Street Tree Census

Data Mining

Table Of Contents

Street Tree Census

Introduction

TreesCount! 2015 is the third citizen participatory inventory of street trees in New York City. Every ten years, NYC Parks has worked with volunteers to record the location, size, species, and condition of all public curbside trees. Volunteer street tree inventories promote increased awareness of the importance of the urban forest and support municipal urban forest management. New York City's prior street tree inventories in 1995 and 2005 led to advances in customer service, funding for routine street tree pruning, the quantification of the ecological and economic benefits of trees, and a major urban greening campaign called MillionTreesNYC.

From May 2015 to October 2016, over 2,200 citizen mappers spent almost 12,000 hours using high-tech mapping tools with survey wheels, tape measures, and tree identification keys to creating a spatially accurate digital inventory of NYC's street trees. The simple and intuitive mapping method was designed by a local non-profit, TreeKIT. The mapping technique leveraged a municipal geospatial dataset of curb edges to solve urban locational accuracy issues. The data collection method was integrated by the software company Azavea into a web application featuring online training modules, event management, and community engagement tools to provide a seamless volunteer experience. The TreesCount! user experience was designed to scale for thousands of non-technical volunteers to collect standardized and consistent data with minimal training. To inspire public engagement, the web app featured real-time inventory metrics for individuals as well as partner community groups, and a progress map on the status of the data collection campaign. Powered by the public, TreesCount!2015 demonstrated that citizen science can support the collection of high-quality spatial data for municipal urban forest management and ongoing citizen engagement.

Data collectors recorded eleven variables on each treincludinggd biological, structural, and infrastructural information. To learn more about each variable collected in the census, Data source: https://data.cityofnewyork.us/Environment/2015-Street-Tree-Census-Tree-Data/uvpi-gqnh

Objective

Applying the pre-processing methods on different classifying models and comparing the accuracy of each model to present the most accurate model to predict the health of a street trees in New York city.

Description

The dataset has 45 attributes for 684,000 records.
17 of these attributes are numerical, 15 are nominal, 12 of them are binary, and 1 is the date. However, after cleaning and analyzing it, we dropped 24 attributes and due to its large size, we only took 10,000 records.
11 of these attributes are nominal, 10 of them are binary, 2 numerical, and 1 date.

Rdundant attributes

Have not had a redundant attribute.

Dataset Attribute

Attribute	Type	Discerption
tree_id	Nominal	Unique identification number for each tree point.
block_id	Nominal	Identifier linking each tree to the block in the blackface table/shapefile that it is mapped on.
created_at	Date	The data tree points were collected in the census software.
tree_dbh	Numeric	The tree's diameter isis approximately 54" / 137cm above the ground. Data were collected for both living and dead trees; for stumps, use stump_diam
stump_diam	Numeric	Diameter of the stump measured through the center, rounded to the nearest inch
curb_loc	Nominal	Location of tree bed about the curb; trees are either along the curb (OnCurb) or offset from the curb (OffsetFromCurb)
Status	Nominal	Indicates whether the tree is alive, standing dead, or a stump.
Health	Nominal	Indicates the user's perception of tree health.
spc_latin	Nominal	The scientific name for the species, e.g. "Acer rubrum".
steward	Nominal	Indicates the number of unique signs of stewardship observed for this tree. Not recorded for stumps or dead trees.
Guards	Nominal	Indicates whether a guard is present, and if the user felt it was a helpful or harmful guard. Not recorded for dead trees and stumps.
sidewalk	Binary	Indicates whether one of the sidewalk flags immediately adjacent to the tree was damaged, cracked, or lifted. Not recorded for dead trees and stumps
user_type	Nominal	This field describes the category of users who collected this tree point's data.
root_stone	Binary	Indicates the presence of a root problem caused by paving stones in tree bed
root_grate	Binary	Indicates the presence of a root problem caused by metal grates in tree beds.

Attribute	Type	Discerption
root_other	Binary	Indicates the presence of other root problems
trunk_wire	Binary	Indicates the presence of a trunk problem caused by wires or rope wrapped around the trunk
trnk_light	Binary	Indicates the presence of a trunk problem caused by lighting installed on the tree
trnk_other	Binary	Indicates the presence of other trunk problems
brch_light	Binary	Indicates the presence of a branch problem caused by lights (usually string lights) or wires in the branches
brch_shoe	Binary	Indicates the presence of a branch problem caused by sneakers in the branches
brch_other	Binary	Indicates the presence of other branch problems
address	Nominal	Nearest estimated address to the tree
problems	Nominal	Describes potential problems for each tree

Phase 1: Summarizing Properties

Importing dataset

The main tool used in this project is RapidMiner. Which enterprise-ready data science platform amplifies the collective impact of our datasets. First, import the dataset at the extension.CSV by the read CSV operator. Then selected the attribute operator.

Report The Data Summarizing Properties

The Attribute	type	Summarizing Properties	Frequency
tree_id	Nominal	-	10000
block_id	Nominal	-	10000
created_at	Date	May 19, 2015- Sep 29, 2016	499 Days
tree_dbh	Numeric	count 10000.000000 mean 11.213300 std 8.718314 min 0.000000 25% 4.000000 50% 9.000000 75% 16.000000 max 132.000000	(0-13.2) =6,791 (13.3-26.4) =2553 (26.5 – 39.6) =602 (39.7-52.8) =47
stump_diam	Numeric	count 10000.000000 mean 0.447200 std 3.340378 min 0.000000 25% 0.000000 50% 0.000000 75% 0.000000 max 79.000000	(0-7.9) =9,802 (8-15.8) =72 (15.9-23.7) =58 (23.8-31.6) =37 (31.7-39.5) =23
curb_loc	Nominal	count 10000 unique 2 top OnCurb freq 9612	OnCurb 9612 OffsetFromCurb 388

Status	Nominal	count 10000 unique 3 top Alive freq 9515	Alive 9515 Stump 258 Dead 227
Health	Nominal	count 9515 unique 3 top Good freq 7673	Good 7673 Fair 1446 Poor 396
spc_latin	Nominal	count 9514 unique 118 top Platanus x acerifolia freq 1272	Platanus x acerifolia 1272 Gleditsia triacanthos 952 Pyrus calleryana 862 Quercus palustris 794 Acer platanoides 499 ... Crataegus crusgalli 1 Castanea mollissima 1 Pseudotsuga menziesii 1 Acer buergerianum 1 Larix laricina 1
steward	Nominal	count 2371 unique 3 top 1or2 freq 2039	1or2 2039 3or4 302 4orMore 30
Guards	Nominal	count 1113 unique 3 top Helpful freq 757	Helpful 757 Harmful 259 Unsure 97
sidewalk	Binary	unique 2 top NoDamage freq 6770	NoDamage 6770 Damage 2745
user_type	Nominal	unique 3 top TreesCount Staff freq 4340	TreesCount Staff 4340 Volunteer 3213 NYC Parks Staff 2447
root_stone	Binary	count 10000 unique 2 top No freq 7952	No 7952 Yes 2048
root_grate	Binary	count 10000 unique 2 top No freq 9952	No 9952 Yes 48
root_other	Binary	count 10000 unique 2 top No freq 9533	No 9533 Yes 467
trunk_wire	Binary	count 10000 unique 2 top No freq 9789	No 9789 Yes 211

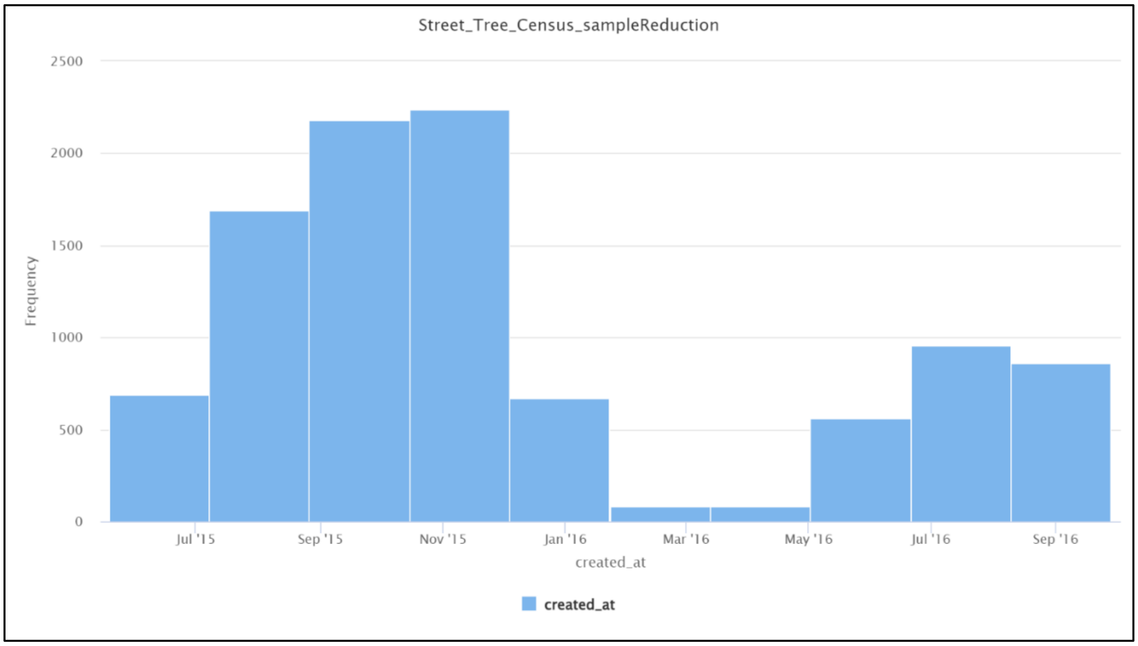
trnk_light	Binary	count 10000 unique 2 top No freq 9985	trnk_light No 9985 Yes 15
trnk_other	Binary	count 10000 unique 2 top No freq 9529	No 9529 Yes 471
brch_light	Binary	count 10000 unique 2 top No freq 9117	No 9117 Yes 883
brch_shoe	Binary	count 10000 unique 2 top No freq 9995	No 9995 Yes 5
brch_other	Binary	count 10000 unique 2 top No freq 9633	No 9633 Yes 367
address	Nominal	-	-
problems	Nominal	count 3301 unique 80 top Stones freq 1384	Stones 1384 BranchLights 388 Stones, BranchLights 279 RootOther 187 TrunkOther 162 ... Stones,WiresRope,TrunkOther,BranchLights,BranchOther 1 Stones,WiresRope,TrunkOther 1 TrunkLights,TrunkOther 1 Sneakers 1 Sneakers,BranchOther 1

Phase 1: Statistics

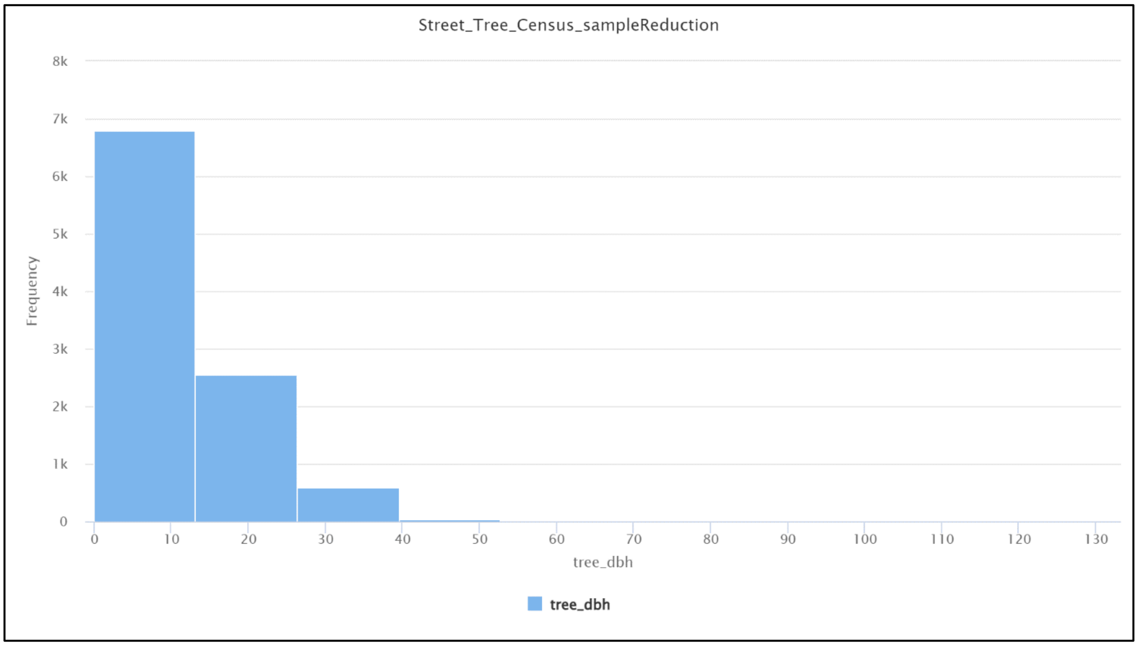
Statistics for TreesCount!

In the histograms below we are presenting the number of Trees in the US, health, length, and status of its alive or dead including the trees' species.
All those attributes are numeric and nominal data. And give you an overview of the status of the patients included in the study.

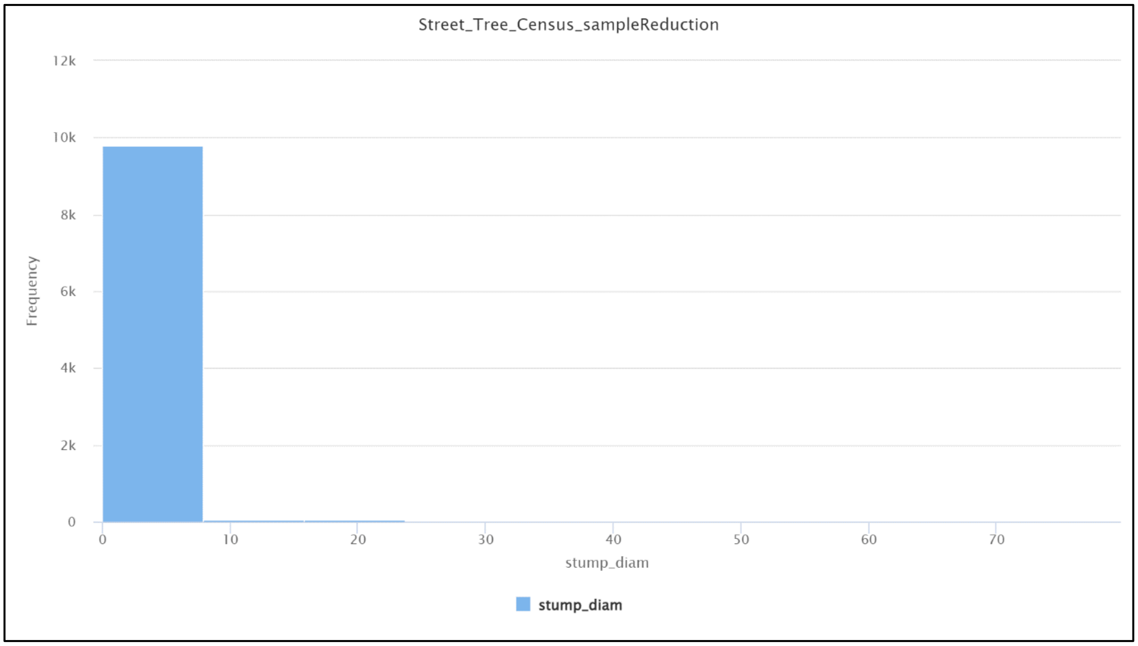
1- Trees created date
In the event trees were mapped on paper and entered into the software at a later time, this date is for the time data entry was completed. The same creation date is applied to all trees on a given block.



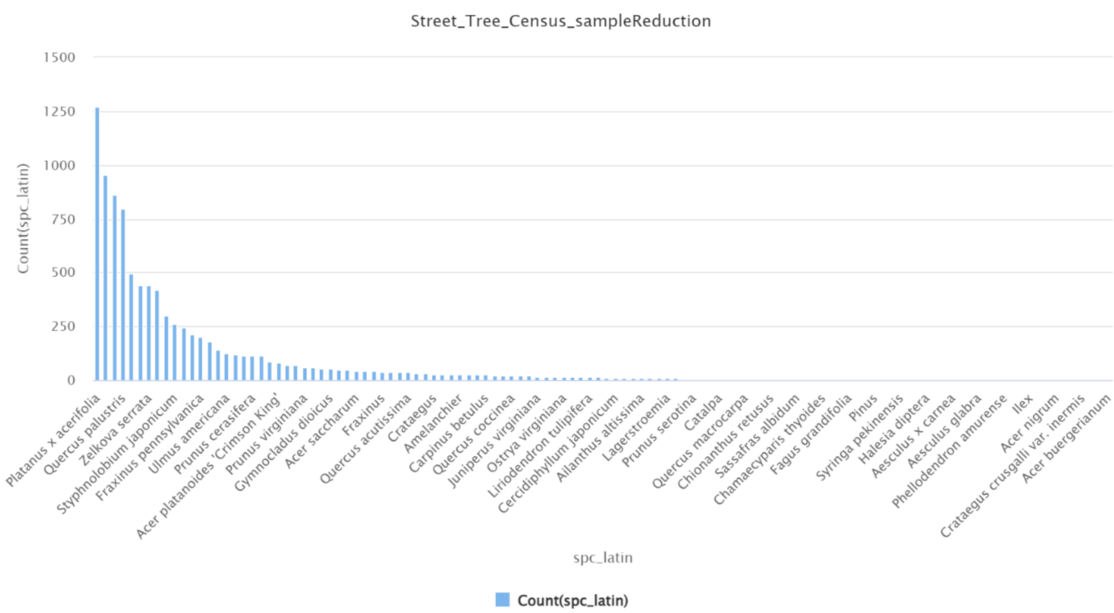
2- Trees length
Because standard measuring tapes are more accessible than forestry-specific measuring tapes designed to measure diameter, users originally measured tree circumference in the field. To better match other forestry datasets, this circumference value was subsequently divided by 3.14159 to transform it into diameter. Both the field measurement and processed value were rounded to the nearest whole inch



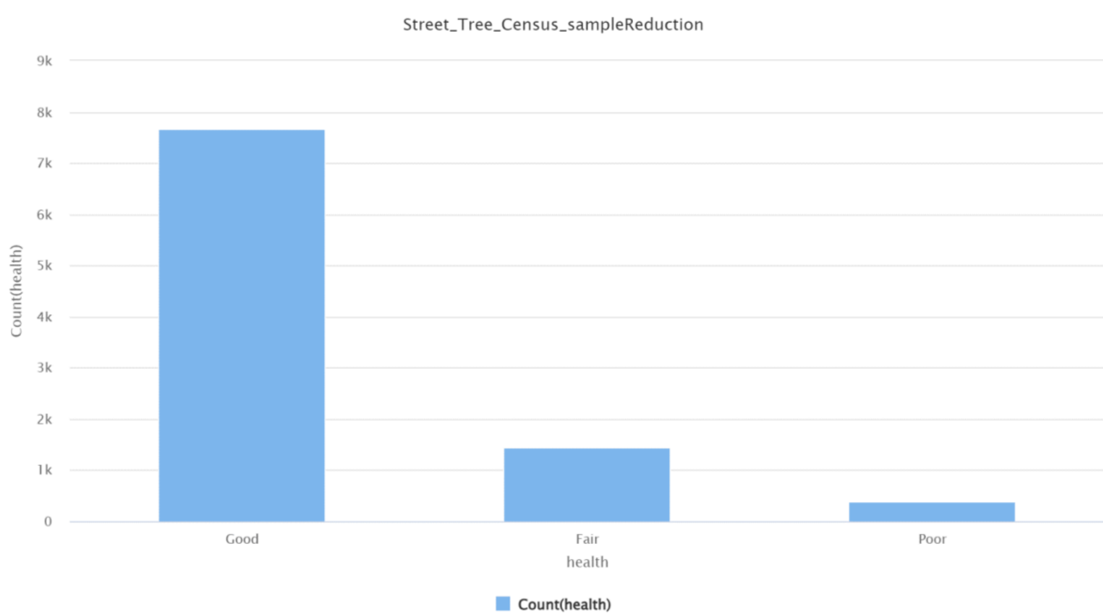
3-Diameter of trees stump
The diameter of the stump was measured through the center, rounded to the nearest inch. This only applies to records where "status" is "Stump." Diameter can be directly measured on stumps since a flat cross-section is accessible.



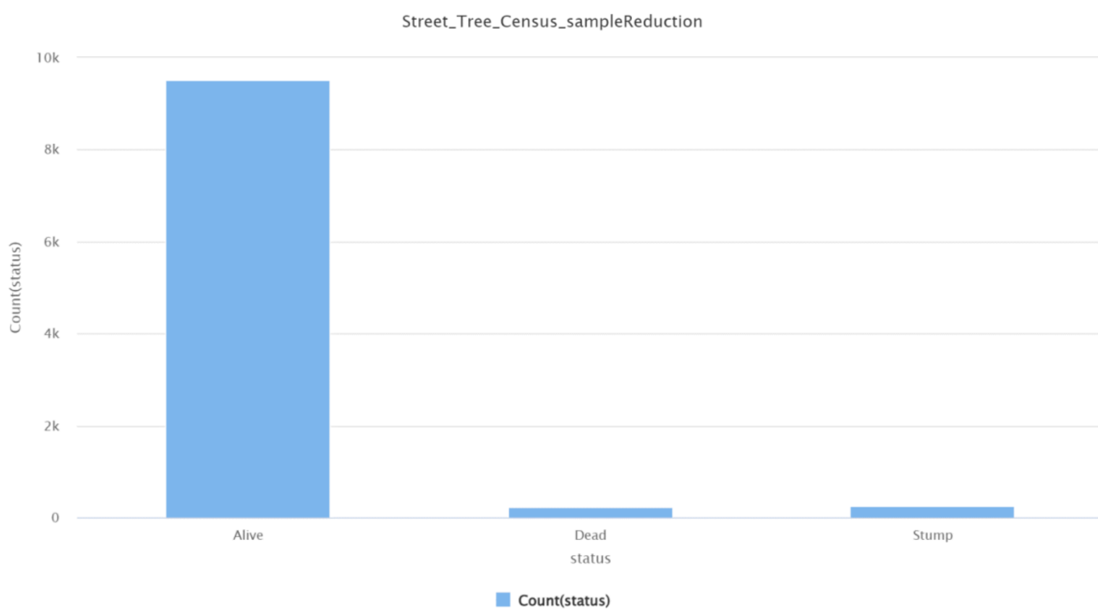
4-Scientific / Latin name of tree species:
The scientific name for the species, e.g. "Acer rubrum", is a list of common tree species found and planted in New York City.



5-Tree Health:
Indicates the user's perception of tree health, Field left blank if the tree is dead or stump.



6-Tree Status:
Indicates whether the tree is alive, standing dead, or a stump.



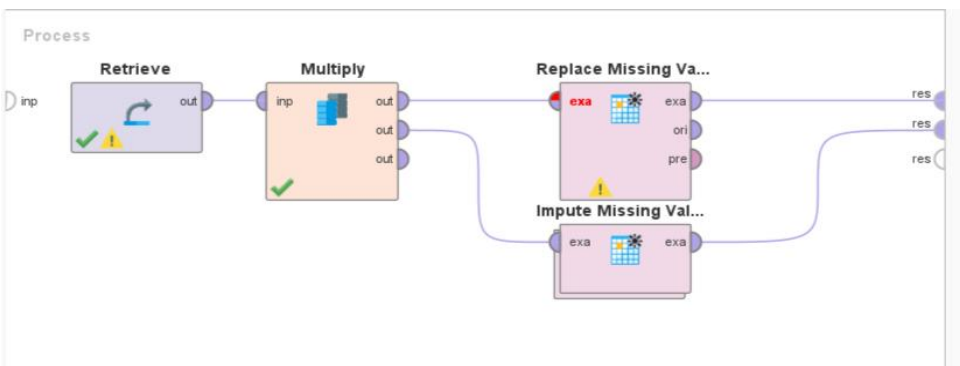
Pre-processing

Handling Missing Values

The dataset contains 25,068 missing values in total. needed to be replaced and handle these missing values before analyzing the data.

In this data mining course, we learned two ways to handle missing values:

- **Replace Missing Values**: This operator allows you to select attributes to make replacements in, and to specify a regular expression. We replaced the missing values with the average of the attribute.
- **Imputing Missing Values:** This operator estimates values for the missing values by applying a model learned for missing values.



We used both the "Replace Missing Values" and "Imputing Missing Values" operators to handle missing values of the numeric attributes.
For the nominal attributes, we handled the missing values in the following way:

- Problems: There are 66.99% of missing values in the "Problems" column. This is because if the tree does not have any problems, then the value is recorded as missing. We handled this problem by using the special value "None".

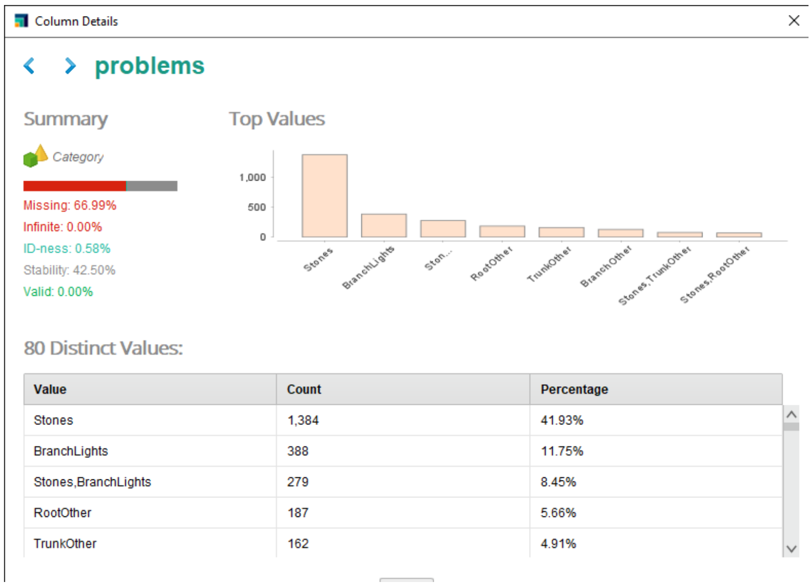
- Sidewalk: There is a 4.85% missing value in the "Sidewalk" column. We solved this by using the strategy.

- Health: There is a 4.85% missing value in the "Health" column. This is because if the tree status is considered dead or stumped, then the health record will be missing. We also found that 200 or more alive trees were recorded as missing. We solved this problem by using the specific value "unknown_healthStatus".

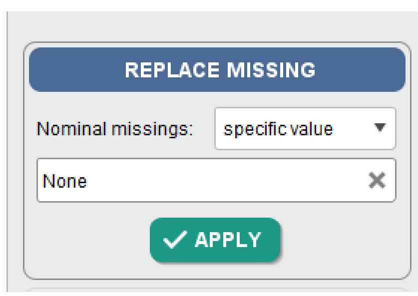
- Spc_latin: There is a 4.86% missing value in the "Spc_latin" column. This is because some tree species are missing. We solved this problem by using the specific value "unknown_spc".

- Steward: There is a 76.29% missing value in the "Steward" column. We solved this by using the "most frequent" strategy.

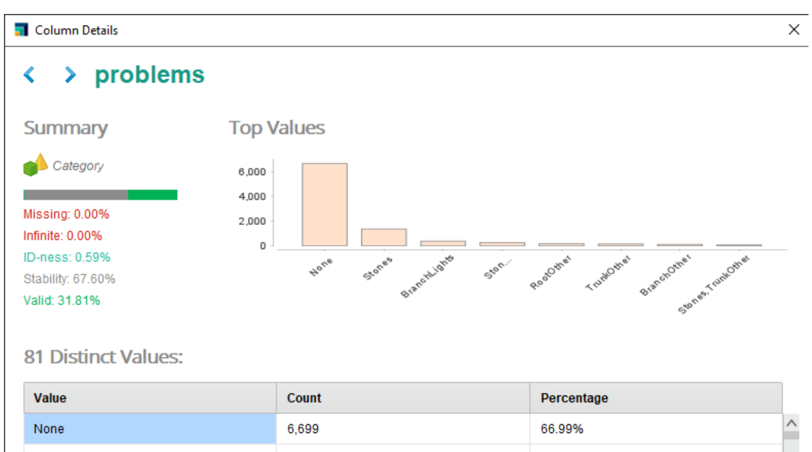
- Problem column:
  - Before cleaning



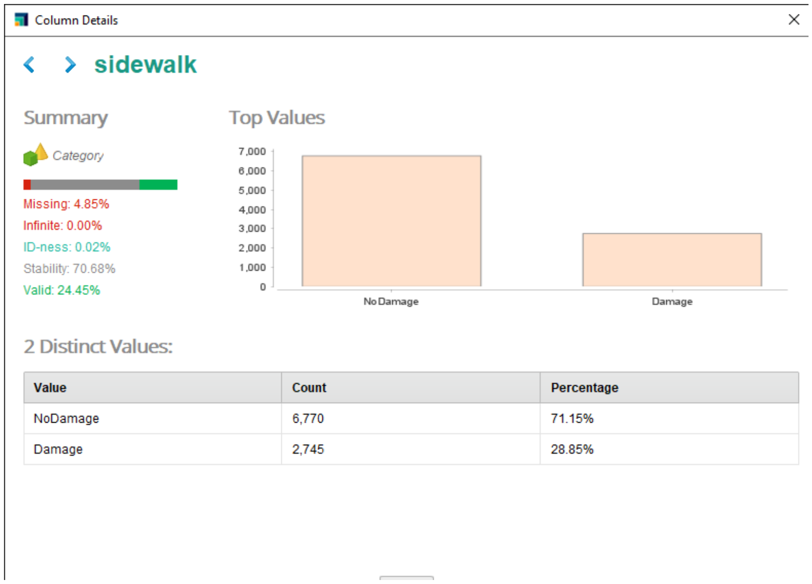
  - Method used to clean



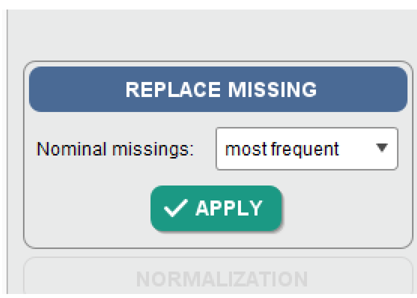
  - After cleaning



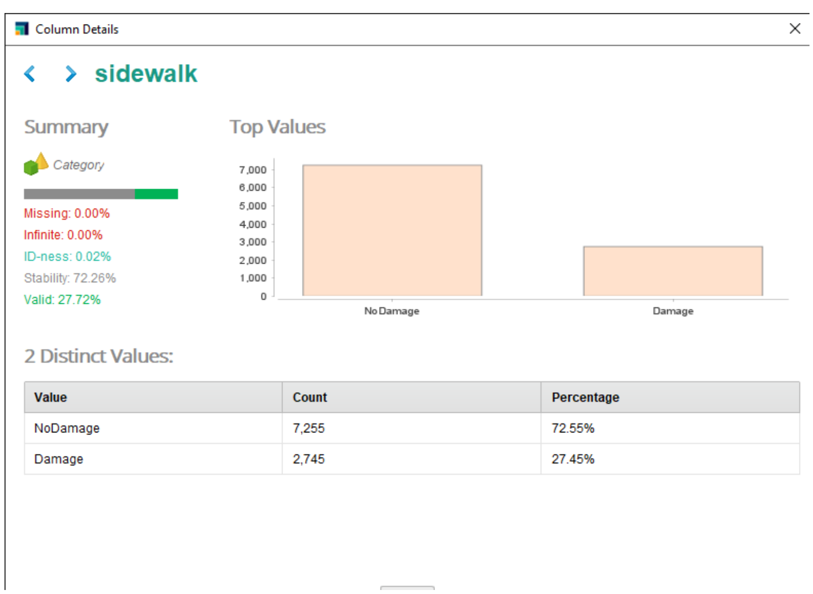
- Sidewalk column:
  - Before cleaning



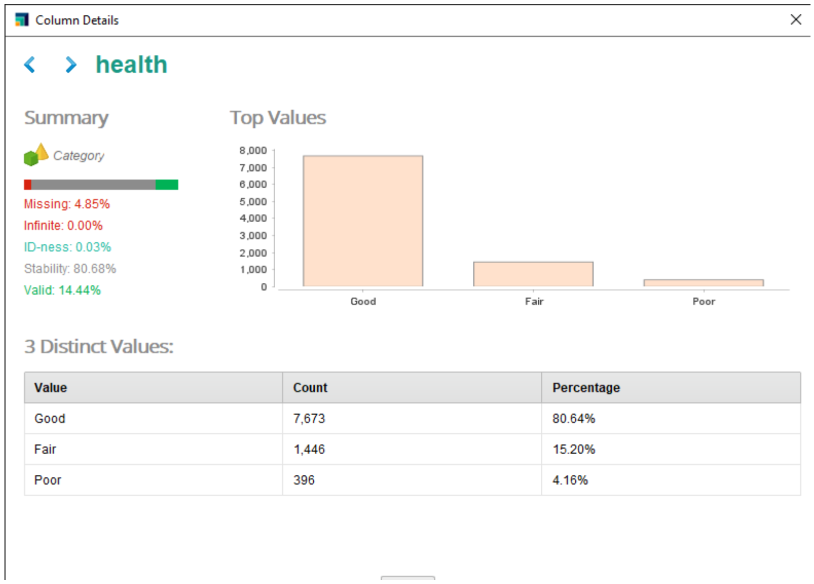
  - Method used to clean



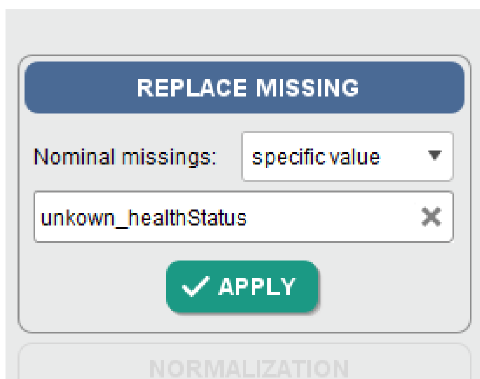
  - After cleaning



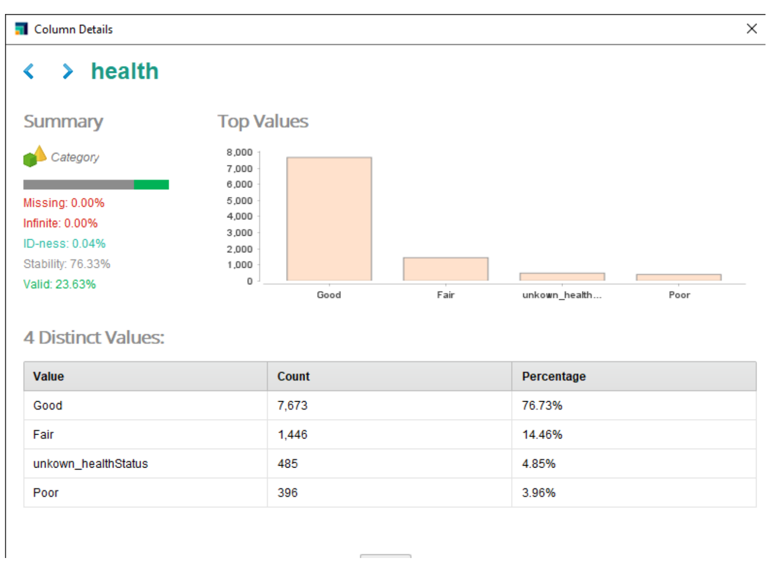
- Health column:
  - Before cleaning



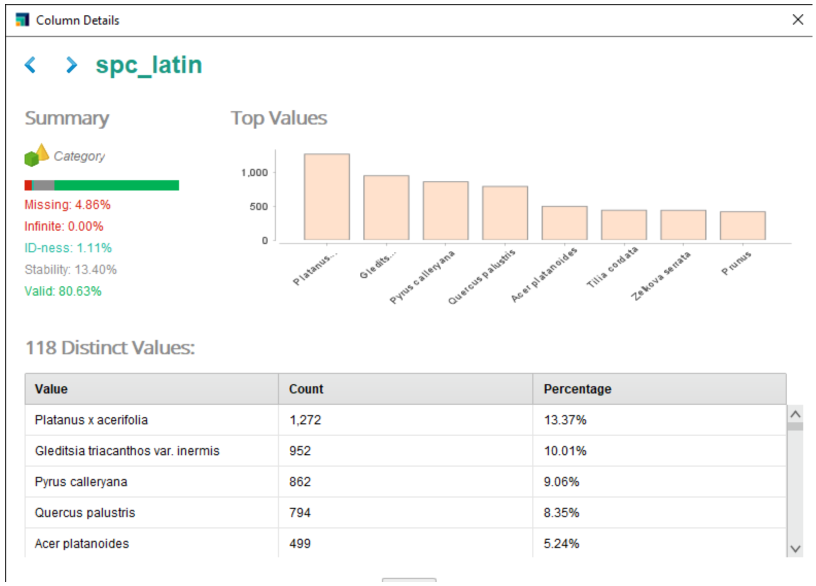
  - Method used to clean



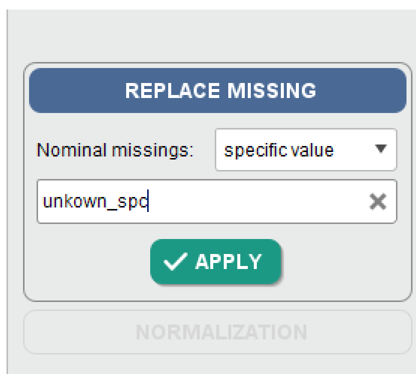
  - After cleaning



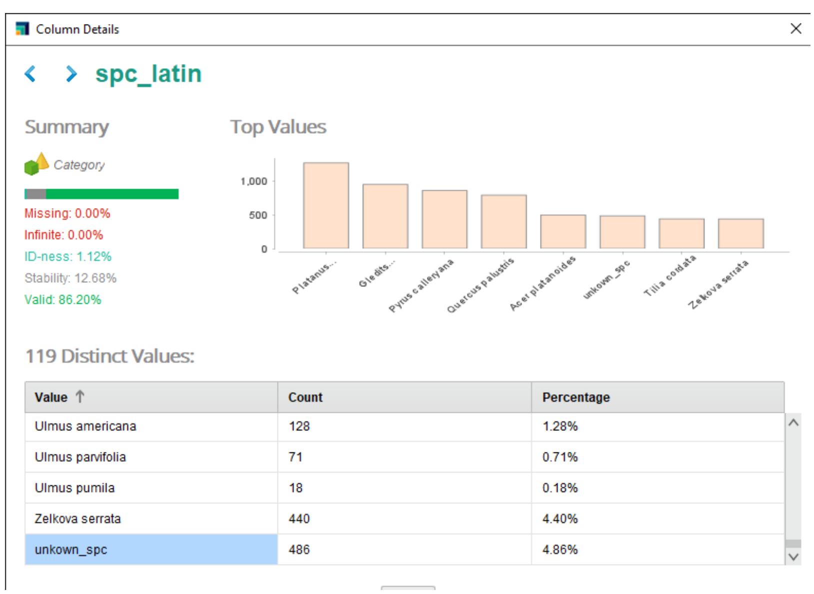
- Spc_latin column:
  - Before cleaning



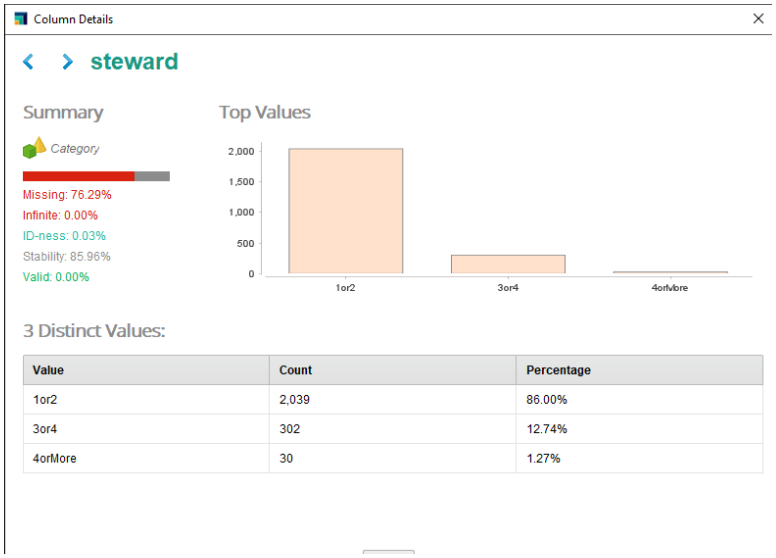
  - Method used to clean



  - After cleaning

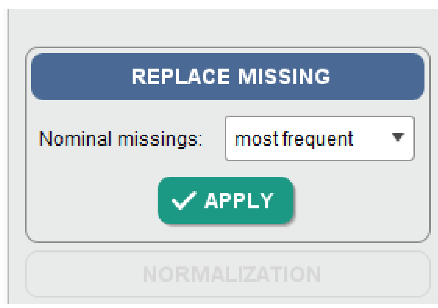


- Steward Column:
  - Before cleaning

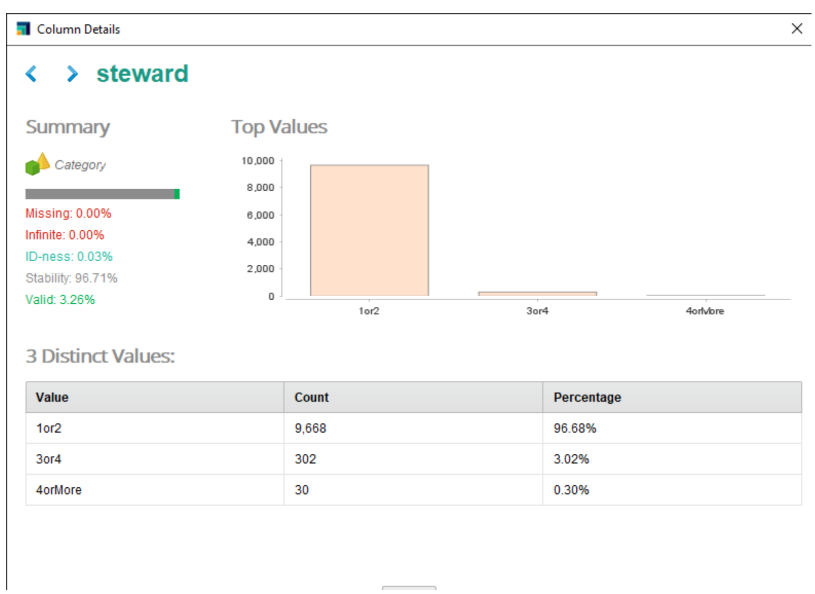


Value	Count	Percentage
1or2	2,039	86.00%
3or4	302	12.74%
4orMore	30	1.27%

  - Method used to clean



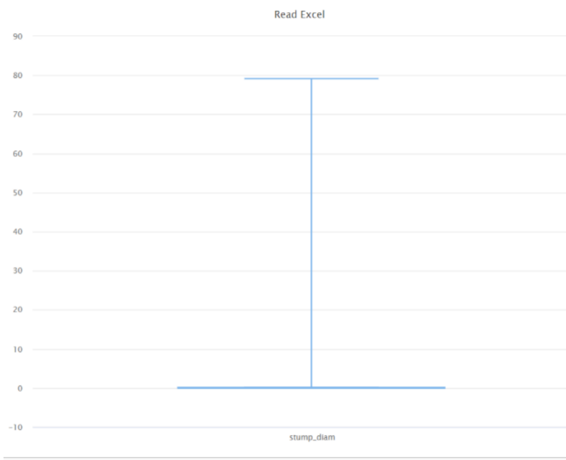
  - After cleaning

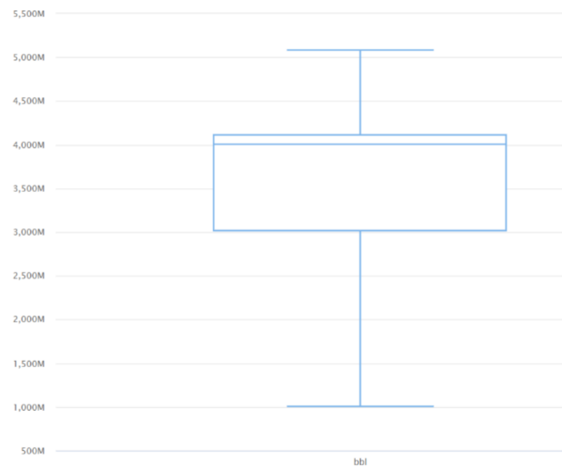


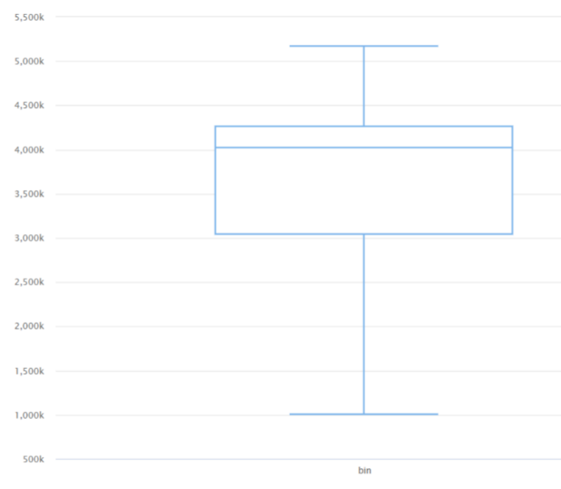
Value	Count	Percentage
1or2	9,668	96.68%
3or4	302	3.02%
4orMore	30	0.30%

Detecting Outliers

An outlier is a value that lies in the extremes of a data series and thus can affect the overall observation. Outliers are also termed extremes because they lie on either end of the data.
To find the outliers, we used the boxplot. As shown by the plot, the attribute does not have any outliers that needed to be eliminated.

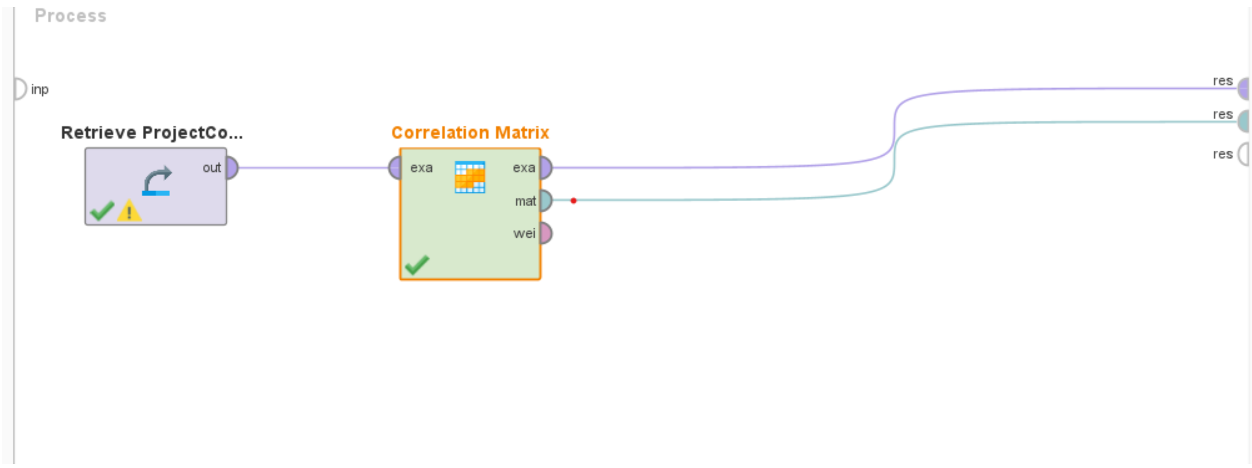


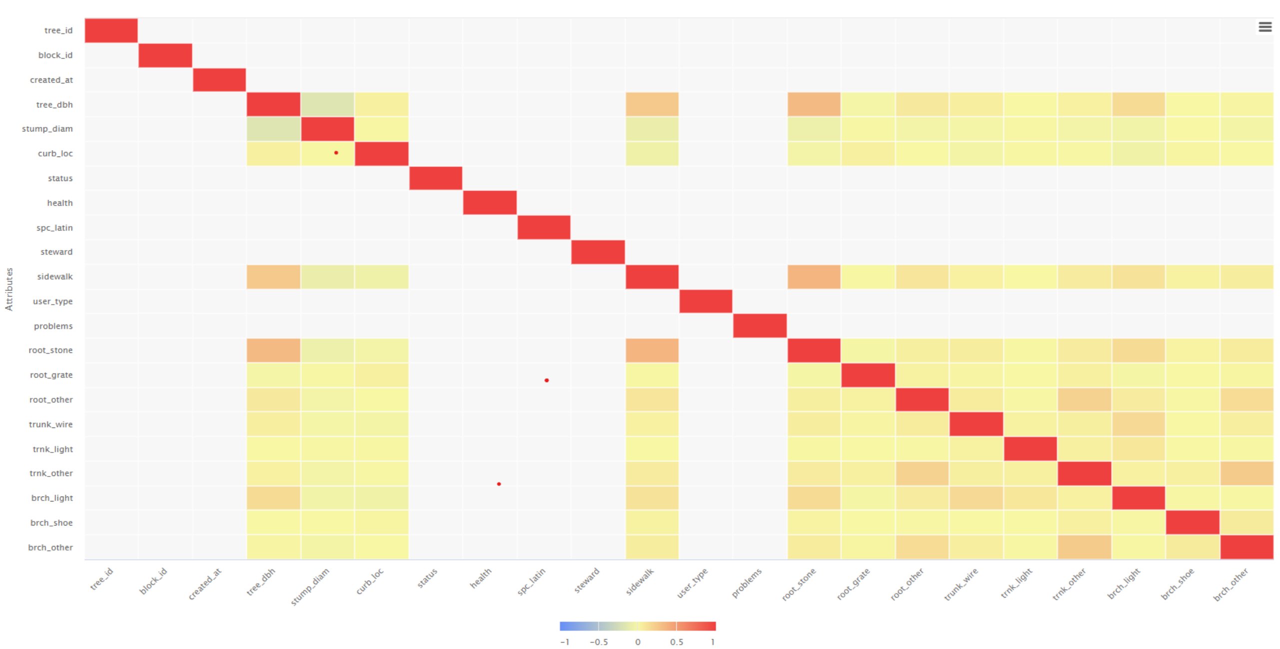




Correlation Matrix

The following correlation matrix will help us determine the strength of the relationships between the traits. We will discuss the correlation coefficients between the variables. The matrix displays different forms of correlations. The first table shows the most highly associated quality, and the second table shows the most adversely associated quality.





Corrélation Matrix Coefficient - Positive Values :

As we can see, there is a potential positive correlation between the tree_dbh and root_stone attributes, with a coefficient of 0.335.

Tree_dbh: "Diameter at breast height of tree"
Root_stone: "Root problems caused by paving stones in the tree bed."

It is more accurate to say that there is a potential positive correlation between tree height and the likelihood of encountering root-stone problems. As trees grow taller, their roots naturally extend further into the soil in search of water and nutrients. This increased exploration can lead to a higher chance of encountering stones or other underground obstacles. When tree roots encounter stones, they may face challenges such as obstruction, distortion, and limited nutrient absorption.

Our research has found that the height of a tree can impact root-stone problems in several ways:

- **Imbalanced growth:** When roots encounter stones, they may have to navigate around or grow over them. This can lead to an imbalanced root system, with roots growing unevenly or in irregular patterns. Such imbalances can affect the overall stability and health of the tree.
- **Reduced nutrient availability:** Stones in the soil can limit the ability of tree roots to absorb water and nutrients. As roots come into contact with stones, their capacity to access the surrounding soil for essential resources may be compromised. This can result in nutrient deficiencies and hinder the tree's overall growth and vigor.

The tree's height and root-stone problems are important factors for tree health and stability. Stones in the soil pose challenges, but proactive measures can mitigate them. With careful site selection, soil preparation, and maintenance, we can ensure tree longevity and vitality. This enables flourishing roots and the enduring benefits of trees for future generations.

First Att...	Second ...	Cor... ↓
tree_dbh	root_stone	0.335
postcode	bin	0.328
postcode	communit...	0.313
postcode	boro_ct	0.311
postcode	bbl	0.301
postcode	borocode	0.296
st_senate	latitude	0.260
st_senate	y_sp	0.259
longitude	census tr...	0.214
x_sp	census tr...	0.214
st_assem	latitude	0.204
st_assem	y_sp	0.204
root_other	trnk_other	0.201
boro_ct	census tr...	0.179
census tr...	bin	0.177
communit...	census tr...	0.172
census tr...	bbl	0.170
trunk_wire	brch_light	0.165

Attributes	tree_dbh	root_st...
tree_dbh	1	0.335
root_stone	0.335	1

Corrélation Matrix Coefficient - Négative Values :

We can see here that there is a negative correlation between the attributes of Sidewalk and root_stone, with a coefficient of -0.351.

First Att...	Second ...	Cor...
sidewalk	root_stone	-0.351
trnk_other	brch_other	-0.240
tree_dbh	sidewalk	-0.232
st_assem	census tr...	-0.232
st_senate	census tr...	-0.218
borocode	x_sp	-0.201
borocode	longitude	-0.201
x_sp	bbl	-0.189
longitude	bbl	-0.189
communit...	x_sp	-0.182
communit...	longitude	-0.182
boro_ct	x_sp	-0.179
boro_ct	longitude	-0.179
cncldist	st_senate	-0.177
st_senate	council di...	-0.173
tree_dbh	stump_di...	-0.172
x_sp	bin	-0.164

Sidewalk damage immediately adjacent to trees is often caused by root problems. As tree roots grow, they can encounter stones and other underground obstacles. This can cause the roots to shift and exert pressure on the sidewalk, leading to cracking, lifting, or upheaval.

The relationship between sidewalk damage and root problems is generally considered a negative correlation. This means that as root problems increase, the likelihood of sidewalk damage also tends to increase.
There are a few things that can be done to prevent sidewalk damage caused by root problems. One is to avoid planting trees too close to sidewalks. Another is to install root barriers around trees. Root barriers are made of materials that roots cannot penetrate, such as plastic or metal.

If sidewalk damage has already occurred, it can be repaired by a professional. Repairs typically involve removing the damaged concrete and replacing it with new concrete.

Attributes	sidewalk	root_st...
sidewalk	1	-0.351
root_stone	-0.351	1

Stage 2

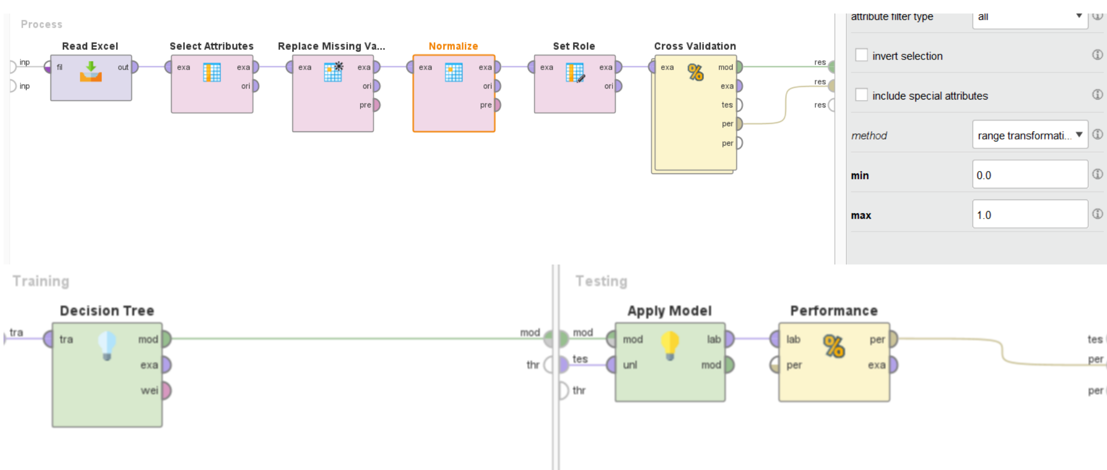
Decision Tree Model Construction

The current objective is to obtain a very accurate model. This will be done using a Decision Tree classifier, which identifies a small number of predetermined classes.

A. Normalization

1. Decision Tree - Normalized Range

The normalized range operator is used to set the value of each feature in the training split data set to a range of [0, 1]. This results in an accuracy of 81.57%.



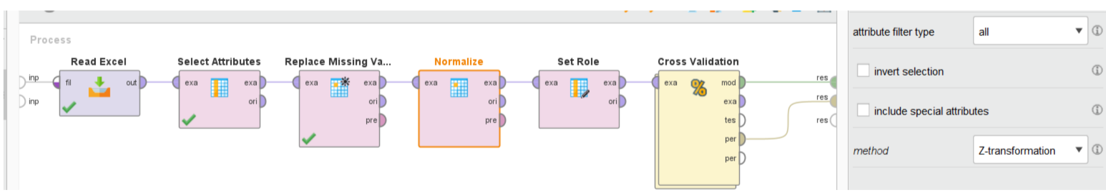
accuracy: 81.57% +/- 0.09% (micro average: 81.57%)

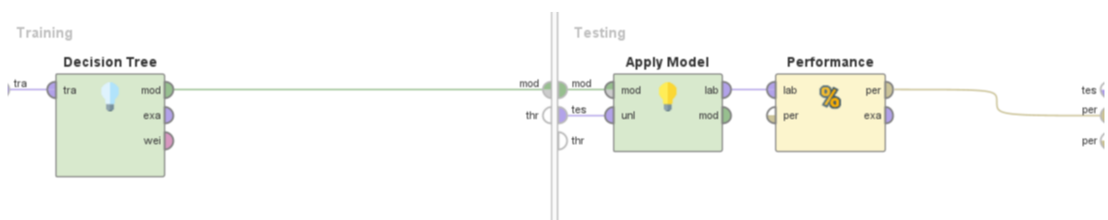
	true Good	true Fair	true unkown_healthStatus	true Poor	class precision
pred. Good	7667	1441	0	394	80.69%
pred. Fair	4	5	0	2	45.45%
pred. unkown_healthStatus	0	0	485	0	100.00%
pred. Poor	2	0	0	0	0.00%
class recall	99.92%	0.35%	100.00%	0.00%	

PerformanceVector

```
PerformanceVector:
accuracy: 81.57% +/- 0.09% (micro average: 81.57%)
ConfusionMatrix:
True:    Good    Fair    unkown_healthStatus    Poor
Good:    7667    1441    0       394
Fair:    4       5       0       2
unkown_healthStatus:     0       0       485     0
Poor:    2       0       0       0
```

2. **Decision Tree - Normalize Z-Score**
   The normalization Z-score operator can help us remove outliers that could have a negative impact on our results. It does this by determining and changing the number so that it sits in the range of [-1, 0, 1]. As shown, our accuracy remains at 81.57%, which is not greater than the accuracy of the normalization range method.





accuracy: 81.57% +/- 0.09% (micro average: 81.57%)

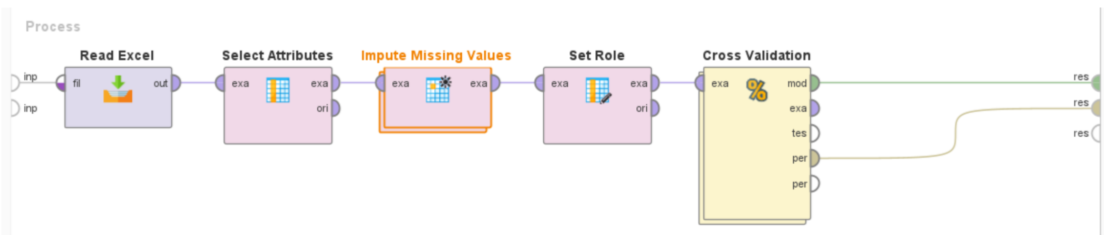
	true Good	true Fair	true unkown_healthStatus	true Poor	class precision
pred. Good	7667	1441	0	394	80.69%
pred. Fair	4	5	0	2	45.45%
pred. unkown_healthStatus	0	0	485	0	100.00%
pred. Poor	2	0	0	0	0.00%
class recall	99.92%	0.35%	100.00%	0.00%	

PerformanceVector

```
PerformanceVector:
accuracy: 81.57% +/- 0.09% (micro average: 81.57%)
ConfusionMatrix:
True:    Good    Fair    unkown_healthStatus     Poor
Good:    7667    1441    0       394
Fair:    4       5       0       2
unkown_healthStatus:     0       0       485     0
Poor:    2       0       0       0
```

B. Decision Tree Imputation of Missing Values

The k-Nearest Neighbor (KNN) algorithm was used to impute numerical missing values. The KNN value used to fill in all missing values was 5, and the accuracy was 81.57%.



accuracy: 81.57% +/- 0.09% (micro average: 81.57%)

	true Good	true Fair	true unkown_healthStatus	true Poor	class precision
pred. Good	7667	1441	0	394	80.69%
pred. Fair	4	5	0	2	45.45%
pred. unkown_healthStatus	0	0	485	0	100.00%
pred. Poor	2	0	0	0	0.00%
class recall	99.92%	0.35%	100.00%	0.00%	

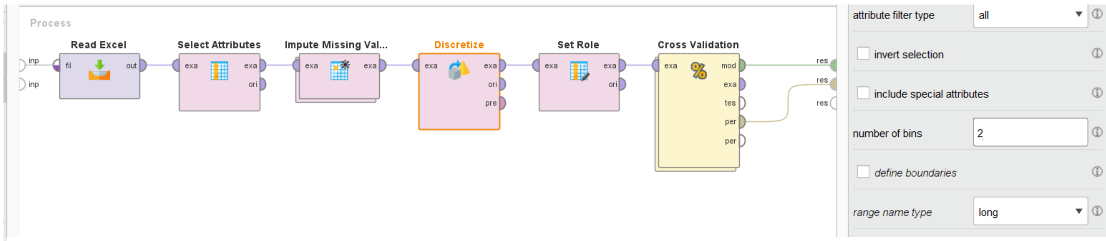
PerformanceVector

```
PerformanceVector:
accuracy: 81.57% +/- 0.09% (micro average: 81.57%)
ConfusionMatrix:
True:    Good    Fair    unkown_healthStatus     Poor
Good:    7667    1441    0       394
Fair:    4       5       0       2
unkown_healthStatus:     0       0       485     0
Poor:    2       0       0       0
```

C. Discretize

1. Impute with 2 bins

The Discretize operator divides a continuous property's range into intervals, which reduces the amount of data and increases data efficiency. By including two bins in this pre-processing step, the accuracy has increased to 81.57%.



accuracy: 81.57% +/- 0.24% (micro average: 81.57%)

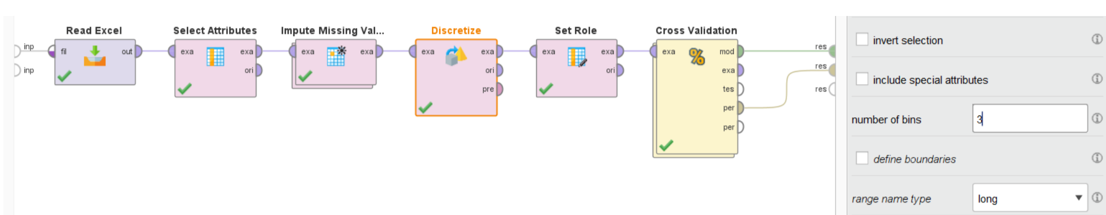
	true Good	true Fair	true unkown_healthStatus	true Poor	class precision
pred. Good	7636	1409	0	372	81.09%
pred. Fair	36	36	0	24	37.50%
pred. unkown_healthStatus	0	0	485	0	100.00%
pred. Poor	1	1	0	0	0.00%
class recall	99.52%	2.49%	100.00%	0.00%	

PerformanceVector

```
PerformanceVector:
accuracy: 81.57% +/- 0.24% (micro average: 81.57%)
ConfusionMatrix:
True:   Good    Fair    unkown_healthStatus     Poor
Good:   7636    1409    0       372
Fair:   36      36      0       24
unkown_healthStatus:    0       0       485     0
Poor:   1       1       0       0
```

2. impute with 3 bin

The accuracy of this approach is 81.59%.



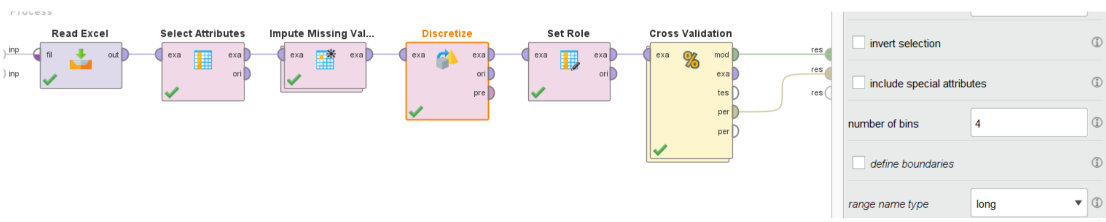
accuracy: 81.59% +/- 0.31% (micro average: 81.59%)

	true Good	true Fair	true unkown_healthStatus	true Poor	class precision
pred. Good	7640	1412	0	376	81.04%
pred. Fair	32	34	0	20	39.53%
pred. unkown_healthStatus	0	0	485	0	100.00%
pred. Poor	1	0	0	0	0.00%
class recall	99.57%	2.35%	100.00%	0.00%	

PerformanceVector

```
PerformanceVector:
accuracy: 81.59% +/- 0.31% (micro average: 81.59%)
ConfusionMatrix:
True:   Good    Fair    unkown_healthStatus     Poor
Good:   7640    1412    0       376
Fair:   32      34      0       20
unkown_healthStatus:    0       0       485     0
Poor:   1       0       0       0
```

3. **impute with 4 bins.**
   The accuracy for 4 bins was 81.54%, which was less than the accuracy for 3 bins.



accuracy: 81.54% +/- 0.24% (micro average: 81.54%)

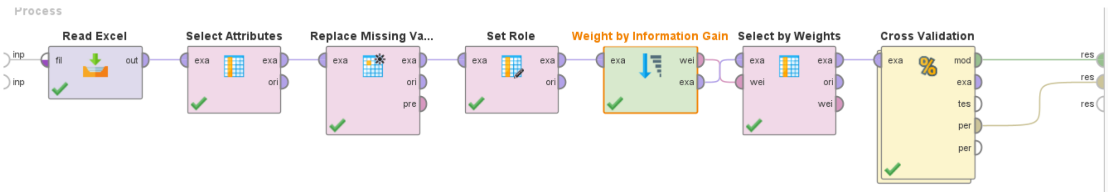
	true Good	true Fair	true unkown_healthStatus	true Poor	class precision
pred. Good	7641	1414	0	377	81.01%
pred. Fair	29	25	0	16	35.71%
pred. unkown_healthStatus	0	0	485	0	100.00%
pred. Poor	3	7	0	3	23.08%
class recall	99.58%	1.73%	100.00%	0.76%	

PerformanceVector

```
PerformanceVector:
accuracy: 81.54% +/- 0.24% (micro average: 81.54%)
ConfusionMatrix:
True:   Good    Fair    unkown_healthStatus     Poor
Good:   7641    1414    0       377
Fair:   29      25      0       16
unkown_healthStatus:    0       0       485     0
Poor:   3       7       0       3
```

D. Reducing Dimensionality by Information Gain

As shown, the accuracy of the model was 76.73%. This is the worst result of all the models tested.



accuracy: 76.73% +/- 0.05% (micro average: 76.73%)

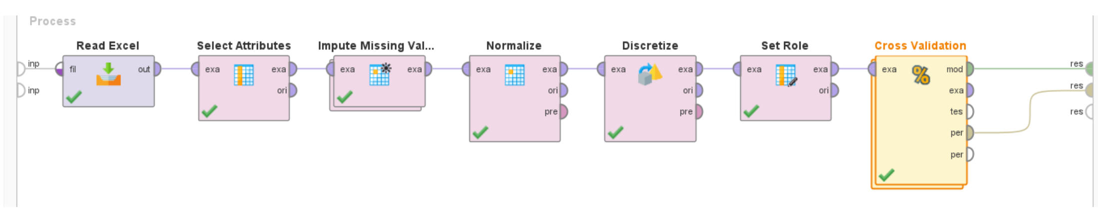
	true Good	true Fair	true unkown_healthStatus	true Poor	class precision
pred. Good	7673	1446	485	396	76.73%
pred. Fair	0	0	0	0	0.00%
pred. unkown_healthStatus	0	0	0	0	0.00%
pred. Poor	0	0	0	0	0.00%
class recall	100.00%	0.00%	0.00%	0.00%	

PerformanceVector

```
PerformanceVector:
accuracy: 76.73% +/- 0.05% (micro average: 76.73%)
ConfusionMatrix:
True:   Good    Fair    unkown_healthStatus     Poor
Good:   7673    1446    485     396
Fair:   0       0       0       0
unkown_healthStatus:    0       0       0       0
Poor:   0       0       0       0
```

E. Exclude Dimensionality Reduction

Here, We excluded the reducing operator phase in this process to make it certain that may the accuracy get better without reducing any variables. The ACU is 81.57 witch is good.



accuracy: 81.57% +/- 0.24% (micro average: 81.57%)

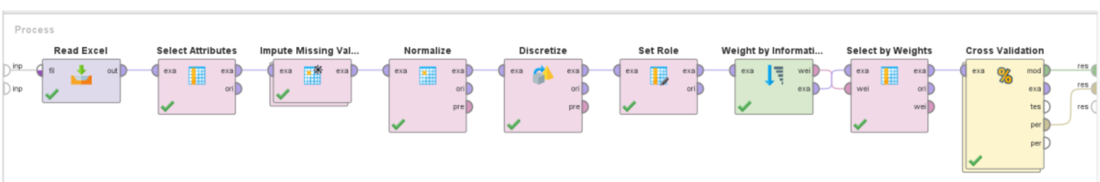
	true Good	true Fair	true unkown_healthStatus	true Poor	class precision
pred. Good	7636	1409	0	372	81.09%
pred. Fair	36	36	0	24	37.50%
pred. unkown_healthStatus	0	0	485	0	100.00%
pred. Poor	1	1	0	0	0.00%
class recall	99.52%	2.49%	100.00%	0.00%	

PerformanceVector

```
PerformanceVector:
accuracy: 81.57% +/- 0.24% (micro average: 81.57%)
ConfusionMatrix:
True:   Good    Fair    unkown_healthStatus     Poor
Good:   7636    1409    0       372
Fair:   36      36      0       24
unkown_healthStatus:    0       0       485     0
Poor:   1       1       0       0
```

F. Including Dimensionality reduction

As shown, the accuracy of the model was 76.73%. This is one of two worst result of all the models tested.



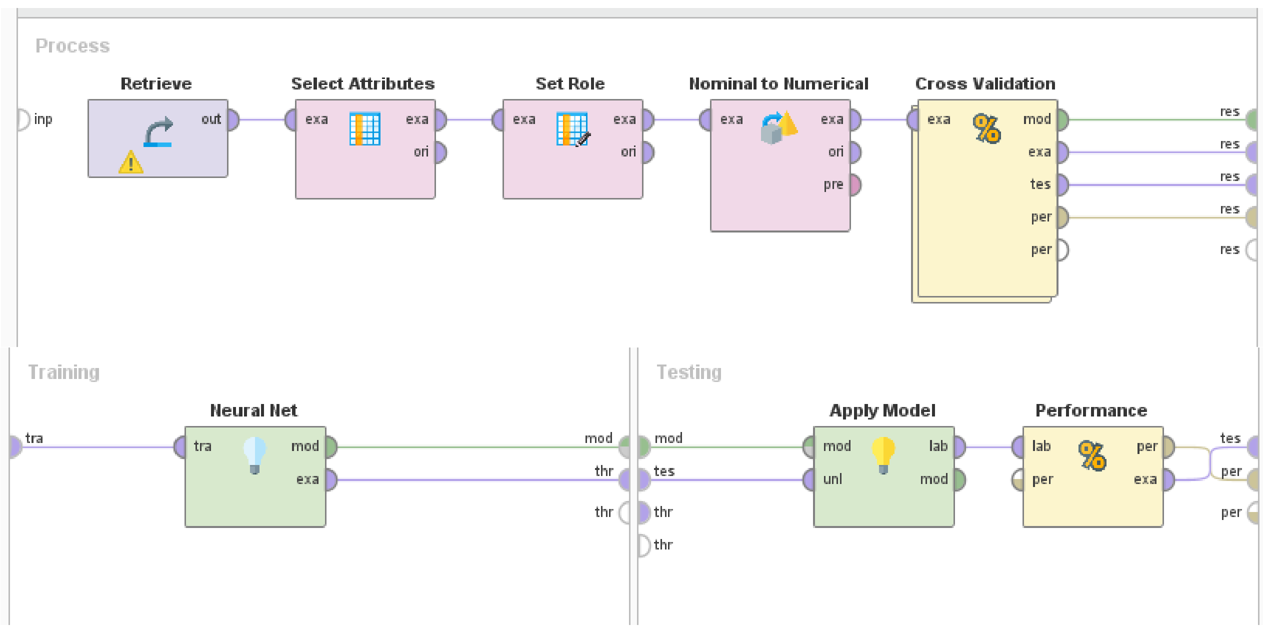
accuracy: 76.73% +/- 0.05% (micro average: 76.73%)

	true Good	true Fair	true unkown_healthStatus	true Poor	class precision
pred. Good	7673	1446	485	396	76.73%
pred. Fair	0	0	0	0	0.00%
pred. unkown_healthStatus	0	0	0	0	0.00%
pred. Poor	0	0	0	0	0.00%
class recall	100.00%	0.00%	0.00%	0.00%	

PerformanceVector

```
PerformanceVector:
accuracy: 76.73% +/- 0.05% (micro average: 76.73%)
ConfusionMatrix:
True:    Good    Fair    unkown_healthStatus    Poor
Good:    7673    1446    485     396
Fair:    0       0       0       0
unkown_healthStatus:     0       0       0       0
Poor:    0       0       0       0
```

Neural Network Classifier



The performance table is displayed in the result window:

accuracy: 81.58% +/- 0.11% (micro average: 81.58%)

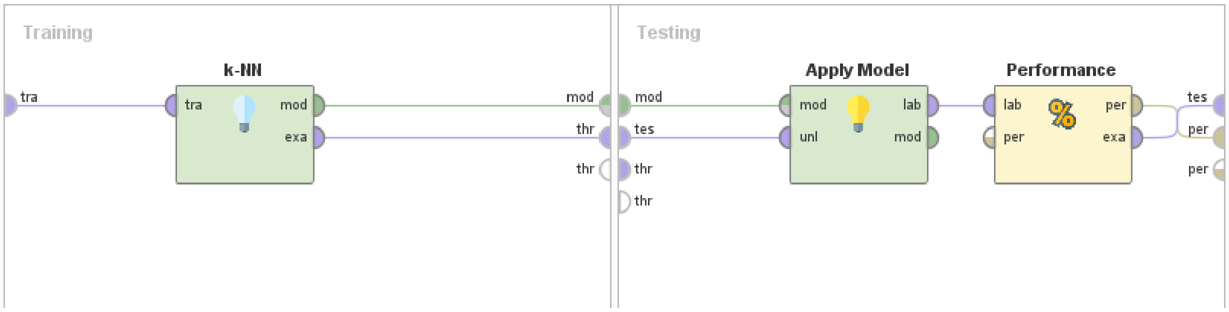
	true Good	true Fair	true unkown_healthStatus	true Poor	class precision
pred. Good	7672	1445	0	396	80.65%
pred. Fair	0	1	0	0	100.00%
pred. unkown_healthStatus	1	0	485	0	99.79%
pred. Poor	0	0	0	0	0.00%
class recall	99.99%	0.07%	100.00%	0.00%	

PerformanceVector

```
PerformanceVector:
accuracy: 81.58% +/- 0.11% (micro average: 81.58%)
ConfusionMatrix:
True:   Good    Fair    unkown_healthStatus     Poor
Good:   7672    1445    0       396
Fair:   0       1       0       0
unkown_healthStatus:    1       0       485     0
Poor:   0       0       0       0
```

The accuracy figure is 81.57% with recalls 99% & 100% . That means our tree health classifier is very effective at determining the variables that have an impact on the survival of healthy trees.
In another experiment, we added more hidden layers, but each time the accuracy decreased.

k-nearest neighbors (KNN)



The performance table is displayed in the result window:

accuracy: 76.77% +/- 0.84% (micro average: 76.77%)

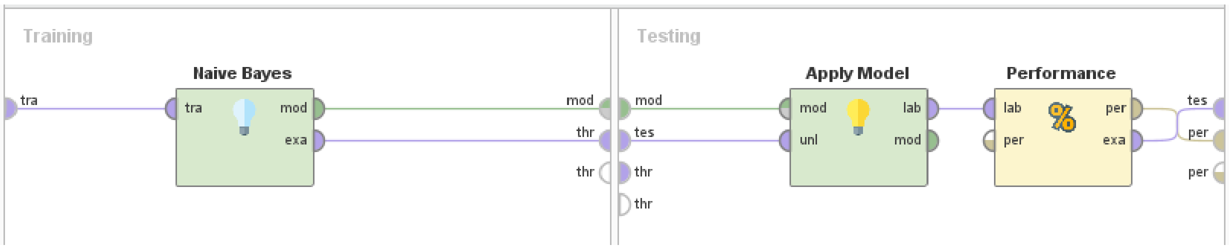
	true Good	true Fair	true unkown_healt...	true Poor	class precision
pred. Good	7381	1347	278	372	78.71%
pred. Fair	273	94	5	23	23.80%
pred. unkown_heal...	2	0	201	0	99.01%
pred. Poor	17	5	1	1	4.17%
class recall	96.19%	6.50%	41.44%	0.25%	

PerformanceVector

```
PerformanceVector:
accuracy: 76.77% +/- 0.84% (micro average: 76.77%)
ConfusionMatrix:
True:   Good    Fair    unkown_healthStatus     Poor
Good:   7381    1347    278     372
Fair:   273     94      5       23
unkown_healthStatus:    2       0       201     0
Poor:   17      5       1       1
```

Algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. Therefore, it is make sense that the accuracy is 76.77% and that means it's as good as neural network.

Naive Bayes



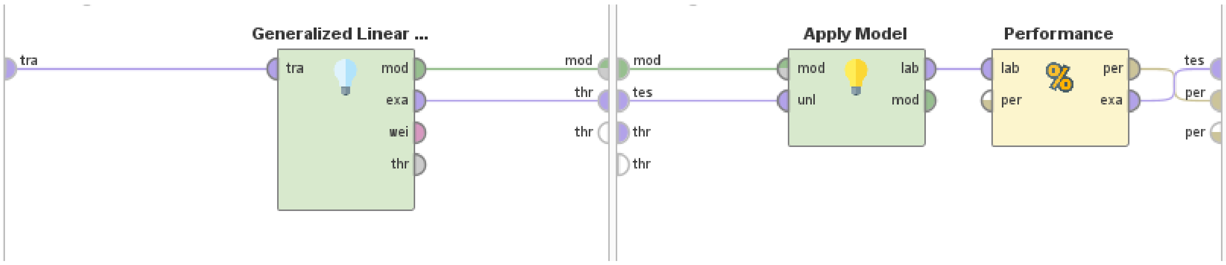
The performance table is displayed in the result window:

Accuracy is 81.6%:

	true Good	true Fair	true unkown_heal...	true Poor	class precision
pred. Good	2193	409	1	117	80.62%
pred. Fair	0	0	0	0	0.00%
pred. unkown_he...	0	0	137	0	100.00%
pred. Poor	0	0	0	0	0.00%
class recall	100.00%	0.00%	99.28%	0.00%	

We got a high accuracy here, which is 81.6%, but view to other models, Neural Network is better.

Generalized Linear Model



The performance table is displayed in the result window:

Accuracy is 78.7%:

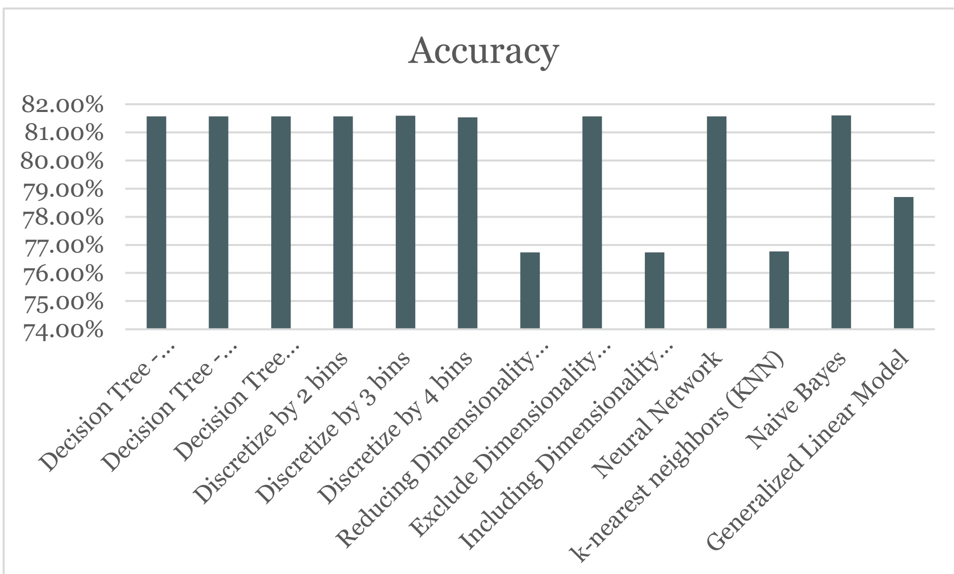
	true Good	true Fair	true unkown_h...	true Poor	class precision
pred. Good	2109	401	3	104	80.59%
pred. Fair	17	4	0	0	19.05%
pred. unkown_h...	5	0	131	0	96.32%
pred. Poor	70	8	1	3	3.66%
class recall	95.82%	0.97%	97.04%	2.80%	

Accuracy in generalized linear model is 78.7% with recalls 95.82% 97.04% that mean is better than Knn but not the best model.

Result

Experiments	Accuracy
Decision Tree - Normalized Range	81.57%
Decision Tree - Normalize Z-Score	81.57%
Decision Tree Imputation of Missing Values	81.57%
Discretize by 2 bins	81.57%
Discretize by 3 bins	81.59%
Discretize by 4 bins	81.54%
Reducing Dimensionality by Information Gain	76.73%
Exclude Dimensionality Reduction	81.57%
Including Dimensionality reduction	76.73%
Neural Network	81.57%
k-nearest neighbors (KNN)	76.77%
Naive Bayes	81.6%
Generalized Linear Model	78.7%
Gradient boosted Trees	81.6%
Deep Learning	81.6%

Conclusion


Accuracy
82.00%
81.00%
80.00%
79.00%
78.00%
77.00%
76.00%
75.00%
74.00%
Decision Tree -...
Decision Tree -...
Decision Tree...
Discretize by 2 bins
Discretize by 3 bins
Discretize by 4 bins
Reducing Dimensionality...
Exclude Dimensionality...
Including Dimensionality...
Neural Network
k-nearest neighbors (KNN)
Naive Bayes
Generalized Linear Model