Project Final Report

Venkata Sri Ambica Mokarala

Objective:

Find out the best model/algorithm that predicts one out of three functionality labels (Functional, non-functional, and functional needs repairs) of water pipes in the given dataset using SAS enterprise miner.

Methodology:

To Predict the functionality of the water pipes we will use three classification algorithms in this project. The three classification models that are used to predict the output in this project are Logistic regression, Decision tree and Neural Networks.

As shown in the variable summary, there are 31 nominal variables and 9 interval variables, and timeId variable. There are a total of 47520 observations in the given dataset. The dataset contains different fields that explain different features of the water pipes. Features such as basin (Geographic water basin), scheme_management (Who operates the waterpoint), source (The source of the water), construction_year (Year the waterpoint was constructed) etc. are some of the helpful features to find out the functionality and location of the water pipes in the dataset. We can also reject “num private” and “recorded_by” as there is not much useful information in these data fields.

```
Variable Summary

         Measurement    Frequency
Role        Level         Count

ID         NOMINAL          1
INPUT      INTERVAL         9
INPUT      NOMINAL         30
TIMEID     INTERVAL         1



The CONTENTS Procedure

Data Set Name        EMWS2.FIMPORT2_DATA                              Observations          47520
Member Type          DATA                                             Variables             41
Engine               V9                                               Indexes               0
Created              30/04/2022 04:13:52                              Observation Length    600
Last Modified        30/04/2022 04:13:52                              Deleted Observations  0
Protection                                                            Compressed            NO
Data Set Type                                                         Sorted                NO
Label
Data Representation  SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64
Encoding             utf-8  Unicode (UTF-8)
```

Fig. Variable summary of the dataset.

Before modeling our data, it is essential to understand and clean the data to get the best results. We perform basic EDA on the dataset to analyze the data, find out missing values, detect anomalies in the data, and find patterns. “StatExplorer” node is used to analyze the variables.

<u>Exploratory data Analysis:</u>

Dataset contains a total of 39 variables. The dataset consists of input variables of both nominal and interval datatypes and the target variable of the dataset is “status_group” which has 3 labels that are to be predicted by the algorithms. The Status_group variable of the dataset has 3 values functional, non-functional, and functional needs repair.

Results - Node: StatExplore Diagram: final

File Edit View Window

Output

Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
TRAIN	basin	INPUT	9	0	Lake Victoria	17.27	Pangani	15.03
TRAIN	extraction_type	INPUT	18	0	gravity	44.91	nira/tanira	13.82
TRAIN	extraction_type_class	INPUT	7	0	gravity	44.91	handpump	27.82
TRAIN	extraction_type_group	INPUT	13	0	gravity	44.91	nira/tanira	13.82
TRAIN	funder	INPUT	1697	2876	Government Of Tanzania	15.26		6.05
TRAIN	installer	INPUT	1924	2889	DWE	29.34		6.08
TRAIN	lga	INPUT	125	0	Njombe	4.21	Moshi Rural	2.12
TRAIN	management	INPUT	12	0	vwc	68.30	wug	10.95
TRAIN	management_group	INPUT	5	0	user-group	88.42	commercial	6.04
TRAIN	payment	INPUT	7	0	never pay	42.76	pay per bucket	15.20
TRAIN	payment_type	INPUT	7	0	never pay	42.76	per bucket	15.20
TRAIN	permit	INPUT	3	2439	TRUE	65.29	FALSE	29.57
TRAIN	public_meeting	INPUT	3	2689	TRUE	85.74	FALSE	8.60
TRAIN	quality_group	INPUT	6	0	good	85.51	salty	8.78
TRAIN	quantity	INPUT	5	0	enough	55.85	insufficient	25.47
TRAIN	quantity_group	INPUT	5	0	enough	55.85	insufficient	25.47
TRAIN	region	INPUT	21	0	Iringa	8.95	Shinyanga	8.37
TRAIN	scheme_management	INPUT	13	3102	VWC	62.00	WUG	8.76
TRAIN	scheme_name	INPUT	2537	22523		47.40	K	1.14
TRAIN	source	INPUT	10	0	shallow well	28.49	spring	28.49
TRAIN	source_class	INPUT	3	0	groundwater	77.08	surface	22.43
TRAIN	source_type	INPUT	7	0	shallow well	28.49	spring	28.49
TRAIN	subvillage	INPUT	6001	54	Madukani	0.89	Majengo	0.87
TRAIN	ward	INPUT	2076	0	Igosi	0.51	Imalinyi	0.43
TRAIN	water_quality	INPUT	7	0	soft	85.51	salty	8.23
TRAIN	waterpoint_type	INPUT	7	0	communal standpipe	47.95	hand pump	29.61
TRAIN	waterpoint_type_group	INPUT	6	0	communal standpipe	58.11	hand pump	29.61
TRAIN	wpt_name	INPUT	6001	0	none	5.84	Shuleni	2.82
TRAIN	status_group	TARGET	3	0	functional	54.30	non functional	38.41

Fig. StatExplorer output (class variables)

Number of levels:

There are some nominal input variables with more than 128 levels. The variables with more than 128 levels which are not required for the analysis are Funder (Who funded the well), installer (Organization that installed the well), Scheme_name (Who operates the waterpoint), Subvillage (Geographic location), ward (Geographic location), wpt_name (Name of the waterpoint if there is one).

Missing Values:

The variables funder, installer, permit, public_meeting, scheme_management, scheme_name, subvillage are some of the variables with missing variables.

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
amount_tsh	INPUT	322.0476	3200.623	47520	0	0	0	350000	57.23017	4638.376
construction_year	INPUT	1303.353	950.7639	47520	0	0	1986	2013	-0.64118	-1.58846
district_code	INPUT	5.63931	9.661285	47520	0	0	3	80	3.948458	16.05677
gps_height	INPUT	668.7454	692.9722	47520	0	-63	370	2770	0.462198	-1.29129
latitude	INPUT	-5.705	2.943503	47520	0	-11.6494	-5.01771	-2E-8	-0.15402	-1.05715
longitude	INPUT	34.09132	6.538403	47520	0	0	34.91166	40.34519	-4.20379	19.36255
num_private	INPUT	0.504566	13.25385	47520	0	0	0	1776	89.07841	10076.14
population	INPUT	179.5283	472.773	47520	0	0	25	30500	13.53763	468.8981
region_code	INPUT	15.32652	17.6188	47520	0	1	12	99	3.16241	10.20446

Fig. StatExplorer output. (Interval variables)

Skewness:

There are some interval variables that do not have a skewness value between +3 to -3. Means these variables are not normally distributed. The variables are amount_tsh, longitude, num_private and population.

To run the data through the models, It is important to clean data of missing values because not all algorithms/models are good at handling the missing values. So, to handle the missing values and to clean the data we use “Replacement” node and “StatExplorer” as shown in below fig.

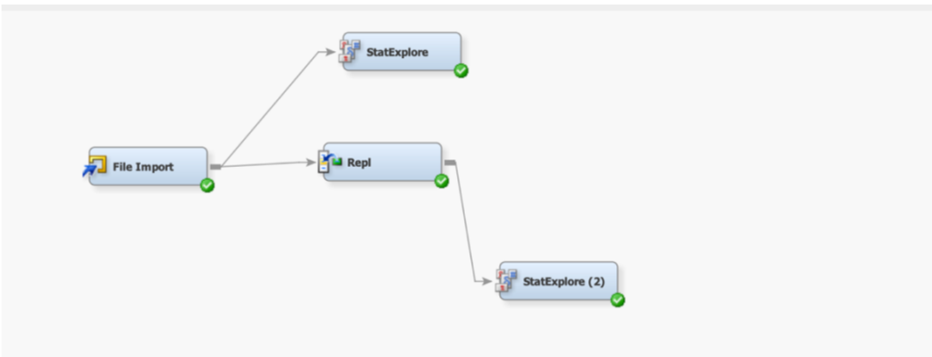

Fig. Diagram.

Variable	Formatted Value	Replacement Value	Frequency Count	Type	Character Unformatted Value	Numeric Val
payment_type	unknown		6521	C	unknown	.
payment_type	on failure		3154	C	on failure	.
payment_type	annually		2886	C	annually	.
payment_type	other		844	C	other	.
payment_type	_UNKNOWN_	_DEFAULT_	.	C		.
permit	TRUE		31028	C	TRUE	.
permit	FALSE		14053	C	FALSE	.
permit		_MISSING_	2439	C		.
permit	_UNKNOWN_	_DEFAULT_	.	C		.
public_meeting	TRUE		40743	C	TRUE	.
public_meeting	FALSE		4088	C	FALSE	.
public_meeting		_MISSING_	2689	C		.
public_meeting	_UNKNOWN_	_DEFAULT_	.	C		.
quality_group	good		40633	C	good	.
quality_group	salty		4173	C	salty	.
quality_group	unknown		1490	C	unknown	.

Fig. Replacement editor window

The empty values in the dataset are set to “_MISSING_” in the replacement editor window of the replacement node. And the data/variables after the replacement can be explored using StatExplorer connected to the replacement node. The output of StatExplorer after replacing the missing values is shown below.

Output

(maximum 500 observations printed)

Data Role=TRAIN

Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
TRAIN	REP_permit	INPUT	3	2439	TRUE	65.29	FALSE	29.57
TRAIN	REP_public_meeting	INPUT	3	2689	TRUE	85.74	FALSE	8.60
TRAIN	REP_scheme_management	INPUT	13	3102	VWC	62.00	WUG	8.76
TRAIN	basin	INPUT	9	0	Lake Victoria	17.27	Pangani	15.03
TRAIN	extraction_type	INPUT	18	0	gravity	44.91	nira/tanira	13.82
TRAIN	extraction_type_class	INPUT	7	0	gravity	44.91	handpump	27.82
TRAIN	extraction_type_group	INPUT	13	0	gravity	44.91	nira/tanira	13.82
TRAIN	lga	INPUT	125	0	Njombe	4.21	Moshi Rural	2.12
TRAIN	management	INPUT	12	0	vwc	68.30	wug	10.95
TRAIN	management_group	INPUT	5	0	user-group	88.42	commercial	6.04
TRAIN	payment	INPUT	7	0	never pay	42.76	pay per bucket	15.20
TRAIN	payment_type	INPUT	7	0	never pay	42.76	per bucket	15.20
TRAIN	quality_group	INPUT	6	0	good	85.51	salty	8.78
TRAIN	quantity	INPUT	5	0	enough	55.85	insufficient	25.47
TRAIN	quantity_group	INPUT	5	0	enough	55.85	insufficient	25.47
TRAIN	region	INPUT	21	0	Iringa	8.95	Shinyanga	8.37
TRAIN	source	INPUT	10	0	shallow well	28.49	spring	28.49
TRAIN	source_class	INPUT	3	0	groundwater	77.08	surface	22.43
TRAIN	source_type	INPUT	7	0	shallow well	28.49	spring	28.49
TRAIN	water_quality	INPUT	7	0	soft	85.51	salty	8.23
TRAIN	waterpoint_type	INPUT	7	0	communal standpipe	47.95	hand pump	29.61
TRAIN	waterpoint_type_group	INPUT	6	0	communal standpipe	58.11	hand pump	29.61
TRAIN	status_group	TARGET	3	0	functional	54.30	non functional	38.41

Fig. Output of StatExplorer after replacement.

Variable worth and chi-square plots of the StatExplorer output are shown below. The larger the Chi-square value, the greater the probability that there really is a significant difference. From analyzing the plots, we can find quantity and quantity_group are the two variables with the highest worth in the dataset. variables lga, Quantity and quantity_group have the high chi-square values.

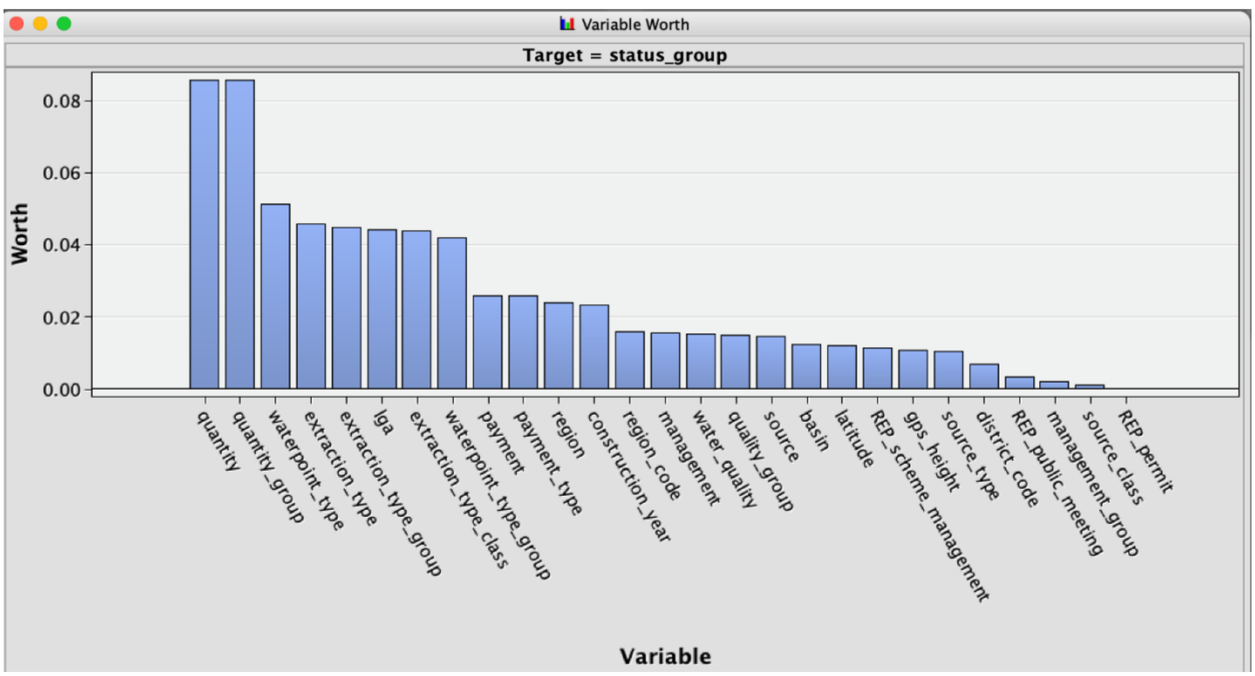

Fig. Variable worth plot of statexplorer 2

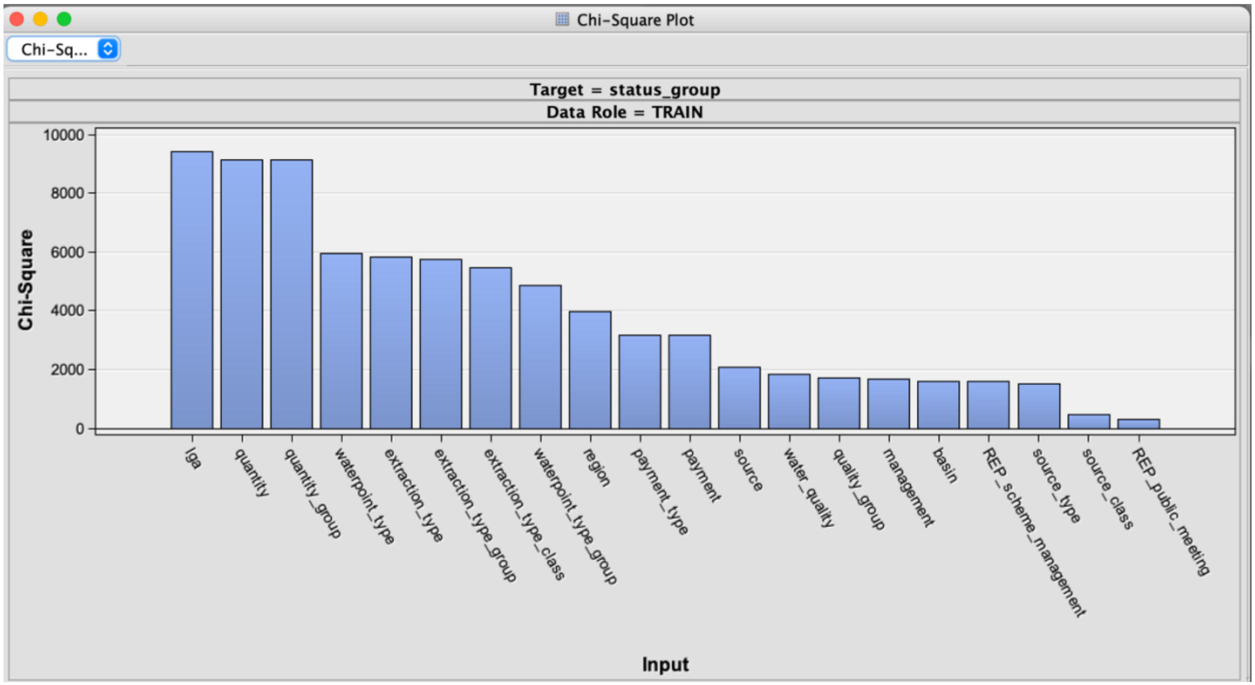

Fig. Chi-Square plot of statexplorer 2

After rejecting the input variables which have more than 128+ levels, and variables with skewness not in the range of +3 to -3 and replacing the missing values. Now our data is ready to be modelled.

Data Partitioning:

Data partitioning is a technique used to distribute data into multiple datasets to improve the performance of the model. The main goal of any classification algorithm is to find out how accurately a model can predict unseen instances. So, we use data partitioning and validation datasets to find the accuracy of the model. In this project, cleaned data is partitioned 70% into training (for preliminary model fitting, to find the best model weights using this data set), 15% into validation (assessing the adequacy of the model, and for model fine-tuning), and 15% into test data. In this way, we can prevent our model from overfitting, and this will accurately evaluate our model. The data partition node is connected to the replacement node as shown below to check the discrepancy.

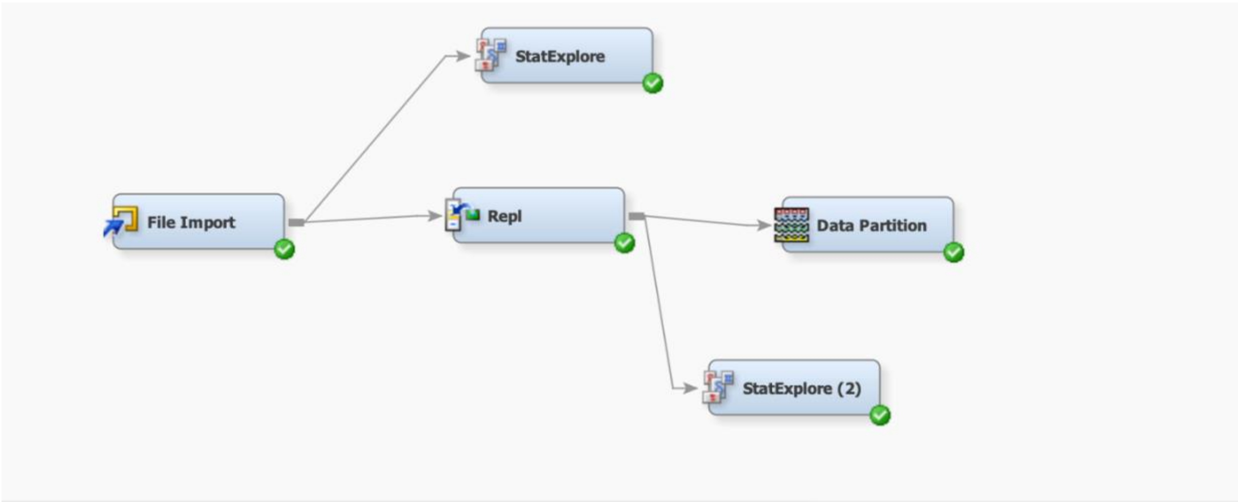

Fig. Diagram.

Now this partitioned data is used for the Logistic regression, Decision tree and Neural Networks algorithm to predict the target variable.

<u>Model 1- Decision Tree & Random Forest:</u>

A decision tree is a flowchart and a specific type of probability tree that starts with one main idea and divides it into branches based on the decisions. Tree construction is performed in top-down, recursive, divide-and-conquer manner. In complex decisions or when different factors including the uncertainty involved, then decision trees are the best model to deal with. It is very helpful in analyzing quantitative data and deciding based on numbers. A decision tree includes some symbols like alternative branches, decision nodes, chance nodes, and end nodes. These symbols combinedly explain the outcomes or decisions of the model. In decision tree, if all the data belongs to one class, then we call it as a pure node. The color of the branch represents the purity level.

To clearly lay out the issue, analyze the possible consequences of our decision, and provide a framework to qualify our values of functional, nonfunctional, and the water pumps that need repair, let's start with the Decision Tree for analysis.

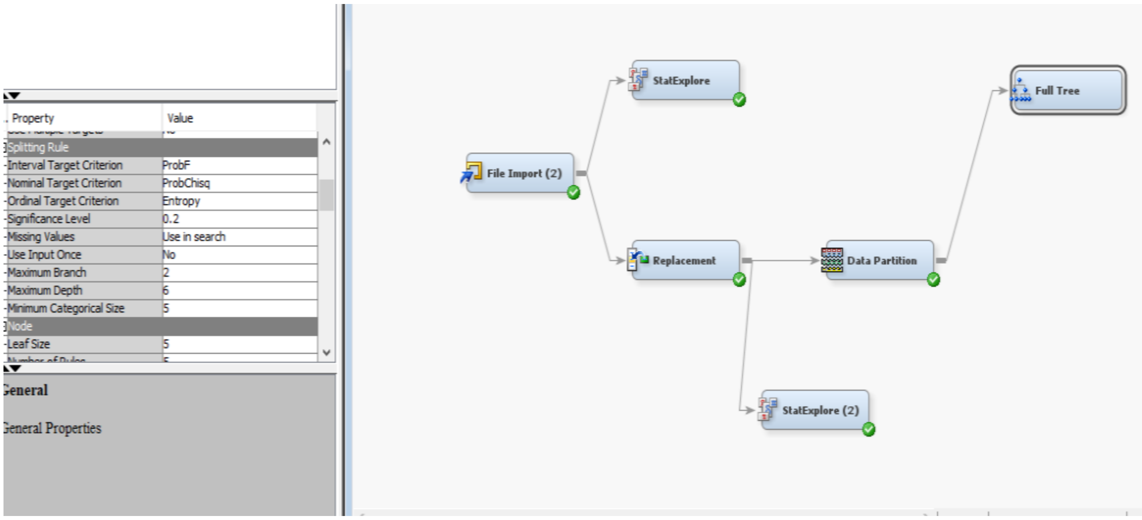

Fig. Diagram representing connection of Decision Tree node

There are two types of decision trees.
1. Full Tree
2. Pruned Tree

1.1. Full-Tree:

In the full tree, we run the algorithm completely and let it grow fully. In this analysis by default, we have the maximum branches to 2 and depth to 6. That means the tree will grow up to 6 generations of splits and we are analyzing the assessment measure based on the misclassification rate.

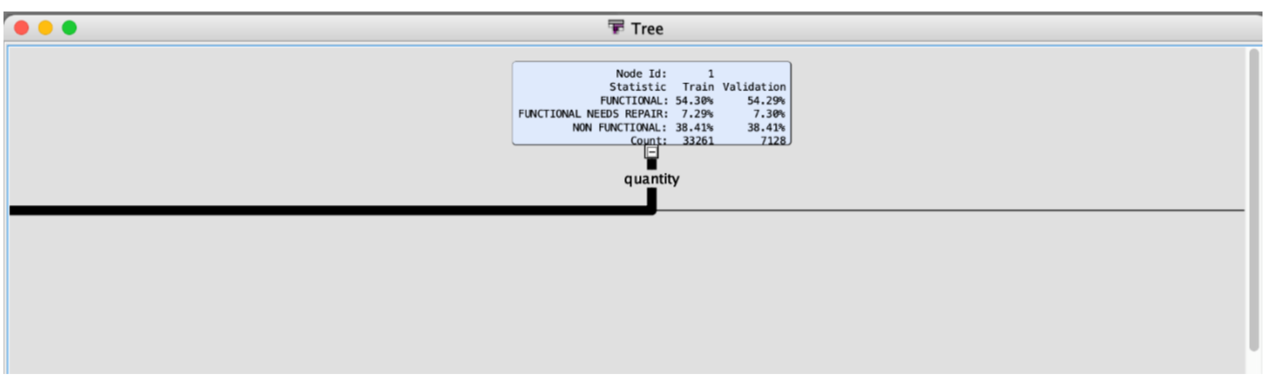

Fig. Tree window

From the results above in the Tree window, we can observe quantity is the first variable to split and we can also see that the ratio is divided into functional is 54.30%, functional needs repair is 7.29%, and nonfunctional is 38.42% for training data and for validation data set 54.29% as functional, 7.30% as functional needs repair and 38.42% as non-functional.

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
status_group		_NOBS_	Sum of Frequencies	33261	7128	7131
status_group		_MISC_	Misclassification Rate	0.283635	0.2883	0.28341
status_group		_MAX_	Maximum Absolute...	0.997912	1	1
status_group		_SSE_	Sum of Squared Err...	12688.49	2764.797	2776.308
status_group		_ASE_	Average Squared E...	0.127161	0.129293	0.129776
status_group		_RASE_	Root Average Squa...	0.356596	0.359573	0.360245
status_group		_DIV_	Divisor for ASE	99783	21384	21393
status_group		_DFT_	Total Degrees of Fr...	66522	.	.

Fig. Fit statistics

From the fit statistics window, we can observe the misclassification rate for training, validation, and test datasets. The Misclassification rate for training data is 0.283635 (28.3635%) so the accuracy is 1- 0.2836 = 0.7164 I.e., **1.64%**. Whereas the misclassification rate for validation data set is 0.2883(28.83%) so the accuracy can be calculated as 1- 0.2883= 0.7117 I.e.,**71.17%** Coming to the test dataset, the misclassification rate is 0.28341(28.341%) which means the accuracy is 1- 0.28341 = 0.71659 I.e.,**71.659%**.

The below figure represents the importance of the variables that are contributing more to our model prediction and the number of splitting rules for each variable. As "quantity" is the first variable to split it represents 1.0000 which means 100% of importance and the remaining variables follow in the order.

Variable Importance

Variable Name	Label	Number of Splitting Rules	Importance	Validation Importance	Ratio of Validation to Training Importance
REP_quantity	Replacement: quantity	3	1.0000	1.0000	1.0000
REP_waterpoint_type	Replacement: waterpoint_type	3	0.7087	0.6772	0.9555
REP_extraction_type	Replacement: extraction_type	6	0.4318	0.4052	0.9382
REP_payment	Replacement: payment	3	0.3542	0.3586	1.0125
REP_source	Replacement: source	7	0.3523	0.3642	1.0336
REP_basin	Replacement: basin	7	0.3391	0.3466	1.0222
REP_management	Replacement: management	8	0.3029	0.2703	0.8923
REP_quality_group	Replacement: quality_group	3	0.1244	0.1308	1.0513
gps_height		2	0.0365	0.0000	0.0000

Fig. Variable importance

<u>1.2. Pruned Tree:</u>

The only drawback in the full decision tree is "Overfitting". As we allow the tree to grow fully, the number of branches increases which results in noises and outliers, and due to that Overfitting issue occurs. So, to stop that issue we use pruned tree which is the modification of full tree. In pruned tree, we reduce the number of child nodes from the branch nodes.

It is of two types:

1. Pre-Pruning or early stopping
2. Post-Pruning

In Pre-Pruning, we stop the tree before it completes the classification whereas, in post-pruning we prune the tree after it completes the classification. Here in SAS enterprise miner, by selecting the assessment method on misclassification rate measure, it automatically performs the pruning process. It stops the growth of tree at a minimum misclassification rate in the validation set. So, in the end, it results in a smaller number of nodes compared to full tree model.

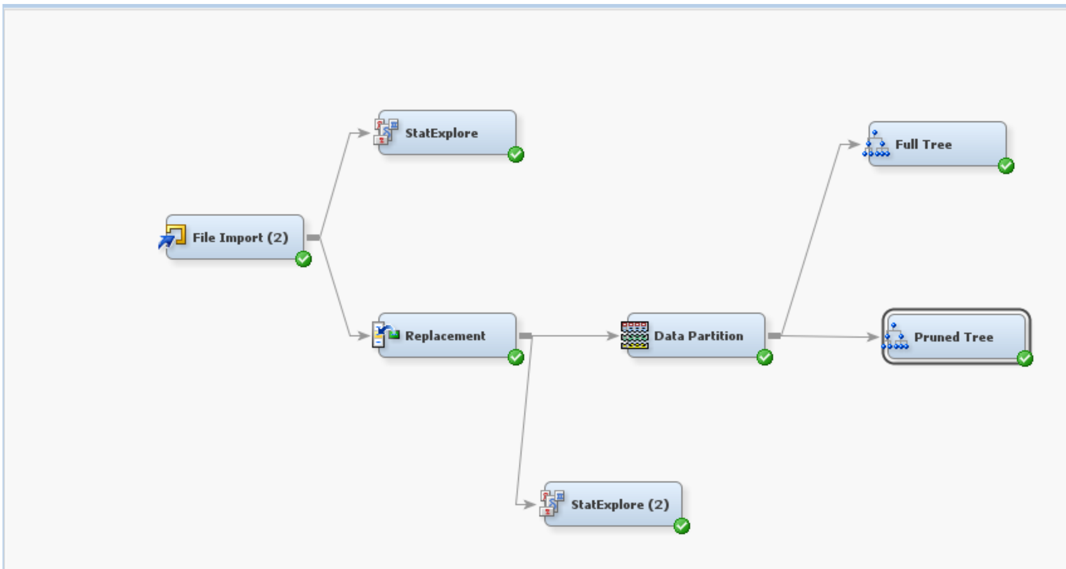

Fig. Adding Pruned

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
status_group		_NOBS_	Sum of Frequencies	33261	7128	7131
status_group		_MISC_	Misclassification Rate	0.284958	0.288159	0.284532
status_group		_MAX_	Maximum Absolute...	0.997572	0.997572	0.997572
status_group		_SSE_	Sum of Squared Err...	13211.21	2860.287	2850.351
status_group		_ASE_	Average Squared E...	0.132399	0.133758	0.133238
status_group		_RASE_	Root Average Squa...	0.363867	0.36573	0.365017
status_group		_DIV_	Divisor for ASE	99783	21384	21393
status_group		_DFT_	Total Degrees of Fr...	66522	.	.

Fig. Fit statistics for Pruned tree

From the fit statistics of Pruned decision tree model, we can observe the misclassification rates. The misclassification rate for train data is 0.284958(28.4958%) which means the accuracy is 1-0.284958 = 0.715042 I.e., **71.50%**. For the validation data, it is 0.288159 (28.8159%) so the accuracy will be 1-0.288159 = 0.711841 I.e., **71.18%**. Whereas, for test data the misclassification rate is 0.284532 so the accuracy is 1-0.284532 = 0.715468 I.e., **71.5468%.**

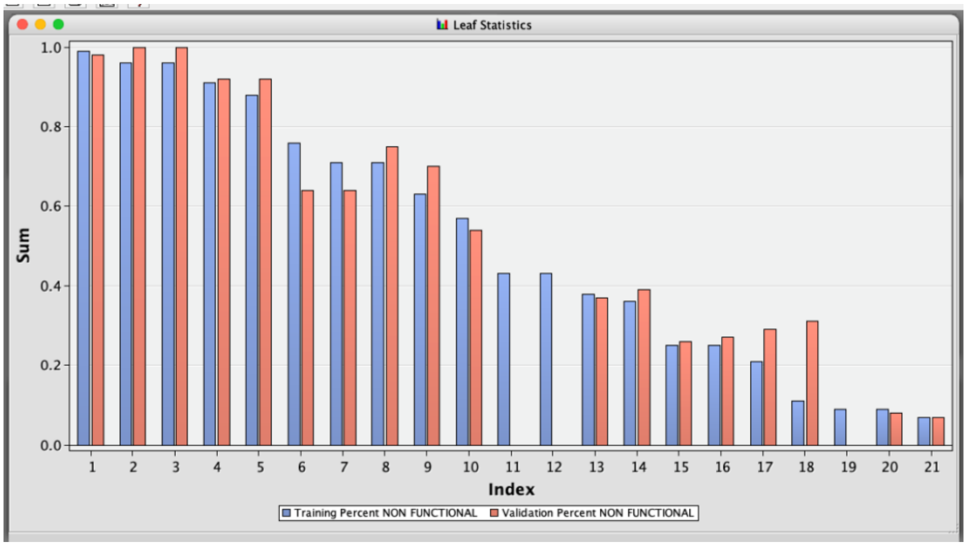

Fig. leaf statistics shows leaves of the decision tree

Along with the number of branches, we can also observe the number of leaves in the model using leaf statistics. From the above fig, we can see the leaf statistics plot of the pruned decision tree. So, the index on the X-axis represents the number of leaves in the optimal tree. Here in this model, the number of leaves is 21.

Variable Importance

Variable Name	Label	Number of Splitting Rules	Importance	Validation Importance	Ratio of Validation to Training Importance
quantity		2	1.0000	1.0000	1.0000
waterpoint_type_group		1	0.6989	0.6701	0.9588
lga		4	0.5372	0.5354	0.9967
construction_year		2	0.3109	0.2274	0.7312
extraction_type		1	0.2580	0.1989	0.7710
region		2	0.2143	0.1909	0.8906
source		1	0.1786	0.2216	1.2407
extraction_type_group		2	0.1421	0.1303	0.9171
quality_group		1	0.1195	0.1323	1.1077
management		3	0.0903	0.0840	0.9293
latitude		1	0.0522	0.0548	1.0498

Fig. Variable Importance after Pruned in output window

By comparing the variable importance of both full tree and pruned tree, we can observe the changes in the number of splitting rules (especially for lga variable) and minor changes in the importance of the variables as well.

<u>1.3. HP Forest:</u>

The Hp forest model is one kind of decision tree. As it uses the methodology of the decision tree and creates random forests in a high-performance environment. It creates a predictive model for regression. Random forest is an ensemble learning method for classification and regression tasks. The random forest will have several trees different from each other based on training, validation, and test sets.

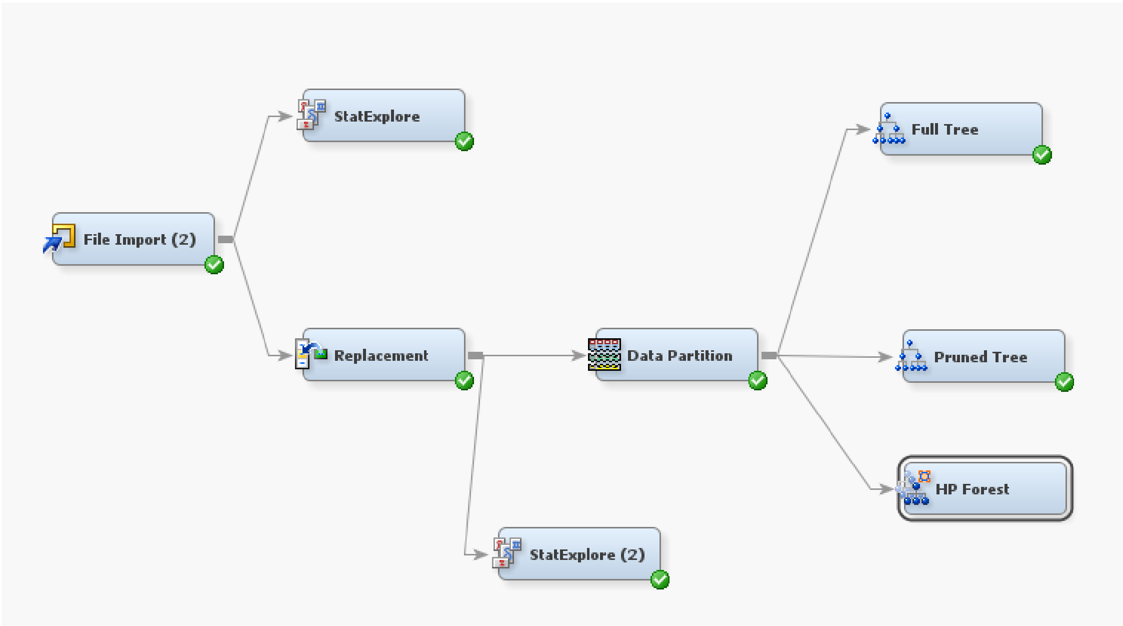

Fig. Connecting HP Forest node

The Random branch assessment method is used to calculate variable importance (margin for a class target and absolute error for an interval target) based on the validation data when available. Hence, selecting the variable importance method to Random assessment.

```
Event Classification Table

Data Role=TRAIN Target=status_group Target Label=' '

  False       True       False      True
Negative    Negative    Positive   Positive

  4216       19266       1219       8560


Data Role=VALIDATE Target=status_group Target Label=' '

  False       True       False      True
Negative    Negative    Positive   Positive

   950        4087        303       1788
```

Fig. Confusion Matrix

The above confusion matrix displays as per training and validation set shows what's predicted and what's actual and how many are rightly and how many are wrongly classified.

Classification Table

Data Role=TRAIN Target Variable=status_group Target Label=' '

Target	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
FUNCTIONAL	FUNCTIONAL	74.5934	94.2248	17017	51.1620
FUNCTIONAL NEEDS REPAIR	FUNCTIONAL	7.3160	68.8247	1669	5.0179
NON FUNCTIONAL	FUNCTIONAL	18.0906	32.3028	4127	12.4079
FUNCTIONAL	FUNCTIONAL NEEDS REPAIR	20.3288	0.7530	136	0.4089
FUNCTIONAL NEEDS REPAIR	FUNCTIONAL NEEDS REPAIR	66.3677	18.3093	444	1.3349
NON FUNCTIONAL	FUNCTIONAL NEEDS REPAIR	13.3034	0.6966	89	0.2676
FUNCTIONAL	NON FUNCTIONAL	9.2750	5.0221	907	2.7269
FUNCTIONAL NEEDS REPAIR	NON FUNCTIONAL	3.1905	12.8660	312	0.9380
NON FUNCTIONAL	NON FUNCTIONAL	87.5345	67.0006	8560	25.7358

Data Role=VALIDATE Target Variable=status_group Target Label=' '

Target	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
FUNCTIONAL	FUNCTIONAL	73.7057	93.4367	3616	50.7295
FUNCTIONAL NEEDS REPAIR	FUNCTIONAL	7.3583	69.4231	361	5.0645
NON FUNCTIONAL	FUNCTIONAL	18.9360	33.9299	929	13.0331
FUNCTIONAL	FUNCTIONAL NEEDS REPAIR	16.7939	0.5685	22	0.3086
FUNCTIONAL NEEDS REPAIR	FUNCTIONAL NEEDS REPAIR	67.1756	16.9231	88	1.2346
NON FUNCTIONAL	FUNCTIONAL NEEDS REPAIR	16.0305	0.7670	21	0.2946
FUNCTIONAL	NON FUNCTIONAL	11.0952	5.9948	232	3.2548
FUNCTIONAL NEEDS REPAIR	NON FUNCTIONAL	3.3955	13.6538	71	0.9961
NON FUNCTIONAL	NON FUNCTIONAL	85.5093	65.3031	1788	25.0842

Fig. Classification table for our target variable displaying result as per training and validation set

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
status_group		_ASE_	Average Squared E...	0.101605	0.107767	0.108594
status_group		_DIV_	Divisor for ASE	99783	21384	21393
status_group		_MAX_	Maximum Absolut...	0.992996	0.994376	0.994376
status_group		_NOBS_	Sum of Frequencies	33261	7128	7131
status_group		_RASE_	Root Average Squ...	0.318755	0.328279	0.329536
status_group		_SSE_	Sum of Squared Er...	10138.41	2304.497	2323.147
status_group		_DISF_	Frequency of Class...	33261	7128	7131
status_group		_MISC_	Misclassification R...	0.217672	0.229517	0.231524
status_group		_WRONG_	Number of Wrong ...	7240	1636	1651

Fig. Fit Statistics of HPforest

As we see the misclassification rate here for validation set, it's lower than the decision and pruned tree, I.e., 0.229517 which makes the accuracy 1- 0.229517 = **77.04%.**

Impute:

Decision trees can handle the missing values automatically as they are robust to outliers as well, but Logistic regression and Neural networks are not good at handling the missing values. So, before we perform Logistic regression and Neural networks on our dataset, we need to impute the missing values in the dataset. The missing values in the dataset must be imputed by proper estimation methods like mean, median etc. before running the algorithm.

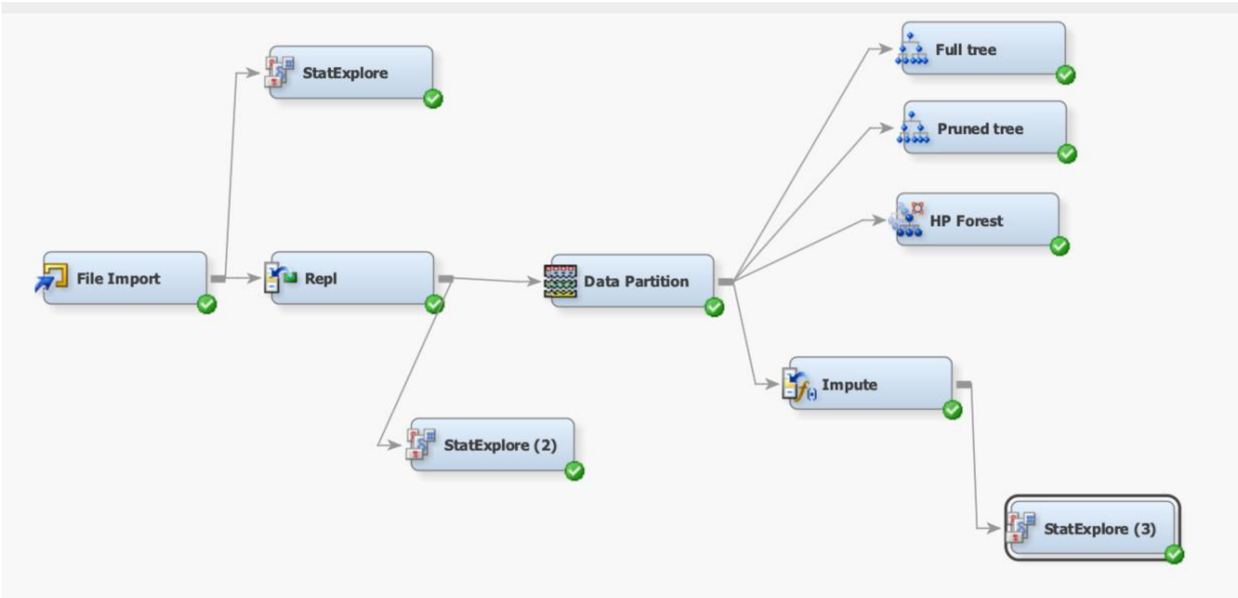

Fig.Diagram

In this project, the missing values of the interval variables are imputed with median as it is more robust to outliers. And missing values of the class variables are imputed with tree surrogate. Tree surrogates are like decision trees for a particular class variable. A StatExplore node is used to analyze the output after imputing the variables. After imputing the missing values of the variables, the output of the StatExplorer is as shown below. We can see there are no missing values for the interval variables and skewness of the variables is in between +3 and –3.

Data Role=TRAIN

Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
TRAIN	IMP_REP_permit	INPUT	2	0	TRUE	70.35	FALSE	29.65
TRAIN	IMP_REP_public_meeting	INPUT	2	0	TRUE	89.96	FALSE	10.04
TRAIN	IMP_REP_scheme_management	INPUT	11	0	VWC	67.02	WUG	9.77
TRAIN	basin	INPUT	9	0	Lake Victoria	17.20	Pangani	15.10
TRAIN	extraction_type	INPUT	18	0	gravity	44.76	nira/tanira	14.00
TRAIN	extraction_type_class	INPUT	7	0	gravity	44.76	handpump	27.94
TRAIN	extraction_type_group	INPUT	13	0	gravity	44.76	nira/tanira	14.00
TRAIN	lga	INPUT	124	0	Njombe	4.23	Moshi Rural	2.17
TRAIN	management	INPUT	12	0	vwc	68.34	wug	10.85
TRAIN	management_group	INPUT	5	0	user-group	88.44	commercial	5.95
TRAIN	payment	INPUT	7	0	never pay	42.65	pay per bucket	15.44
TRAIN	payment_type	INPUT	7	0	never pay	42.65	per bucket	15.44
TRAIN	quality_group	INPUT	6	0	good	85.66	salty	8.64
TRAIN	quantity	INPUT	5	0	enough	55.92	insufficient	25.56
TRAIN	quantity_group	INPUT	5	0	enough	55.92	insufficient	25.56
TRAIN	region	INPUT	21	0	Iringa	8.92	Shinyanga	8.38
TRAIN	source	INPUT	10	0	shallow well	28.63	spring	28.49
TRAIN	source_class	INPUT	3	0	groundwater	77.13	surface	22.39
TRAIN	source_type	INPUT	7	0	shallow well	28.63	spring	28.49
TRAIN	water_quality	INPUT	7	0	soft	85.66	salty	8.11
TRAIN	waterpoint_type	INPUT	7	0	communal standpipe	47.81	hand pump	29.69
TRAIN	waterpoint_type_group	INPUT	6	0	communal standpipe	58.22	hand pump	29.69
TRAIN	status_group	TARGET	3	0	functional	54.30	non functional	38.41

Fig. Output of StatExplorer3 node after imputing (class variables)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
construction_year	INPUT	1305.368	950.1515	33261	0	0	1986	2013	-0.6458	-1.58255
district_code	INPUT	5.676077	9.745351	33261	0	0	3	80	3.925735	15.84388
gps_height	INPUT	668.5112	692.9329	33261	0	-63	368	2770	0.466621	-1.28062
latitude	INPUT	-5.70437	2.945734	33261	0	-11.6494	-5.01936	-2E-8	-0.15342	-1.05895
region_code	INPUT	15.42449	17.8314	33261	0	1	12	99	3.132608	9.94236

Fig. Output of StatExplorer3 node after imputing (interval variables)

Model 2. Logistic Regression:

Logistic regression is a supervised algorithm model used to predict a dependent categorical target variable. It is a statistical analysis method to predict a binary outcome based on prior observations. It can also be used to predict one or more nominal data and that is called multinominal logistic regression. If an item can be classified into multiple classes, then it is represented as an ordinal type.

We have three different types of models in logistic regression. They are forward, backward, and stepwise regression models. In forward regression, the model analyzes the input variables from top to bottom and rejects the variables at the end. The backward regression is the reverse of the forward process, it analyzes from bottom to top. The stepwise regression model is a combination of both forward and backward. It analyzes the input variables and rejects the unnecessary variables parallelly. In this project, we used a stepwise regression model.

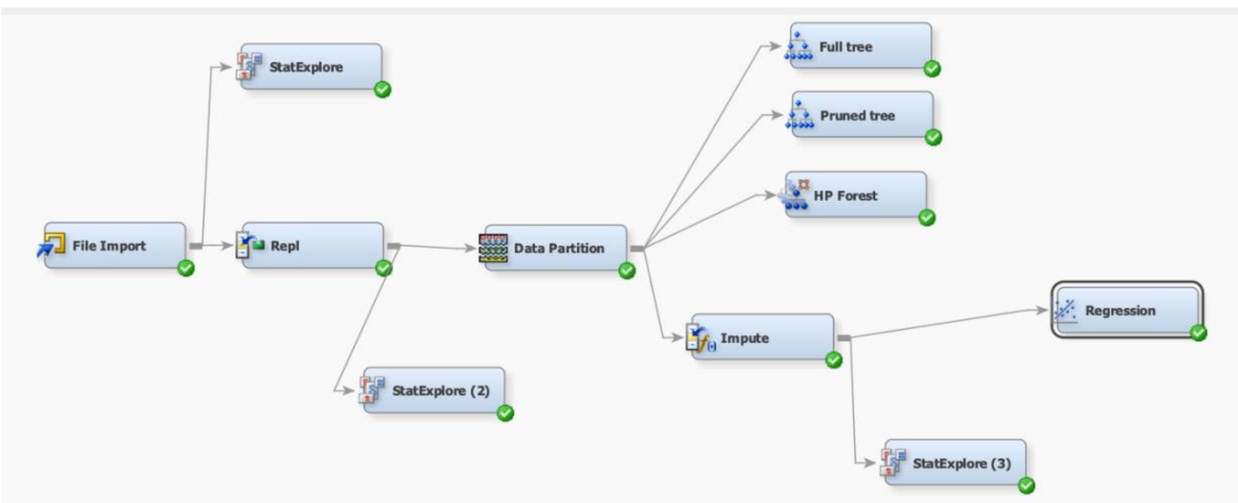

Fig. Diagram

The output shows which iteration is selected by the model and as we can see all the variables/predictors selected by the model are significant and model. As per the output, the selected model is based on step 13.

```
The selected model, based on the misclassification rate for the validation data, is the model trained in Step 13. It consists of the following effects:

Intercept  IMP_REP_public_meeting  IMP_REP_scheme_management  REP_gps_height  REP_population  basin  extraction_type  lga  management  payment  quality_group  quantity


    Likelihood Ratio Test for Global Null Hypothesis: BETA=0

    -2 Log Likelihood          Likelihood
  Intercept     Intercept &        Ratio
       Only      Covariates     Chi-Square       DF     Pr > ChiSq

  16889.124       11532.103      5357.0208      258         <.0001
```

Type 3 Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
IMP_REP_public_meeting	2	12.7277	0.0017
IMP_REP_scheme_management	16	45.2501	0.0001
REP_gps_height	2	12.3050	0.0021
REP_population	2	29.3586	<.0001
basin	16	49.0225	<.0001
extraction_type	16	143.9780	<.0001
lga	140	649.8479	<.0001
management	16	57.9000	<.0001
payment	12	271.7252	<.0001
quality_group	6	51.7649	<.0001
quantity	8	408.2020	<.0001
source	14	141.8869	<.0001
waterpoint_type	8	226.2581	<.0001

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
status_group		_AIC_	Akaike's Informatio...	40791.91	.	.
status_group		_ASE_	Average Squared E...	0.116777	0.11825	0.119508
status_group		_AVERR_	Average Error Func...	0.400749	0.406869	0.410979
status_group		_DFE_	Degrees of Freedo...	66120	.	.
status_group		_DFM_	Model Degrees of F...	402	.	.
status_group		_DFT_	Total Degrees of Fr...	66522	.	.
status_group		_DIV_	Divisor for ASE	99783	21384	21393
status_group		_ERR_	Error Function	39987.91	8700.477	8792.078
status_group		_FPE_	Final Prediction Error	0.118197	.	.
status_group		_MAX_	Maximum Absolute...	0.999492	0.999775	0.999791
status_group		_MSE_	Mean Square Error	0.117487	0.11825	0.119508
status_group		_NOBS_	Sum of Frequencies	33261	7128	7131
status_group		_NW_	Number of Estimat...	402	.	.
status_group		_RASE_	Root Average Sum ...	0.341726	0.343875	0.3457
status_group		_RFPE_	Root Final Predictio...	0.343798	.	.
status_group		_RMSE_	Root Mean Square...	0.342763	0.343875	0.3457
status_group		_SBC_	Schwarz's Bayesian...	44452.24	.	.
status_group		_SSE_	Sum of Squared Err...	11652.34	2528.659	2556.641
status_group		_SUMW_	Sum of Case Weigh...	99783	21384	21393
status_group		_MISC_	Misclassification Rate	0.249572	0.249299	0.257467

Fig. Fit statistics output of Logistic regression

From the above output of fit statistics, we can see the misclassification rate for training data is 0.249572 so the accuracy is **75.0428%.** The misclassification rate for the validation data set is 0.249299 so the accuracy is **75.0701%.** The misclassification rate for test data is 0.257467 and the accuracy will be **74.2533%.**

Model 3-Neural Networks:

Neural networks are a class of flexible nonlinear regression, discriminant, and data reduction models. The Neural Network node provides a variety of feedforward networks that are commonly called backpropagation. Backpropagation refers to the method for computing the error gradient for a feedforward network, a straightforward application of the chain rule of elementary calculus.

Most connections in a network have an associated numeric value called a weight or parameter estimate. The training methods attempt to minimize the error function by iteratively adjusting the values of the weights. The value produced by the combination function is transformed by an activation function, which involves no weights or other estimated parameters.

The Neural Network node also provides a variety of conventional methods for nonlinear optimization that are usually faster and more reliable than the algorithms from the neural network literature. we also standardize the inputs before running the model to get better results.

<u>Neural network node:</u>

The misclassification rate is 0.2493(24.93%) for training data, 0.2643(26.43%) for validation data and 0.2617(26.17%) for test data. That means the accuracy for validation set is **73.83%**

Results - Node: Neural Network Diagram: waterpump

File Edit View Window

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
status_group		_DFT_	Total Degrees of Freedom	19460	.	.
status_group		_DFE_	Degrees of Freedom for Error	18894	.	.
status_group		_DFM_	Model Degrees of Freedom	566	.	.
status_group		_NW_	Number of Estimated Weights	566	.	.
status_group		_AIC_	Akaike's Information Criterion	13063.57	.	.
status_group		_SBC_	Schwarz's Bayesian Criterion	17521.45	.	.
status_group		_ASE_	Average Squared Error	0.118087	0.123877	0.121892
status_group		_MAX_	Maximum Absolute Error	0.997276	0.996248	0.997586
status_group		_DIV_	Divisor for ASE	29190	6252	6258
status_group		_NOBS_	Sum of Frequencies	9730	2084	2086
status_group		_RASE_	Root Average Squared Error	0.343638	0.351962	0.34913
status_group		_SSE_	Sum of Squared Errors	3446.953	774.481	762.7973
status_group		_SUMW_	Sum of Case Weights Times ...	29190	6252	6258
status_group		_FPE_	Final Prediction Error	0.125162	.	.
status_group		_MSE_	Mean Squared Error	0.121624	0.123877	0.121892
status_group		_RFPE_	Root Final Prediction Error	0.353782	.	.
status_group		_RMSE_	Root Mean Squared Error	0.348747	0.351962	0.34913
status_group		_AVERR_	Average Error Function	0.408755	0.429126	0.423959
status_group		_ERR_	Error Function	11931.57	2682.896	2653.133
status_group		_MISC_	Misclassification Rate	0.249332	0.264395	0.261745
status_group		_WRONG_	Number of Wrong Classificati...	2426	551	546

Fig: Fit statistics of neural networks

Fig: Iteration plot of misclassification rate

We can also observe the iteration plot for misclassification rate. From the above graph, we can observe that the iteration point is selected at point 46 to avoid overfitting of the model.

Auto neural network node:

In the neural network model, it undergoes only a single hidden layer and no iterations. So, we prefer auto-neural networks to select a greater number of iterations and more hidden layers. More the hidden layers, the more accuracy and performance of the model.

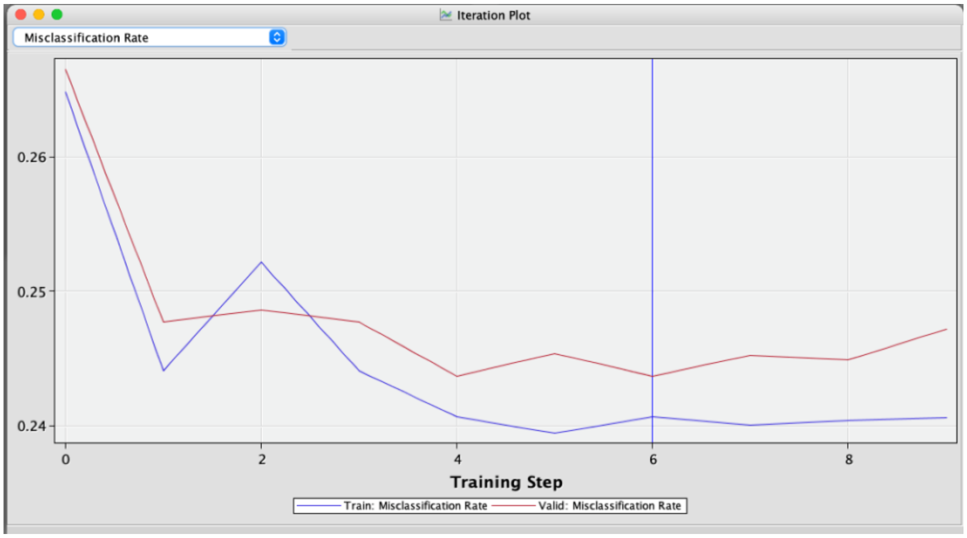

Fig: Iteration plot for Misclassification rate of Auto-Neural node

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
status_group		_DFT_	Total Degrees of Fr...	66522	.	.
status_group		_DFE_	Degrees of Freedo...	64280	.	.
status_group		_DFM_	Model Degrees of F...	2242	.	.
status_group		_NW_	Number of Estimat...	2242	.	.
status_group		_AIC_	Akaike's Informatio...	42872.88	.	.
status_group		_SBC_	Schwarz's Bayesian...	63286.93	.	.
status_group		_ASE_	Average Squared E...	0.112167	0.114643	0.11614
status_group		_MAX_	Maximum Absolut...	0.99949	0.999722	0.999104
status_group		_DIV_	Divisor for ASE	99783	21384	21393
status_group		_NOBS_	Sum of Frequencies	33261	7128	7131
status_group		_RASE_	Root Average Squa...	0.334914	0.33859	0.340793
status_group		_SSE_	Sum of Squared Er...	11192.39	2451.533	2484.586
status_group		_SUMW_	Sum of Case Weigh...	99783	21384	21393
status_group		_FPE_	Final Prediction Err...	0.119992	.	.
status_group		_MSE_	Mean Squared Error	0.11608	0.114643	0.11614
status_group		_RFPE_	Root Final Predicti...	0.346398	.	.
status_group		_RMSE_	Root Mean Square...	0.340705	0.33859	0.340793
status_group		_AVERR_	Average Error Func...	0.384724	0.394215	0.398255
status_group		_ERR_	Error Function	38388.88	8429.904	8519.875
status_group		_MISC_	Misclassification Ra...	0.240642	0.247194	0.251437
status_group		_WRONG_	Number of Wrong ...	8004	1762	1793

Fig. Fit statistics of Auto neural node

The misclassification rate for validation set of Autoneural network is 0.247194(24.71%). So, the accuracy is 1- 0.247194 i.e., **75.28%** for Autoneural Network.

Conclusion:

Model Comparison:

The model comparison node is used to assess which model is best as per validation classification rate.

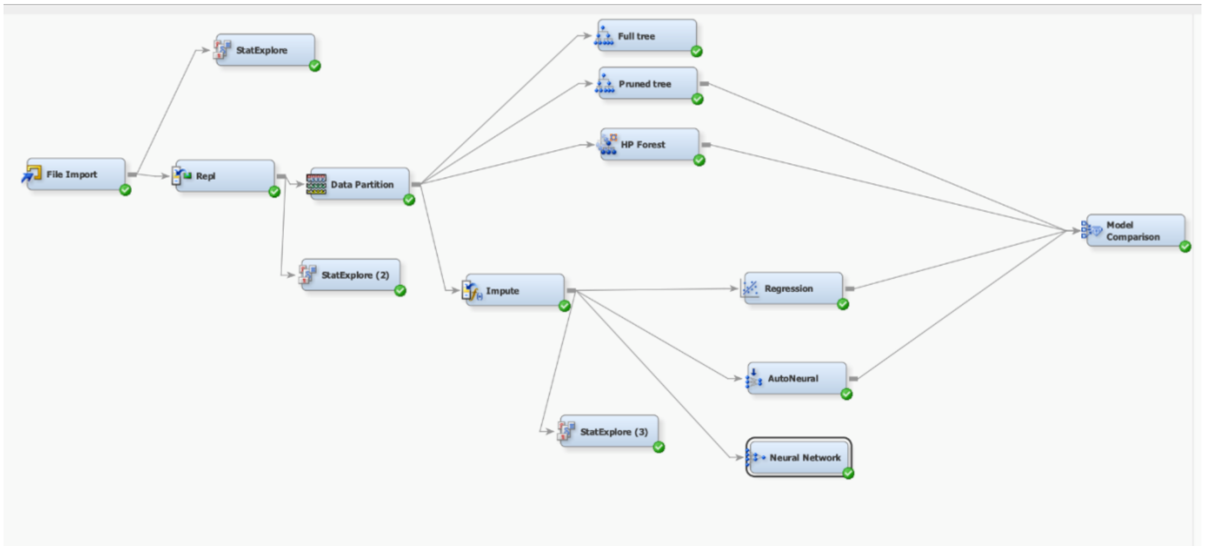

Fig. Diagram

Fit Statistics

Selected Model	Predecess or Node	Model Node	Model Descriptio n	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate	Train: Average Squared Error	Train: Divisor for ASE	Train: Maximum Absolute Error	Train: Sum of Frequencie s	Train: Root Average Squared Error	Train: Sum of Squared Errors	Tra Fre of Cla Cas
Y	HPDMFo...	HPDMFo...	HP Forest	status_g...		0.229517	0.101605	99783	0.992996	33261	0.318755	10138.41	
	AutoNeu...	AutoNeu...	AutoNeu...	status_g...		0.247194	0.112167	99783	0.99949	33261	0.334914	11192.39	
	Reg	Reg	Regressi...	status_g...		0.249299	0.116777	99783	0.999492	33261	0.341726	11652.34	
	Tree	Tree	Pruned t...	status_g...		0.288159	0.132399	99783	0.997572	33261	0.363867	13211.21	

Fig: Fit Statistics of Model Comparison Node

The misclassification rates and accuracy of the models HPForest, Autoneural, Logistic regression and Decision tree are as follows

	HPForest	**AutoNeural**	**Logistic regression**	**Decision tree**
Misclassification rates%	**22.95%**	**24.71%**	**24.92%**	**28.85%**
Accuracy%	**77.05%**	**75.3%.**	**75.08%**	**71.15%**

As we can see from the above statistics, we can see that HPForest accuracy is higher than all other models, and the model comparison node selected HPForest as shown in the above fit statistics figure. Based on the model comparison node, we see that the misclassification rate for HP Forest algorithm is lowest, and Accuracy is higher among Decision/Prune tree, Logistic Regression and Neural Network.

Therefore, HP forest is the best model in our case.