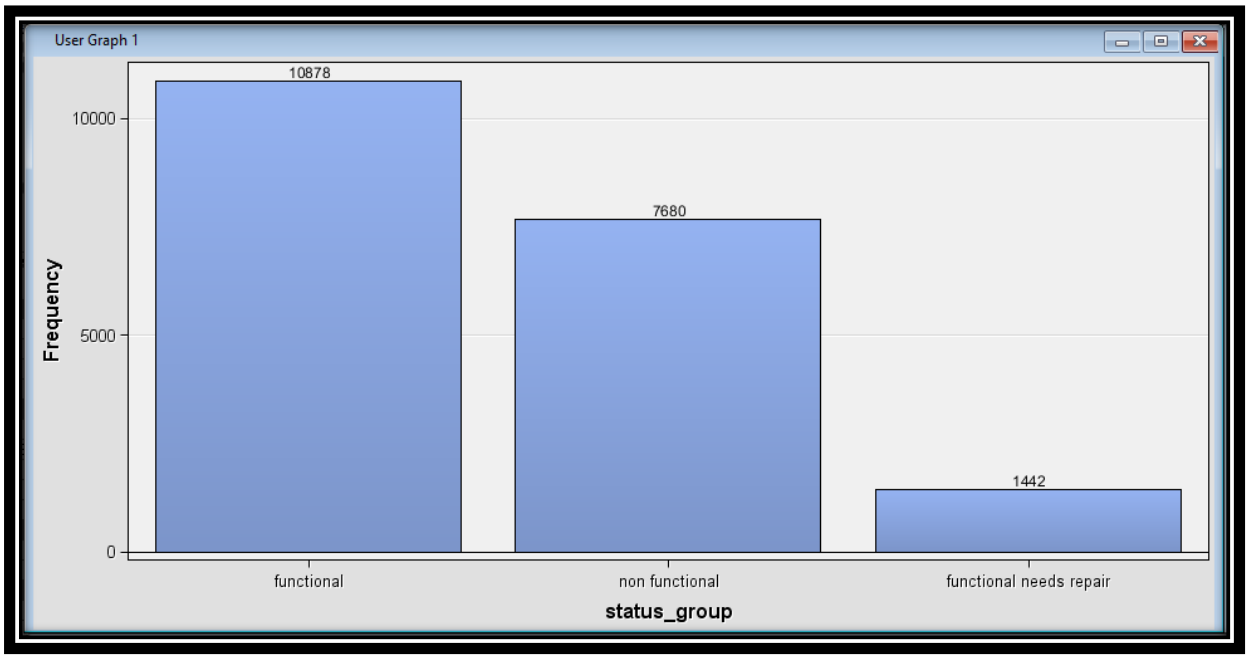


Figure 15: Status Group Bar Chart

The target of the predictive modeling is status_group. The status group is the target variable that determines the pump status. The number of levels of status_group is 3 which are functional, non-functional, and functional needs repair. The mode of the status group is functional with 10878 observations and the functional-needs repair is the min with 1442 observations.

Quantity

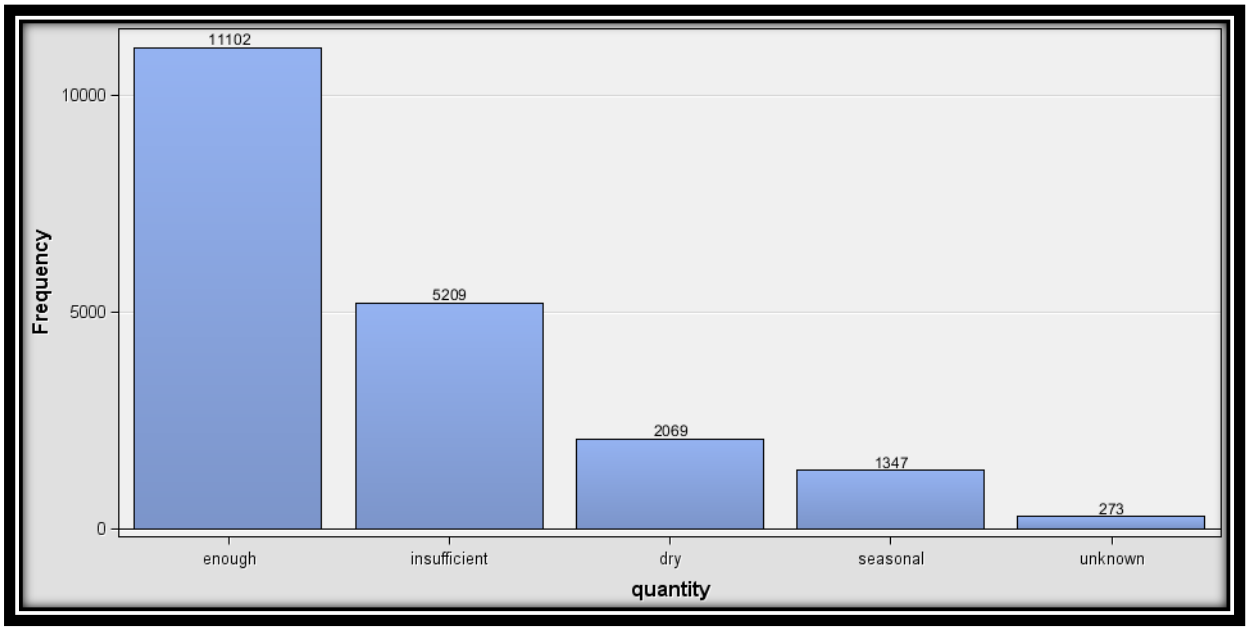


Figure 16: Quantity

There are 5 levels for quantity. Quantity variables refer to the quantity of water points for the pump and there are 5 categories in this dataset which are enough, insufficient, dry, seasonal, and unknown. The unknown quantity will be replaced with a mode value which is enough. The mode of the quantity is enough quantity with 11102 observations.

Quantity Group

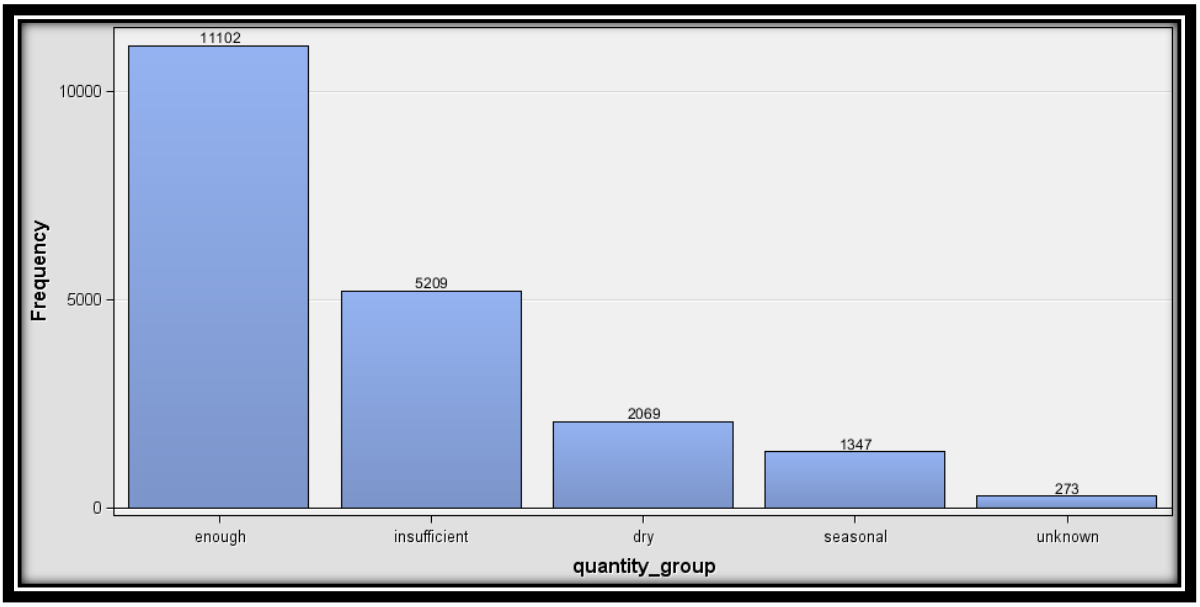


Figure 17: Quantity Group

There are 5 types of quantity which are enough, insufficient, dry, seasonal, and unknown. The types of quantity groups are the same as quantity. Therefore, the quantity group will be dropped before data modeling. Most of the pumps having enough quantity of water. The pump with an unknown quantity of water will be replaced with mode.

Waterpoint type

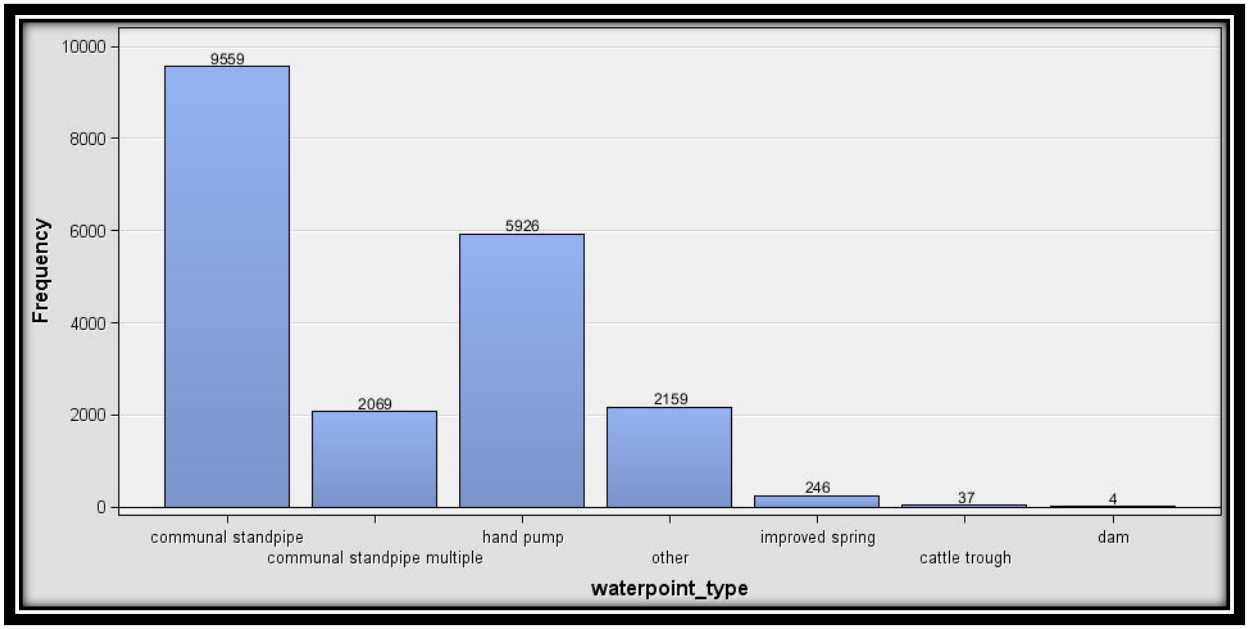


Figure 18: Waterpoint Type

There are 7 types of waterpoint which are communal standpipe, communal standpipe multiple, hand pump, other, improve spring, cattle trough, and DAM. Waterpoint types refer to the kind of water point of the pump. The mode of the waterpoint type is communal standpipe with 9559 observations. The min of the waterpoint type is the dam which only has 4 observations.

Waterpoint type group

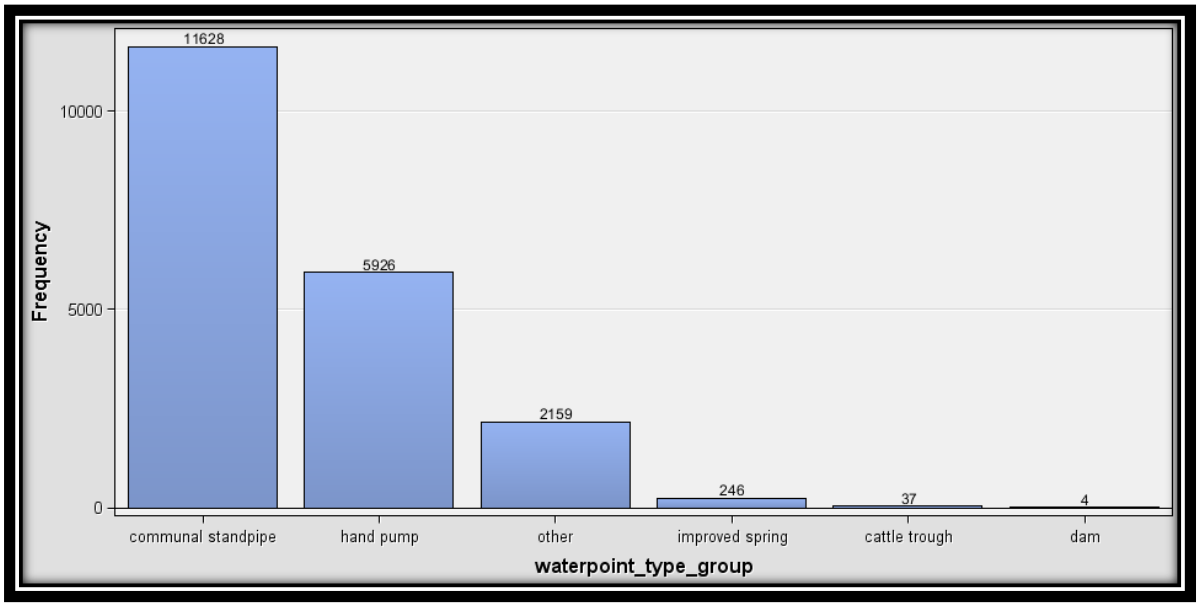


Figure 19: Waterpoint type group

There are 5 groups of waterpoint types which are communal standpipe, hand pump, other, improved spring, cattle trough. The difference between the waterpoint type and waterpoint type group is the communal_standpipe and communal_standpipe_multiple was combined into one group in waterpoint type variable.

Extraction Type

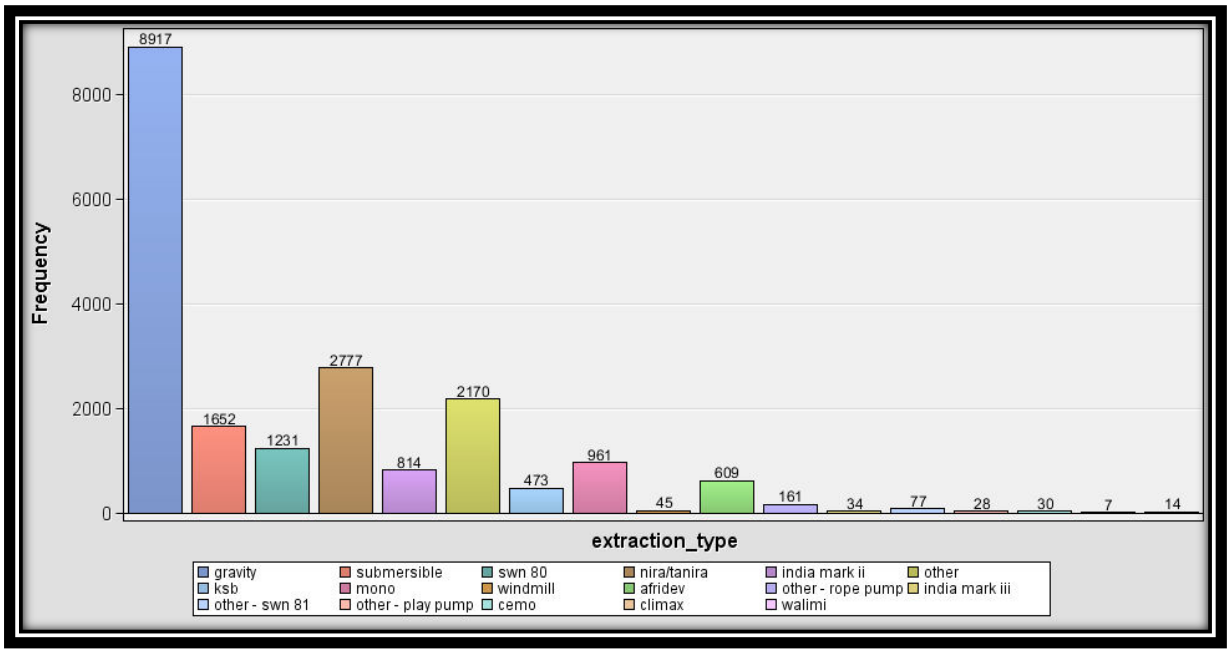


Figure 20: Extraction Type

There are 17 types of extraction which are gravity, submersible, swn 80, nira/tanira, india mark ii, other, ksb, mono, windmill, afridev, other – rope pump, india mark iii, other – swn 81, other play pump, cemo,climax, and walimi . The mode of extraction type is gravity which having 8917 observations. The min is the pump with extraction type climax which only has 7 observations.

Extraction Type Class

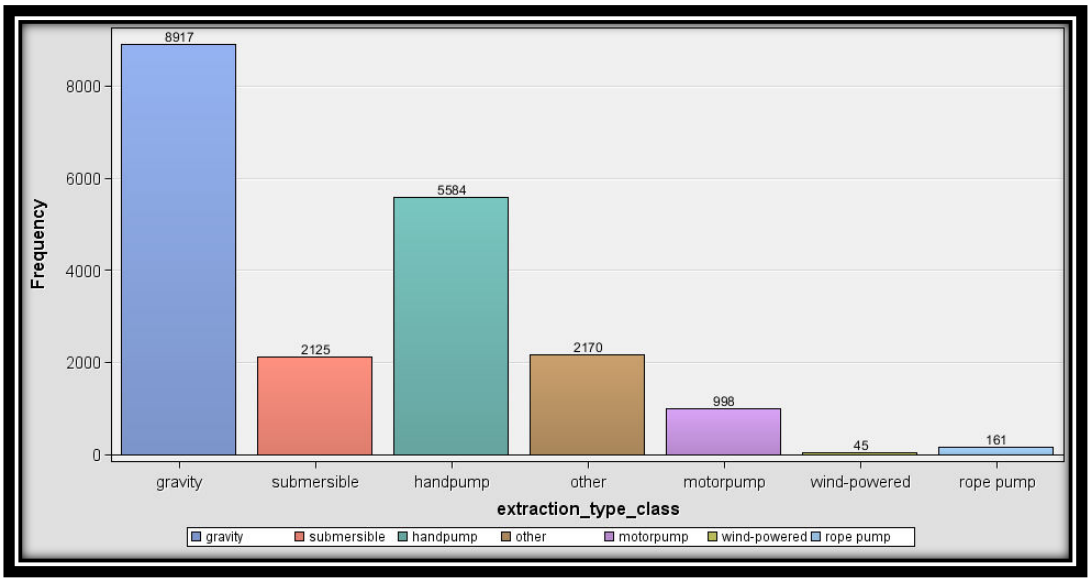


Figure 21: Extraction Type Class

There are 7 classes of extraction types which are gravity, submersible, hand pump, other, motorpump, wind-powered, rope pump. The mode of extraction type is gravity and the min of extraction type is wind-powered.

Extraction type group

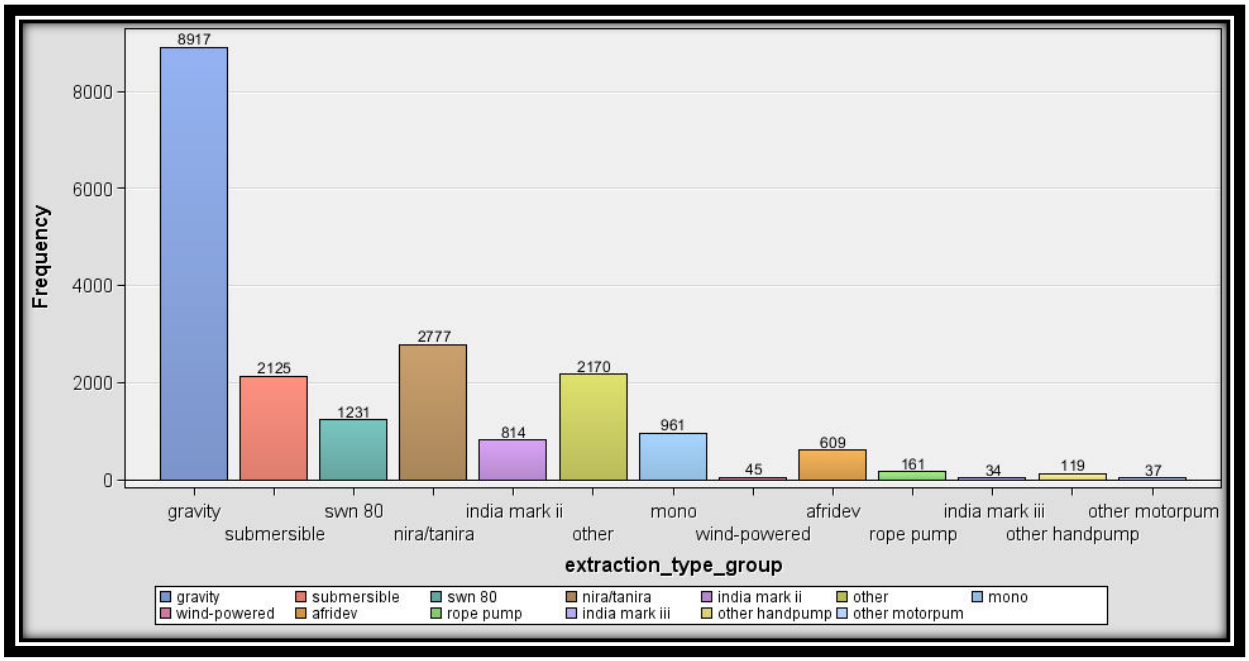


Figure 22: Extraction Type Group

There are 13 groups of extraction types which are gravity, submersible, swn 80, nira/tanira, india mark ii, other, mono, wind-powered, afridev, rope_pump, india mark iii, other handpump and other motorpum. The mode is gravity with 8917 observation and the min is india mark iii with 34 observations.

Region

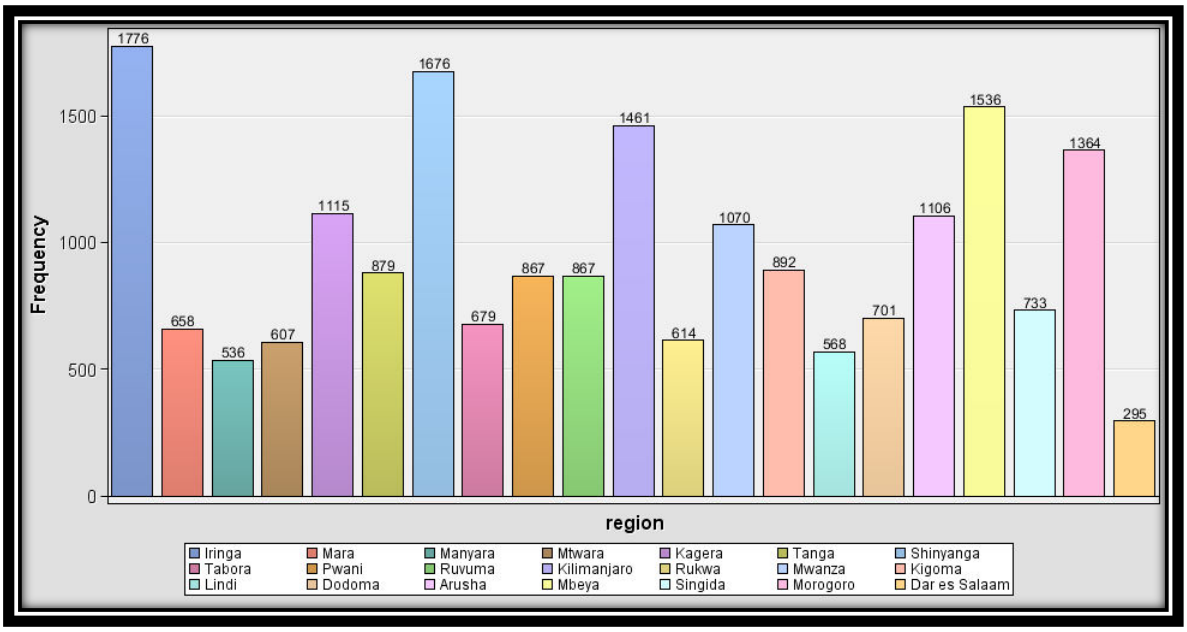


Figure 23: Region

There are 21 region in this dataset which are Iringa, Mara, Manyara, Mtwara, Kagera, Tanga, Shinyanga, Tabora, Pwani, Ruvuma, Kilimanjaro, Rukwa, Mwanza, Kigoma, Lindi, Dodoma, Arusha, Mbeya, singida, Morogoro, Dar es Salaam. The mode of region is 1776 observations in lringa and the min is 295 observations in Dar es Salaam.

Payment type

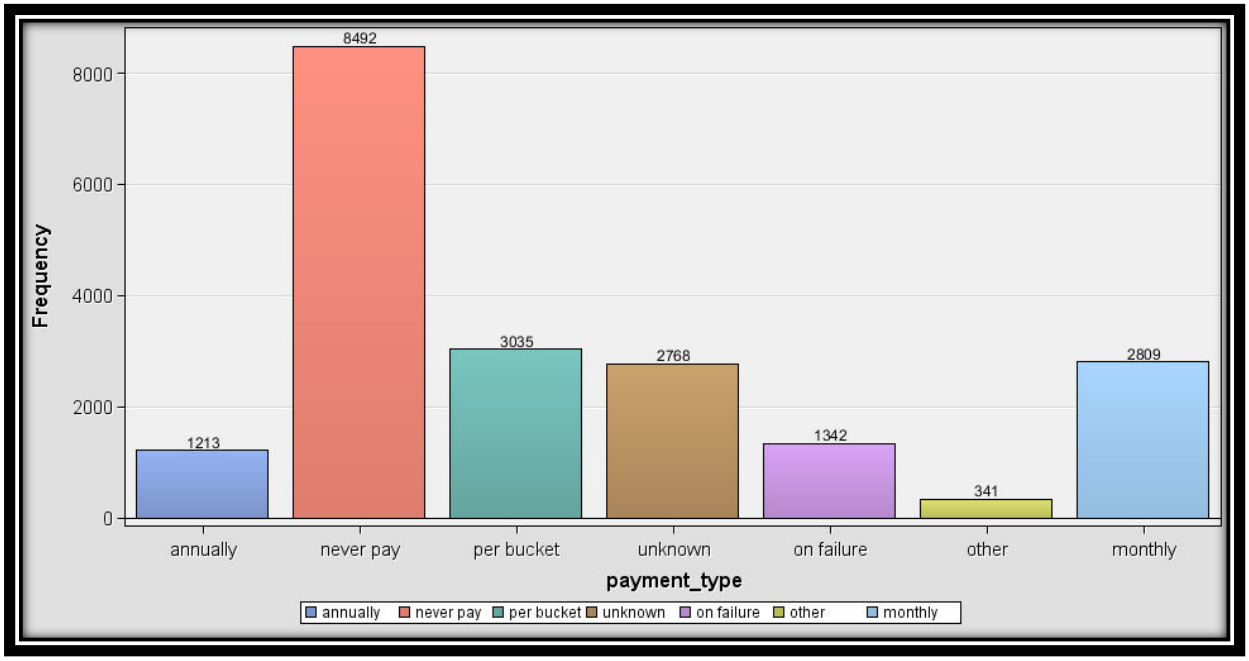


Figure 24: Payment Type

There are 7 payment types which are annually, never pay, per bucket, unknown, on failure, other, monthly. The mode of payment type is never pay with 8492 observations.

Payment

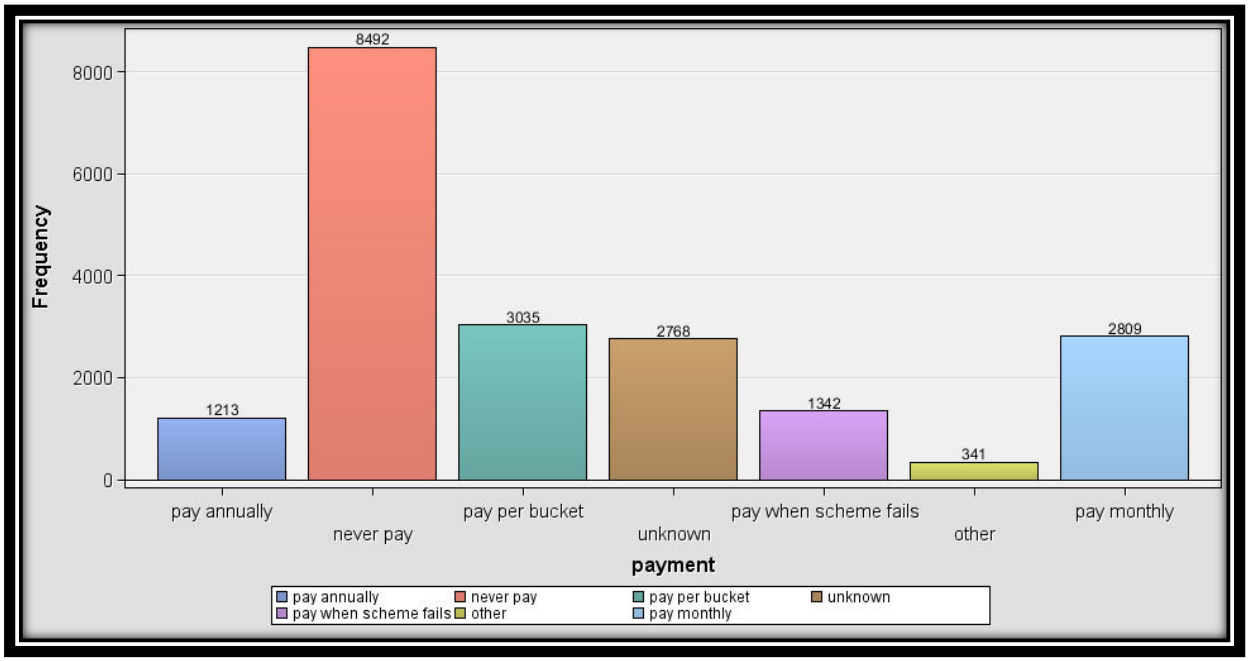


Figure 25: Payment

There are 7 types of payment which are pay annually, never pay, pay per bucket, unknown, pay when scheme fails, other, pay monthly. The payment and payment types are repeated variables.

Source

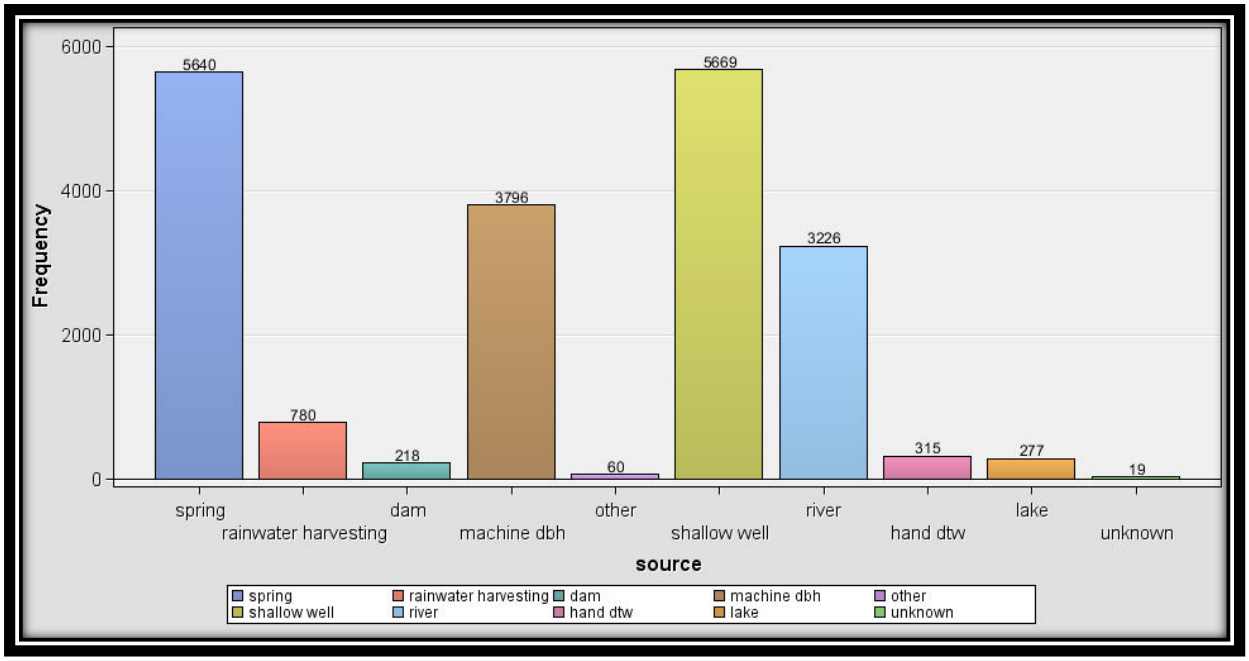


Figure 26: Source

There are 9 sources of water which are spring, rainwater harvesting, darn, machine dbh, other, shallow well, river, hand dtw, lake, and unknown. The mode of source of waterpoint is shallow well with 5669 observations.

Source class

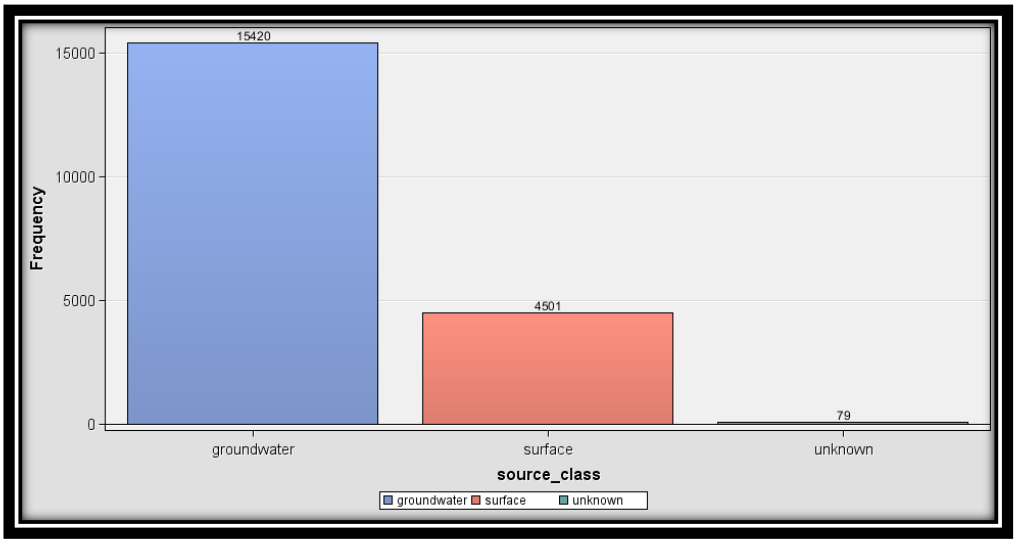


Figure 27: Source Class

There are 3 classes of source class which are groundwater, surface, unknown. The mode of source class is groundwater which means that most of the sources of water have come from groundwater with the 15420 observations.

Source type

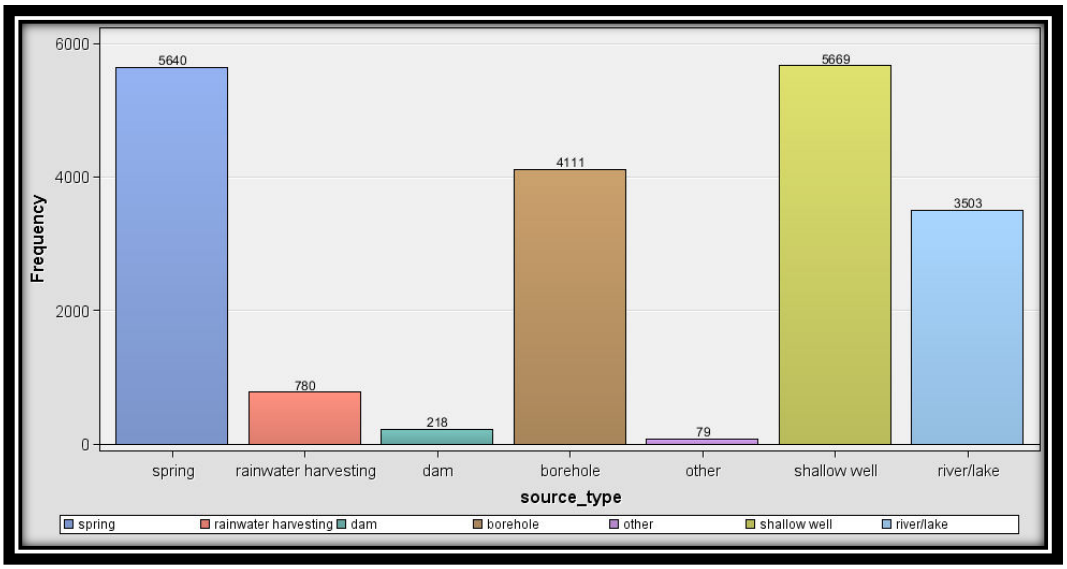


Figure 28: Source Type

There are 7 types of sources of water which are spring, rainwater harvesting, dam, borehole, other, shallow well, river/lake. Most of the pump sources of water come from the shallow well with 5669 observations.

Water quality

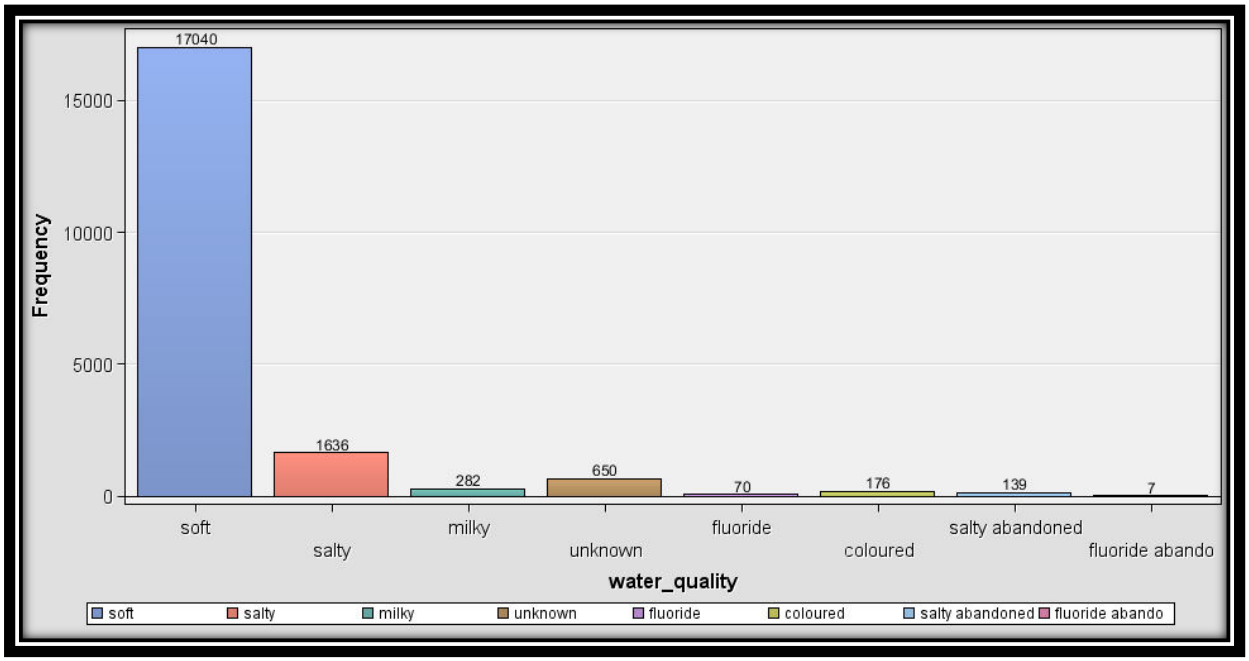


Figure 29: Water Quality

There are 7 types of water quality which are soft, salty, milky, unknown, fluoride, coloured, salty abandoned. The mode of water quality is soft with 17040 observations.

Quality Group

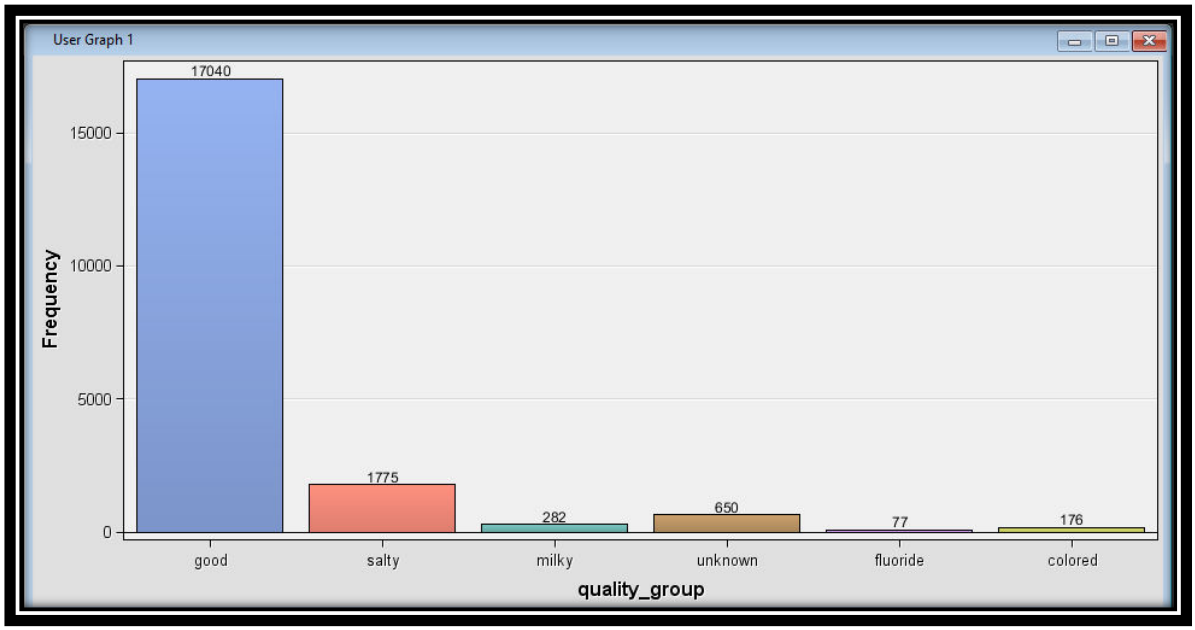


Figure 30: Quality Group

There are 6 groups of water quality which are good, salty, milky, unknown, fluoride, colored. The mode of the quality group is good with 17040 observations. The min of the quality group is fluoride with 77 records.

Management

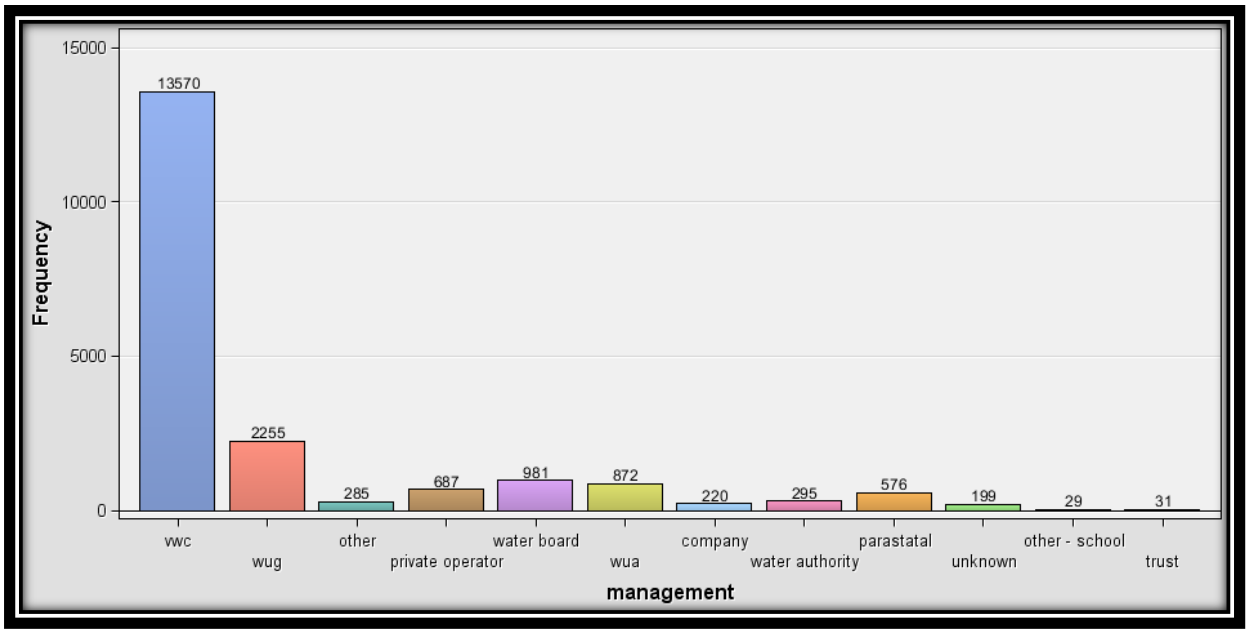


Figure 31: Management Bar Chart

There are 12 levels in management which are vwc. Wug, other, private operator, water board, wua, company, water quality, parastatal, unknown, other-school, trust. Most of the water pumps are manage by vwc with 13670 records. The min of management is other-school with 29 records.

Management Group

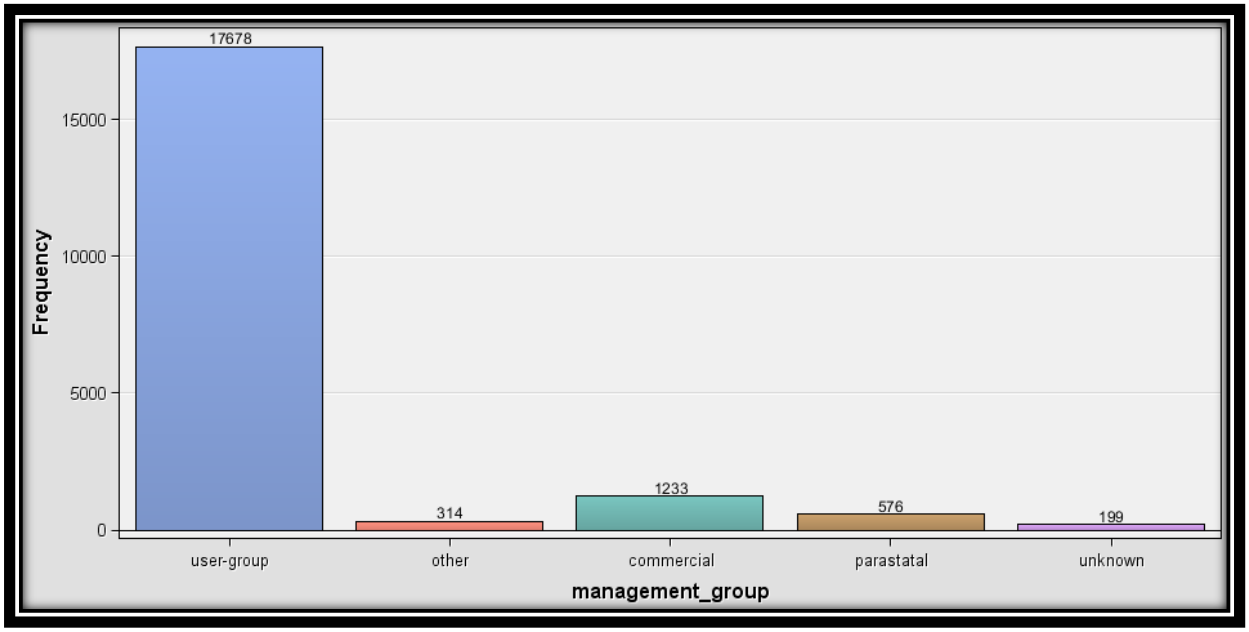


Figure 32: Management Group

There are 5 management groups which are user-group, other, commercial, parastatal, unknown. The mode of management group is user-group with 17676 records.

Waterpoint type group

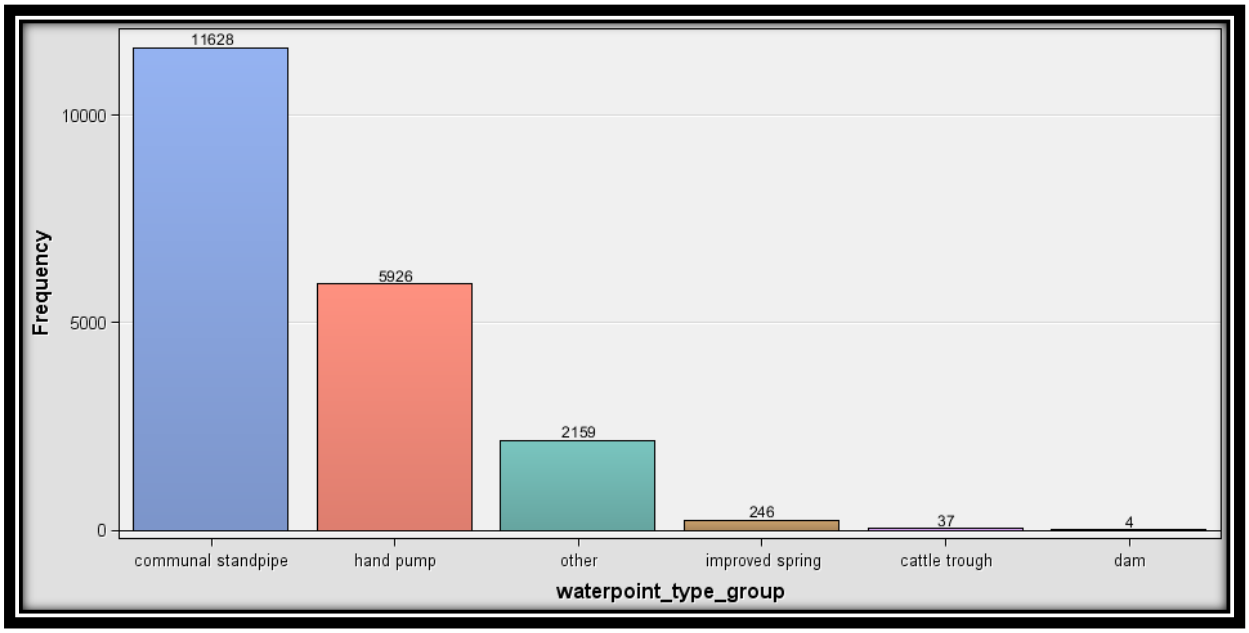


Figure 33: Waterpoint Type Group

There are 5 groups of waterpoint types which are communal standpipe, hand pump, other, improved spring, cattle trough. The communal standpipe and communal standpipe multiple were combined into one group. The mode of the waterpoint type group is 11628 records of the communal standpipe.

Amount Tsh

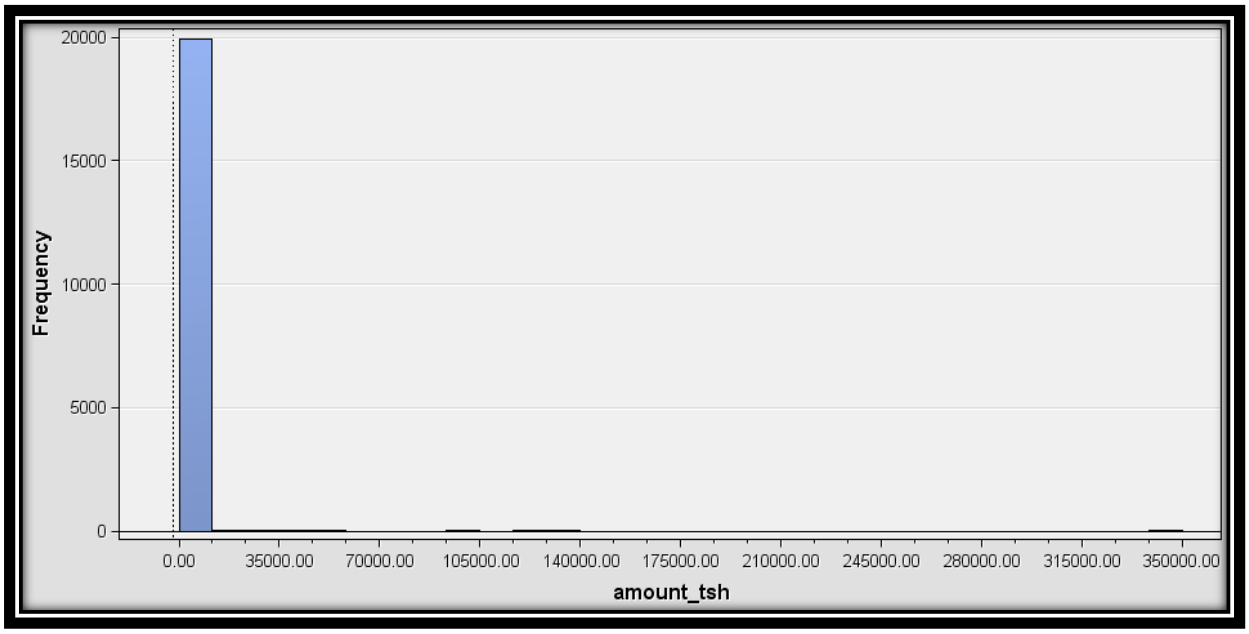


Figure 34: amount_tsh

There is no missing value found in Amount tsh because the total static head can be 0 which means the pump is not on. However, the amount tsh is high negatively skewed which most of the values are between 0 and 1333. The mean of amount_tsh is 317.6504 and the maximum of amount_tsh is 350000.

Construction Year

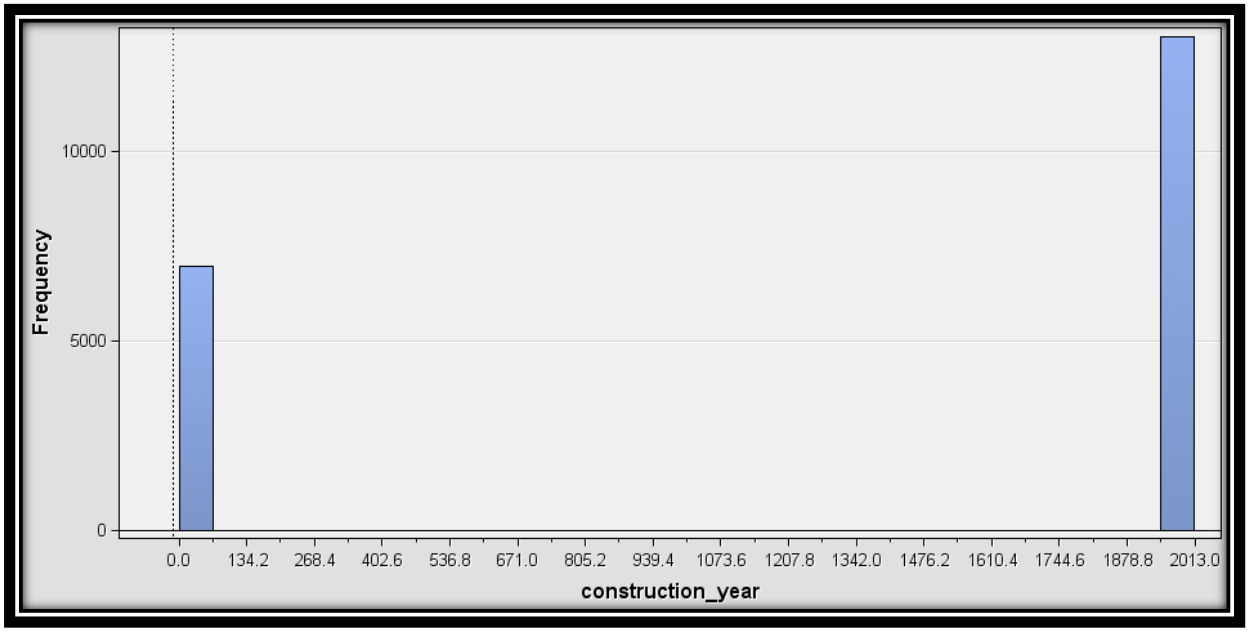


Figure 35: Construction Year

There are 698 missing values in Construction Year as the year cannot be zero. The transformation should be done in data preprocessing. The maximum construction year in the dataset is 2013.

Quantity Vs Status Group

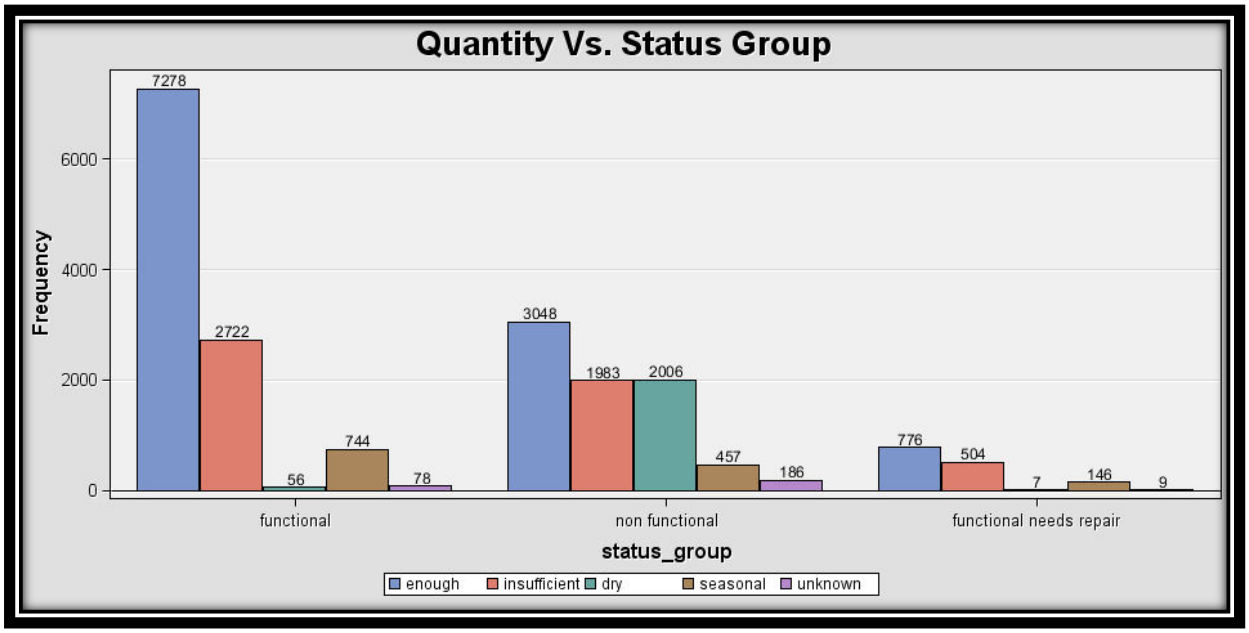


Figure 36: Quantity Vs. Status Group

Most of the pumps with enough quantity are functional. Besides, it also shows that most of the non-functional pumps are having enough quantity of water.

Waterpoint Type Vs Status Group

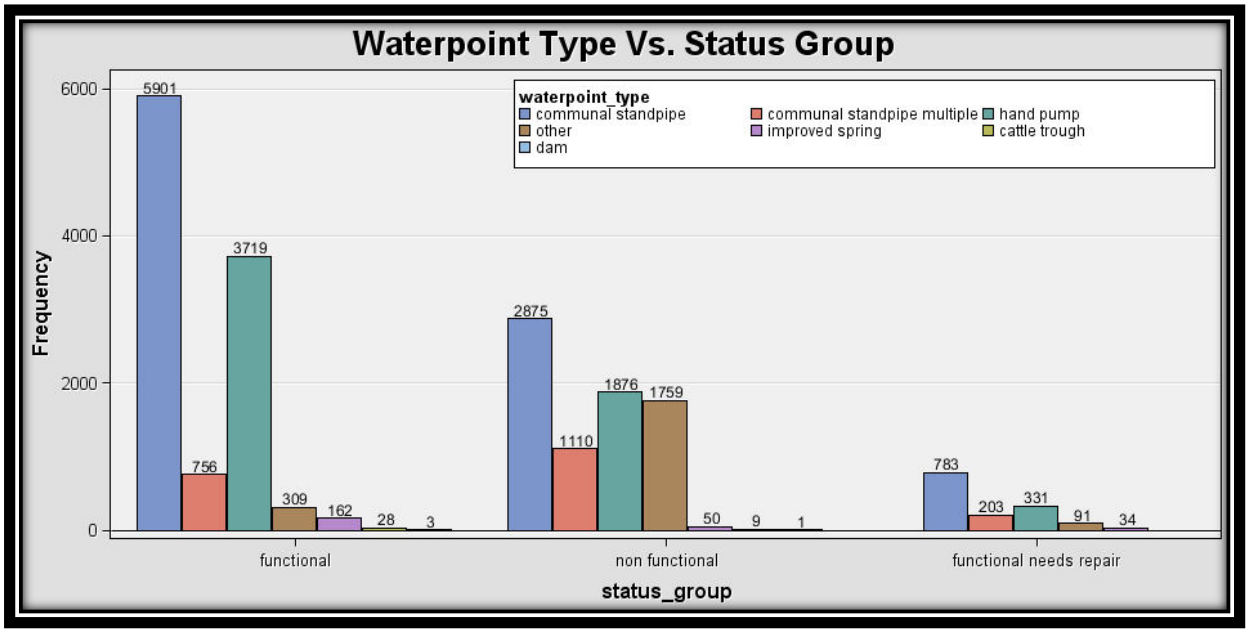


Figure 37: Waterpoint Vs. Status Group

Most of the communal standpipe are functional. Besides, it also shows that most of the non-functional pumps are using communal standpipe. There is no significant difference to show which waterpoint types are functional or non-functional.

Extraction type Vs Status Group

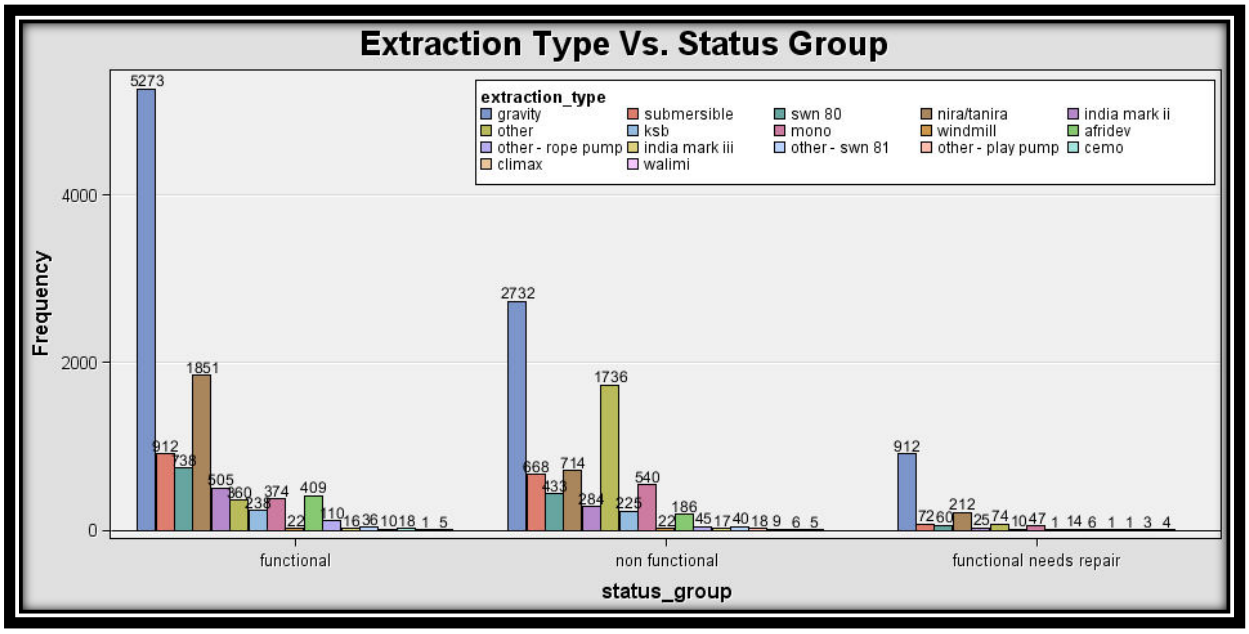


Figure 38: Extraction Type Vs. Status Group

Figure 36 shows that most of the pumps with gravity extraction are functional. Besides, most of the pumps with other extraction types are non-functional.

Region Vs. Status Group

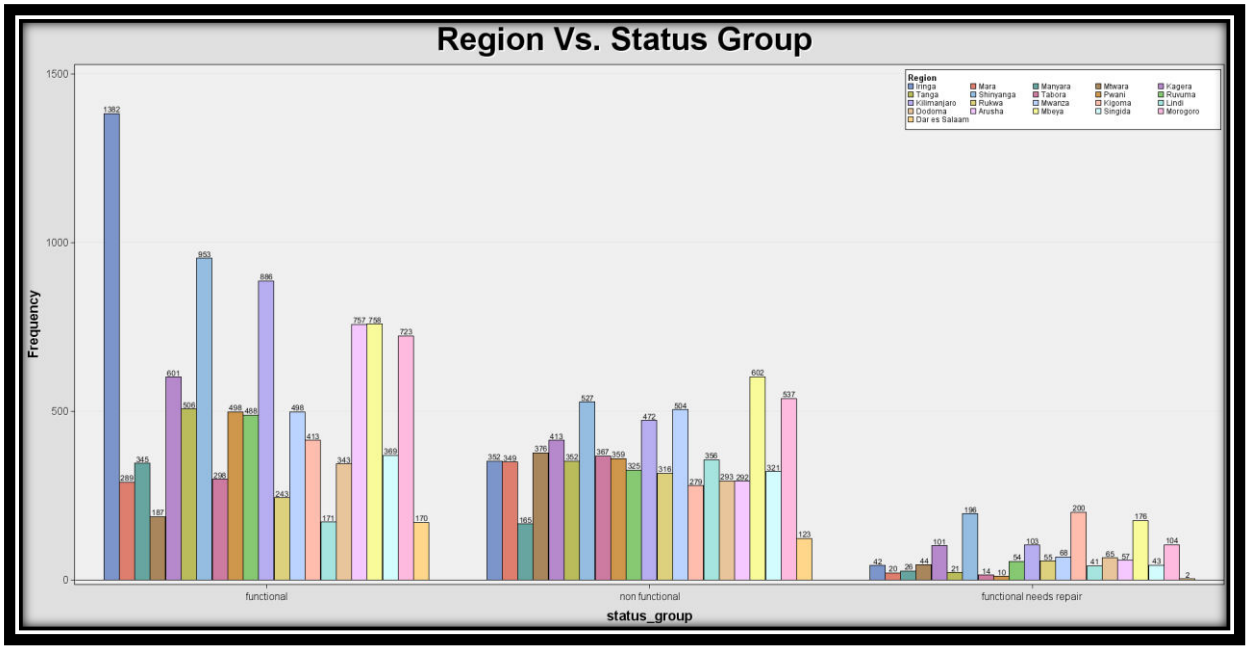


Figure 39: Region Vs. Status Group

Figure 37 shows that most of the pumps from Iringa are functional. Besides, Shiyanga, Kilimanjaro, and Mbeya are having a similar number of non-functional pumps.

3.0 Modify (Data Pre-processing):

Pre-processing data is a crucial stage in the data mining process. In data mining and machine learning initiatives, the phrase "garbage in and trash out" is particularly important. Data collection methods are frequently uncontrolled, leading to out-of-range numbers, impossible data combinations, and missing information, among other things (Techopedia, 2019).

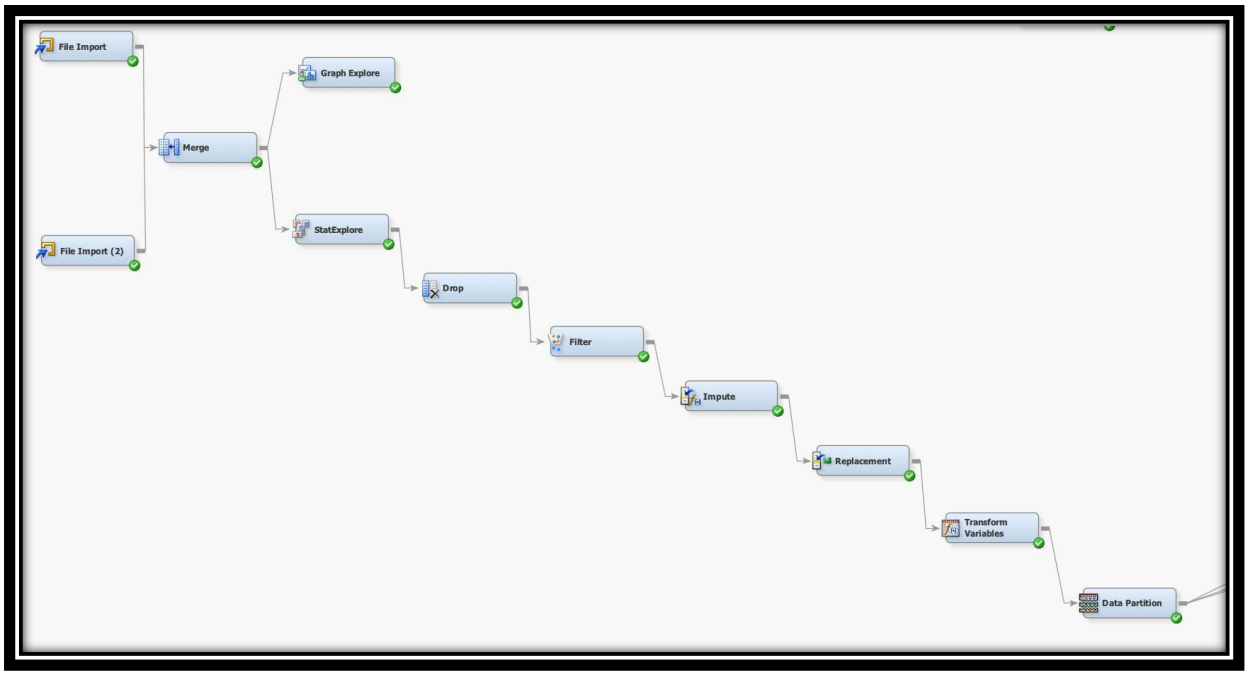


Figure 40: Data Pre-Processing Process Flow Diagram

3.1 Drop:

Here we will be dropping some of the variable because in the exploratory data analysis some of the columns are like one another. To remove the repeated values, we will be dropping the similar columns.

basin	**Default**	Input	Nominal
construction_ye	**Default**	Input	Nominal
date_recorded	**Default**	Time ID	Interval
district_code	**Default**	Input	Interval
extraction_type	**Default**	Input	Nominal
extraction_type	**Yes**	Input	Nominal
extraction_type	**Yes**	Input	Nominal
gps_height	**Default**	Input	Interval
id	**Default**	ID	Nominal
latitude	**Default**	Input	Interval
longitude	**Default**	Input	Interval
management	**Default**	Input	Nominal
management_gr	**Yes**	Input	Nominal
num_private	**Yes**	Input	Interval
payment	**Yes**	Input	Nominal
payment_type	**Default**	Input	Nominal
permit	**Yes**	Input	Nominal
population	**Default**	Input	Interval
public_meeting	**Default**	Input	Nominal
quality_group	**Yes**	Input	Nominal
quantity	**Default**	Input	Nominal
quantity_group	**Yes**	Input	Nominal
region	**Default**	Input	Nominal
scheme_manage	**Default**	Input	Nominal
source	**Default**	Input	Nominal
source_class	**Default**	Input	Nominal
source_type	**Default**	Input	Nominal
status_group	**Default**	Target	Nominal
water_quality	**Default**	Input	Nominal
waterpoint_type	**Default**	Input	Nominal
waterpoint_type	**Yes**	Input	Nominal

Figure 41: Variable selection

Figure above shows the columns which have been dropped.

3.2: Filter

The process of selecting a smaller subset of your data set for viewing or analysis is known as data filtering. Filtering is usually (but not always) temporary: the entire data set is saved, but just a portion of it is used in the calculation (Facer, 2018).

So, we do filter to remove the outliers in the data sets.

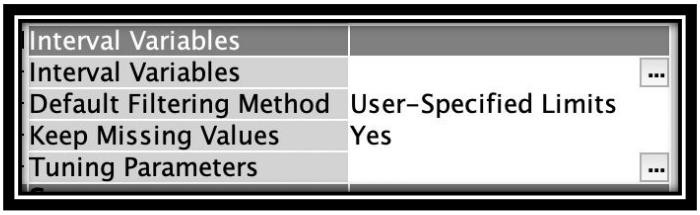


Figure 42: Interval Variables

The intervals variable set-in default filtering method is User-Specified Limits, we have set it for User-Specified Limits is because the filtering process can be set by the user.

Figure 43: Longtiude

In the figure above here we drop value until 2 in longitude as it seems to be outlier because most of the records of this column are more than 20.

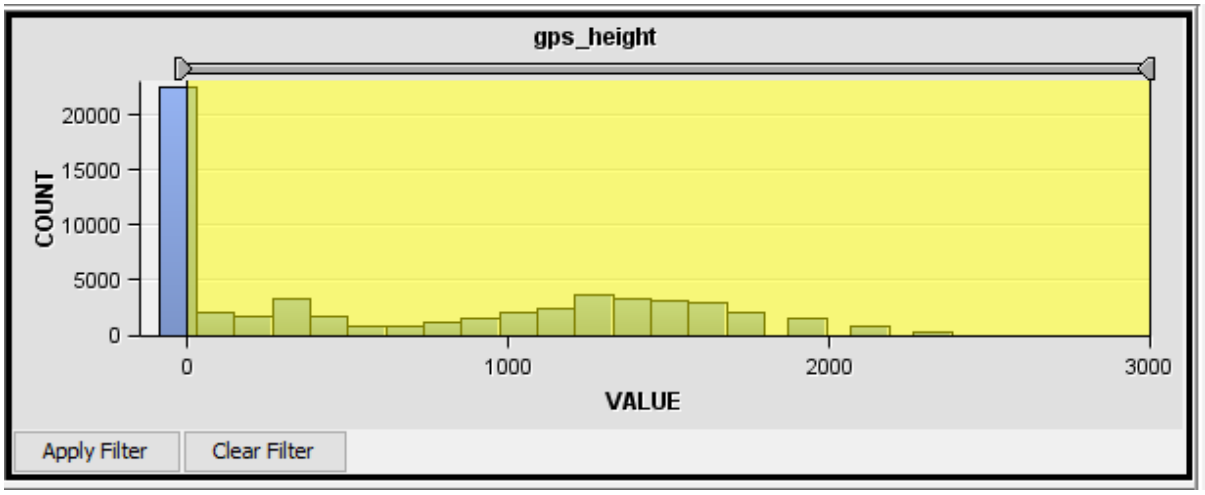
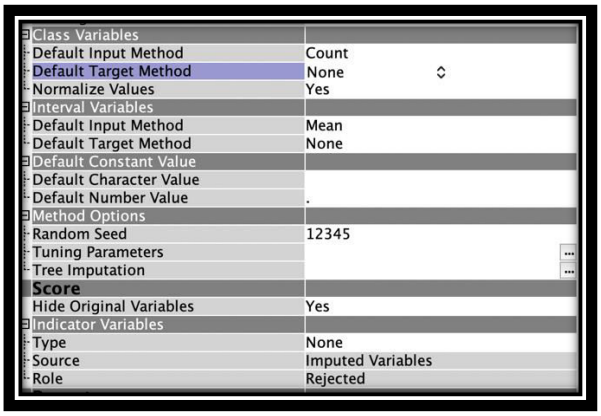


Figure 44: gps_height

In the figure 44 we are not taking gps height less than 0 as height need to be more than 0 in any case.

3.2 Impute:

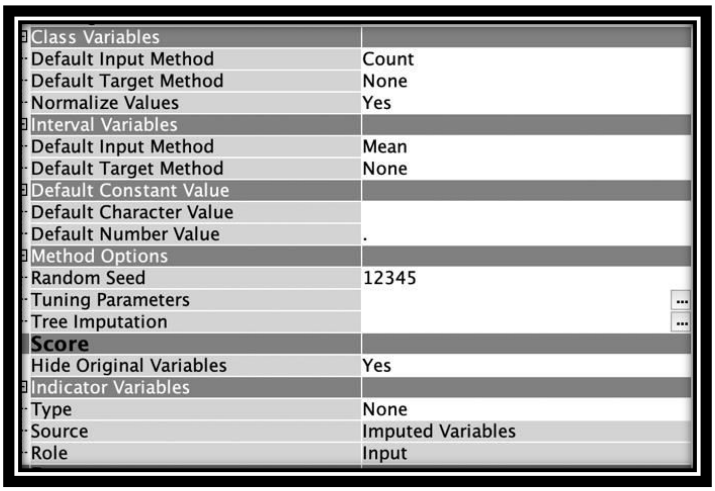
Imputation is the process of replacing missing data with replaced values in statistics. It's called "unit imputation" when substituting for a data point, and "item imputation" when substituting for a component of a data point (Kleinbort, 2019).



Class Variables	
Default Input Method	Count
Default Target Method	None
Normalize Values	Yes
Interval Variables	
Default Input Method	Mean
Default Target Method	None
Default Constant Value	
Default Character Value	
Default Number Value	.
Method Options	
Random Seed	12345
Tuning Parameters	
Tree Imputation	
Score	
Hide Original Variables	Yes
Indicator Variables	
Type	None
Source	Imputed Variables
Role	Rejected

Figure 45: Properties

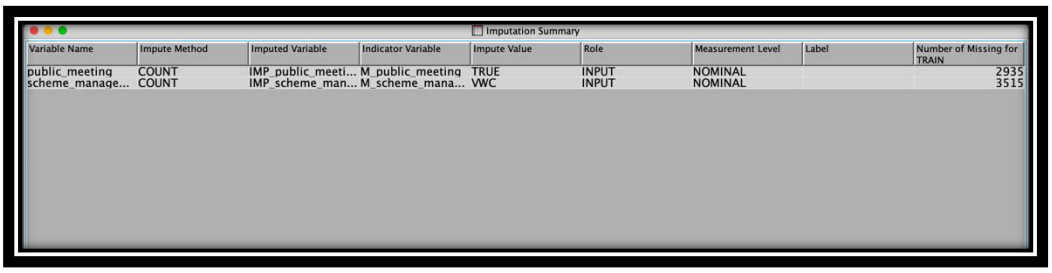
Here in the above figure the default input method is Count this is because the categorical missing values will be replaced by the mode of the column.



Class Variables	
Default Input Method	Count
Default Target Method	None
Normalize Values	Yes
Interval Variables	
Default Input Method	Mean
Default Target Method	None
Default Constant Value	
Default Character Value	
Default Number Value	.
Method Options	
Random Seed	12345
Tuning Parameters	...
Tree Imputation	...
Score	
Hide Original Variables	Yes
Indicator Variables	
Type	None
Source	Imputed Variables
Role	Input

Figure 46: Properties

The Role will be changed to input because during modelling there could be a possibility of rejection of variables if the role as rejected.



Imputation Summary

Variable Name	Impute Method	Imputed Variable	Indicator Variable	Impute Value	Role	Measurement Level	Label	Number of Missing for TRAIN
public_meeting	COUNT	IMP_public_meeti...	M_public_meeting	TRUE	INPUT	NOMINAL		2935
scheme_manage...	COUNT	IMP_scheme_man...	M_scheme_mana...	VWC	INPUT	NOMINAL		3515

Figure 47: Imputation Summary

Here with the help of count we have replaced 2935 missing values of *public_meeting* with mode and 3515 values of *scheme_management*.

3.2.1: Public_meeting:

Before imputation of the column *public_meeting*. We can see there are some unknown vales in public_meeting. So in order to remove the unknown values we will be imputation those into True because we are imputation with mode.

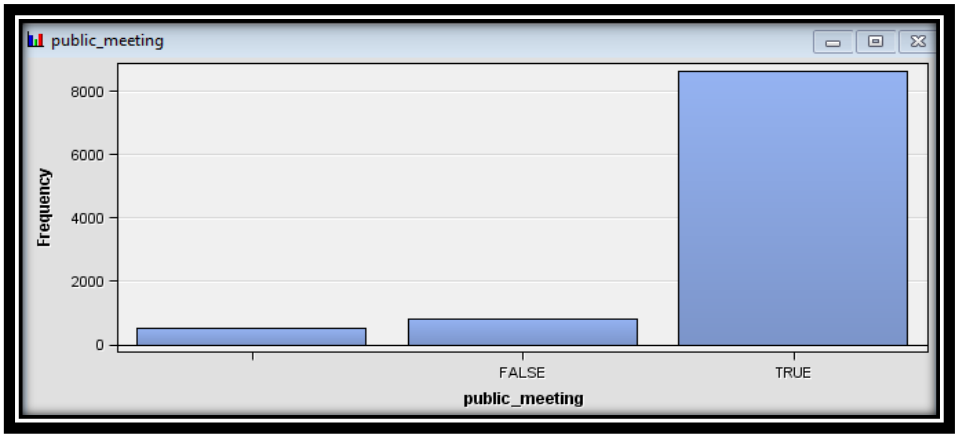


Figure 48: Public_Meeting

After Imputation with mode we can see the uknown values have been imputed to true in figure 47.

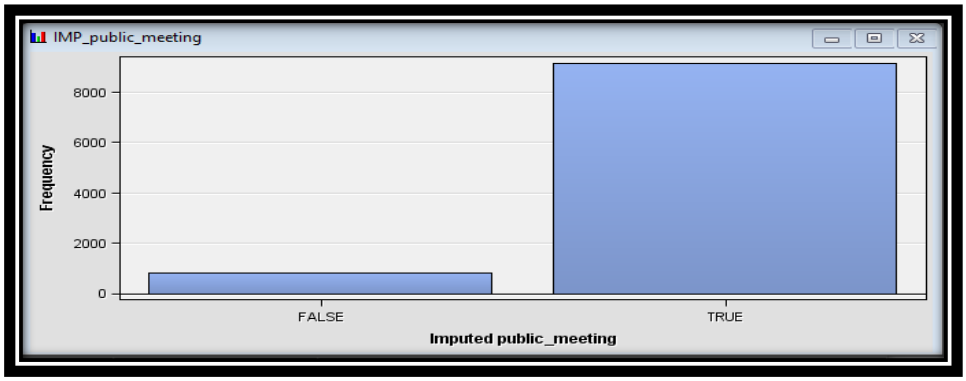


Figure 49: Public_Meeting(Imputed)

3.2.2: Scheme_managemt:

Before imputation of the column scheme_managemt.

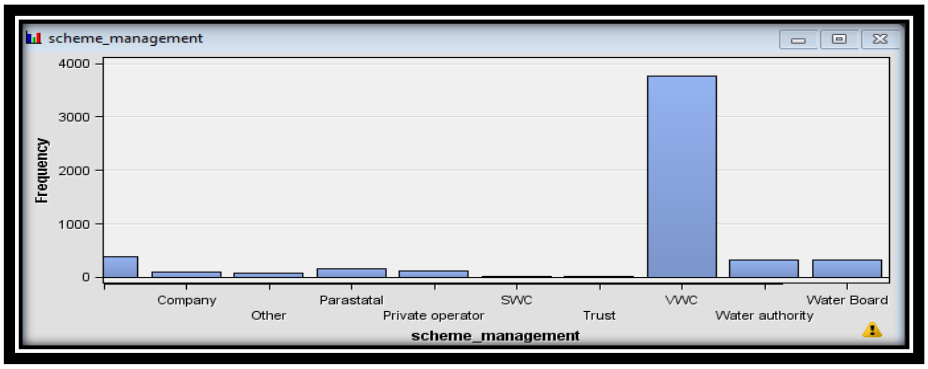


Figure 50: Scheme_Managemt

After Imputation with mode in the coloumn scheme_managemt.

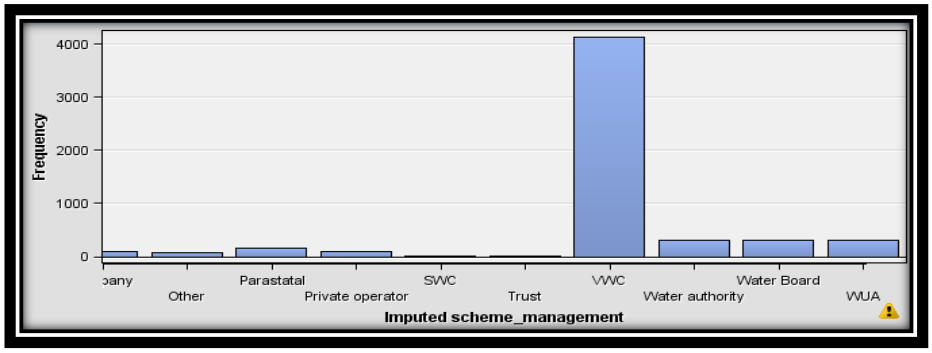


Figure 51: Scheme_management(Imputed)

3.3: Replacement

Train	
Interval Variables	
Replacement Editor	
Default Limits Method	None
Cutoff Values	
Class Variables	
Replacement Editor	
Unknown Levels	Mode (most frequent level)
Score	
Replacement Values	User-Specified
Hide	No
Report	
Replacement Report	Yes
Status	
Create Time	8/4/21 5:46 AM
Run ID	137f9b3d-8652-7243-aa02-2af0817efe0a

Figure 52: Train Varibales

The Default method is set to none because it won't filter the interval variables, the replacement values will be set as User-Specified because the replaced value can be specified by the user itself. Here in unknown level of class variable we used mode method so that it will fill up all the unknown value with mode.

Variable	Formatted Value	Replacement Value	Frequency Count	Type	Character Unformatted Value	Numeric Value
source	lake		639C		lake	.
source	other		198C		other	.
source	unknown		63C		unknown	.
source	_UNKNOWN_	_DEFAULT_	. C			.
source_class	groundwater		42946C		groundwater	.
source_class	surface		12885C		surface	.
source_class	unknown		261C		unknown	.
source_class	_UNKNOWN_	_DEFAULT_	. C			.
source_type	spring		16938C		spring	.
source_type	shallow well		14996C		shallow well	.
source_type	borehole		11012C		borehole	.
source_type	river/lake		10037C		river/lake	.
source_type	rainwater harvesting		2200C		rainwater harvesting	.
source_type	dam		648C		dam	.
source_type	other		261C		other	.
source_type	_UNKNOWN_	_DEFAULT_	. C			.
status_group	functional		30607C		functional	.
status_group	non functional		21596C		non functional	.
status_group	functional needs repair		3889C		functional needs repair	.
status_group	_UNKNOWN_	_DEFAULT_	. C			.
water_quality	soft		48384C		soft	.
water_quality	salty		4380C		salty	.
water_quality	unknown		1632C		unknown	.
water_quality	milky		799C		milky	.
water_quality	coloured		466C		coloured	.
water_quality	salty abandoned		221C		salty abandoned	.
water_quality	fluoride		198C		fluoride	.
water_quality	fluoride abando		12C		fluoride abando	.
water_quality	_UNKNOWN_	_DEFAULT_	. C			.
waterpoint_type	communal standpipe		27801C		communal standpipe	.
waterpoint_type	hand pump		15809C		hand pump	.
waterpoint_type	other		5797C		other	.
waterpoint_type	communal standpipe multiple		5784C		communal standpipe multiple	.
waterpoint_type	improved spring		779C		improved spring	.
waterpoint_type	cattle trough		115C		cattle trough	.
waterpoint_type	dam		7C		dam	.
waterpoint_type	_UNKNOWN_	_DEFAULT_	. C			.

Figure 53: Replacement

The above figure we have set some of the variables as default because using default the value that is defied will be unknown levels of property.

<u>After replacement</u>

IMP_public_meeting	Unknown	C	.	TRUE	Imputed public_meeting
IMP_scheme_management	Unknown	C	.	VWC	Imputed scheme_management
basin	Unknown	C	.	Pangani	
extraction_type	Unknown	C	.	gravity	
management	Unknown	C	.	vwc	
payment_type	Unknown	C	.	never pay	
quantity	Unknown	C	.	enough	
region	Unknown	C	.	Iringa	
source	Unknown	C	.	spring	
source_class	Unknown	C	.	groundwater	
source_type	Unknown	C	.	spring	
status_group	Unknown	C	.	functional	
water_quality	Unknown	C	.	soft	
waterpoint_type	Unknown	C	.	communal standpipe	

Figure 54: After Replacement

3.4: Transformation

Here the in the Score section hide is set to Yes because the transformed data is handled that after we transform the data the data will not change during the modelling. But if it is set to No the transformed data performed in the meta data will be used not be used during the modelling.

Interval Targets	None
Class Inputs	None
Class Targets	None
Treat Missing as Level	No
Sample Properties	
Method	First N
Size	Default
Random Seed	12345
Optimal Binning	
Number of Bins	4
Missing Values	Use in Search
Grouping Method	
Cutoff Value	0.1
Group Missing	No
Number of Bins	Variables
Add Minimum Value to Offset Value	Yes
Offset Value	1
Score	
Use Meta Transformation	Yes
Hide	Yes
Reject	No
Report	
Summary Statistics	Yes

Figure 55: Properties

amount_tsh	**Log**	**4**	Input	Interval
basin	**Default**	**4**	Rejected	Nominal
construction_ye	**Default**	**4**	Rejected	Nominal
district_code	**Log**	**4**	Input	Interval
extraction_type	**Default**	**4**	Rejected	Nominal
gps_height	**Default**	**4**	Input	Interval
latitude	**Default**	**4**	Input	Interval
longitude	**Default**	**4**	Input	Interval
management	**Default**	**4**	Rejected	Nominal
payment_type	**Default**	**4**	Rejected	Nominal
population	**Equalize**	**4**	Input	Interval

Figure 56: Data

3.4.1: amount_tsh

Before transformation:

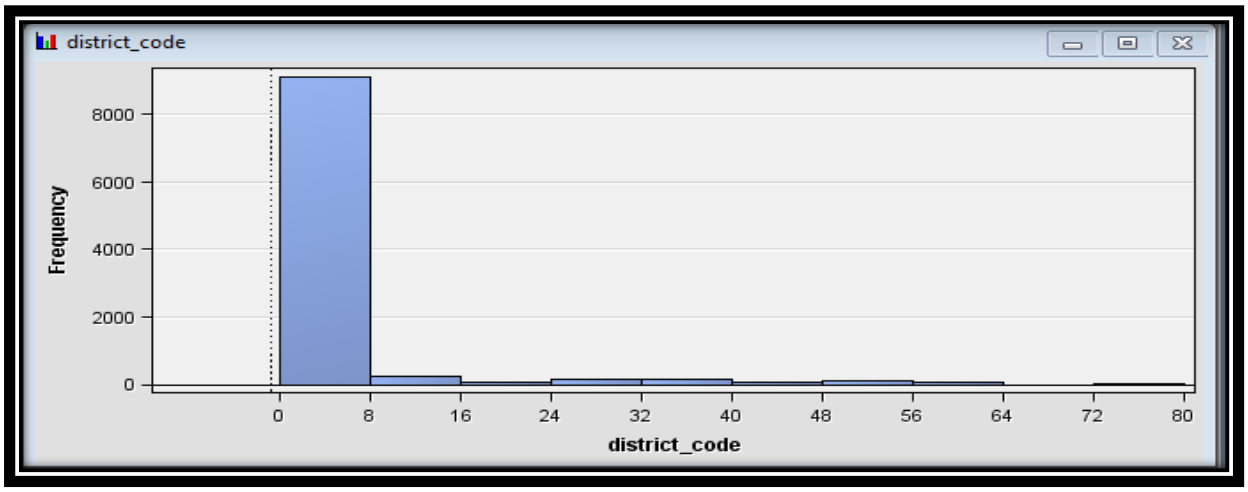


Figure 57: District_code

As amount_tsh and distinct_code does not follow normal bell curve shape we can use log transformation with the help of which we can make it as normal bell shape histogram. The above figure 55 shows the histogram of district code before transformation. Similarly in figure 56 we can see histogram of this same column after log transformation.

After transformation:

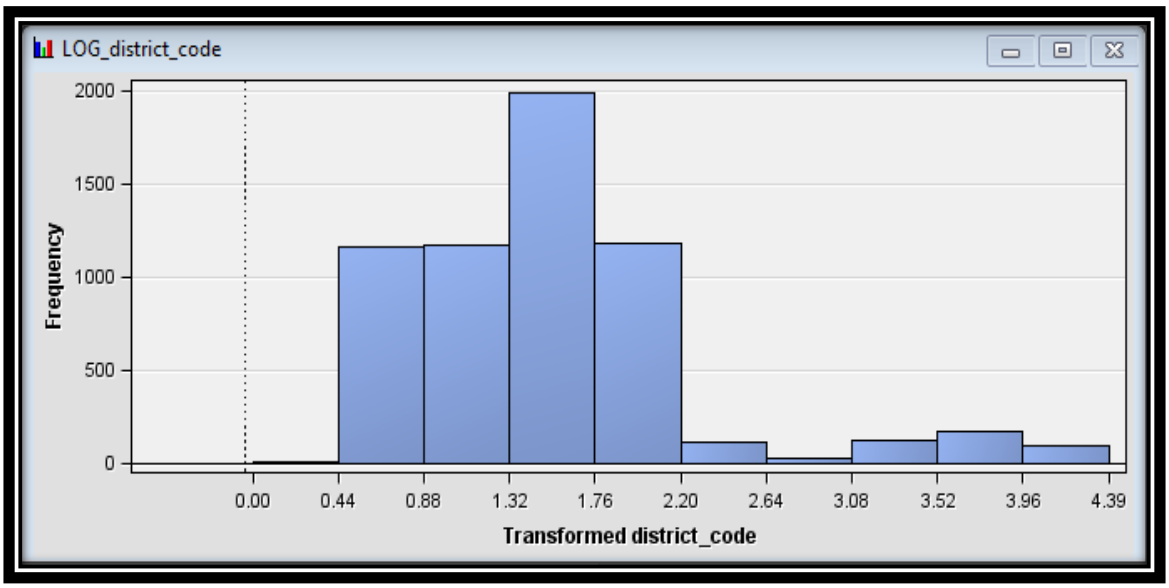


Figure 58: District_Code(Transformed)

3.4.2: population

Before Transformation:

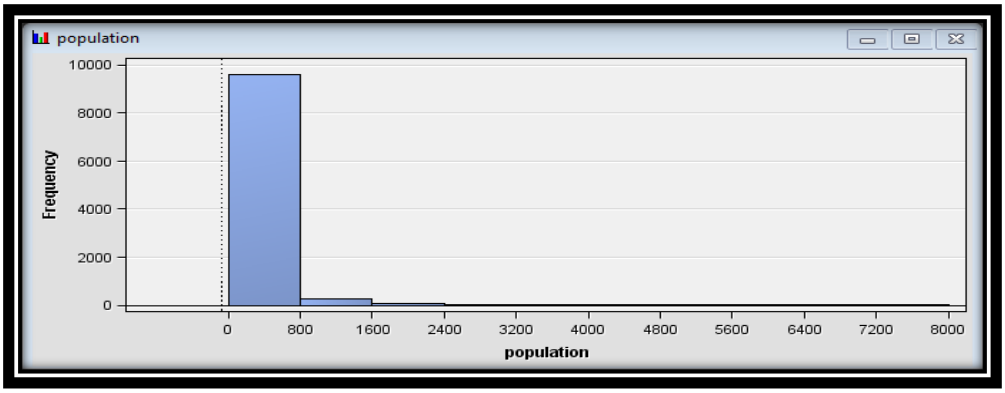


Figure 59: Population

In the above figure 59 we can see the histogram is not appropriate as it is not equalizing. So, to get rid of this issue we can use equalize process to transform it. Equalize process will automatically spread this value by which it will improve contrast. In figure 21 we can see the population histogram after transformation.

After Transformation:

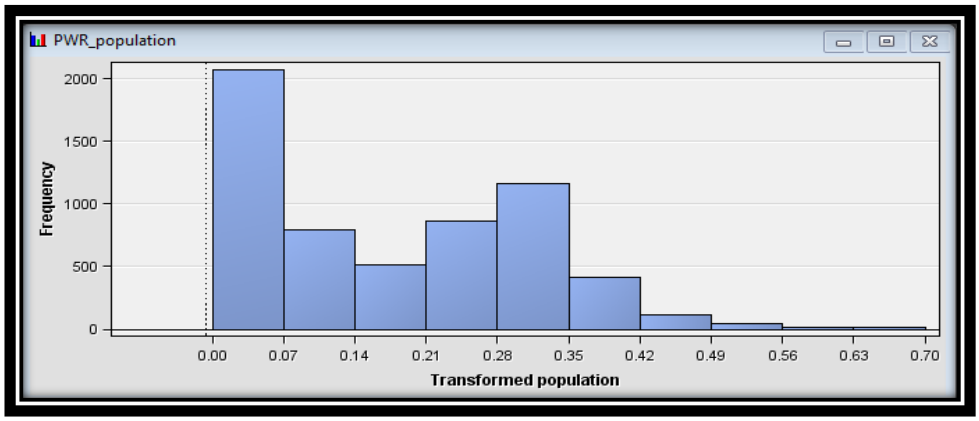


Figure 60: Population(Transformed)

3.4.2: construction_year

Before Transformation:

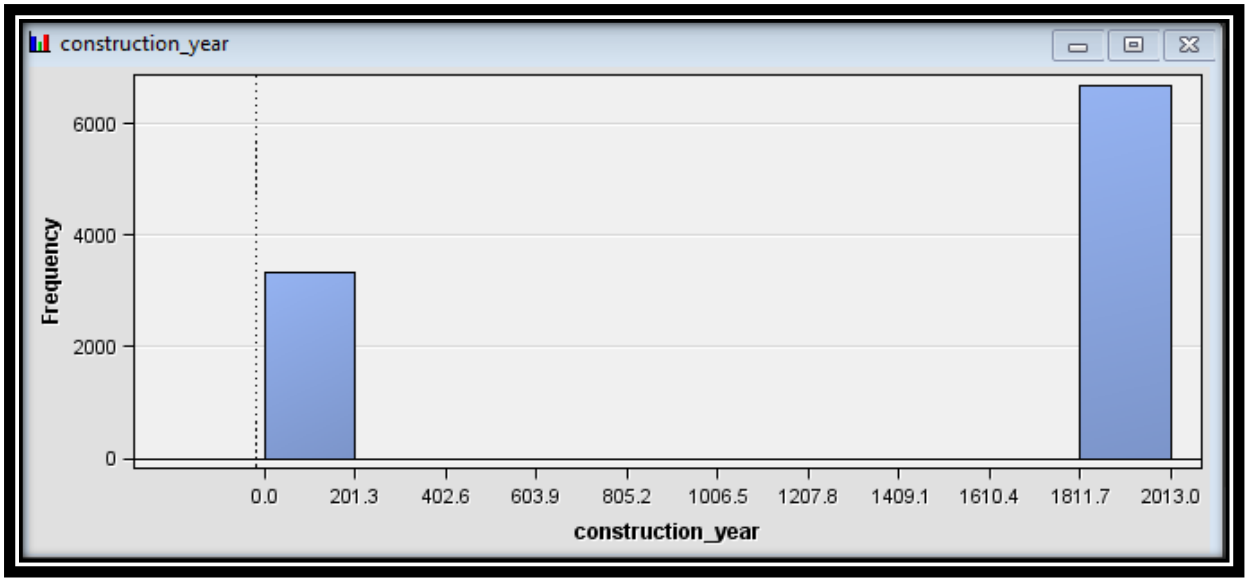


Figure 61: Construction Year

For the transformation of the year, we are using quantile method as by log and other method are not appropriate for the transformation of year. Hear with the help of quantile, we can make group of year together. We can see figure 62 which is after transformation of the year.

After transformation:

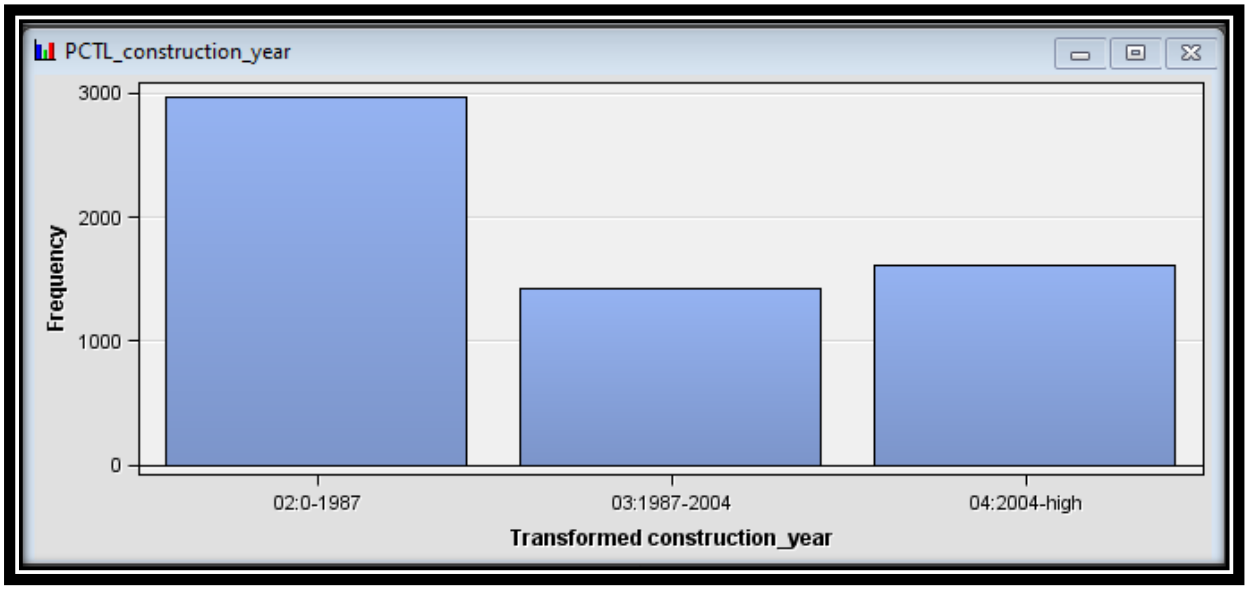


Figure 62: Construction Year (Transformed)

3.4.1: Data partition

Data Set Allocations	
Training	50.0
Validation	50.0
Test	0.0

Figure 63: Data Allocations

As for the data partition we set the training data as 50, validation data as 50 and the test data as 0.

Here train data is used for preliminary model fitting, so this way we can find the best model weights using the data. Then the validation data is used for adequacy of the data in the model comparison, As for the test data it is used to generate the final, unbiased data or the error in the model. (SAS Help Centre, 2017)

4.0 Model

4.1 NEURAL NETWROK (Nowshad)

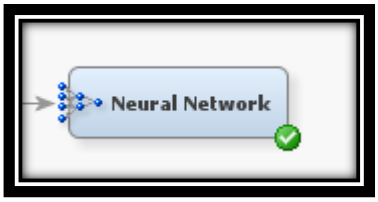


Figure 64: Neural Network

A neural network is an algorithm series which undertake to detect fundamental relationships within a dataset with the help of a process which imitate the process that human brain works ("Neural Network Definition," 2021).

In a neural network we generally find an input layer, an output layer that is target and between these two layers (input and output) there is another hidden layer. These layers relate to each other through nodes by this a neural network ("Neural Networks - What are they and why do they matter?" 2021).

Neural Network (before any Pruning)

Here in the below figure below I didn't do any changes where we can see number of hidden Units are 3 and architecture method is Multilayer perception.

Property	Value
Architecture	Multilayer Perceptron
Direct Connection	No
Number of Hidden Units	3
Randomization Distribution	Normal
Randomization Center	0.0
Randomization Scale	0.1
Input Standardization	Standard Deviation
Hidden Layer Combination Function	Default
Hidden Layer Activation Function	Default
Hidden Bias	Yes
Target Layer Combination Function	Default
Target Layer Activation Function	Default
Target Layer Error Function	Default

Figure 65: Validation

Role	Measurement Level	Frequency Count
INPUT	INTERVAL	6
INPUT	NOMINAL	12
REJECTED	NOMINAL	16
TARGET	NOMINAL	1

Figure 66: Measurement level

From this above figure we can see there are 6 intervals 12 nominals variables are used for the prediction of the model other than that all other variables are rejected except the target variables which is status group.

Figure 67: Iteration Plot

Here in the figure 67, We can see the misclassification rate is 0.25227 and iteration number is 111 with the data pre-processing that we have done in the group part. As we know using regression mode with neural network it can increase good prediction rate. In order to see if it will be good to use regression with neural, we will be testing three different regression model with neural network to get low misclassification rate. However, if it doesn't work, I will only use neural network.

Model name	figure	Misclassification rate and iteration

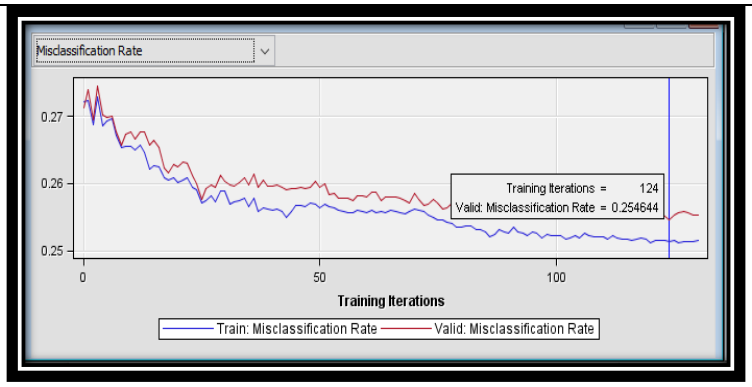
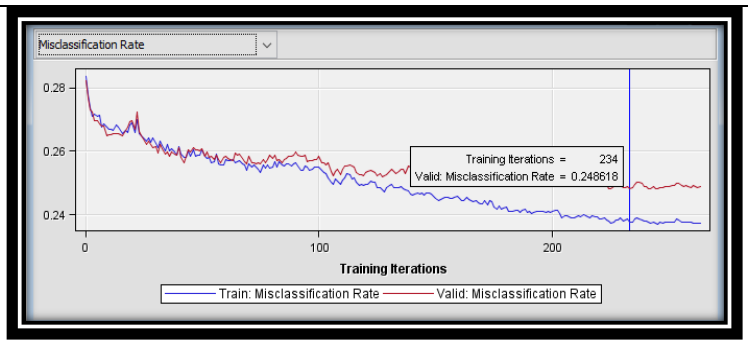
Forward	 *Figure 68: Forward Iteration*	MSC rate = 0.251596 Iteration = 127
Backward	 *Figure 69: Backward Iteration*	MSC rate = 0.254644 Iteration = 124
stepwise	 *Figure 70: StepWise Iteration*	MSC rate = 0.248618 Iteration = 234

Table 5: iteration

From the above table we can clearly see that using stepwise regression with neural network can reduce MSC rate. Here by using stepwise regression with neural network the MSC rate become 24.8 from 25.2. So, to get good prediction I will be using stepwise regression with neural networks further.

Neural Network (after Pruning)

In this particular model while I was increasing number of hidden unit we are getting low Msc rate which is shown in table below.

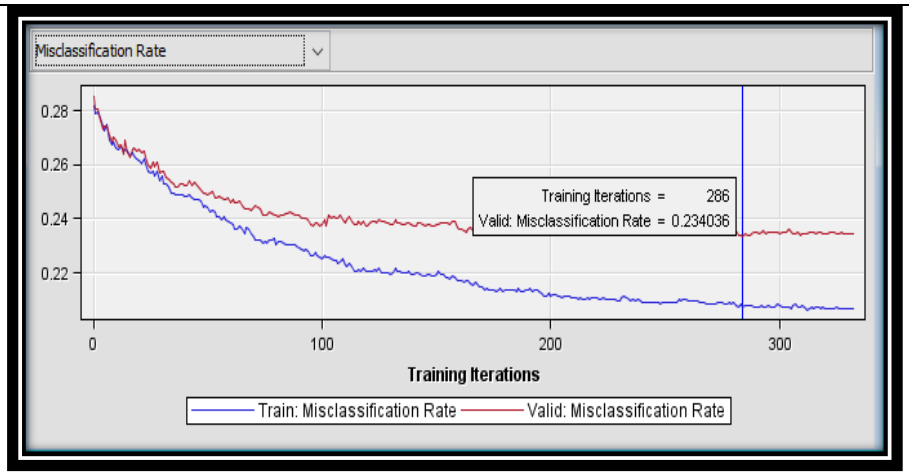
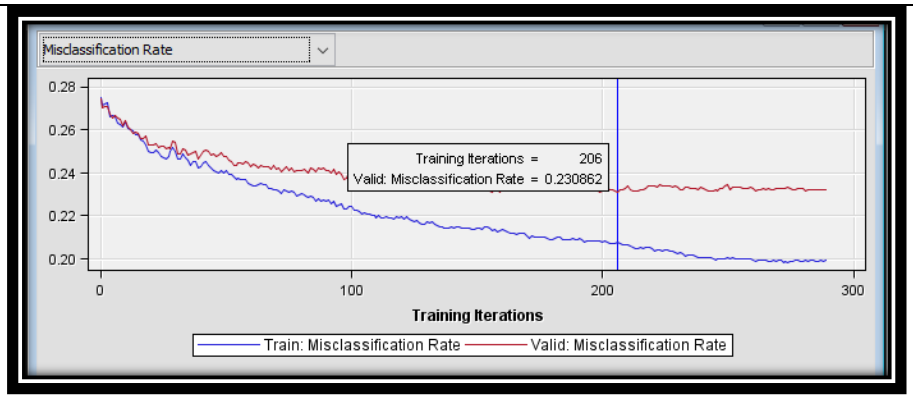
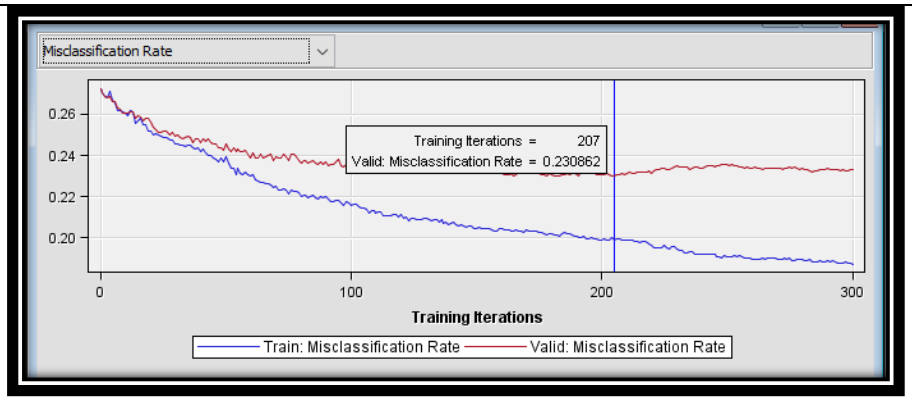
Number of Hidden units	Figure	MSC rate and iteration
20 Units	 *Figure 71: 20 Units*	Msc rate = 0.234036 Iteration = 266
30 Units	 *Figure 72: 30 Units*	Msc rate = 0.230862 Iteration = 206
40 Units	 *Figure 73: 40 Units*	Msc rate = 0.230862 Iteration = 208

Table 6: Iteration

From the above table, we see misclassification rate of both 30 and 40 hidden units are same that is 0.230862 but we will be taking 30 units as its iteration number is less than 40 hidden units.

Now to get better prediction rate I will be doing some changing in data partition part than I will be selecting best train and validation number.

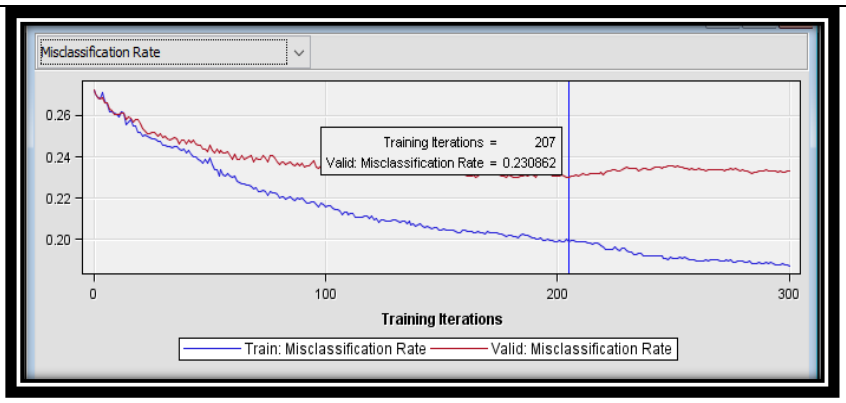
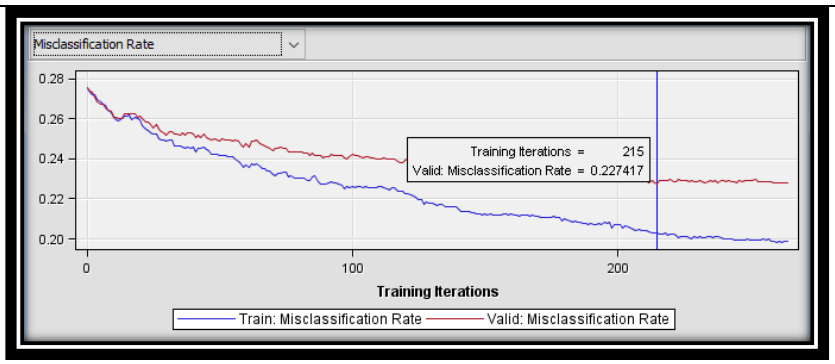
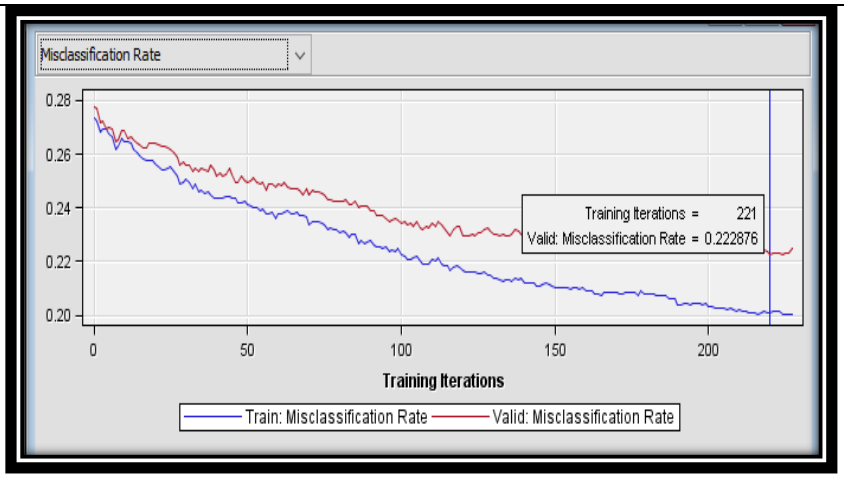
Data partition	figure	Msc and iteration
Training = 50 Validation = 50	 *Figure 74: iteration Plot*	Msc rate = 0.230862 Iteration = 208
Training = 40 Validation = 60	 *Figure 75: iteration Plot*	Msc rate = 0.227417 Iteration = 221
Training = 30 Validation = 70		Msc rate = 0.222876 Iteration = 221

	Figure 76 iteration Plot	

Table 7: Iteration

From the above table if we notice using training as 30 and validation as 70 this model will have best misclassification rate which is 0.222876. Now I will make some changes in imputation and other filtering method to know the best and final misclassification of this model.

Here after doing some imputation for instance instead of mean using median and doing any transformation is not required as misclassification rate is not increasing.

So, in neural network best misclassification rate is around 22 which is shown in figure 77.

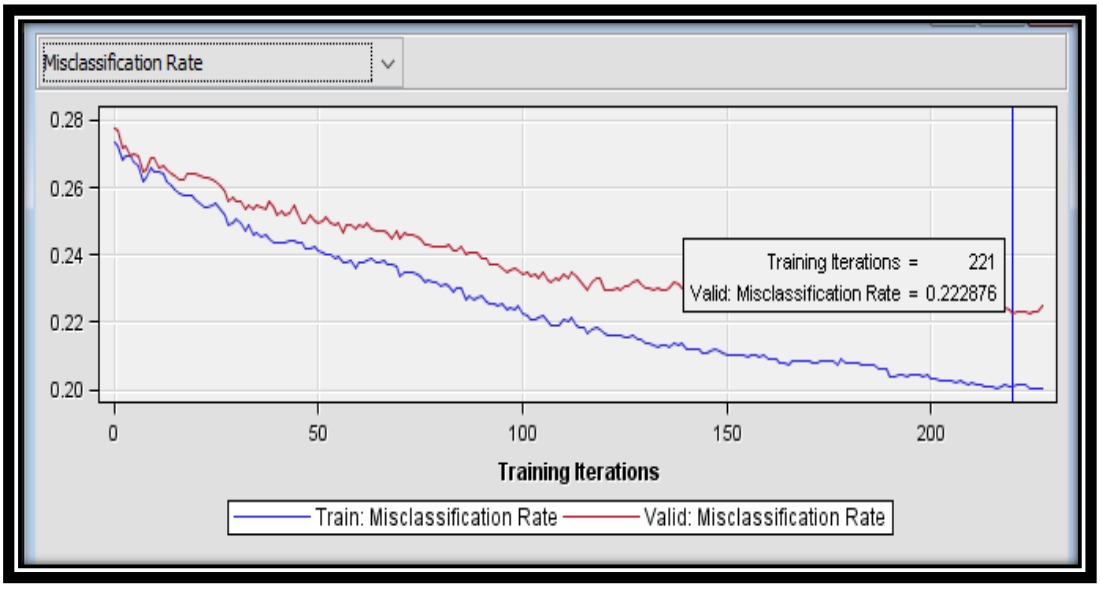


Figure 77: Missclassification Rate

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
REP_status_group	Replacement: status_group	_DFT_	Total Degrees of Freedom	78524	.
REP_status_group	Replacement: status_group	_DFE_	Degrees of Freedom for Error	74122	.
REP_status_group	Replacement: status_group	_DFM_	Model Degrees of Freedom	4402	.
REP_status_group	Replacement: status_group	_NW_	Number of Estimated Weights	4402	.
REP_status_group	Replacement: status_group	_AIC_	Akaike's Information Criterion	47928.89	.
REP_status_group	Replacement: status_group	_SBC_	Schwarz's Bayesian Criterion	88740.53	.
REP_status_group	Replacement: status_group	_ASE_	Average Squared Error	0.09561	0.106292
REP_status_group	Replacement: status_group	_MAX_	Maximum Absolute Error	0.999713	0.99998
REP_status_group	Replacement: status_group	_DIV_	Divisor for ASE	117786	50490
REP_status_group	Replacement: status_group	_NOBS_	Sum of Frequencies	39262	16830
REP_status_group	Replacement: status_group	_RASE_	Root Average Squared Error	0.309209	0.326024
REP_status_group	Replacement: status_group	_SSE_	Sum of Squared Errors	11261.53	5366.668
REP_status_group	Replacement: status_group	_SUMW_	Sum of Case Weights Times Freq	117786	50490
REP_status_group	Replacement: status_group	_FPE_	Final Prediction Error	0.106966	.
REP_status_group	Replacement: status_group	_MSE_	Mean Squared Error	0.101288	0.106292
REP_status_group	Replacement: status_group	_RFPE_	Root Final Prediction Error	0.327057	.
REP_status_group	Replacement: status_group	_RMSE_	Root Mean Squared Error	0.318258	0.326024
REP_status_group	Replacement: status_group	_AVERR_	Average Error Function	0.332169	0.375012
REP_status_group	Replacement: status_group	_ERR_	Error Function	39124.89	18934.36
REP_status_group	Replacement: status_group	_MISC_	Misclassification Rate	0.201136	0.222698
REP_status_group	Replacement: status_group	_WRONG_	Number of Wrong Classifications	7897	3748

Figure 78: Fit Statisitics

The above figure shows final fit statistics table where we can see a good rate of status group (that is target) for both train and validation data in some of the method which are Mean square error, Average Squared error, Root mean squared error and misclassification.

Data Role=VALIDATE Target Variable=REP_status_group Target Label=Replacement: status_group

Target	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
FUNCTIONAL	FUNCTIONAL	76.9224	89.1200	8183	48.6215
FUNCTIONAL NEEDS REPAIR	FUNCTIONAL	6.8904	62.7568	733	4.3553
NON FUNCTIONAL	FUNCTIONAL	16.1873	26.5741	1722	10.2317
FUNCTIONAL	FUNCTIONAL NEEDS REPAIR	29.3399	1.3069	120	0.7130
FUNCTIONAL NEEDS REPAIR	FUNCTIONAL NEEDS REPAIR	52.5672	18.4075	215	1.2775
NON FUNCTIONAL	FUNCTIONAL NEEDS REPAIR	18.0929	1.1420	74	0.4397
FUNCTIONAL	NON FUNCTIONAL	15.1997	9.5731	879	5.2228
FUNCTIONAL NEEDS REPAIR	NON FUNCTIONAL	3.8043	18.8356	220	1.3072
NON FUNCTIONAL	NON FUNCTIONAL	80.9960	72.2840	4684	27.8313

Figure 79: Target Variable Validation

From the given table, It can be noticed that the pump which are functional are quite high in number as well as in prediction which is 8181 and nearly 48% and in the same way one of the other target levels can be noticed though this is lower than function but still it will have a great impact which totally damaged which total number and prediction is 4684 and around 28%.

```
Data Role=VALIDATE Target=REP_status_group Target Label=Replacement: status_group

  False       True        False       True
Negative    Negative    Positive    Positive

  1796        9251        1099        4684
```

Figure 80: Outcomes

The above figure shows the result of the confusion matrix where the value of True negative is high which is more than 9 thousands of the total records which means most of the pump seems function whereas around 1796 pump which are somehow not functional or needs repairs shows as function.

To be more exact we can see it with the help of a 3*3 confusion matrix:

3*3 confusion matrix information taken from figure **			
	functional	Functional needs repair	Non-function
TRUE POSITIVE	8186	215	4684
FALSE POSITIVE	733 + 1722 = 2455	120 + 74=194	879+220=1099
TRUE NEGATIVE	215+74+220 +4684 = 5193	8183+1722+879+4684 =15468	8183+733 120+215=9251
FALSE NEGATIVE	120+879=99 9	733+220=953	1722+74=1796

Table 8: Confusion Matrix

Now with the help of above confusion matrix we can know exactly what the actual percentage of the levels is we need to repair, and which are function and non-functional.

Functional	8186/ (8186+2455) =0.769
Non-functional	215/ (215+194) = 0.52
Needs repair	4684/ (4684+1099) = 0.80

Table 9: Confusion Matrix (Result)

From this table it now clears almost 76% are functional which means there is less possibility that these pumps need repair on the other hand there is a possibility of 80% of pump needs repair it is also under function and at the end nearly 52% pump must need to be changed.

Data Role=VALIDATE Output Type=CLASSIFICATION

Variable	Numeric Value	Formatted Value	Frequency Count	Percent
I_REP_status_group	.	FUNCTIONAL	10638	63.2086
I_REP_status_group	.	FUNCTIONAL NEEDS REPAIR	409	2.4302
I_REP_status_group	.	NON FUNCTIONAL	5783	34.3613

Figure 81: Validation (Score Method)

Now to be more precise we can see from score what is the actual rate. From this above classification table, it is seen almost 63% pump are now working where ae nearly 3% needs repair and 35% are almost damaged.

According to the figure 77 the prediction rate using neural networks is around 78% depending on misclassification level.

4.2 HP TREE (Nowshad)

Figure 82: HP Tree

Hp tree helps us to visualize and create a tree model by which we can determine importance of the variable. This tree is basically built through PROCHSPLIT which stands and design for the high-performance environment of data mining ("SAS Help Center," 2021).

HP TREE (Before Pruning)

Splitting Rule	
Interval Target Criterion	Variance
Nominal Target Criterion	Entropy
Interval Bins	100
Minimum Distance	0.01
Significance Level	0.2
Bonferroni	No
Missing Values	Largest
Use Input Once	No
Maximum Branch	2
Maximum Depth	10
Minimum Categorical Size	5
Node	
Leaf Size	5
Surrogate Rules	0
Validation	

Figure 83: Splitting Rule

Here in the figure 84 shown below I didn't change anything that is keeping preprocessing and data partition as default (group part), putting interval Bins as 100, target citation as entropy in splitting rule of HP Tree I got misclassification rate as 0.247264 (shown in figure 75) which is pretty good. Now I will do pruning to see if I could make a good prediction rate.

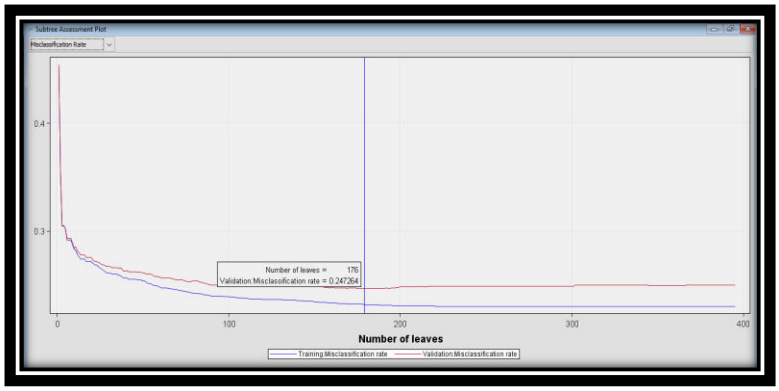


Figure 84: Missclassifaction Rate

Model Events

Target	Event	Measurement Level	Number of Levels	Order	Label
REP_status_group	NON FUNCTIONAL	NOMINAL	3	Descending	Replacement: status_group

Figure 85: Model Events

In the figure 85 it shows our target variable which is status_group and in which we have 3 levels that are functional, not functional, and functional but need to repair.

Variable Summary

Role	Measurement Level	Frequency Count
ID	INTERVAL	1
ID	NOMINAL	1
INPUT	INTERVAL	6
INPUT	NOMINAL	14
REJECTED	NOMINAL	14
TARGET	NOMINAL	1
TIMEID	INTERVAL	1

Figure 86: Variable Summary

Here it is mentioned that in Hp tree model it rejected 14 nominal value while taking 20 variable as input 2 variable as Id which belongs to interval class and there is one target value which is Status-group.

HP TREE (After Pruning)

Subtree	
Subtree Method	C4.5
Selection Method	Automatic
Confidence	0.25
Nominal Target Assessment	Misclassification
Minimum Subtree	No
Assessment Threshold Value	1.0
Number of Leaves	1
Cross Validation Folds	10
Cross Validation Seed	12345

Figure 87: Sub-Tree

In subtree section if we make subtree method as C4.5 (it works by creating subtree based on the algorithm of C4.5) we can have around 0.245 misclassification rate which is shown in figure 88.

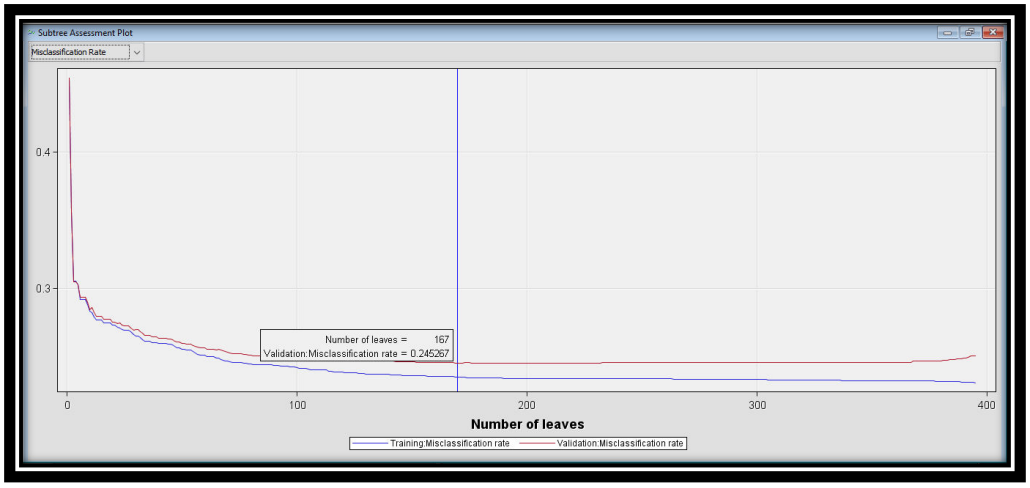
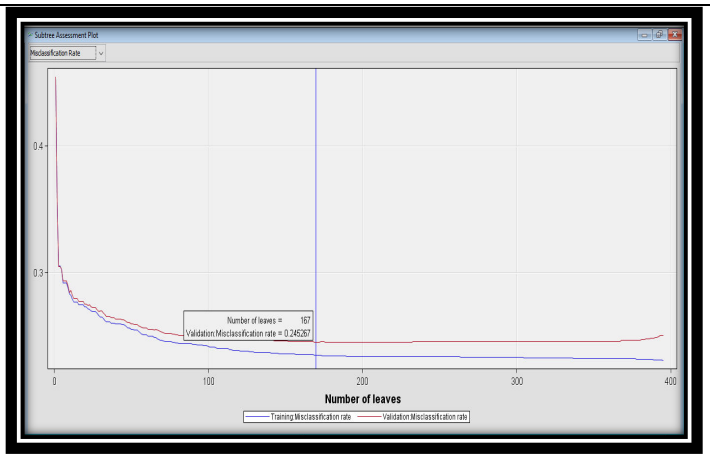
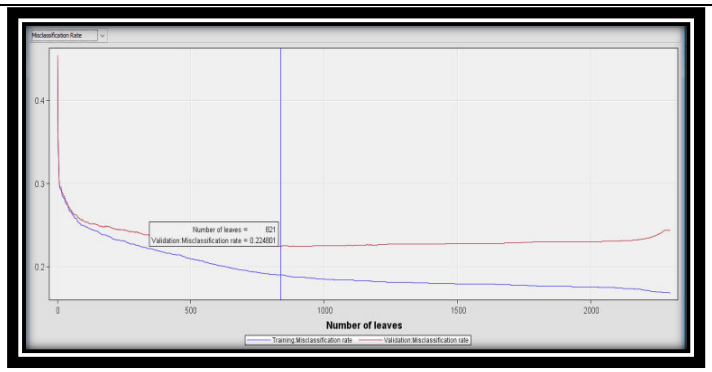


Figure 88: Number of leaves

Suitable maximum branch

Number of branches	Figure	Msc rate and iteration
2	 *Figure 89: iteration Plot*	Msc rate = 0.24567 Iteration = 821
3	 *Figure 90: iteration Plot*	Msc rate = 0.224801 Iteration = 821

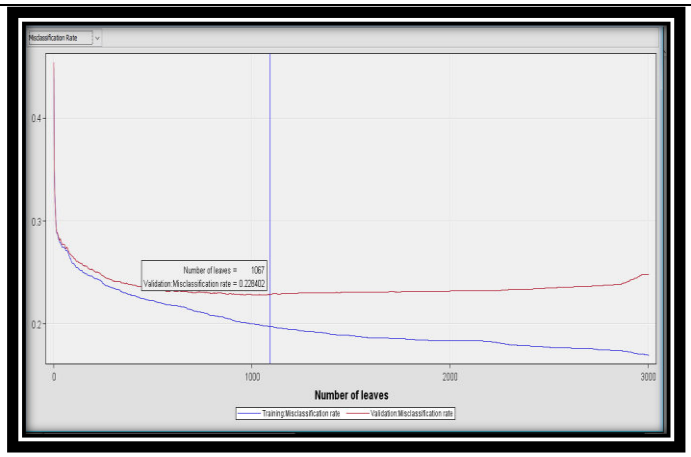
4	 *Figure 91: iteration Plot*	Msc rate = 0.228402 Iteration = 1067

Table 10: Itration table

From the above table shown above using maximum number of branches as 3 will be good as it has less misclassification rate that is 0.224 than any other branches number. So, for this further I will be using 3 maximum branch number.

Now I will see if changing maximum depth can increase prediction rate or not. Table with figure shown below:

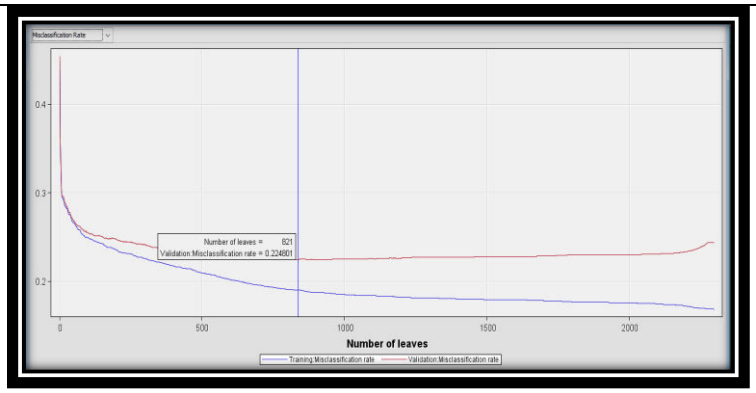
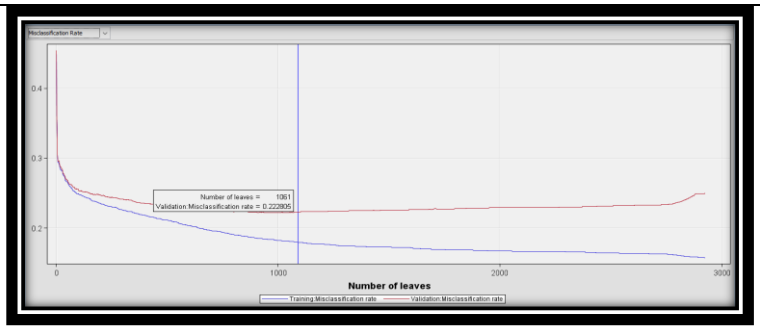
Number of depth	Figure	Msc rate and iteration
10	 *Figure 92: iteration Plot*	Msc rate = 0.224801 Iteration = 821
15		Msc rate = 0.222805 Iteration = 1061

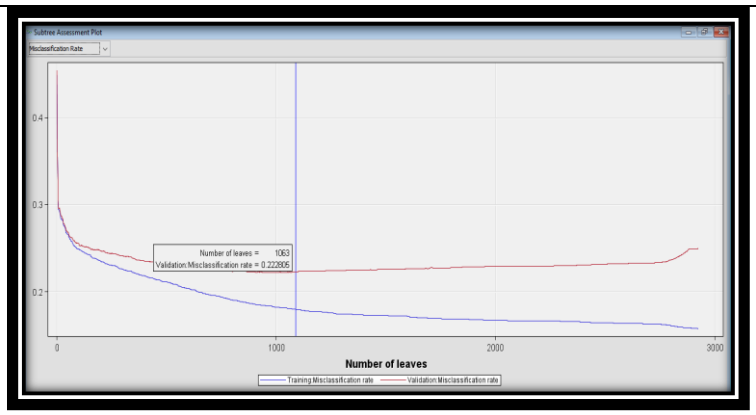
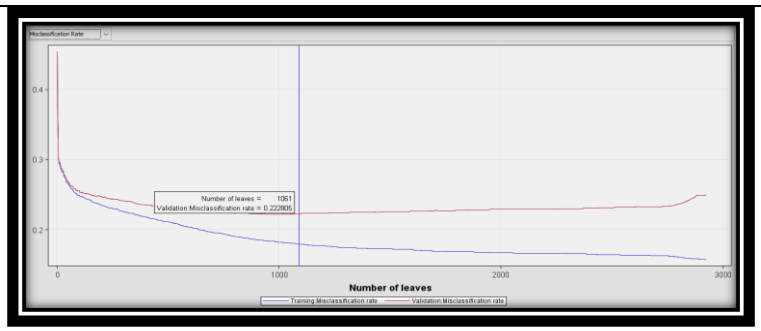
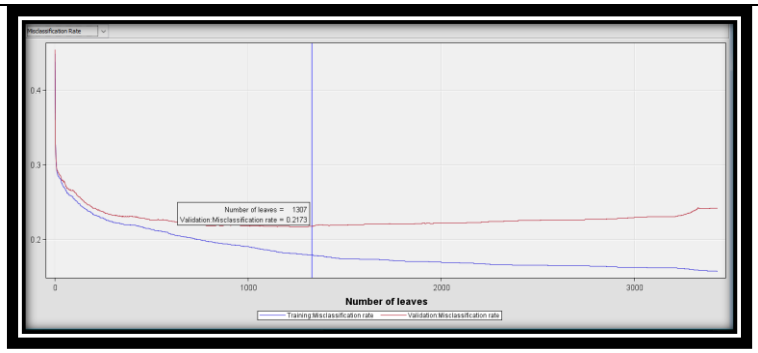
	Figure 93: iteration Plot	
20	 *Figure 94: iteration Plot*	Msc rate = 0.222805 Iteration = 1063

Table 11: Iteration table

Here we can use maximum depth as 15 as misclassification does not increase if we increase more than 15 depth numbers. So, our misclassification is now 0. 222805.Now I need to see appropriate data partition.

Data partition	Figure	Msc rate and iteration
Training = 50 Validation = 50	 *Figure 95: iteration Plot*	Msc rate = 0.222805 Iteration = 1061
Training = 60 Validation = 40	 *Figure 96: iteration Plot*	Msc rate = 0.2173 Iteration = 1307

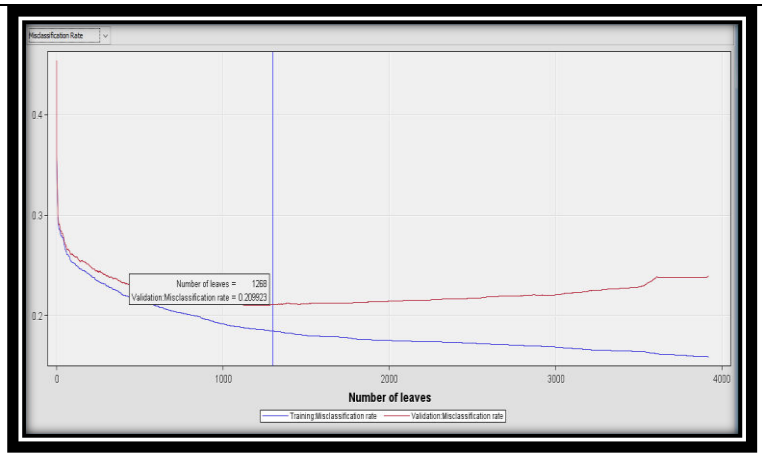
Training = 70 Validation = 30	 *Figure 97: iteration Plot*	Msc rate = 0.209923 Iteration = 1268

Table 12: Iteration Table

From the above table we see for good prediction rate we can use training data as 70 and validation as 30 as we are having less Misclassification rate on that which is 0.209923.

After data partition when I checked data pre-processing if I could improve prediction rate but I could see any significant changes there so for hp tree best misclassification is around 20 that is prediction rate will be nearly 80%

Finally, to know the actual prediction rate I can use some BarCharts so that it would be easy to understand.

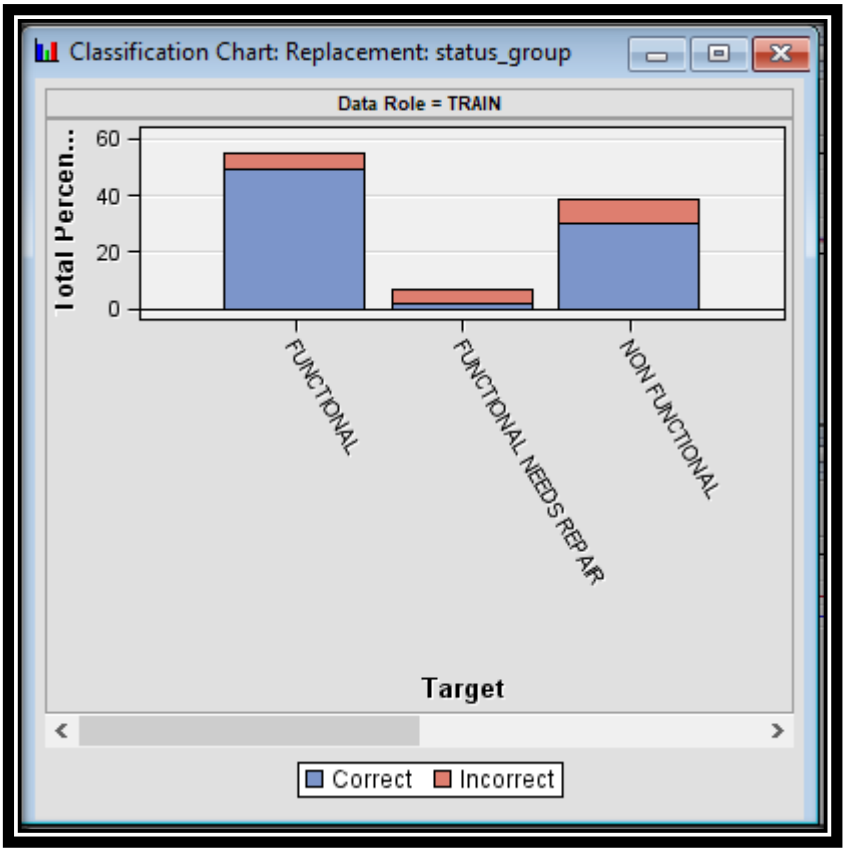


Figure 98: Replacament Graph

The above figure shows the actual value of both validation and train data where blue colour indicates actual percentage functional, needs repair and non-functional pump whereas pink colour shows it this must could be incorrect though they are with functional, Non-functional and need repairs.

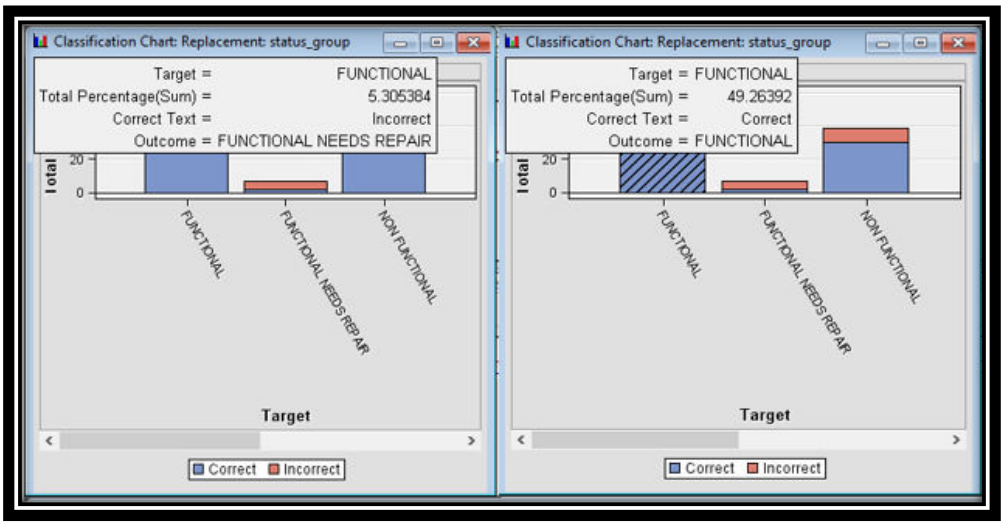


Figure 99: Bar Chat for Status_Group (Functional)

Figure above shows though in the hp tree there are around 54.5% functional pump, in that there is a possibility of having 5.3% pump which needs repairs.

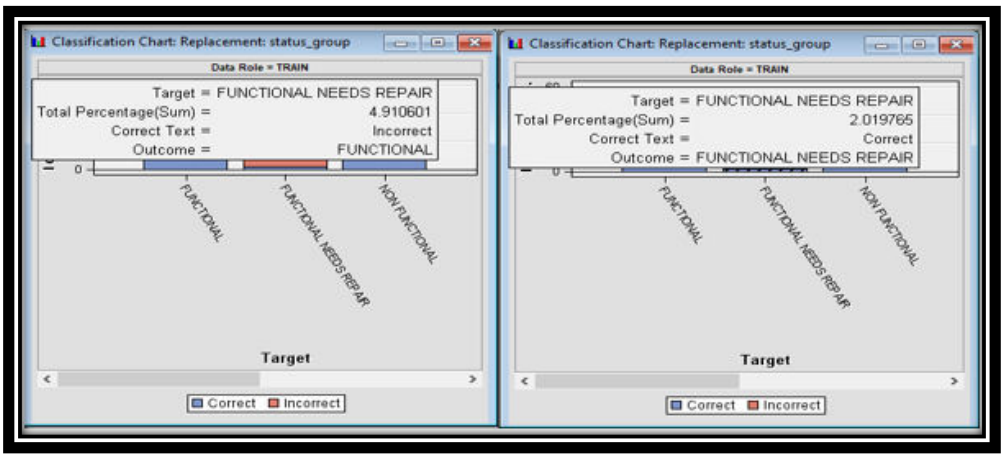


Figure 100: Bar Chat for Status_Group (Functional needs repair)

Again, as I explain before here in functional needs repair levels, they can be closely 5% of the outcome could be function from around 7% of records which are functional but needs repair.

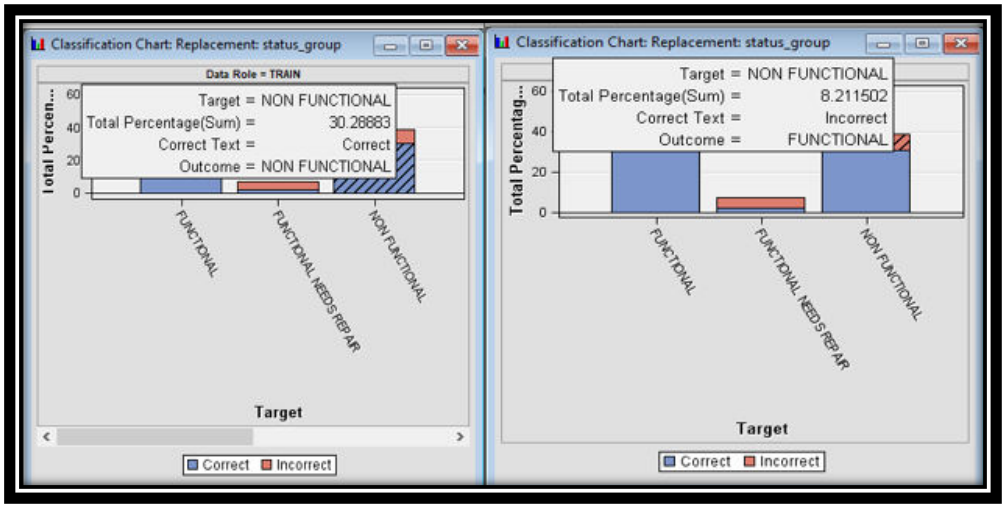


Figure 101: Bar Chat for Status_Group (non-Functional)

As figure mentioned above there are almost 40% data which are non-functional, but from these records there we could find more than 8% pump which could be totally functional and other than that rest 32% would be non-functional.

So, from the above finding there are around 62% pump currently functional, and 30.288 % are among non-functional and rest of the other pump which are around 7.4 % are functional but needs repair if we do model with following hp tree and the prediction rate of this would around 80% correct which is shown in figure 97.

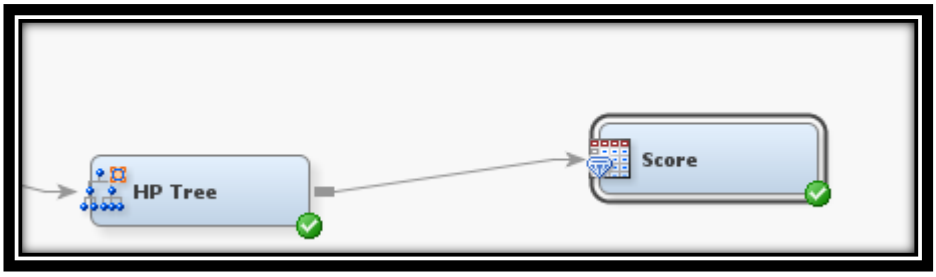


Figure 102: Score Implemtation

For the further confirmation we can apply score method in this model shown in above figure.

Data Role=VALIDATE Output Type=CLASSIFICATION

Variable	Numeric Value	Formatted Value	Frequency Count	Percent
I_REP_status_group	.	FUNCTIONAL	10269	61.0160
I_REP_status_group	.	FUNCTIONAL NEEDS REPAIR	481	2.8580
I_REP_status_group	.	NON FUNCTIONAL	6080	36.1260

Figure 103: Score Method

Here as we mentioned with the help of BarCharts the above figure shows almost there similar prediction rate for validation which are around 61 for functional, around 3 functional but needs repairs and rest of the pump are non-functional which are 36%.

4.3 Decision Tree (Tarun Aitha TP058015)

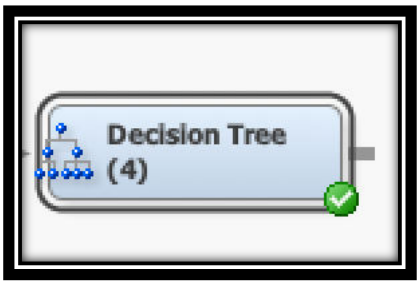


Figure 104: Decision Tree

A decision tree is a decision-making aid that employs a tree-like model of decisions and their potential results, such as chance event outcomes, resource costs, and utility. It's one method to show an algorithm made up entirely of conditional control statements. (Lucidchart, 2019)

Model Events

Target	Event	Measurement Level	Number of Levels	Order	Label
REP_status_group	NON FUNCTIONAL	NOMINAL	3	Descending	Replacement: status_group

Figure 105: Model Events

Well for the model even for the 3 way decision tree the measurement level is Nominal and the even is non-functional and the target variable is Status group.

Decision tree(Before Pruning)

Splitting Rule	
Interval Target Criterion	ProbF
Nominal Target Criterion	ProbChisq
Ordinal Target Criterion	Entropy
Significance Level	0.2
Missing Values	Use in search
Use Input Once	No
Maximum Branch	2
Maximum Depth	6
Minimum Categorical Size	5

Figure 106: Splitting Rule

Here the splitting rule is set to default where the maximum branch is set as 2 and the depth is set to 6.

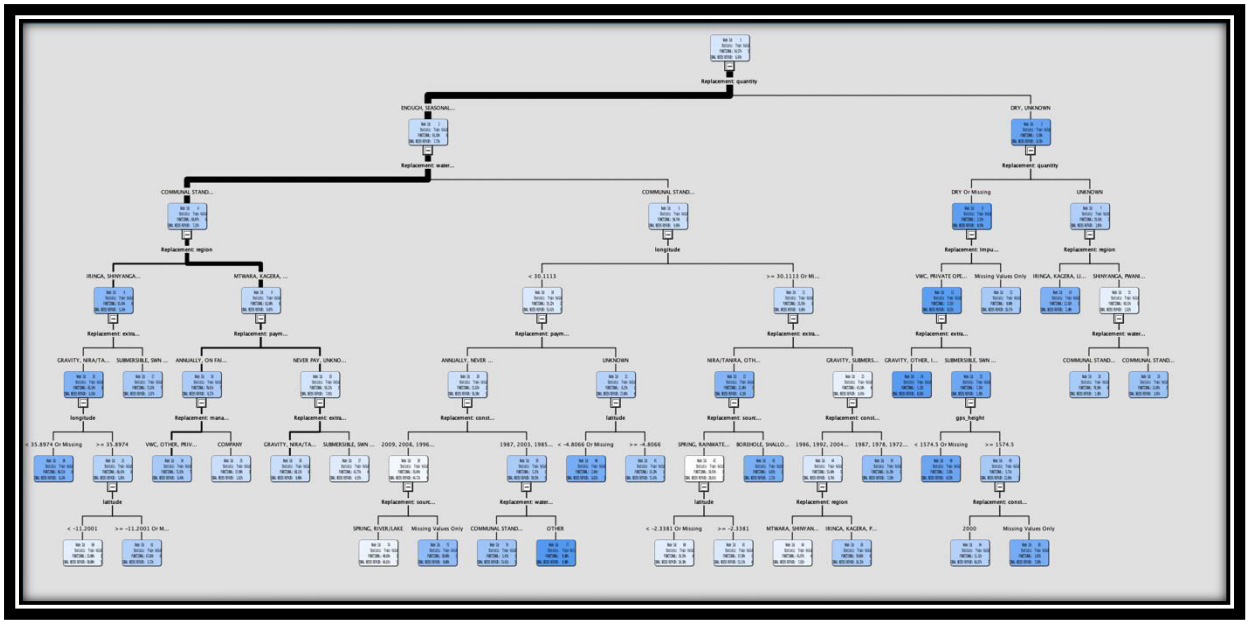


Figure 107: Decsion Tree

Here the full decision tree before pruning.

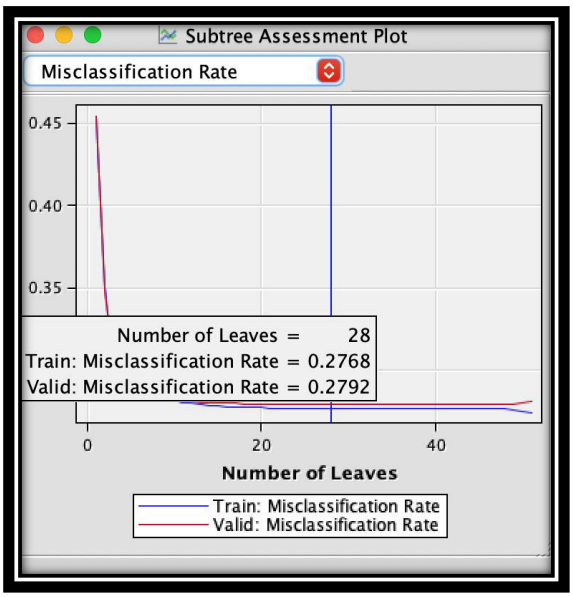


Figure 108: Misclassification Rate

Here the misclassification rate is 0.27 which is 73% accurate, and the number of leaves is 28. Now we will the decision tree after we prune it.

Decision tree (After pruning)

While pruning in the splitting rule the maximum branch is set to 2 and the maximum depth is set 10.

Splitting Rule	
Interval Target Criterion	ProbF
Nominal Target Criterion	ProbChisq
Ordinal Target Criterion	Entropy
Significance Level	0.2
Missing Values	Use in search
Use Input Once	No
Maximum Branch	2
Maximum Depth	10
Minimum Categorical Size	5

Figure 109: Model Modification

The full decision tree looks like

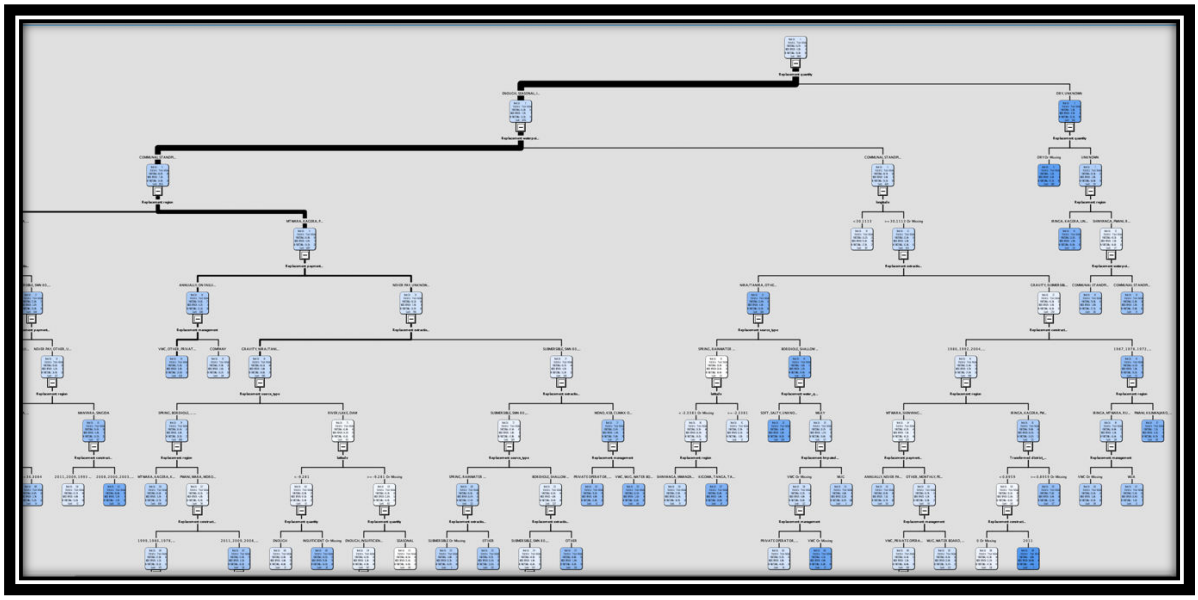


Figure 110: Decision Tree(Before Pruning)

The misclassification rate is 0.25 which is 75% accurate.

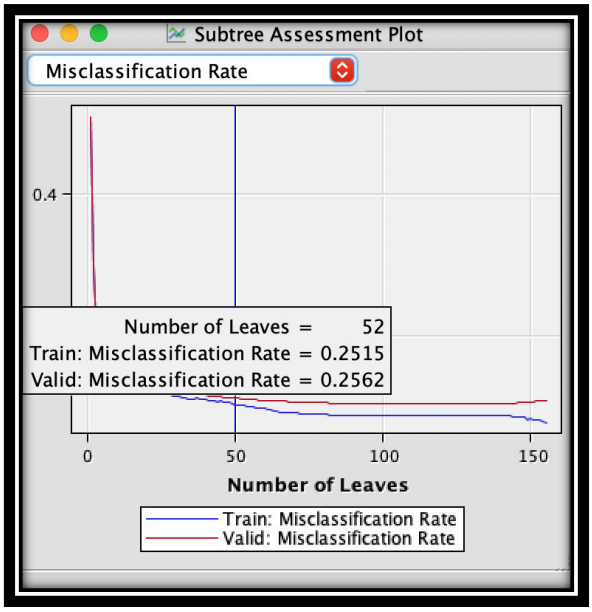


Figure 111: Misclassification Rate

The Misclassification rate has decreased when the maximum number of branches have been decreased. But when we look at the decision tree we it is too big and it might lead to overfitting.

In order to avoid it we I will be pruning the tree again with less number of maximum depth to avoid overfitting.

Splitting Rule	
Interval Target Criterion	ProbF
Nominal Target Criterion	ProbChisq
Ordinal Target Criterion	Entropy
Significance Level	0.2
Missing Values	Use in search
Use Input Once	No
Maximum Branch	3
Maximum Depth	7
Minimum Categorical Size	5

Figure 112: Model Modification

Now I have set the maximum depth as 7 and maximum branch is 3. The decision tree looks like,

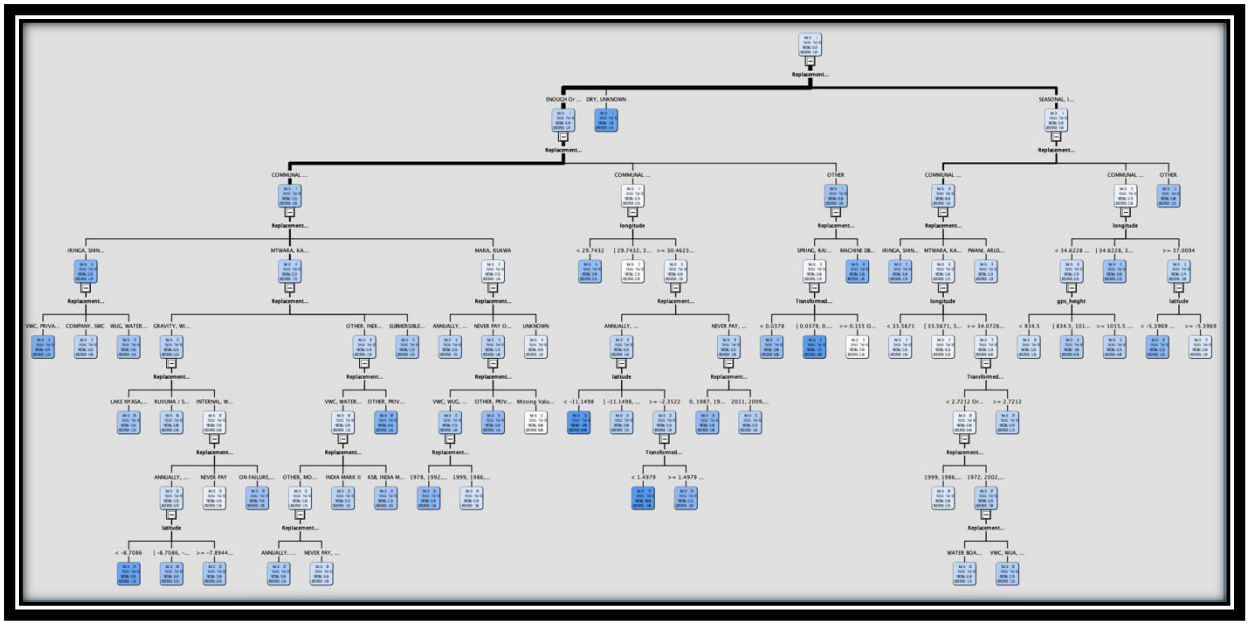


Figure 113: Decision Tree (After Pruning)

The misclassification rate has been decreased by 0.1 which is 74% accurate.

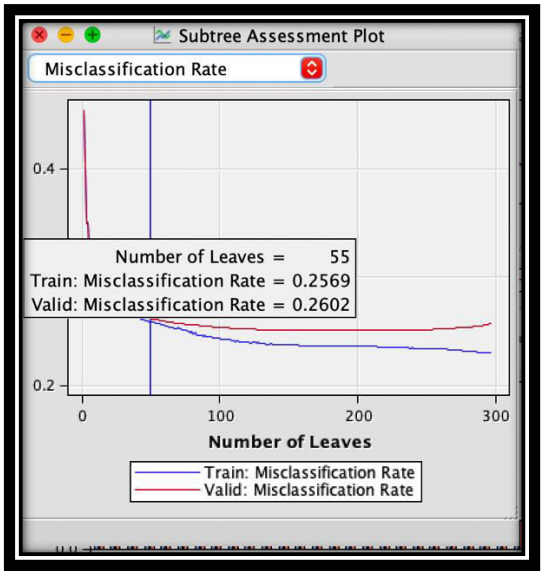


Figure 114: Misclassification Rate

Summary of the experiment:

Decision tree model	**Leaves**	**Validation Misclassification**
Default decision tree(5)	28	0.2678
Decision tree with the	52	0.2515

maximum depth. (10)		
Decision tree with depth 6	51	0.2668
Decision tree with branches 3	55	0.2569

Table 13: Summary of the iterations

So from the experiments done we I have noticed that the tree having maximum depth of 10 is having the less misclassification rate but I will not taking that as the best model because it has the maximum depth and it may cause some overfitting which might lead to problem in the model.

So the preferred model that I will be using is decision tree with 3 way branch. Compared to all others the model has less misclassification rate.

So my conclusion, the best model would be decision tree with 3 way branches and the maximum depth is 7.

From the above experiments I have made a conclusion which will be shown below,

Target	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
FUNCTIONAL	FUNCTIONAL	70.8542	92.3610	14134	50.3940
FUNCTIONAL NEEDS REPAIR	FUNCTIONAL	7.2488	74.3445	1446	5.1556
NON FUNCTIONAL	FUNCTIONAL	21.8969	40.4482	4368	15.5739
FUNCTIONAL	FUNCTIONAL NEEDS REPAIR	23.0769	0.5293	81	0.2888
FUNCTIONAL NEEDS REPAIR	FUNCTIONAL NEEDS REPAIR	58.6895	10.5913	206	0.7345
NON FUNCTIONAL	FUNCTIONAL NEEDS REPAIR	18.2336	0.5926	64	0.2282
FUNCTIONAL	NON FUNCTIONAL	14.0423	7.1097	1088	3.8792
FUNCTIONAL NEEDS REPAIR	NON FUNCTIONAL	3.7816	15.0643	293	1.0447
NON FUNCTIONAL	NON FUNCTIONAL	82.1760	58.9592	6367	22.7012

Figure 115: Classification Table

Here the pumps which are fully functional are 70.8542% and there is a total of 14134 which is 50% of the total pumps, the outcome percentage of it is 92%. The pumps which are functional but needs repair is total 5.1556% which makes the count of 1446. But the pumps which are non-functinal makes upto 22.7012% which the count makes 6367.

In order to intreptet properly I will be doing a 3X3 conclusion matrix which is below,

Result in validation				
Result	**Result/Result in validation**	**Functional**	**Functional Needs Repair**	**Non-Functional**
	Functional	14134	1446	4368

	Functional Needs Repair	81	206	64
	Non-Functional	1088	293	6367

Table 14: Result Matrix

	Functional	**Functional Needs Repair**	**Non-Functional**
True Positve	14134	206	6367
False Positive	1446+4368=5814	81+64=145	1088+293=1318
True Negitaive	206+64+293+6367=6930	14134+4368+1309+6367=26178	14134+1446+81+206=15867
False Negitave	81+1088=1169	1446+293=1759	4368+64=4432

Table 15: Final Outcome

After this the accurte the result will be

- Functional= 14134/(14134+5814)= 0.708 which is 70%.
- Functional but need repair = 206/(206+145) = 0.586 which is 50%.
- Non-fuctional = 6367/(6367+1318) = 0.828 which is 80%.

4.4 Stepwise Regression (Tarun Aitha TP058015)

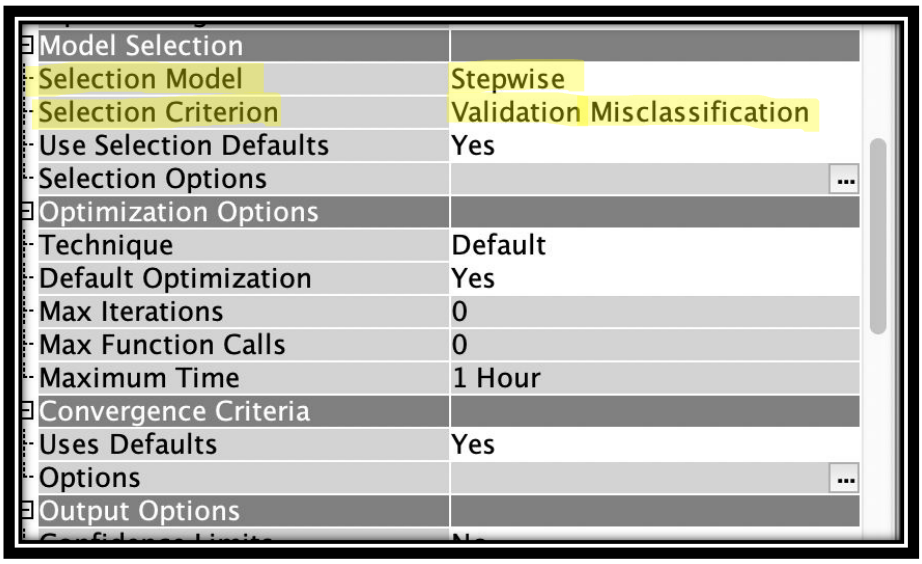
Stepwise Regression (Before Pruning)

Regression is a statistical technique used in finance, investing, and other fields to assess the strength and nature of a connection between one dependent variable (typically represented by Y) and a set of other variables (known as independent variables) (Beers, 2020).

Figure 116: StepWise Regression

Stepwise regression:

Stepwise regression is the iterative creation of a regression model in which the independent variables to be utilised in the final model are chosen step by step. It entails incrementally adding or eliminating potential explanatory factors, with each iteration requiring statistical significance assessment (HAYES, 2019).



Model Selection	
Selection Model	Stepwise
Selection Criterion	Validation Misclassification
Use Selection Defaults	Yes
Selection Options	...
Optimization Options	
Technique	Default
Default Optimization	Yes
Max Iterations	0
Max Function Calls	0
Maximum Time	1 Hour
Convergence Criteria	
Uses Defaults	Yes
Options	...
Output Options	

Figure 117: Model Modification

Here the default values have been which is the selection model is stepwise and the selection criterion is validation misclassification.

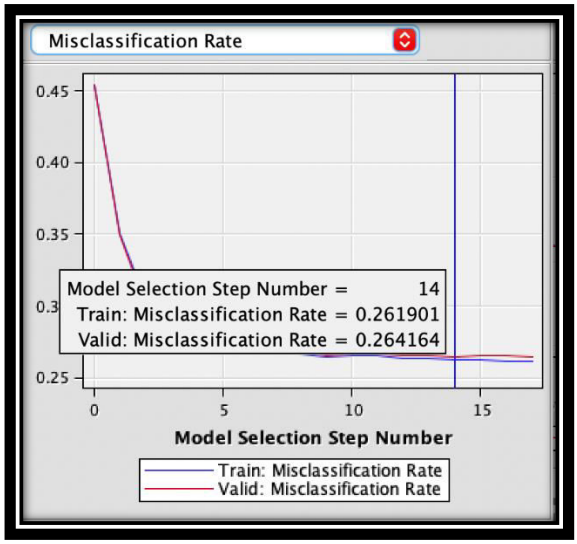
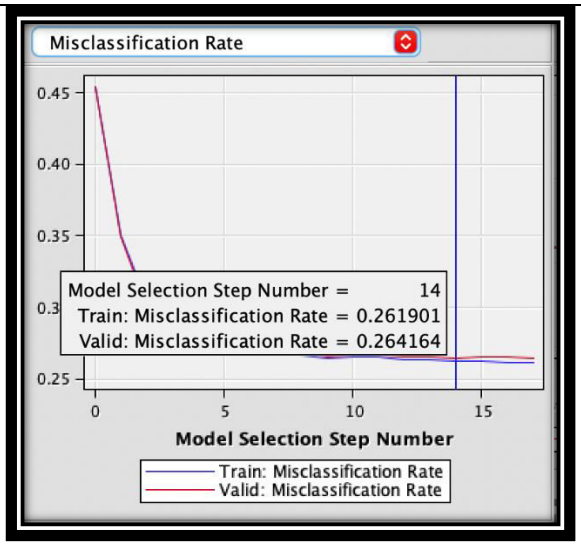


Figure 118: MIsclassification Rate

Here the model step number is 14 and the misclassification rate is 0.261901 which is 74% accurate.

In step wise regression if the techniques hanged for a good prediction rate we cannot see any significant changes. So, I will be changing the train and validation rate which is showed below.

Data Patrion	**Figure**	**Misclassification Rate**
Train=50 Validation=50	 *Figure 119: Iteration Plot*	Misclassification rate: 0.261901 Step number= 14

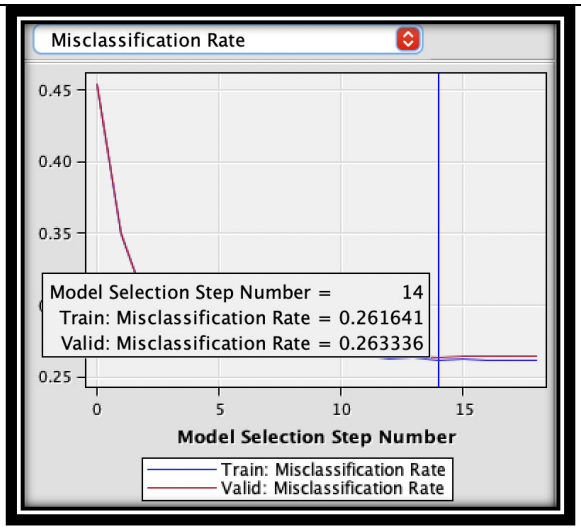
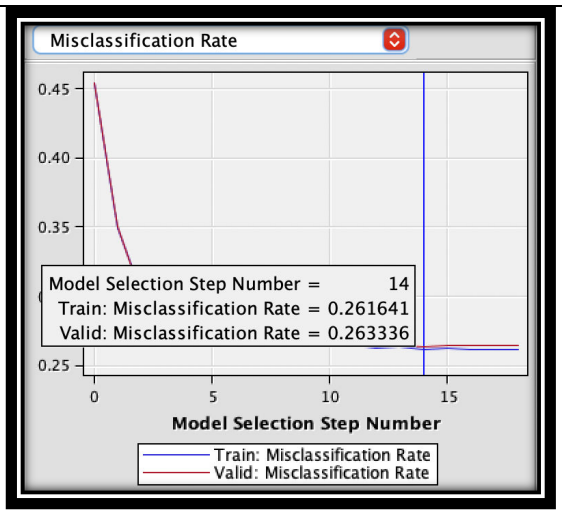
Train=60 Validation=40	 *Figure 120: Iteration Plot*	Misclassification Rate: 0.261641 Step number =14
Train = 70 Validation = 30	 *Figure 121: Iteration Plot*	Misclassification Rate: 0.261641 Step Number: 14

Table 16: Variation in data partiation

As we can see the misclassification rate remains the same so, now I will be increasing the validation rate and reducing the training rate.

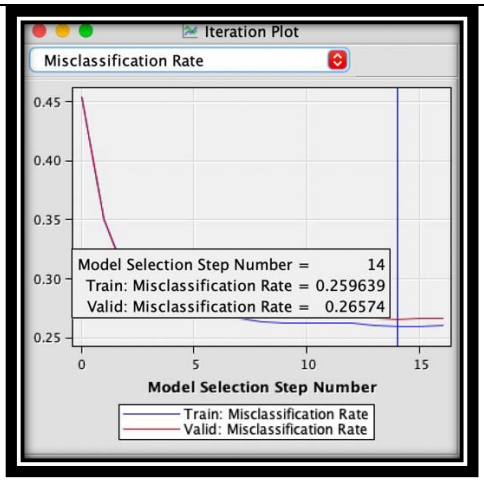
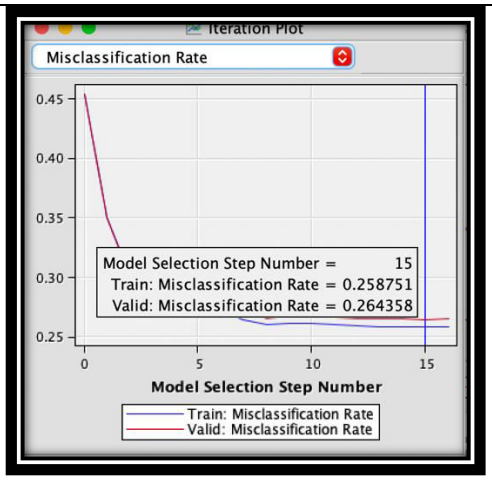
Data Patrion	Figure	Misclassification Rate
Train: 40 Validation : 60	 *Figure 122: Iteration Plot*	Misclassification Rate: 0.259639 Step number: 14
Train=30 Validation: 70	 *Figure 123: Iteration Plot*	Misclassification Rate: 0.258751 Step number: 15

Table 17: Using variations is data partion

So, from the above experiments if the validation data is increased we can see some improvements in the misclassification rate.

I have also done some data pre-processing after this and I have not seen any significant improvement, the misclassification rate has just been increasing. So, there will no significant improvement after this so I will be using the default settings for the modelling.

Now to see the outcome and the result of the final output I will be using score method to view the results,

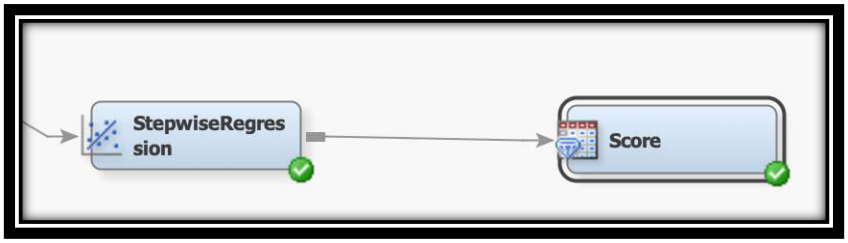


Figure 124: Score

The result after view the socre is,

Data Role=VALIDATE Output Type=CLASSIFICATION

Variable	Numeric Value	Formatted Value	Frequency Count	Percent
I_REP_status_group	.	FUNCTIONAL	26402	67.2405
I_REP_status_group	.	FUNCTIONAL NEEDS REPAIR	383	0.9754
I_REP_status_group	.	NON FUNCTIONAL	12480	31.7840

Figure 125: Validation

From the above figure the results shown are the pumps which fully functional are 67% which the count males upto 26402, the pumps which are functional but needs repair makes upto 0.9% which makes the count to 383, and the non-functional are 31% which the count makes to 12480.

4.5 Forward Logistic Regression – Law Pei Lin (TP051691)

Logistic Regression is a type of regression analysis that is used to describe data and to examine the relationship between one categorical dependent variable and one or more numeric or categorical independent variables (Hoffman, 2019).

$$logit\,(\theta_i) = ln\left(\frac{\theta_i}{1-\theta_i}\right) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_U x_{iU}$$

Figure 126: Logit Link function (MacKenzie et al., 2018)

Figure 126 shows the logit link function which is used by logistic regression.

Model Construction

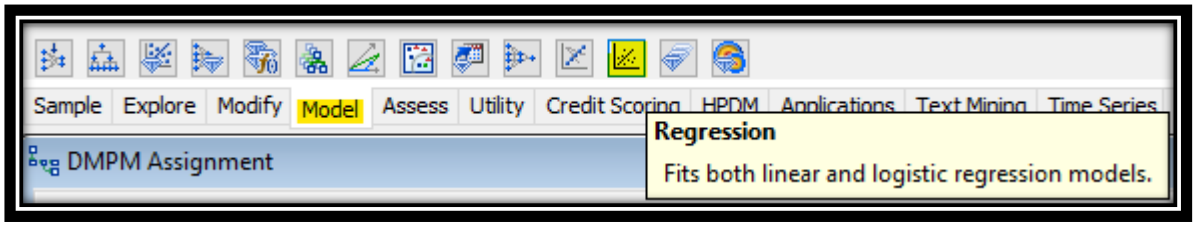
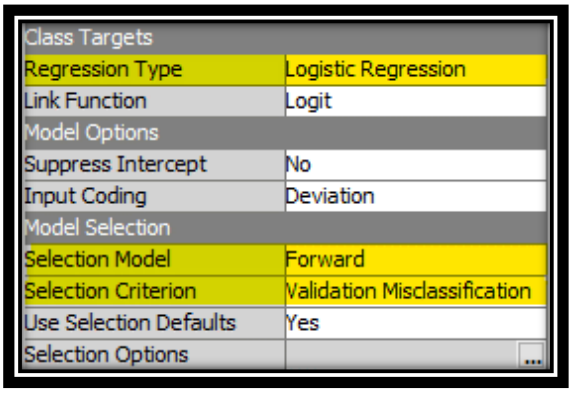


Figure 127: Regression Node

Forward Logistic Regression is created by using Regression Node which is in the Model tab as shown in figure 127.



Class Targets	
Regression Type	Logistic Regression
Link Function	Logit
Model Options	
Suppress Intercept	No
Input Coding	Deviation
Model Selection	
Selection Model	Forward
Selection Criterion	Validation Misclassification
Use Selection Defaults	Yes
Selection Options	...

Figure 128: Properties of Forward Logistic Regression Node

Figure 128 shows the properties of the Forward Logistic Regression node. The regression type is logistic regression because the target is a class variable. Next, the selection model is Forward which the iteration of the regression will begin with an empty model and adds in variables one by one by forward stop. In each forward step, it will add the variable that brings the best improvement to the model (Stephanie, 2017). The selection criterion is set to validation misclassification because the prediction of the dataset is decision.

Model Validation

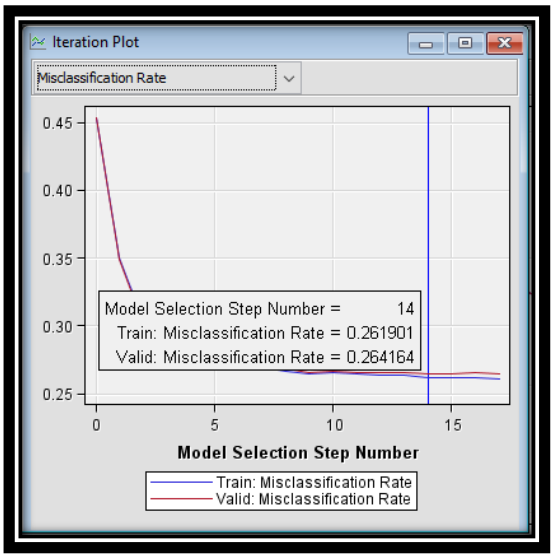


Figure 129: Iteration Plot of Forward Logistic Regression (Before Filtering)

Figure 129 shows that the optimal model is having 0.264164 misclassification rate with iteration 14. Besides, the figure shows that there is no overfitting or underfitting issue.

Summary of Forward Selection

Step	Effect Entered	DF	Number In	Score Chi-Square	Pr > ChiSq	Validation Misclassification Rate
1	REP_quantity	8	1	5276.3500	<.0001	0.3491
2	REP_waterpoint_type	12	2	3036.3869	<.0001	0.3011
3	REP_region	40	3	1716.1198	<.0001	0.3012
4	REP_payment_type	12	4	875.1268	<.0001	0.2855
5	REP_extraction_type	34	5	775.2485	<.0001	0.2774
6	REP_management	22	6	613.7457	<.0001	0.2753
7	PCTL_construction_year	4	7	433.5804	<.0001	0.2687
8	REP_basin	16	8	306.8193	<.0001	0.2679
9	REP_source	18	9	243.0880	<.0001	0.2654
10	REP_IMP_public_meeting	2	10	100.4340	<.0001	0.2664
11	REP_IMP_scheme_management	20	11	131.5485	<.0001	0.2660
12	REP_water_quality	14	12	100.7670	<.0001	0.2653
13	gps_height	2	13	48.3430	<.0001	0.2654
14	PWR_population	2	14	37.6233	<.0001	0.2642
15	longitude	2	15	38.5229	<.0001	0.2650
16	latitude	2	16	25.3996	<.0001	0.2653
17	LOG_district_code	2	17	6.8631	0.0323	0.2646

Figure 130: Summary of Forward Logistic Regression

Figure 130 shows the summary of Forward Regression model which the best iteration is on step 14 which the best model is stops at PWR_population. The reason that it is the best model is the misclassification rate of the model is the lowest and after step 14, the misclassification rates are increasing.

Type 3 Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
PCTL_construction_year	4	326.6860	<.0001
PWR_population	2	37.3425	<.0001
REP_IMP_public_meeting	2	96.9089	<.0001
REP_IMP_scheme_management	20	126.4973	<.0001
REP_basin	16	246.8508	<.0001
REP_extraction_type	34	447.9616	<.0001
REP_management	22	267.4033	<.0001
REP_payment_type	12	791.2865	<.0001
REP_quantity	8	1708.4752	<.0001
REP_region	40	823.9316	<.0001
REP_source	18	237.1151	<.0001
REP_water_quality	14	98.2639	<.0001
REP_waterpoint_type	12	555.8440	<.0001
gps_height	2	42.1071	<.0001

Figure 131: Analysis of Effect

Figure 131 shows the significant variables where the P-value of the variables is less than 0.0001.

REP_extraction_type	nira/tanira vs windmill	non functional	0.215
REP_extraction_type	nira/tanira vs windmill	functional needs repair	0.602
REP_extraction_type	other vs windmill	non functional	1.295
REP_extraction_type	other vs windmill	functional needs repair	1.096
REP_extraction_type	other - mkulima/s vs windmill	non functional	58.405
REP_extraction_type	other - mkulima/s vs windmill	functional needs repair	2.626
REP_extraction_type	other - play pump vs windmill	non functional	1.332
REP_extraction_type	other - play pump vs windmill	functional needs repair	1.017
REP_extraction_type	other - rope pump vs windmill	non functional	0.601
REP_extraction_type	other - rope pump vs windmill	functional needs repair	0.912
REP_extraction_type	other - swn 81 vs windmill	non functional	0.663
REP_extraction_type	other - swn 81 vs windmill	functional needs repair	0.427
REP_extraction_type	submersible vs windmill	non functional	0.537
REP_extraction_type	submersible vs windmill	functional needs repair	1.469
REP_extraction_type	swn 80 vs windmill	non functional	0.421
REP_extraction_type	swn 80 vs windmill	functional needs repair	0.672
REP_extraction_type	walimi vs windmill	non functional	0.570
REP_extraction_type	walimi vs windmill	functional needs repair	0.008

Figure 132: Odds Ratio Estimates of Forward Logistic Regression

Figure 132 shows odds ratio estimates of the model. It shows that the highlighted REP_extraction_type is having the highest point estimate. It means that the "other – skulima/s" tend to be higher accuracy on getting nonfunctional pump compared to "windmill".

REP_quantity	dry vs unknown	non functional	53.091
REP_quantity	dry vs unknown	functional needs repair	6.402
REP_quantity	enough vs unknown	non functional	0.374
REP_quantity	enough vs unknown	functional needs repair	2.103
REP_quantity	insufficient vs unknown	non functional	0.625
REP_quantity	insufficient vs unknown	functional needs repair	3.321
REP_quantity	seasonal vs unknown	non functional	0.536
REP_quantity	seasonal vs unknown	functional needs repair	2.864

Figure 133: Odds Ratio Estimates of Forward Logistic Regression (2)

The second highest point estimate is shown in figure 133. From the figure, it shows that the dry pump tends to be non-functional compared to the pump with an unknown quantity of water.

Event Classification Table

Data Role=TRAIN Target=REP_status_group Target Label=Replacement: status_group

False Negative	True Negative	False Positive	True Positive
3961	15374	1874	6836

Data Role=VALIDATE Target=REP_status_group Target Label=Replacement: status_group

False Negative	True Negative	False Positive	True Positive
3901	15297	1951	6898

Figure 134: Event Classification Table of Forward Logistic Regression

Next, figure 134 shows the event classification table. The number observations getting true positive are 6898 which means that there are 6898 observations are predicted pump are functional. Besides, 15297 observations are getting true negative which also predicted correctly with its negative result. 3901 observations are having false negative and 1951 observations are having false positive which means that there are predicted wrongly.

Experiment by using Default Filter

To improve the accuracy of the regression model, the filter in data pre-processing is changed and used the default filtering method to remove observations from data according to the rare value and standard deviations from the mean.

Class Variables	
Class Variables	
Default Filtering Method	Rare Values (Percentage)
Keep Missing Values	Yes
Normalized Values	Yes
Minimum Frequency Cutoff	1
Minimum Cutoff for Percentage	0.01
Maximum Number of Levels Cutoff	25
Interval Variables	
Interval Variables	
Default Filtering Method	Standard Deviations from the Mean
Keep Missing Values	Yes
Tuning Parameters	

Figure 135: Properties of Filter Node

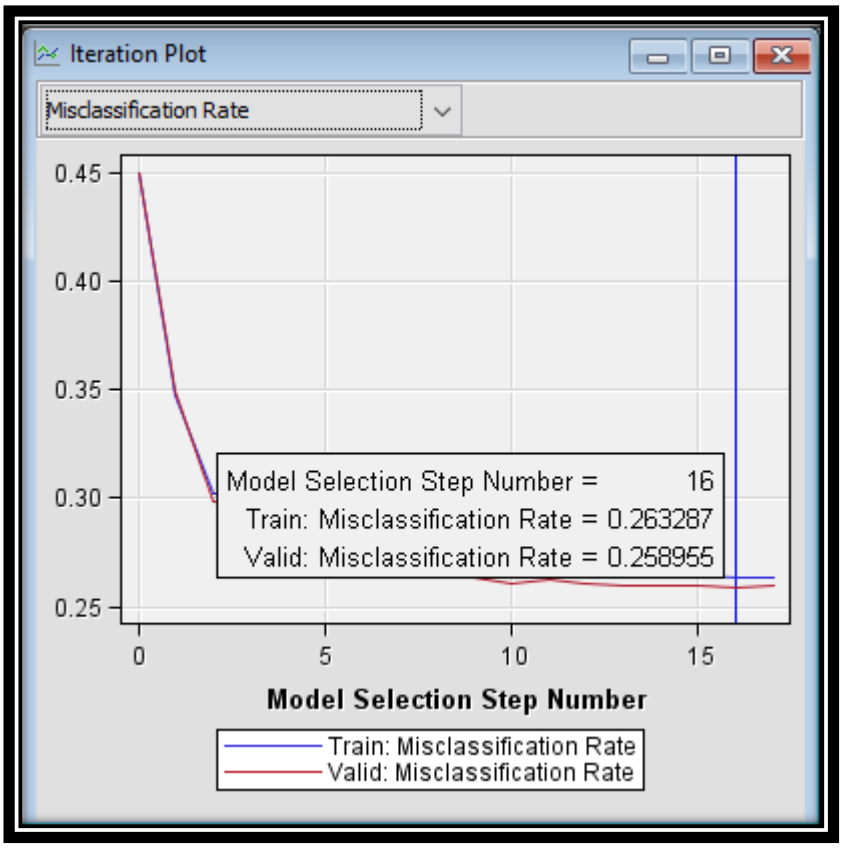


Figure 136: Iteration plot of Forward Logistic Regression (After Filtering)

It brings a better accuracy by having a 0.258955 misclassification rate for the validation model. The number of iterations of the best regression model is 16 steps. The accuracy of the regression model is 0.74.

Summary of Forward Selection

Step	Effect Entered	DF	Number In	Score Chi-Square	Pr > ChiSq	Validation Misclassification Rate
1	REP_quantity	8	1	4919.0321	<.0001	0.3488
2	REP_waterpoint_type	8	2	2826.6480	<.0001	0.2987
3	REP_region	40	3	1508.4372	<.0001	0.2962
4	REP_payment_type	12	4	746.0706	<.0001	0.2790
5	REP_source	14	5	700.3598	<.0001	0.2802
6	PCTL_construction_year	4	6	495.6362	<.0001	0.2744
7	REP_management	16	7	467.8217	<.0001	0.2702
8	REP_extraction_type	16	8	340.7170	<.0001	0.2658
9	REP_basin	16	9	238.7300	<.0001	0.2635
10	REP_IMP_scheme_management	16	10	126.2102	<.0001	0.2609
11	REP_IMP_public_meeting	2	11	78.5537	<.0001	0.2623
12	REP_water_quality	6	12	69.2680	<.0001	0.2605
13	gps_height	2	13	36.5348	<.0001	0.2598
14	longitude	2	14	30.0336	<.0001	0.2593
15	latitude	2	15	23.2575	<.0001	0.2596
16	EXP_population	2	16	11.8150	0.0027	0.2590
17	LOG_amount_tsh	2	17	7.3692	0.0251	0.2594

The selected model, based on the misclassification rate for the validation data, is the model trained in Step 16.

Figure 137: Summary of Forward Logistic Regression (After Filtering)

Figure 137 shows the summary of forwarding regression which the iteration starts from REP_quantity until EXP_population.

Type 3 Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
EXP_population	2	11.7604	0.0028
PCTL_construction_year	4	368.9627	<.0001
REP_IMP_public_meeting	2	77.8322	<.0001
REP_IMP_scheme_management	16	130.3681	<.0001
REP_basin	16	172.2578	<.0001
REP_extraction_type	16	350.9678	<.0001
REP_management	16	211.4271	<.0001
REP_payment_type	12	697.0629	<.0001
REP_quantity	8	1659.0809	<.0001
REP_region	40	690.0689	<.0001
REP_source	14	269.4026	<.0001
REP_water_quality	6	64.6428	<.0001
REP_waterpoint_type	8	482.1552	<.0001
gps_height	2	26.1524	<.0001
latitude	2	23.2701	<.0001
longitude	2	30.8872	<.0001

Figure 138: Analysis of Effects For Forward Logistic Regression (After Filtering)

Figure 138 shows the significant variables with p-value less than 0.0001 which are PCTL_construction_year, REP_IMP_public_meeting, REP_IMP_scheme_management, REP_basin, REP_extraction_type, REP_payment_type, REP_quantity, REP_region, REP_source, REP_water_quality, REP_waterpoint_type, gps_height, latitude, and longitude.

REP_quantity	dry vs unknown	non functional	32.819
REP_quantity	dry vs unknown	functional needs repair	4.524
REP_quantity	enough vs unknown	non functional	0.240
REP_quantity	enough vs unknown	functional needs repair	1.986
REP_quantity	insufficient vs unknown	non functional	0.386
REP_quantity	insufficient vs unknown	functional needs repair	3.177
REP_quantity	seasonal vs unknown	non functional	0.328
REP_quantity	seasonal vs unknown	functional needs repair	2.377

Figure 139: Odds Ratio Estimation Forward Logistic Regression (After Filtering)

Figure 139 shows the odds ratio estimation of the forward regression model. It shows that the highest point estimation is REP_quantity with dry vs unknown for nonfunctional pump (32.819) which means that dry pump is more likely to be non-functional pump compared to pump with unknown water quantity.

REP_region	Mbeya vs Tanga	non functional	2.766
REP_region	Mbeya vs Tanga	functional needs repair	9.739
REP_region	Morogoro vs Tanga	non functional	1.988
REP_region	Morogoro vs Tanga	functional needs repair	7.881
REP_region	Mtwara vs Tanga	non functional	3.100
REP_region	Mtwara vs Tanga	functional needs repair	24.647
REP_region	Mwanza vs Tanga	non functional	1.019
REP_region	Mwanza vs Tanga	functional needs repair	1.256
REP_region	Pwani vs Tanga	non functional	2.672
REP_region	Pwani vs Tanga	functional needs repair	1.012
REP_region	Rukwa vs Tanga	non functional	7.259

Figure 140: Odds Ratio Estimation Forward Logistic Regression (After Filtering) (2)

Figure 140 shows that the second-highest point estimation is REP_region with Mtwara vs Tanga for functional needs repair. The point estimation is 25.647. It means that pumps in Mtwara are more likely to be functional but needs repair compared to the pumps in Tanga.

Data Role=VALIDATE Target Variable=REP_status_group Target Label=Replacement: status_group

Target	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
FUNCTIONAL	FUNCTIONAL	72.2797	90.4034	12774	49.7352
FUNCTIONAL NEEDS REPAIR	FUNCTIONAL	7.9556	76.4962	1406	5.4742
NON FUNCTIONAL	FUNCTIONAL	19.7646	35.9510	3493	13.5999
FUNCTIONAL	FUNCTIONAL NEEDS REPAIR	23.6181	0.3326	47	0.1830
FUNCTIONAL NEEDS REPAIR	FUNCTIONAL NEEDS REPAIR	47.2362	5.1143	94	0.3660
NON FUNCTIONAL	FUNCTIONAL NEEDS REPAIR	29.1457	0.5970	58	0.2258
FUNCTIONAL	NON FUNCTIONAL	16.7563	9.2640	1309	5.0966
FUNCTIONAL NEEDS REPAIR	NON FUNCTIONAL	4.3267	18.3896	338	1.3160
NON FUNCTIONAL	NON FUNCTIONAL	78.9171	63.4520	6165	24.0033

Figure 141: Classification Table of Validate Model

Figure 141 shows the classification table of the validated model. It shows that the prediction of getting correctly for functional pump is 12774 observations with 49.7352%. The functional need repair pump that getting the correct prediction is 94 observations with 0.3660 %. Besides, the non-functional pump that getting a correct prediction is 6165 observations with 24.0033%. It also shows that the number of observations of the wrong prediction. For example, 1406 observations are getting the wrong prediction with the target functional need repair, but the prediction is functional.

Data Role=VALIDATE Target=REP_status_group Target Label=Replacement: status_group

False Negative	True Negative	False Positive	True Positive
3551	14321	1647	6165

Figure 142: Confusion Matrix / Event Classification of Forward Logistic Regression (After Filtering)

Figure 142 shows the confusion matrix of the non-functional pump. The number of false positives is lower than false negative which means the prediction has a higher probability of underestimate rather than overestimate. The model may provide more underestimate predictions such as the pump is non-functional, but the predicted result is functional which will affect the further action toward the pump.

Actual Result				
Predicted Result		**Functional**	**Functional Needs Repair**	**Non-Functional**
	Functional	12774	1406	3493

	Functional Needs Repair	47	94	58
	Non-Functional	1309	338	6165

Table 18: 3x3 confusion matrix of Forward Logistic Regression

	Functional	**Functional Needs Repair**	**Non-Functional**
True Positive	12774	94	6165
False Positive	1406+3493 = 4899	47+58 = 105	1309+338 = 1647
True Negative	94+58+338+6165 = 6655	12774+3493+1309+6165 = 23741	12774+1406+47+94 = 14321
False Negative	47+1309 = 1356	1406+338 = 1744	3493+58 = 3551

Table 19: Multi-class Classification Events of Forward Logistic Regression.

Table 18 shows the 3x3 confusion matrix of the logistic regression model based on figure 141 and table 19 shows the multi-class classification. Table 19 shows that figure 142 is the event classification rate of the Non-Functional pump. 3x3 confusion matrix is created is because the number levels of the target variable in this dataset is 3, which are functional, functional needs repair, and non-functional.

The precision of the model on functional, functional need repair and non-function is calculated as shown in table 16.

Precision of Functional:	12774/ (12774 + 4899) = 0.72
Precision of Functional Needs Repair:	94/ (94+105) = 0.47
Precision of Non-Functional:	6165/ (6165+1647) = 0.79

Table 20: Precision of each target prediction in the logistic regression model.

From table 20, it shows that the precision of functional is 0.72 which means that all pumps that labeled as a functional pump, there are 72% are functional. Besides, the precision of the functional needs repair pump is 0.47 which means there are 47% of the pump are correctly labeled as functional needs repair among all labeled functional needs repair pumps. The precision of the non-functional pump is 0.79 which means that there are 79% correctly labeled as non-functional pumps among all labeled non-functional pumps. The precision of the pump that is functional but needs repair is low.

Several experiments on the data partition (using default filtering) are carried out.

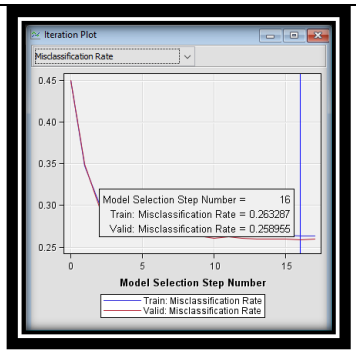
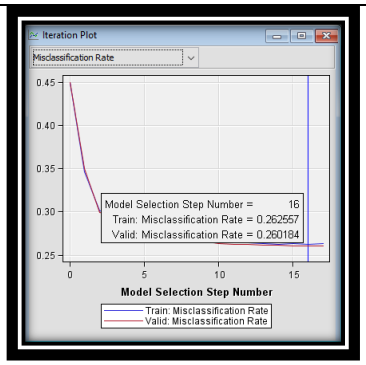
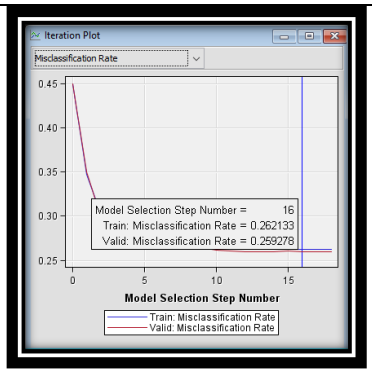
Data Set Allocation	Result	Support Image
Training: 50% Validation: 50%	Misclassification rate: 0.258955 Iteration: 16	 *Figure 143: Iteration Plot (1)*
Training: 60% Validation: 40%	Misclassification rate: 0. 260184 Iteration: 16	 *Figure 144: Iteration Plot (2*
Training: 70% Validation: 30%	Misclassification rate: 0.259278 Iteration: 16	 *Figure 145: Iteration Plot (3)*

Table 21: Experiment with different data partition

The experiments on the data partition with different data set allocations for Forward Logistic Regression found that the effect on the data partition is not much.

4.6 HP Forest (Random Forest) – Law Pei Lin (TP051691)

Random Forest is a tree-based algorithm that consists many of decision trees and establishes the outcome based on the predictions of the decision trees (Mbaabu, 2020).

Model Construction

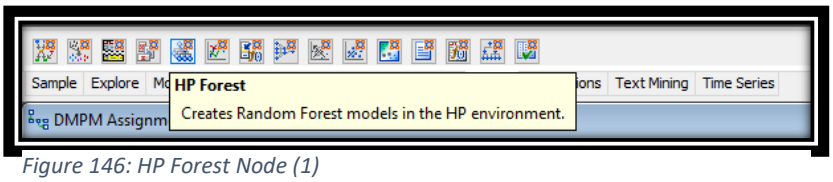


Figure 146: HP Forest Node (1)

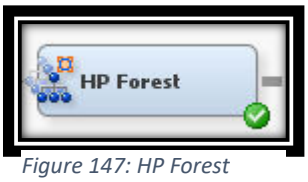


Figure 147: HP Forest Node (2)

Random Forest is created by using HP Forest Node which is in the HPDM tab as shown in the figures above.

Variables	...
Tree Options	
Maximum Number of Trees	100
Seed	12345
Type of Sample	Proportion
Proportion of Obs in Each S	0.6
Number of Obs in Each Sam	.
Splitting Rule Options	
Maximum Depth	50
Missing Values	Use In Search
Minimum Use In Search	1
Number of Variables to Con	.
Significance Level	0.05
Max Categories in Split Sea	30
Minimum Category Size	5
Exhaustive	5000
Node Options	
Method for Leaf Size	Default
Smallest Percentage of Obs	1.0E-5
Smallest Number of Obs in M	1
Split Size	.
Use as Modeling Node	Yes

Figure 148: Default Properties of HP Forest Node

The random forest model is built by using the default options which the maximum number of trees is 100 and the maximum depth is 50.

Several experiments on the data partition are carried out.

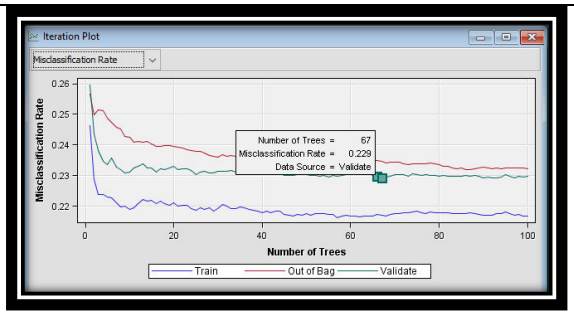
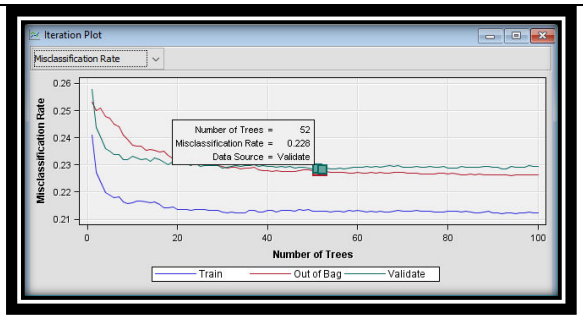
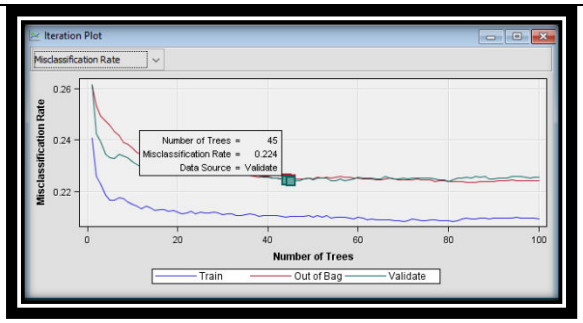
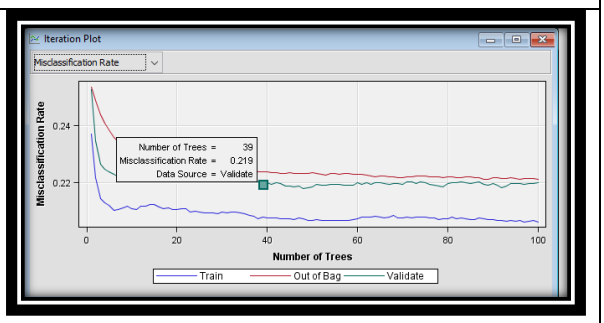
Data Set Allocation	Result	Support Image
Training: 50% Validation: 50%	Misclassification rate: 0.229 Number of Trees: 67	 *Figure 149: Iteration Plot (1)*
Training: 60% Validation: 40%	Misclassification rate: 0.228 Number of Trees: 52	 *Figure 150: Iteration Plot (2)*
Training: 70% Validation: 30%	Misclassification rate: 0.224 Number of Trees: 45	 *Figure 151: Iteration Plot (3)*
Training: 80% Validation: 20%	Misclassification rate: 0.219 Number of Trees: 39	 *Figure 152: Iteration Plot (4)*

Table 22: Experiment with different data partitions.

The experiments on the data partition with different data set allocation for Random Forest show that the data partition 80% for training data and 20% for validation data is the optimal HP Forest model.

Model Validation and Evaluation

Input Variables

Name	Length	Role	Type	RawType	Format Name
LOG_amount_tsh	8	Input	Interval	Num	
LOG_district_code	8	Input	Interval	Num	
PWR_population	8	Input	Interval	Num	
gps_height	8	Input	Interval	Num	
latitude	8	Input	Interval	Num	
longitude	8	Input	Interval	Num	
PCTL_construction_year	36	Input	Classification	Character	
REP_IMP_public_meeting	5	Input	Classification	Character	
REP_IMP_scheme_management	16	Input	Classification	Character	
REP_basin	23	Input	Classification	Character	
REP_extraction_type	17	Input	Classification	Character	
REP_management	16	Input	Classification	Character	
REP_payment_type	10	Input	Classification	Character	
REP_quantity	12	Input	Classification	Character	
REP_region	13	Input	Classification	Character	
REP_source	20	Input	Classification	Character	
REP_source_class	11	Input	Classification	Character	
REP_source_type	20	Input	Classification	Character	
REP_water_quality	15	Input	Classification	Character	
REP_waterpoint_type	27	Input	Classification	Character	

Output Variables

Name	Length	Type	Label
P_REP_status_grouNON_FUNCTIONAL	8	Num	Predicted: REP_status_group=NON FUNCTIONAL
P_REP_status_grouFUNCTIONAL_NEED	8	Num	Predicted: REP_status_group=FUNCTIONAL NEEDS REPAIR
P_REP_status_grouFUNCTIONAL	8	Num	Predicted: REP_status_group=FUNCTIONAL
I_REP_status_group	32	Character	Into: REP_status_group
WARN	4	Character	Warnings

Figure 153: Input Variables and Output Variables used by HP Forest

Figure 153 shows the input variable used by the Random Forest model and the output variable which are Non-functional, Functional Need Repair, and Functional.

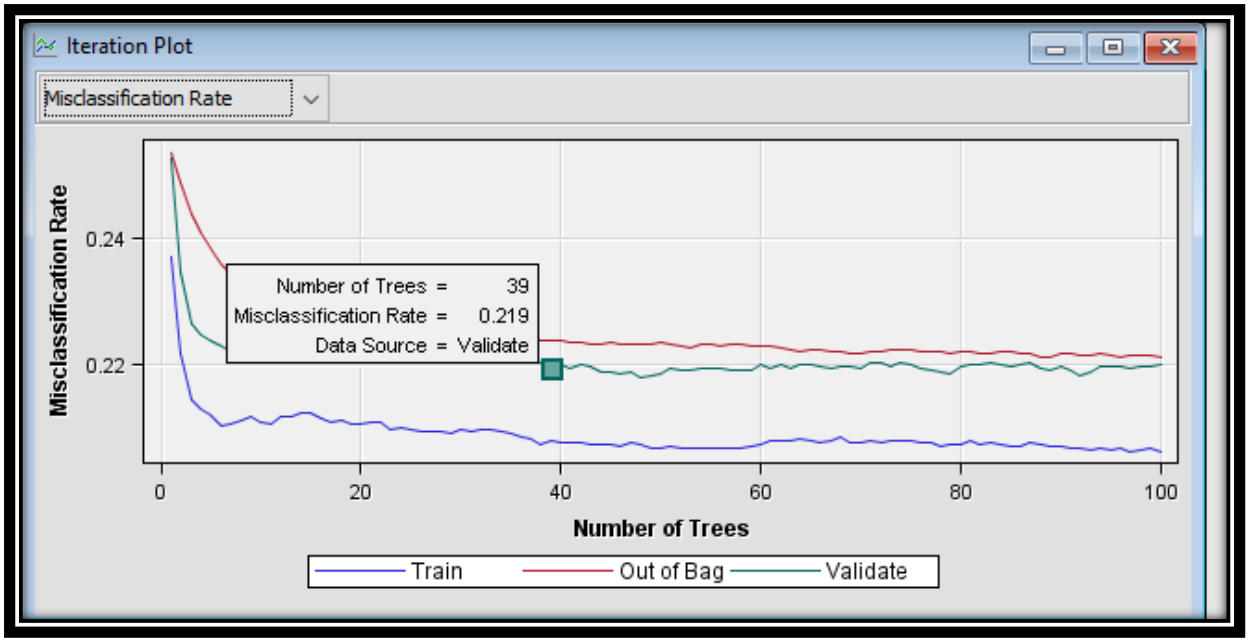


Figure 154: Iteration Plot (HP Forest)

Figure 154 shows the iteration plot of the HP forest, and it shows that there are no node rules which is not interpretable. Besides, it shows that there are 100 trees are created in the models as shown in figure 137. In decision trees, pruning is a method to avoid overfitting, however, random forest no requires to do pruning because there are no node rules (Saraswat, 2017).

Iteration History

Number of Trees	Number of Leaves	Average Square Error (Train)	Average Square Error (Out of Bag)	Average Square Error (Validate)	Misclassification Rate (Train)	Misclassification Rate (Out of Bag)	Misclassification Rate (Validate) ▲	Log Loss (Train)	Log Loss (Out of Bag)	Log Loss (Validate)
57	24358	0.0986	0.105	0.104	0.207	0.223	0.219	0.516	0.550	0.546
59	25206	0.0987	0.105	0.105	0.207	0.223	0.219	0.516	0.550	0.547
77	32868	0.0987	0.105	0.105	0.208	0.222	0.219	0.516	0.549	0.547
91	39032	0.0985	0.105	0.104	0.207	0.222	0.219	0.516	0.548	0.546
41	17322	0.0988	0.106	0.105	0.208	0.224	0.219	0.516	0.551	0.547
51	21693	0.0987	0.105	0.105	0.207	0.223	0.219	0.516	0.550	0.547
97	41530	0.0985	0.105	0.104	0.206	0.221	0.219	0.516	0.548	0.546
39	16523	0.0989	0.106	0.105	0.208	0.224	0.219	0.516	0.551	0.547
54	23041	0.0986	0.105	0.104	0.207	0.223	0.219	0.516	0.550	0.546
55	23485	0.0986	0.105	0.104	0.207	0.223	0.219	0.516	0.550	0.546
56	23916	0.0986	0.105	0.104	0.207	0.223	0.219	0.515	0.549	0.546
70	29896	0.0986	0.105	0.104	0.208	0.222	0.219	0.516	0.549	0.547
76	32492	0.0986	0.105	0.105	0.207	0.222	0.219	0.516	0.549	0.547
61	26035	0.0987	0.105	0.105	0.208	0.223	0.219	0.516	0.549	0.547
63	26875	0.0987	0.105	0.104	0.208	0.222	0.219	0.516	0.549	0.546

Figure 155: Iteration History (Optimal Model)

Iteration History

Number of Trees ▲	Number of Leaves	Average Square Error (Train)	Average Square Error (Out of Bag)	Average Square Error (Validate)	Misclassification Rate (Train)	Misclassification Rate (Out of Bag)	Misclassification Rate (Validate)	Log Loss (Train)	Log Loss (Out of Bag)	Log Loss (Validate)
86	36699	0.0987	0.105	0.104	0.207	0.222	0.220	0.516	0.549	0.547
87	37191	0.0986	0.105	0.104	0.208	0.222	0.220	0.516	0.549	0.547
88	37668	0.0986	0.105	0.104	0.207	0.221	0.219	0.516	0.549	0.547
89	38123	0.0986	0.105	0.104	0.207	0.221	0.219	0.516	0.549	0.546
90	38619	0.0985	0.105	0.104	0.207	0.222	0.220	0.516	0.548	0.546
91	39032	0.0985	0.105	0.104	0.207	0.222	0.219	0.516	0.548	0.546
92	39444	0.0985	0.105	0.104	0.206	0.221	0.218	0.516	0.548	0.546
93	39877	0.0985	0.105	0.104	0.206	0.222	0.219	0.516	0.548	0.546
94	40233	0.0985	0.105	0.104	0.206	0.222	0.220	0.516	0.548	0.546
95	40672	0.0985	0.105	0.104	0.206	0.222	0.220	0.516	0.548	0.546
96	41103	0.0985	0.105	0.104	0.206	0.221	0.220	0.516	0.548	0.546
97	41530	0.0985	0.105	0.104	0.206	0.221	0.219	0.516	0.548	0.546
98	41908	0.0985	0.105	0.104	0.206	0.221	0.220	0.516	0.548	0.546
99	42290	0.0985	0.105	0.104	0.206	0.221	0.220	0.516	0.548	0.546
100	42677	0.0986	0.105	0.104	0.206	0.221	0.220	0.516	0.548	0.546

Figure 156: Iteration History (Total Number of Trees Created)

From the iteration history, it shows that the lowest misclassification rates happen at 39 trees with the misclassification rates 0.219 as shown in figure 136. The accuracy of random forest is 77.6%.

Loss Reduction Variable Importance

Variable	Number of Rules	Gini	OOB Gini	Valid Gini	Margin	OOB Margin	Valid Margin
REP_quantity	3006	0.061404	0.059357	0.057581	0.117468	0.115615	0.112243
REP_waterpoint_type	2124	0.024840	0.023673	0.024710	0.046027	0.045057	0.046334
REP_extraction_type	2406	0.023552	0.021711	0.023334	0.044125	0.042931	0.044764
REP_region	3368	0.020737	0.017186	0.016724	0.037521	0.034479	0.034440
REP_payment_type	2174	0.013739	0.012140	0.012947	0.025070	0.023667	0.024647
REP_source	1656	0.008028	0.007018	0.007411	0.014057	0.013296	0.013679
REP_basin	2316	0.007590	0.005904	0.005869	0.013305	0.011711	0.012006
PCTL_construction_year	2128	0.006576	0.005282	0.005174	0.012052	0.010752	0.010610
LOG_amount_tsh	1390	0.006213	0.005109	0.005676	0.011487	0.010319	0.011085
REP_source_type	1820	0.005793	0.004731	0.004855	0.010003	0.009124	0.009349
longitude	3548	0.008608	0.004656	0.004519	0.014093	0.010070	0.010025
latitude	3474	0.008011	0.004016	0.003816	0.013824	0.009776	0.009533
REP_management	1635	0.005248	0.003884	0.003425	0.009794	0.008698	0.008403
REP_IMP_scheme_management	1860	0.005304	0.003853	0.003257	0.009480	0.008341	0.007806
LOG_district_code	3320	0.005892	0.003779	0.004013	0.009515	0.007325	0.007637
PWR_population	1839	0.003851	0.001993	0.002040	0.006619	0.004767	0.004801
gps_height	2310	0.004405	0.001805	0.001861	0.007479	0.004793	0.004937
REP_water_quality	806	0.001855	0.001323	0.001356	0.003423	0.002962	0.002905
REP_source_class	765	0.001447	0.001026	0.001069	0.002239	0.001852	0.001848
REP_IMP_public_meeting	632	0.001096	0.000590	0.000598	0.001757	0.001247	0.001185

Figure 157: Loss Reduction Variable Importance

Figure 157 shows the variable importance of the model. It shows the rank of importance of how each variable contributes to the predictability of the model (Paul, 2017). The number of rules is referring to the number of splitting rules of the variable (Paul, 2017). The higher the valid margin the better prediction. Therefore REP_quantity is the most significant variable in the random forest model as it provides a better prediction to the model. REP_quantity split 3006 rules in the forest model. The rank of the important variable is according to valid margin which the higher the valid margin the more important variable as shown in Figure 157.

Data Role=VALIDATE Target Variable=REP_status_group Target Label=Replacement: status_group

Target	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
FUNCTIONAL	FUNCTIONAL	75.1952	92.8116	5681	50.6328
FUNCTIONAL NEEDS REPAIR	FUNCTIONAL	7.1211	69.0629	538	4.7950
NON FUNCTIONAL	FUNCTIONAL	17.6837	30.9259	1336	11.9073
FUNCTIONAL	FUNCTIONAL NEEDS REPAIR	16.3934	0.4901	30	0.2674
FUNCTIONAL NEEDS REPAIR	FUNCTIONAL NEEDS REPAIR	66.1202	15.5327	121	1.0784
NON FUNCTIONAL	FUNCTIONAL NEEDS REPAIR	17.4863	0.7407	32	0.2852
FUNCTIONAL	NON FUNCTIONAL	11.7748	6.6983	410	3.6542
FUNCTIONAL NEEDS REPAIR	NON FUNCTIONAL	3.4463	15.4044	120	1.0695
NON FUNCTIONAL	NON FUNCTIONAL	84.7789	68.3333	2952	26.3102

Figure 158: Classification Table of HPForest

```
Data Role=VALIDATE Target=REP_status_group Target Label=Replacement: status_group

 False        True        False        True
Negative    Negative    Positive    Positive

 1368        6370         530         2952
```

Figure 159: Event Classification / Confusion Matrix of HP Forest

Figure 158 shows the classification table and figure 159 shows the event classification table of the non-functional pump.

Actual Result				
Predicted Result		**Functional**	**Functional Needs Repair**	**Non-Functional**
	Functional	5681	538	1336
	Functional Needs Repair	30	121	32
	Non-Functional	410	120	2952

Table 23: 3x3 Confusion Matrix of Validate Data (HP Forest)

	Functional	**Functional Needs Repair**	**Non-Functional**
True Positive	5681	121	2952
False Positive	538+1336= 1874	30+32 = 62	410+120 = 530
True Negative	121+32+120+2952 = 3225	5681+410+1336+2952 = 10379	5681+538+30+121 = 6370
False Negative	30+410 = 440	538+120 = 658	1336+32 = 1368

Table 24: Multi-class Classification Events (HP Forest)

Table 23 shows the 3x3 confusion matrix of the random forest for validated data and table 20 shows the multi-class classification events of validated data. Table 24 shows that figure 159 is the event classification rate of the Non-Functional pump as the value in the confusion matrix shown in figure 159 is the same as the non-functional pump.

The precision of the model on functional, functional need repair, and non-function is calculated as shown in table 25.

Precision of Functional:	5681/ (5681+ 1874) = 0.75
Precision of Functional Needs Repair:	121/ (121+62) = 0.66
Precision of Non-Functional:	2952/ (2952+530) = 0.85

Table 25: Precision of each target prediction in (HP Forest).

From table 25, it shows that the precision of functional is 0.75 which means that all pumps that labeled as a functional pump, there are 75% are functional. Besides, the precision of the functional needs repair pump is 0.66 which means there are 66% of the pump are correctly labeled as functional needs repair among all labeled functional needs repair pumps. The precision of the non-functional pump is 0.85 which means that there are 85% correctly labeled as non-functional pumps among all labeled non-functional pumps. The precision of the pump of all types of pumps in the random forest model is higher than Forward Logistic Regression.

The number of false positives is lower than false negative which means the prediction has a higher probability of underestimate rather than overestimate as shown in figure 159. The model may provide more underestimate predictions such as the pump is non-functional, but the predicted result is functional which will affect the further action toward the pump.

5.0 Assess (Model Comparison)

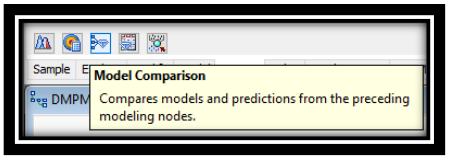


Figure 161: Model Comparison Node (1)

Figure 160: Model Comparison Node (2)

The figure above shows the model comparison node. The node is in Access Tab. Model Comparison node is used to compare the models and select the best fit model to the dataset.

Model Comparison

Fit Statistics
Model Selection based on Valid: Misclassification Rate (_VMISC_)

Selected Model	Model Node	Model Description	Valid: Misclassification Rate	Train: Average Squared Error	Train: Misclassification Rate	Valid: Average Squared Error
Y	HPTree	HP Tree	0.21081	0.08938	0.18427	0.10841
	HPDMForest	HP Forest (Selected)	0.21979	0.09855	0.20612	0.10441
	Neural	Neural Network	0.22216	0.09742	0.20470	0.10621
	Reg3	Foward LR (Selected)	0.25895	0.12216	0.26329	0.12169
	Tree	Decision Tree	0.26245	0.12732	0.26190	0.12875
	Reg	StepwiseRegression	0.26436	0.12082	0.25875	0.12349

Figure 162: Fit Statistics

Figure 162 shows the comparison of misclassification rate and average squared error of each model. From the figure, it shows that HP Tree, HP Forest. and Neural Network is having better misclassification rates which are 0.21, 0.22, and 0.22.

Event Classification Table
Model Selection based on Valid: Misclassification Rate (_VMISC_)

Model Node	Model Description	Data Role	Target	Target Label	False Negative	True Negative	False Positive	True Positive
HPTree	HP Tree	TRAIN	REP_status_group	Replacement: status_group	3224	21825	2321	11892
HPTree	HP Tree	VALIDATE	REP_status_group	Replacement: status_group	1579	9171	1179	4901
Tree	Decision Tree	TRAIN	REP_status_group	Replacement: status_group	4066	15461	1787	6731
Tree	Decision Tree	VALIDATE	REP_status_group	Replacement: status_group	4039	15479	1769	6760
Neural	Neural Network	TRAIN	REP_status_group	Replacement: status_group	3897	21941	2205	11219
Neural	Neural Network	VALIDATE	REP_status_group	Replacement: status_group	1801	9252	1098	4679
Reg	StepwiseRegression	TRAIN	REP_status_group	Replacement: status_group	2336	9235	1114	4142
Reg	StepwiseRegression	VALIDATE	REP_status_group	Replacement: status_group	5464	21321	2826	9654
HPDMForest	HP Forest (Selected)	TRAIN	REP_status_group	Replacement: status_group	5154	25739	1857	12122
HPDMForest	HP Forest (Selected)	VALIDATE	REP_status_group	Replacement: status_group	1368	6370	530	2952
Reg3	Foward LR (Selected)	TRAIN	REP_status_group	Replacement: status_group	3600	14258	1711	6114
Reg3	Foward LR (Selected)	VALIDATE	REP_status_group	Replacement: status_group	3551	14321	1647	6165

Figure 163: Comparison of Event Classification Table

Figure 163 shows the comparison of the confusion matrix of each model. As the confusion matrix refers to the non-functional pump, the selection model will be a lower misclassification rate and lower false negative. The summary of the models finding is shown in Table 26.

Model	**Misclassification Rate**	**False Negative**
HP Tree	Number of leaves: 1307 Misclassification Rate: 0.21	1579
HP Forest	Number of trees: 39 Misclassification Rate: 0.22	1368
Neural Network	Number of Iteration: 221 Hidden Layer: 40 Misclassification Rate: 0.222	1801
Forward Logistic Regression	Number of Iteration: 16 Misclassification Rate: 0.259	3551
Decision Tree	Number of Leaves: 55 Misclassification Rate: 0.262	4039
Stepwise Logistic Regression	Number of Iteration: 15 Misclassification Rate: 0.264	5464

Table 26: Summary of Models

From table 26, the selected model is HP Forest although the misclassification rate is not the best, it is the second-lowest misclassification rate with 0.2198 and the false negative is the lowest compared to other models. The reason for selecting a model that has a lower false negative is because getting a false alarm is better than miss the chance of acknowledging the pump is non-functional.

6.0 Conclusion

Data mining and predictive modelling is a very large subject and there any many applications and uses for it, one of it is which is modelling. We started the project by selecting the data set from ministry of Tanzania and Tarriffa. This project as taught us many concepts in data mining and predictive modelling which was a valuable experience through the project. We as team of three was able to understand how perfectly we need to do the data exploration and data pre-processing while doing any predictive modelling. We also have understood various functions like impute, drop, replacement, filter, and data patriation. In order proceed to the data modelling we need to choose two models each make the report for the modelling.

The modeling is completely different from other, and we have one common goal to find the best model which is finding the least misclassification rate among them. The model can be accepted with over 70% which the misclassification rate must be less than 0.3. The less misclassification rate is more accurate the model is going to be. In oder to achieve the goal everyone will have to change the way the data partition is done and need to change some in the data pre-processing. The report will be drafted individually and then we must gather again to discuss the final stage which is the model comparison.

Now has each one of us has made a report of 2 models we have to compare the model, and to compare them we have used the control point method and joined the models together, then we can use the model comparison node, compare the model, and conclude the best model.

7.0 References:

Biscobing, J. (2018, September 28). *data sampling*. SearchBusinessAnalytics. https://searchbusinessanalytics.techtarget.com/definition/data-sampling

Facer, C. (2018, September 3). What is Data Filtering? Displayr. https://www.displayr.com/what-is-data-filtering/

HAYES, A. (2019). Stepwise Regression. Investopedia. https://www.investopedia.com/terms/s/stepwise-regression.asp

Hoffman, J. I. (2019). Logistic Regression. Basic Biostatistics for Medical and Biomedical Practitioners, 581–589. https://doi.org/10.1016/b978-0-12-817084-7.00033-4

imanuel. (2019, July 25). What is Predictive Modeling ? PAT RESEARCH: B2B Reviews, Buying Guides & Best Practices; Bigtexts.com. https://www.predictiveanalyticstoday.com/predictive-modeling/

Kleinbort, A. (2019, December 2). A Comprehensive Guide To Data Imputation. Medium. https://towardsdatascience.com/a-comprehensive-guide-to-data-imputation-e82eadc22609

MacKenzie, D. I., Nichols, J. D., Royle, J. A., Pollock, K. H., Bailey, L. L., & Hines, J. E. (2018). Fundamental Principals of Statistical Inference. Occupancy Estimation and Modeling, 71–111. https://doi.org/10.1016/b978-0-12-407197-1.00004-1

Mbaabu, O. (2020, December 11). Introduction to Random Forest in Machine Learning. Engineering Education (EngEd) Program | Section. https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/

NetSuite.com. (2020, September 22). Predicting Your Business's Future. Oracle NetSuite. https://www.netsuite.com/portal/resource/articles/financial-management/predictive-Modeling.shtml

Neural Network Definition. (2021). Retrieved August 5, 2021, from Investopedia website: https://www.investopedia.com/terms/n/neuralnetwork.asp

Neural Networks - What are they and why do they matter? (2021). Retrieved August 5, 2021, from Sas.com website: https://www.sas.com/en_my/insights/analytics/neural-networks.html

Paul, U. (2017, February 16). Machine Learning: Running A Random Forest In SAS. LinkedIn. https://www.linkedin.com/pulse/machine-learning-running-random-forest-sas-paul-ulloa-mba/

Saraswat, M. (2017, April 7). Practical Tutorial on Random Forest and Parameter Tuning in R Tutorials & Notes | Machine Learning. HackerEarth. https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/tutorial-random-forest-parameter-tuning-r/tutorial/

Stephanie, G. (2017, September 19). Forward Selection: Definition. Statistics How To. https://www.statisticshowto.com/forward-selection/

SAS Help Center. (2021). Retrieved August 5, 2021, from Sas.com website: https://documentation.sas.com/doc/en/emref/14.3/n0vvj2elvaoz3mn1516yd3dx2deu.htm#:~:text=The%20HP%20Tree%20node%20enables,high%2Dperformance%20data%20mining%20environment.&text=The%20HP%20Tree%20node%20supports%20two%20methods%20of%20model%20validation.

Techopedia. (2019). What is Data Preprocessing? - Definition from Techopedia. Techopedia.com. https://www.techopedia.com/definition/14650/data-preprocessing

SAS Help Centre. (2017, August 27). SAS Help Center. Documentation.sas.com. https://documentation.sas.com/doc/en/emref/14.3/n0u3s4tv5v88cfn1dx0zkfd9pjm5.htm

Selerity. (2019, December 12). Types of predictive analytics models and how they work. Selerity. https://seleritysas.com/blog/2019/12/12/types-of-predictive-analytics-models-and-how-they-work/