



**WIE3007**

**Data Mining and Warehouse**

**Group Assignment**

Name	Matric Number
HEW DING XUAN	U2005416
TAN ZI HAO	U2005328
TIOW CHUN HAN	U2005374
PUA ZHI XIAN	U2005293
MAH SHIRLEY	U2005285

Lecturer : PROFESOR DR. TEH YING WAH

Tutorial Group : 1

# Table of Contents

Dataset.....	1
Dataset Description.....	2
Featuretools.....	3
Star Schema.....	4
SAS Enterprise Miner Diagram.....	5
SEMMA Methodology.....	6
Sample.....	6
Talend Data Integration.....	6
SAS Enterprise Miner.....	6
Explore.....	7
Modify.....	10
Talend Data Prep.....	10
SAS Enterprise Miner.....	12
Model.....	13
Non-Parametric Model.....	14
Decision Tree.....	14
Interactive Decision Tree.....	14
Gradient Boosting.....	16
DBSCAN.....	17
Parametric Model.....	20
Time Series.....	20
Neural Network.....	23
Regression.....	24
Forward Regression.....	24
Backward Regression.....	25
Stepwise Regression.....	26
Asses.....	27
Non-Parametric Model.....	27
Model Comparison.....	27
Score.....	28
Parametric model.....	29
Model Comparison.....	29
Conclusion.....	31
Future works can be done.....	32
GitHub Link.....	33
Presentation Video and Slides Link.....	33
References.....	34

# Dataset

Dataset title : Wine Quality

Dataset source : <https://archive.ics.uci.edu/dataset/186/wine+quality>

Number of instance : 6497

## Snapshot of datasets

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	index	fixed acid	volatile ac	citric acid	residual s	chlorides	free sulfur	total sulfu	density	pH	sulphates	alcohol	quality	category	TasterNar	TastingDate	Country	Price	QuantitySold
	0	0	7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5 red	David	3/14/2021	Spain	59.41	24
	1	1	7.8	0.88	0	2.6	0.098	25	67	0.9968	3.2	0.68	9.8	5 red	Eva	9/16/2020	USA	88.55	88
	2	2	7.8	0.76	0.04	2.3	0.092	15	54	0.997	3.26	0.65	9.8	5 red	Charlie	12/21/2020	Argentina	37.29	53
	3	3	11.2	0.28	0.56	1.9	0.075	17	60	0.998	3.16	0.58	9.8	6 red	Eva	6/2/2022	USA	76.31	85
	4	4	7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5 red	Bob	1/9/2022	USA	68.62	42
	5	5	7.4	0.66	0	1.8	0.075	13	40	0.9978	3.51	0.56	9.4	5 red	Charlie	3/15/2022	Italy	56.38	95
	6	6	7.9	0.6	0.06	1.6	0.069	15	59	0.9964	3.3	0.46	9.4	5 red	Bob	6/7/2022	Australia	44.6	63

These datasets are related to red and white variants of the Portuguese “Vinho Verde” wine. These datasets have multiple details of wine such as volatile\_acidity, category and so on. Based on these variables, we can do classification to estimate the quality of wine in different categories (red, white) based on the materials details given above. Here, we can use decision tree or gradient boosting to get the classification model complete and accurate. While we can also use association rules etc to identify which data given can help to classify which type of materials in wine can lead to good or poor wine.

## Dataset Description

Variables	Data Type	Description
fixed_acidity	double	Acidity in wine
volatile_acidity	double	Volatile in wine
citric_acid	double	% of citric_acid in wine
residual_sugar	double	% of sugar in wine
Chlorides	double	% of chloride in wine
free_sulfure_dioxide	int	% of free sulfur_dioxide in wine
total_sulfur_dioxide	int	% of total sulfur_dioxide in wine
density	double	Density of wine
pH	double	pH of wine
sulphates	double	% of sulphates in wine
alcohol	double	% of alcohol in wine
quality	int	Quality of wine 0 - 10
category	string	Type of wine
TasterName	string	Taster Name
TastingDate	date	Taster taste wine date
Country	string	Country of the wine produced
Price	double	Price of the wine
QuantitySold	int	Quantity sold that year

# Featuretools

```
import pandas as pd
import random

wine_df = pd.read_csv("final_wine.csv", delimiter = ',')
```

First, importing Featuretools and pandas libraries. We will be using pandas to do data importing to be imported to Featuretools for auto features extraction on wine quality datasets.

```
[ ] import featuretools as ft

# Create an EntitySet
es = ft.EntitySet(id='my_entity_set')

#Add data into the EntitySet
es = es.add_dataframe(dataframe_name = "wine", dataframe = wine_df, index = "index")
```

Next, create an entity set and import the wine datasets into the entity set.

```
# Perform Deep Feature Synthesis (DFS)
feature_matrix, feature_defs = ft.dfs(entityset=es,
                                     target_dataframe_name="wine", # Target entity for which you want to generate features
                                     trans_primitives = ['add_numeric', 'multiply_numeric']
                                     )

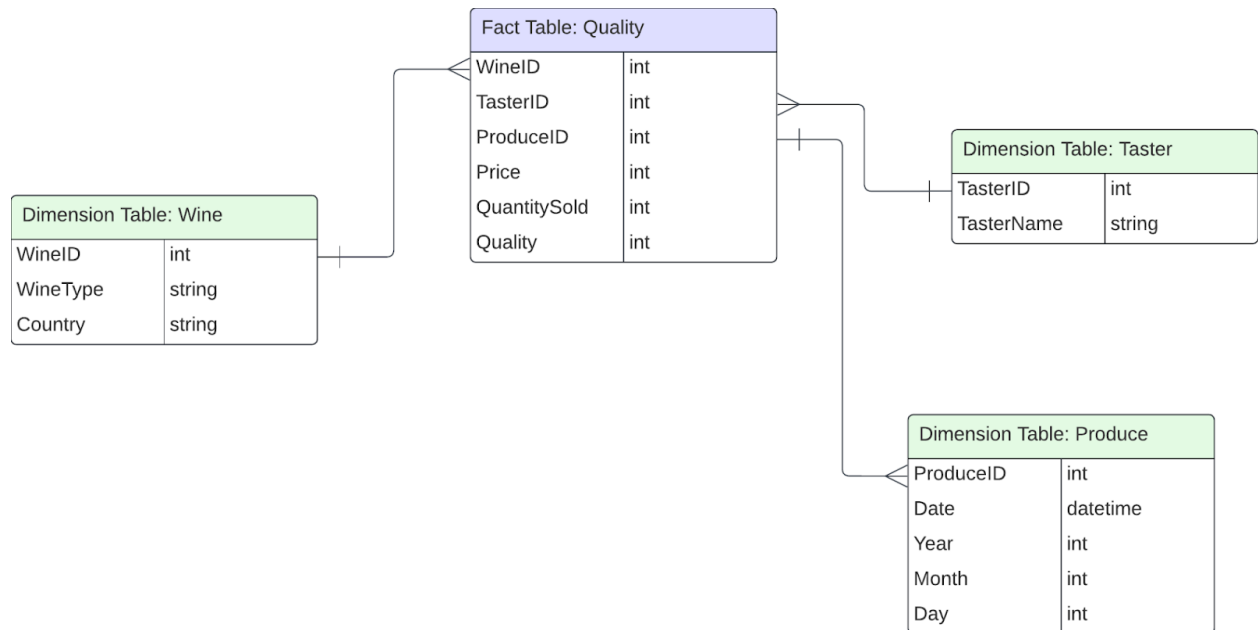
# feature_matrix contains the generated features
# feature_defs contains the definitions of the generated features

feature_matrix
```

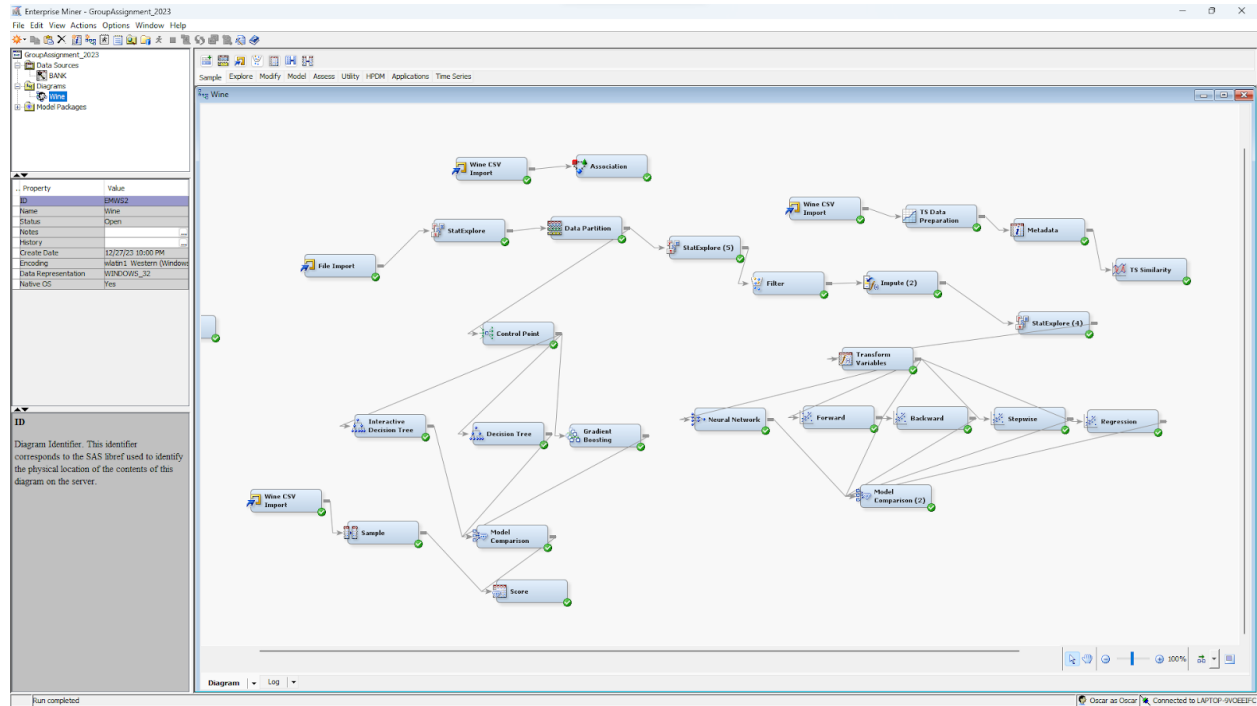
fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	...	quality * sulphates	quality * total sulfur dioxide	quality * volatile acidity	residual sugar * Sequence Tasting	residual sugar * sulphates
7.3	0.650	0.00	1.2	0.065	15.0	21.0	0.99460	3.39	0.47	...	3.29	147.0	4.550	1.2	0.564
8.5	0.490	0.11	2.3	0.084	9.0	67.0	0.99680	3.17	0.53	...	2.65	335.0	2.450	4.6	1.219
7.9	0.430	0.21	1.6	0.106	10.0	37.0	0.99660	3.17	0.91	...	4.55	185.0	2.150	4.8	1.456
6.7	0.675	0.07	2.4	0.089	17.0	82.0	0.99580	3.35	0.54	...	2.70	410.0	3.375	9.6	1.296
6.9	0.685	0.00	2.5	0.105	22.0	37.0	0.99660	3.46	0.57	...	3.42	222.0	4.110	12.5	1.425

We will then run the DFS of Featuretools to extract all features possible from datasets. Here, we can use the features extracted to identify useful information that can then be used in star schema design. From the result from DFS of Featuretools , we decide **quality** as fact while **wine**, **tester** and **produce date** as our dimension table.

# Star Schema



# SAS Enterprise Miner Diagram



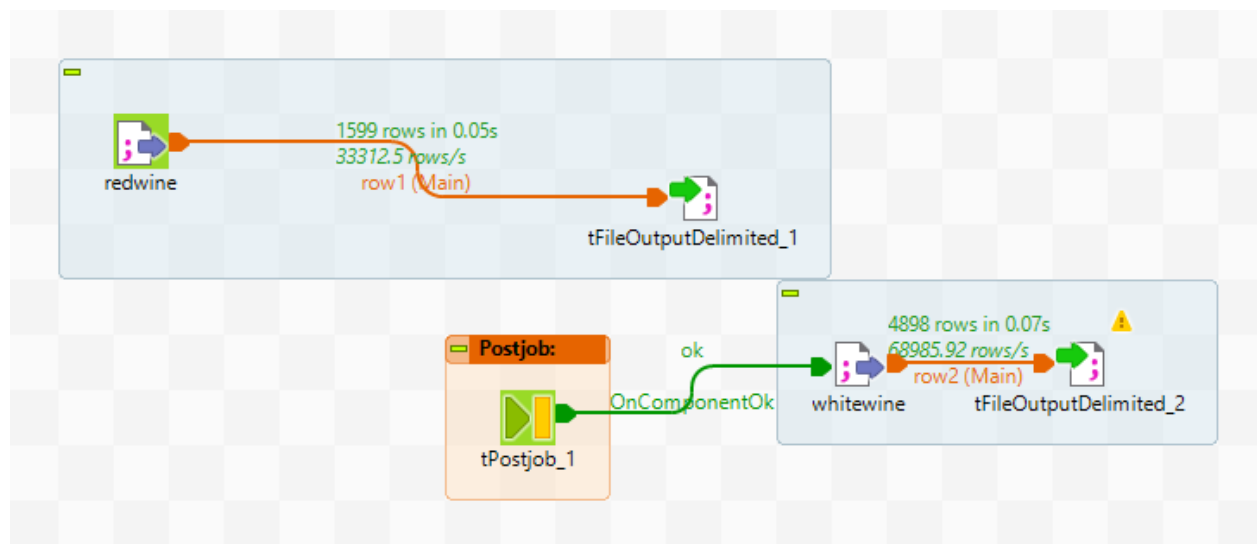
Above is the overview of the SAS Enterprise Miner model diagram that we built to apply SEMMA methodology. Further explanation on this diagram will be continued in the next section.

# SEMMA Methodology

## Sample

### Talend Data Integration

Talend Data Integration is used to join two datasets into one. From the ICS database, the wine data is being separated based on the wine category. In order to make modeling, we have to join the two datasets. Below showing the process of joining datasets into one using **tFileInputDelimited** with **tFileOutputDelimited** function.



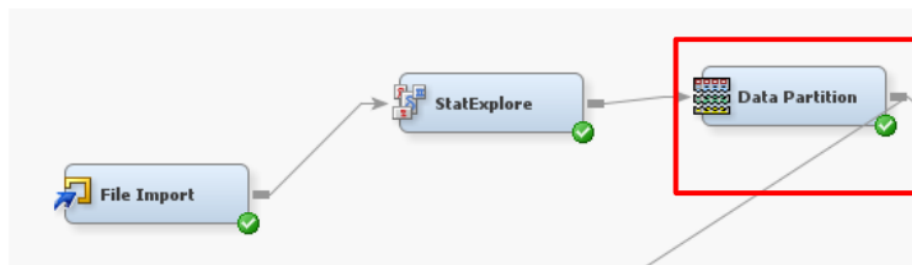
### SAS Enterprise Miner

For the data import, below is the roles and settings for all variables in the datasets. We will be using **quality** as the target role for this dataset which means we will try to identify the quality of wine based on other independent variables.



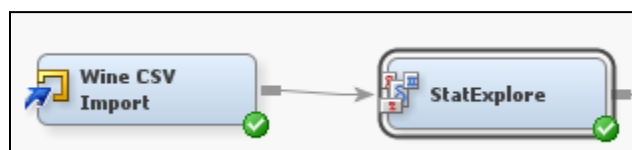
Columns: <input type="checkbox"/> Label		<input type="checkbox"/> Mining					
Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
Country	Rejected	Nominal	No		No	.	.
Price	Rejected	Interval	No		No	.	.
ProduceDate	Rejected	Interval	No		No	.	.
QuantitySold	Rejected	Interval	No		No	.	.
Sequence_Taste	Sequence	Interval	No		No	.	.
TasterName	Input	Nominal	No		No	.	.
TastingDate	Input	Interval	No		No	.	.
alcohol	Input	Interval	No		No	.	.
category	Input	Nominal	No		No	.	.
chlorides	Input	Interval	No		No	.	.
citric_acid	Input	Interval	No		No	.	.
density	Input	Interval	No		No	.	.
fixed_acidity	Input	Interval	No		No	.	.
free_sulfur_dioxide	Input	Interval	No		No	.	.
index	Rejected	Interval	No		No	.	.
pH	Input	Interval	No		No	.	.
quality	Target	Interval	No		No	.	.
residual_sugar	Residual	Interval	No		No	.	.
sulphates	Input	Interval	No		No	.	.
total_sulfur_dioxide	Input	Interval	No		No	.	.
volatile_acidity	Input	Interval	No		No	.	.

The SEMMA sample process in SAS Enterprise Miner begins by loading the wine dataset through the **File Input** node. Statistical exploration via the **StatExplore** node provides insights into the dataset's characteristics, descriptive analysis, guiding decisions for subsequent analysis.



Next, we use the **Data Partition** node to divide the dataset into training and validation sets to ensure the model's robustness and generalizability. For all model training, we will divide the datasets into 80% of the training set and 20% of the test set. This systematic approach sets the foundation for the subsequent stages of the SEMMA methodology, such as modifying variables, building models, and assessing their effectiveness in predicting wine quality.

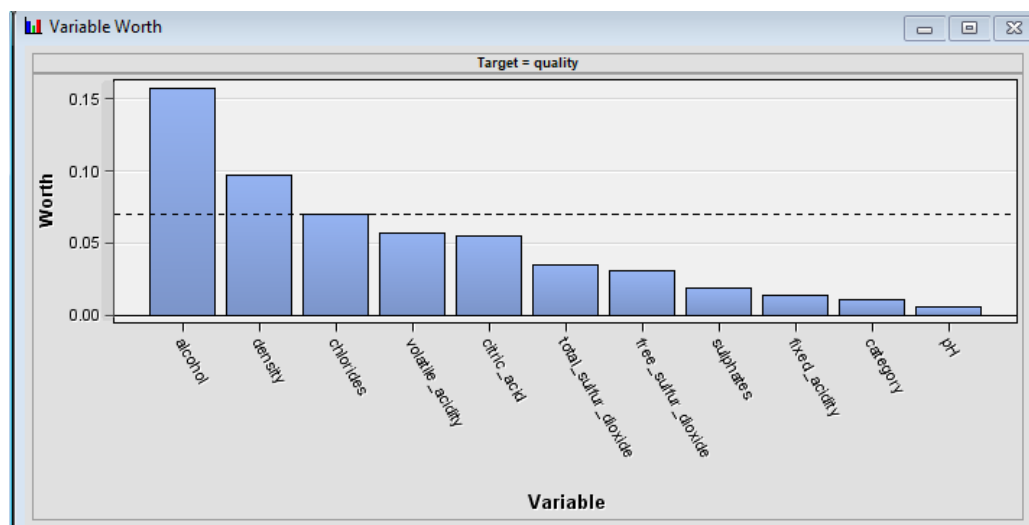
## Explore



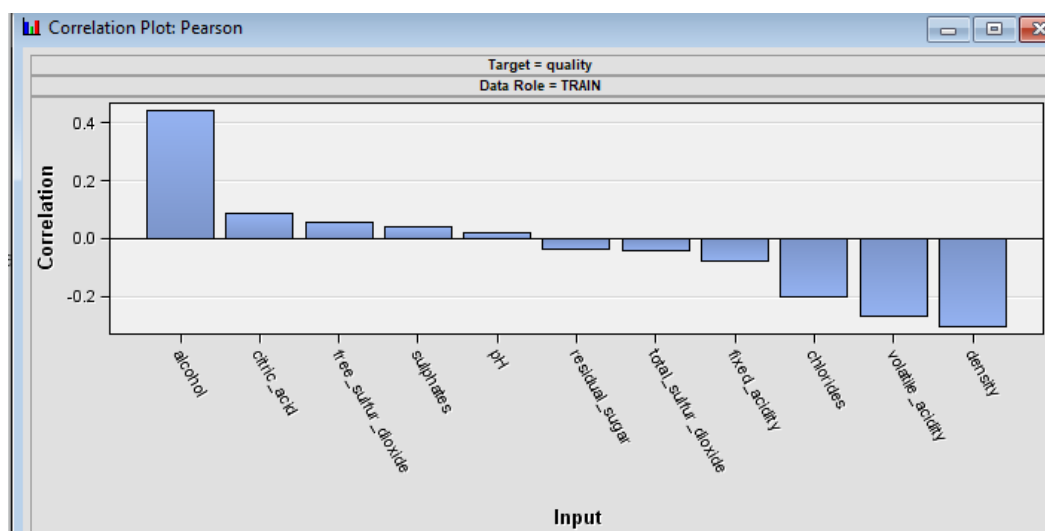
From the **StatExplore**, we can get to know if there are any missing values in the datasets. Below is the result:

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Ske
TastingDate	INPUT	22461.2	317.2113	6497	0	21915	22458	23010	0.0
alcohol	INPUT	10.4918	1.192712	6497	0	8	10.3	14.9	0.9
chlorides	INPUT	0.056034	0.035034	6497	0	0.009	0.047	0.611	5.3
citric_acid	INPUT	0.318633	0.145318	6497	0	0	0.31	1.66	0.4
density	INPUT	0.994697	0.002999	6497	0	0.98711	0.99489	1.03898	0
fixed_acidity	INPUT	7.215307	1.296434	6497	0	3.8	7	15.9	1.1
free_sulfur_dioxide	INPUT	30.52532	17.7494	6497	0	1	29	289	1.2
pH	INPUT	3.218536	0.160836	6487	10	2.72	3.21	4.01	0.1
sulphates	INPUT	0.53135	0.148881	6487	10	0.22	0.51	2	1.7
total_sulfur_dioxide	INPUT	115.7446	56.52185	6497	0	6	118	440	-0.1
volatile_acidity	INPUT	0.339666	0.164636	6497	0	0.08	0.29	1.58	1.4
residual_sugar	RESIDUAL	5.443235	4.757804	6497	0	0.6	3	65.8	1.4
quality	TARGET	5.818378	0.873255	6497	0	3	6	9	0.1

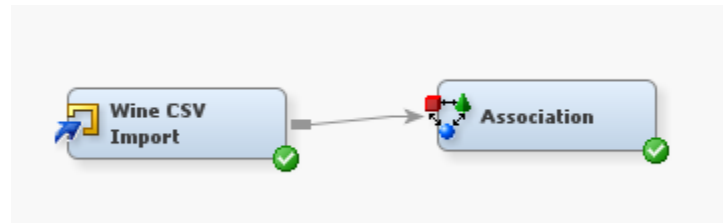
From the result above, we can know that some of the variables contain missing values. Thus we will use impute and filter nodes to do some data cleaning at the modify phase.



Another Explore methodology that is being used is **StatExplore** node where it will be used to analyze the features that can be useful from the dataset.



From this graph, we can see that **alcohol** has the highest correlation with **quality**. Here, we can conclude that we can use alcohol as the source or ID to make models later.



Name	Use	Role	Level
Sequence_Tastin	Yes	Sequence	Interval
category	Yes	ID	Nominal
quality	Yes	Target	Nominal

We use **Association** nodes to perform association discovery on our dataset. Association discovery is the identification of items that occur together in a given event or record. For the association rule, we use the wine category against the quality of wine.

Map	Rule
RULE1	5 ==> 3
RULE2	6 ==> 3
RULE3	7 ==> 3
RULE4	8 ==> 3
RULE5	5 ==> 4
RULE6	6 ==> 4
RULE7	7 ==> 4
RULE8	8 ==> 4
RULE9	8 & 5 & 4 ==> 4
RULE10	5 ==> 5
RULE11	6 ==> 5
RULE12	7 ==> 5
RULE13	8 ==> 5
RULE14	8 & 5 ==> 5
RULE15	6 ==> 5 & 3
RULE16	6 & 4 ==> 5 & 3
RULE17	6 & 5 ==> 5 & 3
RULE18	7 ==> 5 & 3
RULE19	7 & 3 ==> 5 & 3
RULE20	8 ==> 5 & 3
RULE21	8 & 5 & 4 ==> 5 & 3
RULE22	6 ==> 5 & 4
RULE23	6 & 4 ==> 5 & 4
RULE24	7 ==> 5 & 4

Above shows the association rule generated from the **Associate** node. There are several associations between wines which can be identified from this list of rule descriptions.

The use of Featuretools in Python can also be considered as one type of Explore phase in SEMMA. We can use Featuretools to find out significant features that can be used for model training. For more details, refer [Featuretools](#).

Below are the some of the important variables and their roles that will be used for building models:

#### **Potential Predictor Variables**

- fixed\_acidity (double)
- volatile\_acidity (double)
- citric\_acid (double)
- residual\_sugar (double)
- Chlorides (double)
- free\_sulfure\_dioxide (int)
- total\_sulfur\_dioxide (int)
- density (double)
- pH (double)
- sulphates (double)
- alcohol (double)

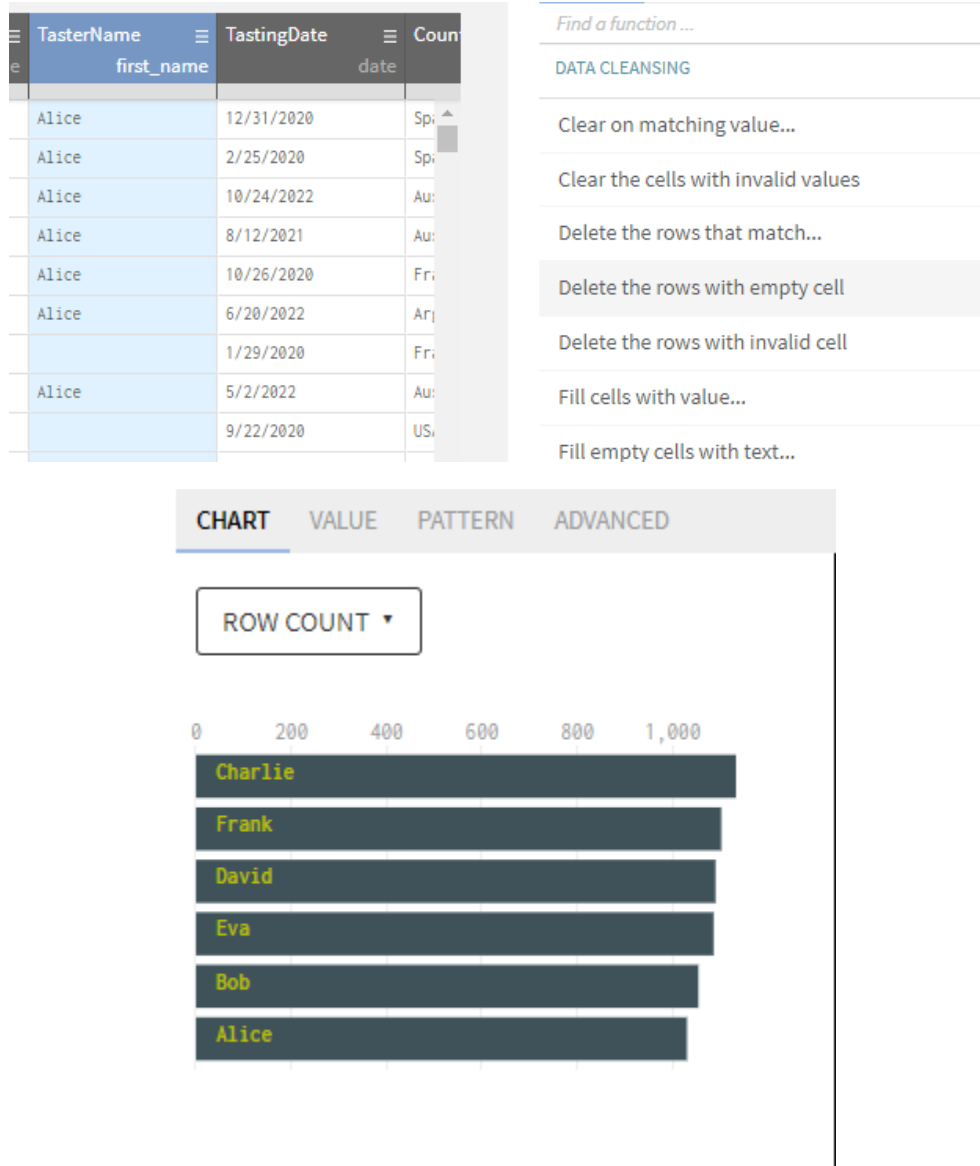
#### **Target Variable**

- quality (int)

## **Modify**

### **Talend Data Prep**

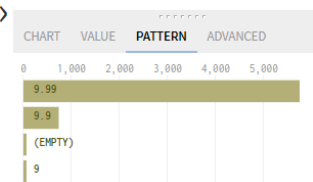
First we can see that there are some empty value in Taster Name, we can use Talend Data Prep **Data Cleansing** function to delete rows with empty cells. We can check the chart also to make sure all values in TasterName is consistent after cleaning..



pH and sulphates column also contains empty value, we can use the same function to remove rows from it. Another way is to fill empty value with certain value. Here we choose to fill it with mean of the values based on the column.

pH	≡ sulphates	≡ alcohol	≡ quality	≡ category	≡ TasterName	≡ TastingDate	≡ Coun
decimal	decimal	decimal	integer	last_name	first_name	date	
3.32	0.44	11.2	5	white	Frank	9/9/2022	Fr
3.31	0.58	11.75	7	white	Frank	12/15/2020	US
3.16	0.5	9.6	6	white	Frank	8/24/2022	Au
3.16	0.5	9.6	6	white	Frank	3/21/2020	Sp
3.16	0.51	9.1	5	white	Frank	8/21/2021	Au
3.32	0.47	10.1	6	white	Frank	8/10/2020	Fr
3.14	0.48	9.8	5	white	Frank	9/27/2021	It
3.04	0.53	8.9	6	white	Frank	8/22/2020	Fr
3.04	0.53	8.9	6	white	Frank	1/10/2020	Sp
3.16	0.43	10.3	5	white	Frank	7/19/2020	US
3.24	0.35	12.1	4	white	Frank	9/3/2020	Ar
3.1	0.45	9.7	5	white	Frank	12/24/2022	Sp
3.18	0.45	11.3	6	white	Frank	6/14/2020	Au
3.48	0.57	10.7	6	white	Frank	1/19/2021	Au
3.23	0.47	9.2	5	white	Frank	6/24/2022	US
3.18	0.6	9.5	5	white	Frank	1/30/2020	Fr
3.05	0.56	11.1	6	white	Frank	7/29/2022	US
3.1	0.48	9.6	5	white	Frank	4/10/2020	It
2.9	0.46	9	6	white	Frank	2/21/2022	Sp
2.9	0.46	9	6	white	Frank	5/21/2020	US
		11	6	white	Frank	5/14/2022	US

Select a row to display its actions

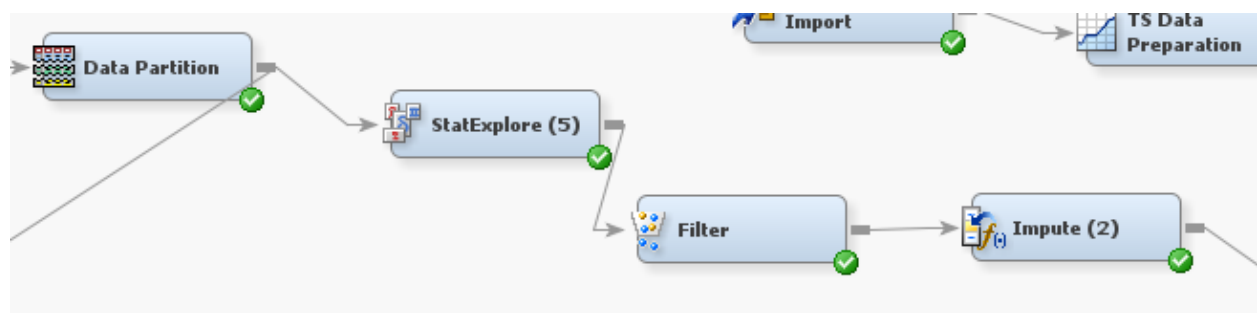


- 1 Delete the rows with empty cell on column TasterName
- 2 Fill empty cells with text on column sulphates
- 3 Fill empty cells with text on column pH

Same to pH, after all the cleaning, Talend Data Prep shows all values are cleaned, as there is no more empty values existing.

## SAS Enterprise Miner

Transformation will be taking place In this phase. Impute node is being used to impute missing values with certain values. For example, in this dataset, we impute the missing value for pH and sulphates with mean. The reason we choose the mean as the replacement method is because both pH and sulphates are numerical and the distribution of the variable is approximately normal.



Name	Use	Method	Use Tree	Role	Level
TasterName	Default	None	Default	Input	Nominal
TastingDate	Default	Default	Default	Input	Interval
alcohol	Default	None	Default	Input	Interval
category	Default	None	Default	Input	Nominal
chlorides	Default	None	Default	Input	Interval
citric_acid	Default	None	Default	Input	Interval
density	Default	None	Default	Input	Interval
fixed_acidity	Default	None	Default	Input	Interval
free_sulfur_diox	Default	None	Default	Input	Interval
pH	Default	Mean	Default	Input	Interval
quality	Default	None	Default	Target	Interval
sulphates	Default	Mean	Default	Input	Interval
total_sulfur_diox	Default	None	Default	Input	Interval
volatile_acidity	Default	None	Default	Input	Interval

Columns: <input type="checkbox"/> Label <input type="checkbox"/> Mining <input type="checkbox"/> Basic <input type="checkbox"/> Statistics					
Name	Report	Filtering Method	Keep Missing Values	Minimum Frequency Cutoff	Number Cutoff
TasterName	No	Default	No	.	
category	No	Default	Default	.	

Next, we use the **Filter** node to remove rows that have missing data that are not suitable to replace by any value. We can use it to exclude certain observations, filtering extreme values from the training data tends to produce better models because the parameter estimates are more stable.

## Model

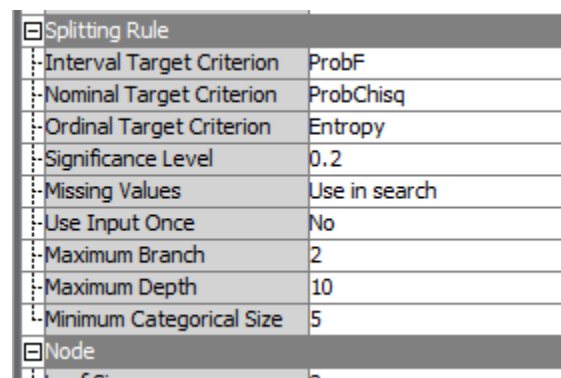
For machine learning model training, we splitted it into 2 different models which are non-parametric model and parametric model.

## Non-Parametric Model

Non parametric models do not need to make assumptions about the relations between the input and output to generate the outcome and this model is suitable for any number of parameters to be set and learned. Thus a missing value would not affect much on the results of this model. The advantages of this model is they can catch complex patterns and relationships without having to follow a hypothesis.

### Decision Tree

We will be using a maximum branch of 2 and maximum depth of 10 for this decision tree model, which will also be the same as an interactive decision tree.



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
Node	
Leaf Size	5

For the splitting rule, here we are using probF as the interval target criterion while ProbChisq will be used for nominal target criterion. The focusing of this project will be on quality which is an interval target. Hence ProbF will be used as the criterion for splitting the node and decision tree.

Decision trees can be used to predict the quality of wine based on the given features. It helps identify the most important features contributing to the quality of wine.

### Interactive Decision Tree

For an interactive decision tree, it's quite similar to the decision tree, but this time we can choose how the decision tree splits its node. In our case, we first split the decision tree using the alcohol which has the highest  $-\log(p)$  which is the logworth value. This can also be proven from the **StatExplore** node that we did just now where alcohol has the highest correlation to quality which has the highest variable worth value among all input variables.



Split Node 1

Target Variable: quality

Variable	Variable Description	-Log(p)	Branches
alcohol	alcohol	200.6735	2
density	density	124.8717	2
chlorides	chlorides	92.5682	2
volatile_acidity	volatile acidity	53.8847	2
citric_acid	citric acid	51.3902	2
total_sulfur_dioxide	total sulfur dioxide	23.4785	2
free_sulfur_dioxide	free sulfur dioxide	20.7973	2
category	category	16.8908	2
fixed_acidity	fixed acidity	13.081	2
sulphates	sulphates	12.4563	2
pH	pH	0.9009	2

Secondly, we will continue splitting the tree by choosing citric acid as the second split node. Here is where interactive decision trees come into mind. We can choose our own split variables.

Split Node 3

Target Variable: quality

Variable	Variable Description	-Log(p)	Branches
volatile_acidity	volatile acidity	67.5749	2
citric_acid	citric acid	22.0443	2
alcohol	alcohol	19.4231	2
chlorides	chlorides	14.5005	2
free_sulfur_dioxide	free sulfur dioxide	11.7987	2
category	category	9.2348	2
total_sulfur_dioxide	total sulfur dioxide	5.3332	2
pH	pH	3.063	2
density	density	3.0335	2
fixed_acidity	fixed acidity	2.7583	2
sulphates	sulphates	1.7413	2

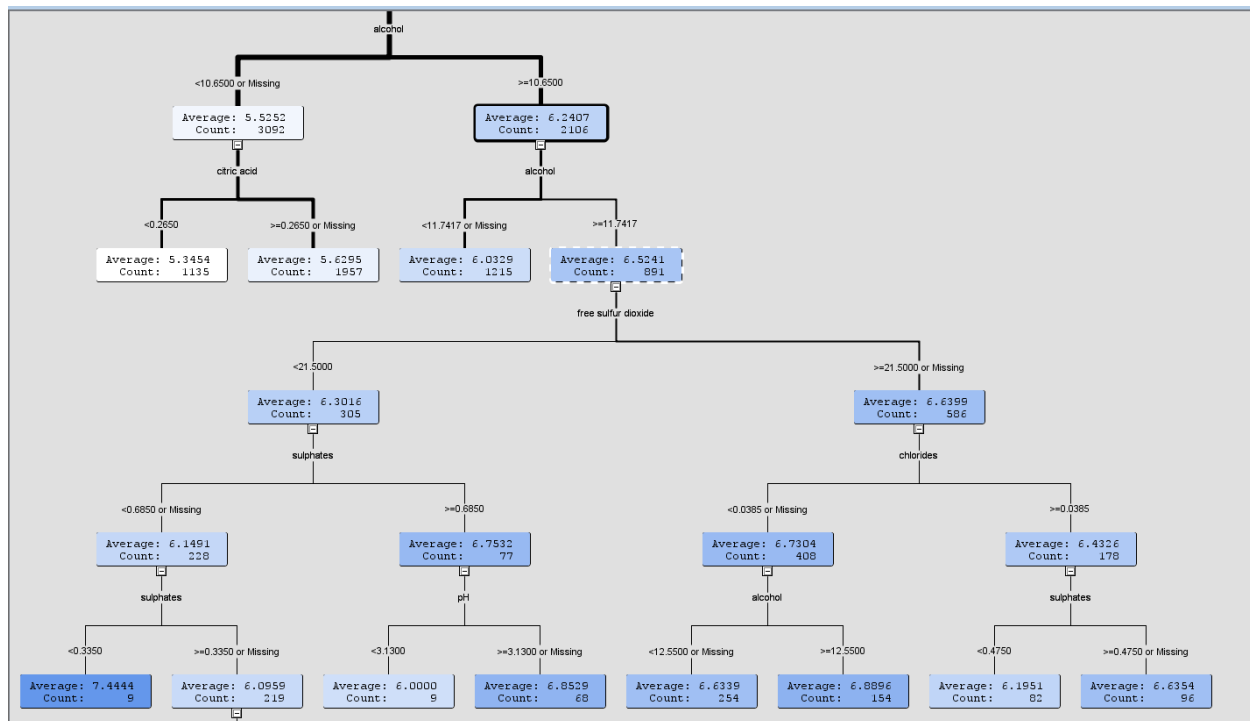
Edit Rule...

OK Cancel Apply Refresh

Following the decision tree, we will then make it automatically split it based on highest to lowest logworth value by choosing the node that has the darkest color (which is having the

highest count under that category). We can further split it to make it more specific to certain characteristics.

Until an interactive decision tree is being made, we will then use it for building models and comparison later. The results comparison will be shown at the assess phase of SEMMA.



## Gradient Boosting

Gradient Boosting is another model provided by SAS Enterprise Miner. Gradient boosting is a boosting approach that resamples the analysis data set several times to generate results that form a weighted average of the re-sampled data set.

General	
Node ID	Boost
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Series Options	
N Iterations	50
Seed	12345
Shrinkage	0.1
Train Proportion	60
Splitting Rule	
Huber M-Regression	No
Maximum Branch	2
Maximum Depth	10
Minimum Categorical Size	5
Reuse Variable	1
Categorical Bins	30
Interval Bins	100

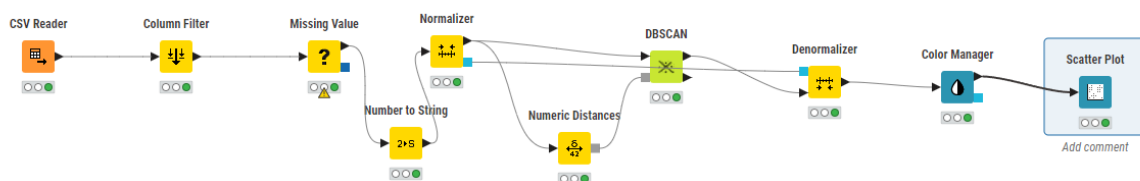
For this model, we will be using a maximum depth of 10 and maximum branch of 2.

Node	
Leaf Fraction	0.1
Number of Surrogate Rules	2
Split Size	.
Split Search	
Exhaustive	5000
Node Sample	20000

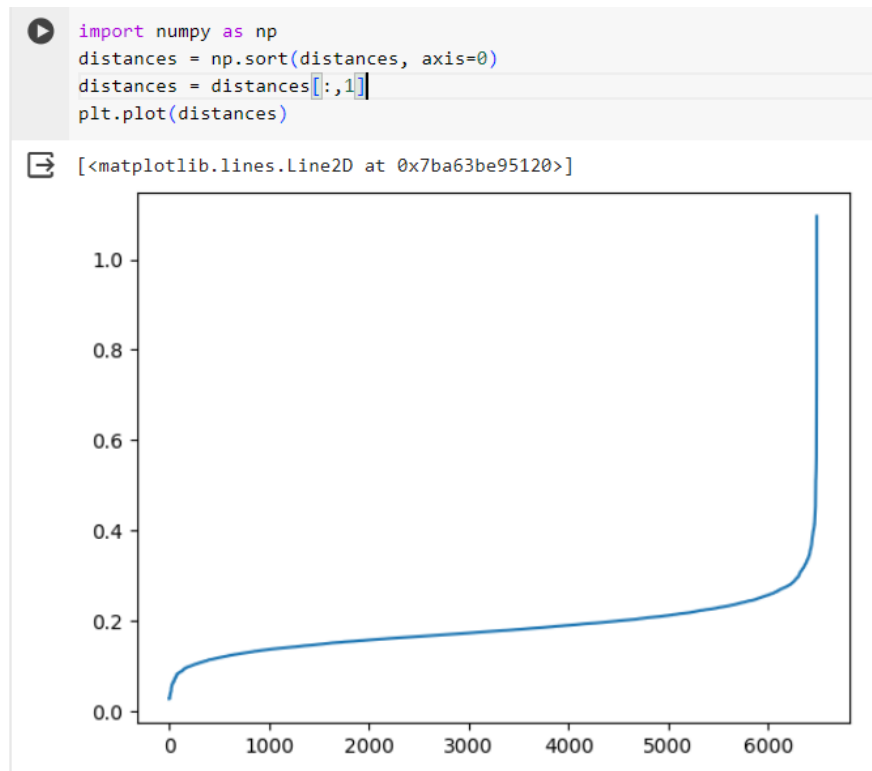
We will also be using 2 surrogate rules which mean 2 backup rules will be selected and used once there is missing data.

## DBSCAN

For DBSCAN, we will compare between ph to alcohol with category of quality. For this model, we will be using KNIME as the tool to do DBSCAN model. The process is of modeling is also implementing SEMMA method.

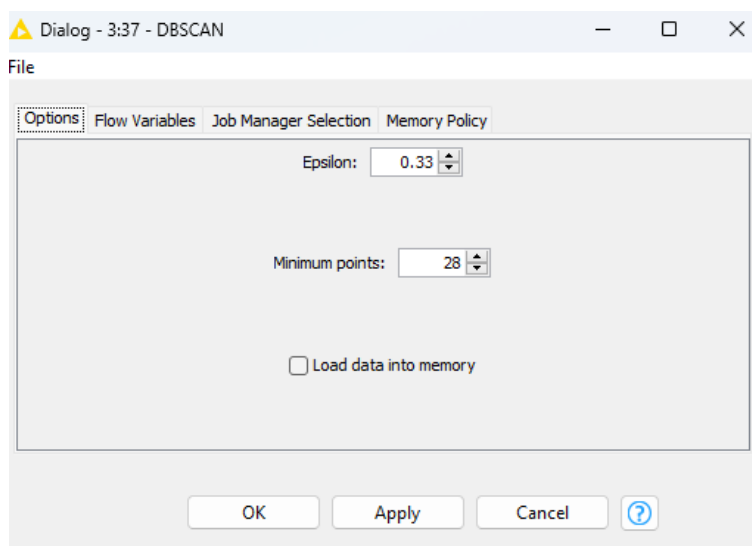


For the epsilon and minimum points configurations, we made a KNN script to get the suitable epsilon. In this case, we get the epsilon around 0.33.



For epsilon, since we normalize it using **Normalizer** node, we will be using value between 0 - 1.0. In this example, we will be using 0.33 as the epsilon.

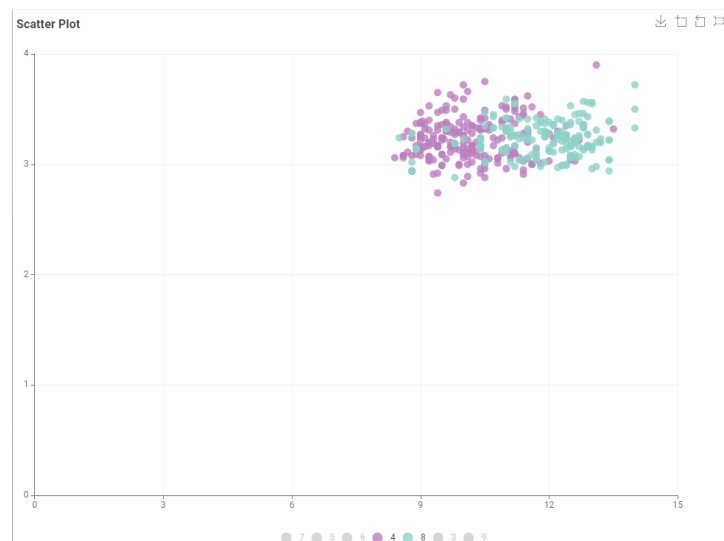
While for minimal points we will use the common calculation which is 2 x column. Here we will be having 14 columns x 2 = 28 minimal points



From the results below, we get to see that only one cluster is being determined. Although we try to adjust the epsilon and minimum points value, we still get one cluster. This can be made as a conclusion that quality of wine does not really affect a certain variable, it might be affected by all variables, but in this case, since we are only comparing between two variables, it is difficult for the model to differentiate the quality of wine based on only one other variables.

#	RowID	Count <i>Number (long)</i>
1	Noise	118
2	Clust...	6369

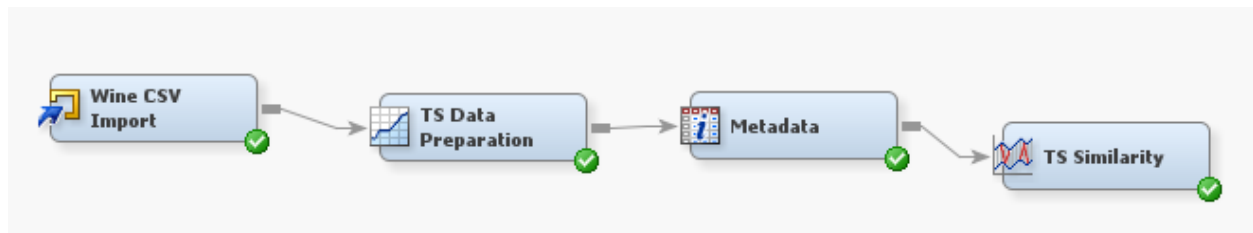
Below show comparison between quality wine of 3 and 9, we can get to see that the scatter point of these wine does not really differentiate well due to dependency to other more variables.



## Parametric Model

### Time Series

For the Time Series model, we have implemented the **Time Series Similarity** node in the SAS Enterprise Miner.



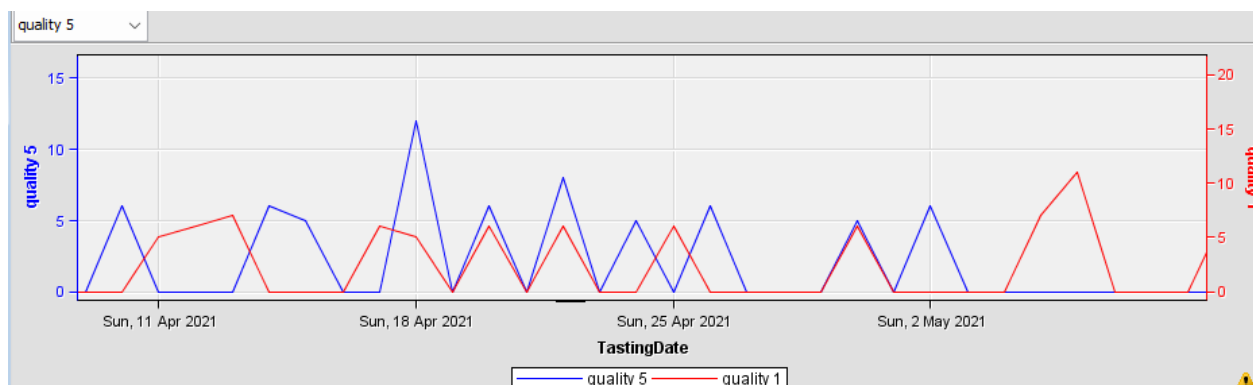
Transpose Options	
-Transpose	Yes
-By Variable	By TSID
-Keep Variable Role	No

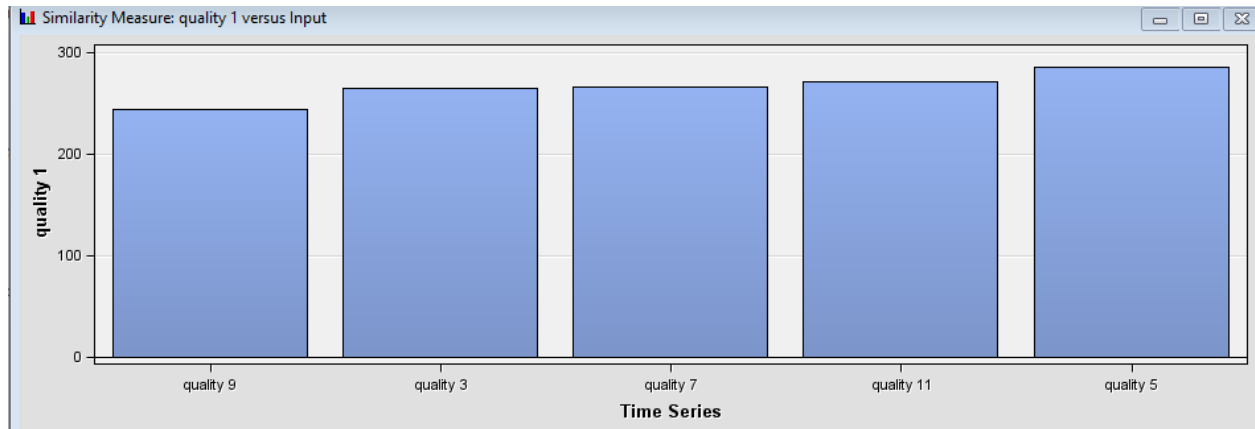
We transpose the datasets first by choosing the **Transpose** option to yes and also delete variable roles after the transformation.

Next we will be changing the metadata of the variables. For example, `_TS_01` is being chosen as the target for this **Time Series** model.

Variables - Meta								
<div>(none) <input type="checkbox"/> not Equal to <input type="checkbox"/> Basic</div>								
Columns: <input type="checkbox"/> Label <input type="checkbox"/> Mining								
Name	Hidden	Hide	Role	New Role	Level	New Level	New Order	New Report
TastingDate	N	Default	Time ID	Default	Interval	Default	Default	Default
_TS_01	N	Default	Input	Target	Interval	Default	Default	Default
_TS_02	N	Default	Input	Default	Interval	Default	Default	Default

Next will be the **TS Similarity** node, we can see that quality 5 has the highest similarity to quality 1 which means that the wine that has quality 1 and 5 has almost the same similarity compared to others although the time is shifted.

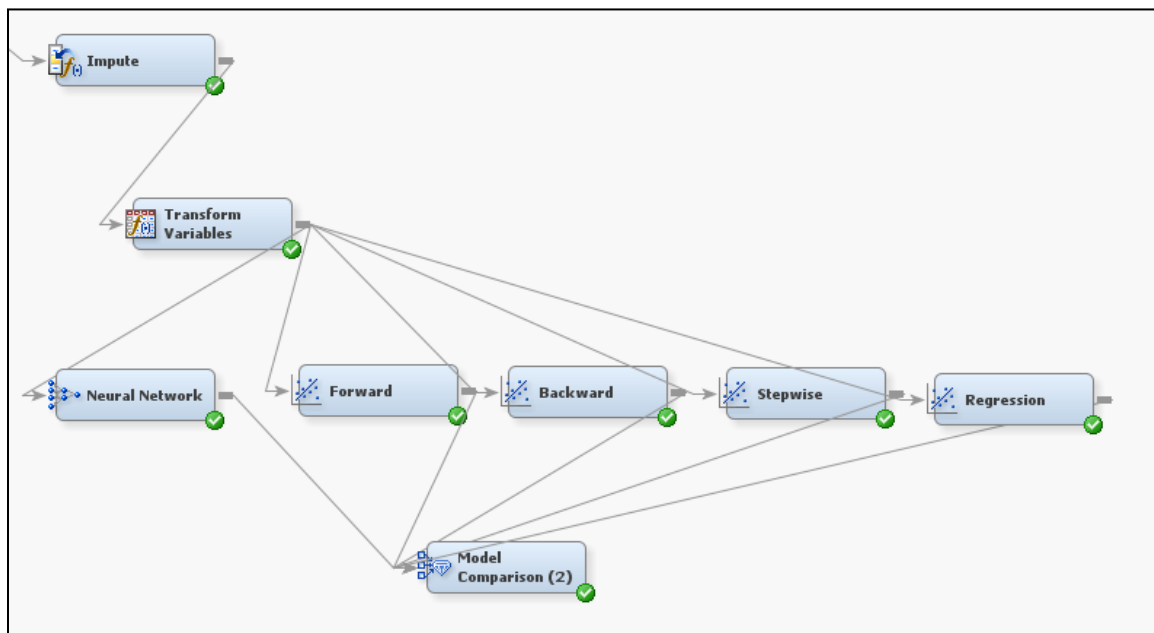




Moving on, we implemented **Neural Network**, **Forward Regression**, **Backward Regression** and **Stepwise Regression**. We will use the **Model Comparison** node to compare the performance of all these models, then choosing the best model which has the lowest mean square errors.

Source	Method	Variable Name	Formula	Number of Levels	Non Missing	Missing	Minimum	Maximum	Mean	Standard Deviation	Skewness	Kurtosis	Label
Input	Original	alcohol			5198	0	8	14.2	10.4852	1.184102	0.553259	-0.54234	
Input	Original	chlorides			5198	0	0.009	0.611	0.056538	0.036718	5.526941	50.87909	
Input	Original	cltrc_acid			5198	0	9	1.23	0.31792	0.1453	0.404003	1.515987	cltrc acid
Input	Original	density			5198	0	0.98711	1.03898	0.994684	0.003006	0.614852	8.258824	
Input	Original	fixed_acidity			5198	0	3.8	15.6	7.212332	1.292129	1.728306	5.081959	fixed acidity
Input	Original	free_sulfur_diox...			5198	0	1	289	30.55454	17.85158	1.340207	9.529849	free sulfur dioxide
Input	Original	pH			5198	0	2.72	4.01	3.216672	0.166532	0.379651	0.349555	
Input	Original	sulphates			5198	0	0.22	2	0.531635	0.151345	1.938087	9.843272	
Input	Original	total_sulfur_diox...			5198	0	6	440	115.6163	56.62922	0.019697	-0.31211	total sulfur dioxide
Output	Original	volatile_acidity			5198	0	0.08	1.58	0.339993	0.164955	1.513671	2.95599	volatile acidity
Output	Computed	LG10_alcohol	log10(alcohol + ...		5198	0	0.954243	1.181844	1.057891	0.043918	0.384923	-0.75459	Transformed alc...
Output	Computed	LG10_chlorides	log10(chlorides ...		5198	0	0.003891	0.207096	0.02365	0.013933	4.619882	36.12072	Transformed chl...
Output	Computed	LG10_cltrc_acid	log10(cltrc_acid...		5198	0	0	0.348305	0.117263	0.047833	-0.09862	0.902598	Transformed clt...
Output	Computed	LG10_density	log10(density + ...		5198	0	0.298222	0.309413	0.299874	0.009543	0.593	7.886648	Transformed de...
Output	Computed	LG10_fixed_acd...	log10(fixed_acid...		5198	0	0.681241	1.220108	0.909694	0.062815	0.973526	2.166137	Transformed fix...
Output	Computed	LG10_free_sulf...	log10(free_sulfu...		5198	0	0.30103	2.462398	1.419184	0.284906	-0.7214	0.131779	Transformed fr...
Output	Computed	LG10_pH	log10(pH + 1)		5198	0	0.570543	0.699838	0.624657	0.016447	0.25727	0.191764	Transformed pH
Output	Computed	LG10_sulphates	log10(sulphates...		5198	0	0.08638	0.477121	0.183217	0.040257	1.169825	3.672794	Transformed su...
Output	Computed	LG10_total_sulf...	log10(total_sulf...		5198	0	0.845098	2.644439	1.987682	0.303855	-1.34938	1.295276	Transformed tot...
Output	Computed	LG10_volatile_a...	log10(volatile_a...		5198	0	0.033424	0.41162	0.12411	0.048885	1.135803	1.25033	Transformed vo...

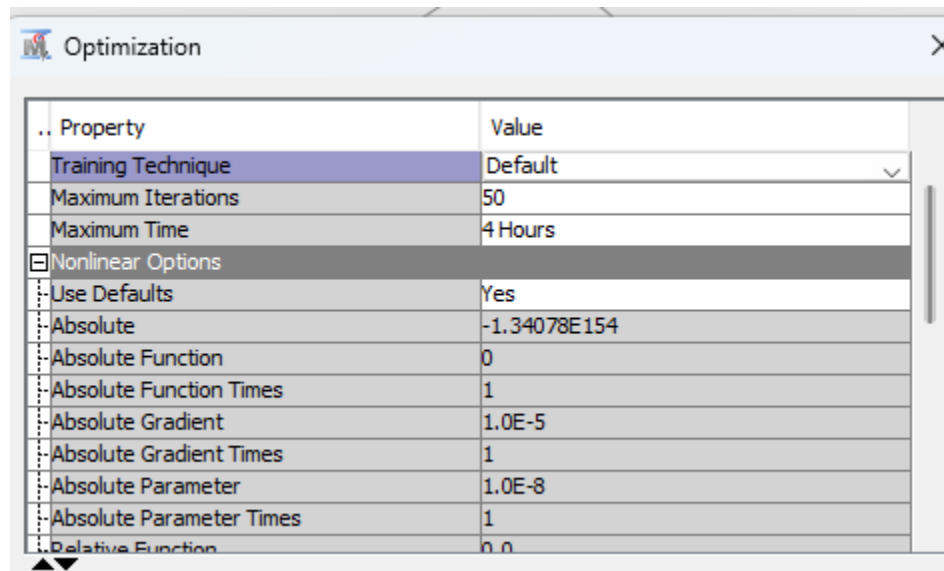
For the Parametric model, we decide to normalize the data before running the model as most of the machine learning models might require data to be normalized first which can help to improve the accuracy of the model training. Here, we used a Transform **Variable** node to calculate log10 of each value of the input variables.





## Neural Network

For the neural network here, we use default values for all configurations as we are showing the example of the SEMMA process in SAS Enterprise Miner.



Below is the result of the Neural **Network** model after training. As we can see here the average square error has been reduced to 0.50097 in the last iterations.

Iter	Restarts	Function Calls	Active Constraints	Objective Function	Objective Function Change	Max Abs Gradient Element	Lambda	and Predicted Change
1*	0	2	0	0.65487	0.1018	0.1183	444E-16	1.000
2	0	3	0	0.58170	0.0732	0.2662	0.00448	0.628
3	0	5	0	0.53942	0.0423	0.1209	0.0171	0.912
4	0	6	0	0.53883	0.000589	0.1460	0.00294	0.0316
5	0	7	0	0.52008	0.0187	0.0733	0.0108	0.821
6	0	8	0	0.51831	0.00177	0.0569	0.00225	0.148
7	0	10	0	0.51140	0.00691	0.0403	0.0101	0.506
8	0	11	0	0.50972	0.00168	0.0294	0.00477	0.309
9	0	12	0	0.50815	0.00157	0.0174	0.00363	0.291
10	0	13	0	0.50685	0.00130	0.0109	0.00230	0.360
11	0	14	0	0.50659	0.000258	0.0111	0.00126	0.0805
12	0	15	0	0.50511	0.00148	0.00633	0.00785	0.666
13	0	16	0	0.50454	0.000577	0.00202	0.00215	0.897
14	0	17	0	0.50409	0.000448	0.00793	0.00057	0.791
15	0	18	0	0.50358	0.000503	0.0105	0.00013	0.692
16	0	19	0	0.50317	0.000413	0.0212	0	0.543
17	0	20	0	0.50246	0.000716	0.00573	0	0.812
18	0	21	0	0.50206	0.000394	0.00576	0	0.734
19	0	22	0	0.50185	0.000216	0.00337	0	0.607
20	0	23	0	0.50178	0.000071	0.00149	0	0.185
21	0	24	0	0.50153	0.000246	0.00293	1.96E-7	0.285
22	0	28	0	0.50125	0.000284	0.00145	0.00095	0.347
23	0	30	0	0.50106	0.000190	0.000966	0.00190	0.652
24	0	31	0	0.50105	8.758E-6	0.000454	0.00025	0.126
25	0	33	0	0.50101	0.000037	0.000214	0.00512	0.517
26	0	34	0	0.50101	2.25E-7	0.000343	0.00089	0.0099
27	0	35	0	0.50099	0.000015	0.000115	0.00755	0.596
28	0	36	0	0.50099	5.799E-6	0.000108	0.00119	1.066
29	0	37	0	0.50098	5.082E-6	0.000165	0.00046	0.598
30	0	39	0	0.50098	4.806E-6	0.000086	0.00792	0.507
31	0	40	0	0.50098	1.992E-6	0.000097	0.00235	0.553
32	0	41	0	0.50098	1.298E-6	0.000143	0.00332	0.278
33	0	42	0	0.50097	1.81E-6	0.000096	0.00448	0.331

## Regression

In the Regression model, we get the result of mean square error = 0.532880. In the Type 3 Analysis of Effects, we can identify which input variable has the highest correlation with the target which is the one that has the highest F value. Here, alcohol and volatile acidity has the highest correlation on the quality of wine.

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	11	1169.817105	106.347010	199.57	<.0001
Error	5186	2763.514369	0.532880		
Corrected Total	5197	3933.331474			
Model Fit Statistics					
R-Square	0.2974	Adj R-Sq	0.2959		
AIC	-3259.9443	BIC	-3257.8888		
SBC	-3181.2720	C(p)	12.0000		
Type 3 Analysis of Effects					
Effect	DF	Sum of Squares	F Value	Pr > F	
LG10_alcohol	1	336.0510	630.63	<.0001	
LG10_chlorides	1	8.0681	15.14	0.0001	
LG10_citric_acid	1	0.0746	0.14	0.7083	
LG10_density	1	8.8641	16.63	<.0001	
LG10_fixed_acidity	1	4.3188	8.10	0.0044	
LG10_free_sulfur_dioxide	1	67.7785	127.19	<.0001	
LG10_pH	1	3.4365	6.45	0.0111	
LG10_sulphates	1	25.1256	47.15	<.0001	
LG10_total_sulfur_dioxide	1	24.2787	45.56	<.0001	
LG10_volatile_acidity	1	138.1699	259.29	<.0001	
category	1	1.3528	2.54	0.1111	

## Forward Regression

In Forward Regression, we are using the **Regression** node but changing the model selection to **Forward**. The result is quite similar to other regression models but there are slight differences in results due to different techniques being used while training the model. Here, we can get to see the mean square errors of **Forward Regression** gives the result of 0.533535. Below is the result of Forward Regression.

Model Selection	
Selection Model	Forward
Selection Criterion	Default
Use Selection Defaults	Yes
Selection Options	...

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	1164.283204	166.326172	311.74	<.0001
Error	5190	2769.048269	0.533535		
Corrected Total	5197	3933.331474			

Model Fit Statistics			
R-Square	0.2960	Adj R-Sq	0.2951
AIC	-3257.5458	BIC	-3255.5408
SBC	-3205.0976	C(p)	14.3849

Type 3 Analysis of Effects				
Effect	DF	Sum of Squares	F Value	Pr > F
LG10_alcohol	1	357.6127	670.27	<.0001
LG10_chlorides	1	7.1881	13.47	0.0002
LG10_density	1	7.0921	13.29	0.0003
LG10_free_sulfur_dioxide	1	77.7614	145.75	<.0001
LG10_sulphates	1	27.9505	52.39	<.0001
LG10_total_sulfur_dioxide	1	44.9553	84.26	<.0001
LG10_volatile_acidity	1	197.1855	369.58	<.0001

## Backward Regression

In Backward Regression, we are using the **Regression** node but changing the model selection to **Backward**. From the result, we can see differences in the impact of variables (inputs) to quality of wine (target). Density seems to play more important roles compared to chlorides which is contrast in Forward Regression. The mean square errors of **Backward Regression** model is 0.532940

Model Selection	
Selection Model	Backward
Selection Criterion	Default
Use Selection Defaults	Yes
Selection Options	...

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	9	1168.436377	129.826264	243.60	<.0001
Error	5188	2764.895097	0.532940		
Corrected Total	5197	3933.331474			

Model Fit Statistics			
R-Square	0.2971	Adj R-Sq	0.2958
AIC	-3261.3479	BIC	-3259.3117
SBC	-3195.7876	C(p)	10.5911

Type 3 Analysis of Effects				
Effect	DF	Sum of Squares	F Value	Pr > F
LG10_alcohol	1	347.5739	652.18	<.0001
LG10_chlorides	1	7.0048	13.14	0.0003
LG10_density	1	10.4097	19.53	<.0001
LG10_fixed_acidity	1	3.4554	6.48	0.0109
LG10_free_sulfur_dioxide	1	74.7629	140.28	<.0001
LG10_pH	1	2.6797	5.03	0.0250
LG10_sulphates	1	30.5763	57.37	<.0001
LG10_total_sulfur_dioxide	1	48.7665	91.50	<.0001
LG10_volatile_acidity	1	187.5581	351.93	<.0001

## Stepwise Regression

Finally for Stepwise Regression, we are using the **Regression** node but changing the model selection to **Stepwise**. Same as above, we will be getting the significant variables to the target from the model and also the mean squared error. Here the mean squared error is 0.533535

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	1164.283204	166.326172	311.74	<.0001
Error	5190	2769.048269	0.533535		
Corrected Total	5197	3933.331474			

Model Fit Statistics			
R-Square	0.2960	Adj R-Sq	0.2951
AIC	-3257.5458	BIC	-3255.5408
SBC	-3205.0976	C(p)	14.3849

Type 3 Analysis of Effects				
Effect	DF	Sum of Squares	F Value	Pr > F
LG10_alcohol	1	357.6127	670.27	<.0001
LG10_chlorides	1	7.1881	13.47	0.0002
LG10_density	1	7.0921	13.29	0.0003
LG10_free_sulfur_dioxide	1	77.7614	145.75	<.0001
LG10_sulphates	1	27.9505	52.39	<.0001
LG10_total_sulfur_dioxide	1	44.9553	84.26	<.0001
LG10_volatile_acidity	1	197.1855	369.58	<.0001

# Asses

## Non-Parametric Model

### Model Comparison

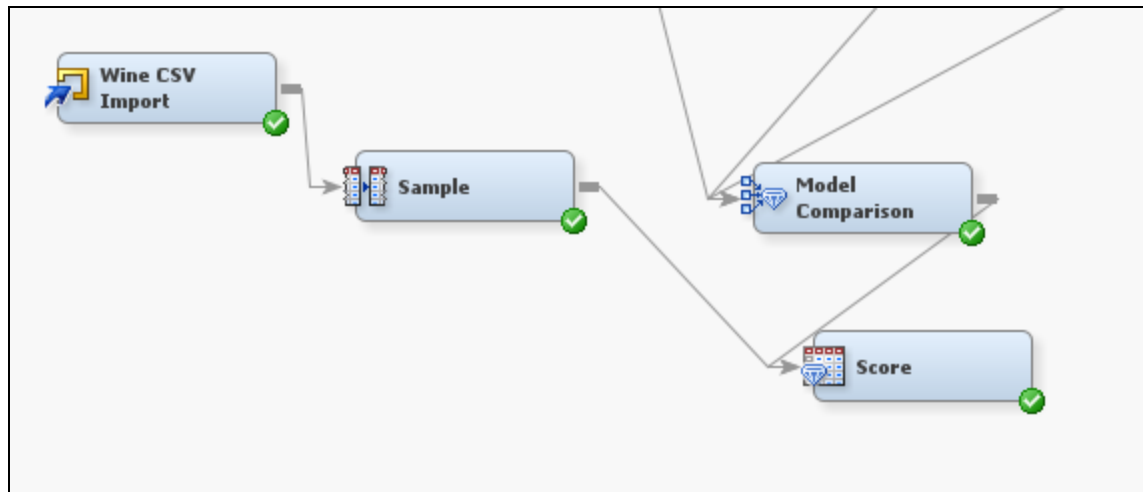
For Decision Tree, Interactive Decision Tree and Gradient Boosting, we have chosen Decision Tree as the best model based on the selection criterion on mean square error.

**Decision Tree** has the lowest mean square error compared to other nodes.

Fit Statistics												
Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Train: Average Squared Error	Train: Sum of Frequencies	Train: Sum of Case Weights Times Freq	Train: Maximum Absolute Error	Train: Sum of Squared Errors	Train: Average Squared Error	Train: Average Squared Error
Y	Tree	Tree	Decision Tr...	quality		0.453383	5198		3.448052	2356.686	0.453383	0
	Tree2	Tree2	Interactive ...	quality		0.470908	5198		3.448052	2447.78	0.470908	0
	Boost	Boost	Gradient Bo...	quality		0.497424	5198	5198	3.221013	2585.611	0.497424	0

We can also try to score the model by getting random sampling from datasets. Below we use 10% of the datasets to predict the quality of the wine and here are the results.

.. Property	Value
<b>General</b>	
Node ID	Smpl
Imported Data	
Exported Data	
Notes	
<b>Train</b>	
Variables	
Output Type	Data
Sample Method	Random
Random Seed	12345
<input checked="" type="checkbox"/> Size	
Type	Percentage
Observations	.
Percentage	10.0
Alpha	0.01
PValue	0.01
Cluster Method	Random
<input checked="" type="checkbox"/> Stratified	
Criterion	Proportional
Ignore Small Strata	No



## Score

EMWS2.Score_SCORE										
category	fixed acidity	volatile acidity	citric acid	residual sugar	free sulfur dioxide	total sulfur dioxide	Observation Number	Warnings	Note	quality
	7.9	0.6	0.06	1.6	15	59	7			5.017442
	7.9	0.43	0.21	1.6	10	37	28			5.382353
	6.9	0.685	0	2.5	22	37	32			5.783699
	7.3	0.45	0.36	5.9	12	87	40			5.783699
	8.8	0.66	0.26	1.7	4	23	51			5.382353
	7.2	0.725	0.05	4.65	4	11	65			5.033333
	7.3	0.67	0.26	1.8	16	51	84			5.112903
	5	1.02	0.04	1.4	41	85	95			5.588235
	7.8	0.41	0.68	1.7	18	69	107		115	5.263006
	7	0.69	0.08	1.8	22	89	120			5.112903
	7.9	1.04	0.05	2.2	13	29	135			5.783699
	8.3	0.715	0.15	1.8	10	52	137			5.112903
	7.6	0.49	0.26	1.6	10	88	148			5.112903
	7.3	0.33	0.47	2.1	5	11	151			5.211765
	6.8	0.6	0.18	1.9	18	86	160			5.112903
	7.8	0.63	0.48	1.7	14	96	166			5.112903
	7.5	0.52	0.42	2.3	8	38	178			5
	5.4	0.835	0.08	1.2	13	93	199			6.706667

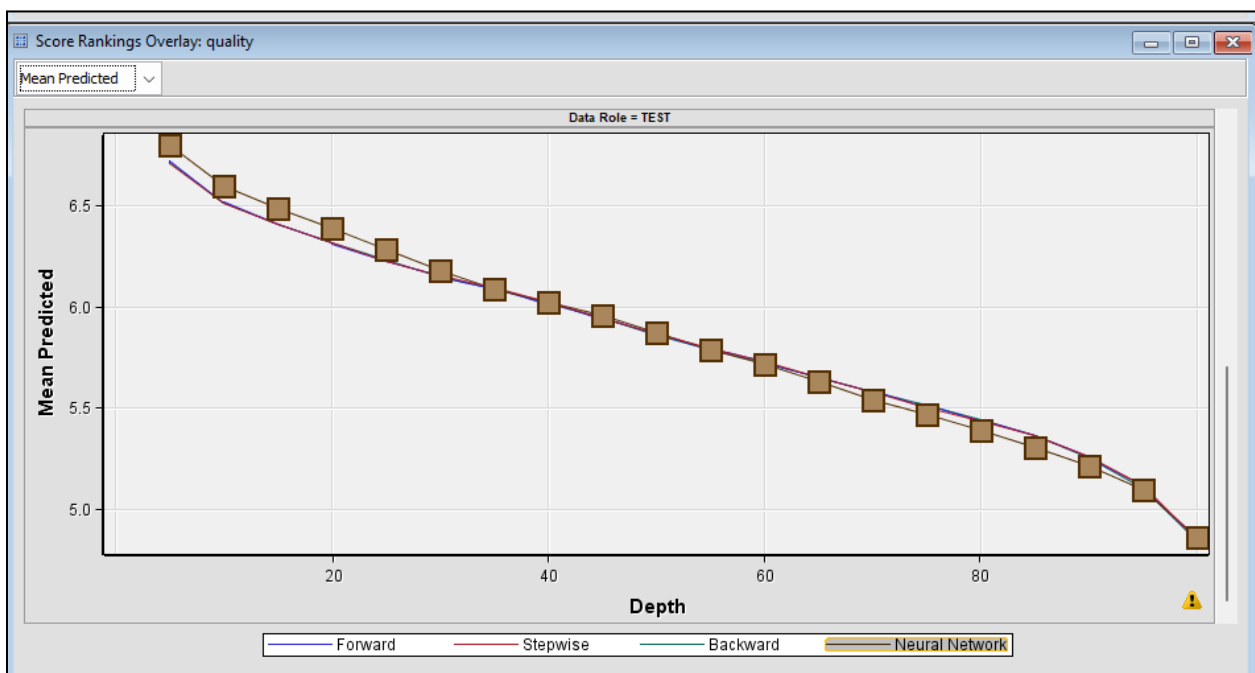
For the Score node, after running the node, we can go to the exported data to view the details of the scoring based on the sample data. As we can see that, the model actually predicts correctly and almost around the correct quality which means that the best model result from model comparison actually produces correct predictions.

## Parametric model

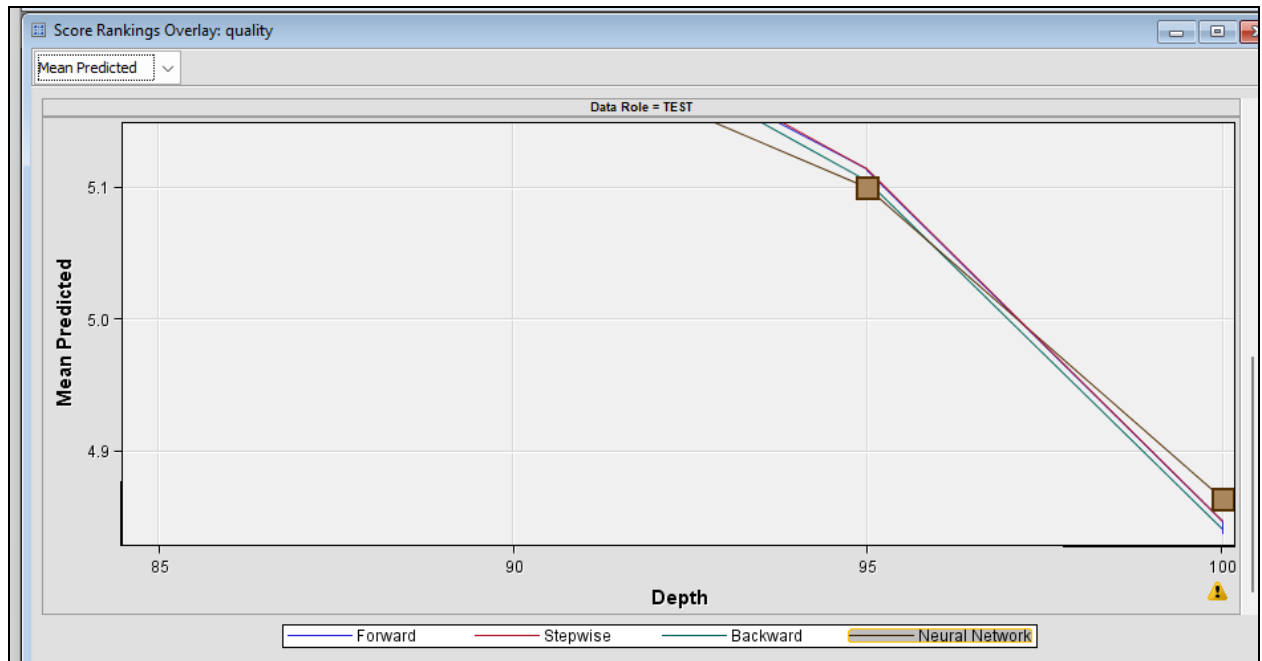
### Model Comparison

Fit Statistics												
Selected Model	Predecessor Node	Model Node	Model Description	Target	Target Label	Selection Criterion: Train: Average Squared Error	Train: Total Degrees of Freedom	Train: Degrees of Freedom for Error	Train: Model Degrees of Freedom	Train: Number of Estimated Weights	Train: Akaike's Information Criterion	Train: Schwarz Bayesian Criterion
Y	Neural	Neural	Neural Net...	quality		0.500974	5198	5158	40	40	-3512.86	
	Reg4	Reg4	Forward	quality		0.53165	5198	5186	12	12	-3259.94	
	Reg2	Reg2	Backward	quality		0.531915	5198	5188	10	10	-3261.35	
	Reg	Reg	Forward	quality		0.532714	5198	5190	8	8	-3257.55	
	Reg3	Reg3	Stepwise	quality		0.532714	5198	5190	8	8	-3257.55	

From the results from model comparison, we can see that the **Neural Network** has the highest accuracy compared to the regression model. The chosen criteria here will be based on the mean squared errors.



To interpret the cumulative lift graph, the larger the area under the line, the better the model. Since the regression charts are very close, and we may not be able to visually differentiate among them, zooming in the graph might see the slight difference between each model.



As of the score ranking between all models, we can see that **Neural Networks** actually win a bit compared to other regressions.



# Conclusion

To sum up, we ran an analysis on a dataset of wines using several input variables and quality as the target variable. We looked at parametric models including neural networks, regression, forward, backward, and stepwise regression, as well as non-parametric models like decision trees, interactive decision trees, and gradient boosting.

In terms of non-parametric models, the decision tree was the most successful. Its capacity to forecast wine quality using feature splits proved to be useful. Decision trees are also appealing in this situation because of their interpretability and simplicity.

However, the neural network beat other models such as stepwise regression, forward regression and backward regression. Given their ability to identify intricate patterns in data, neural networks proved to be more predictive when it came to the target variable of wine quality. The neural network performed well in this research because it was able to capture the complex correlations between the input features.

In conclusion, all models have their own pros and cons depending on what datasets we use. There is no best or worst model when we do data mining, all we have to do is to fine-tune each parameter and figure out configuration that can make the best accuracy for data mining. Data mining with models will make new data to be predictive and make it easier for decision making which can then improve the business.

## Future works can be done

We could try improving our models' intelligence to make better forecasts regarding wine quality in the future. We may experiment with the dataset's attributes, for example, by adding more detailed information about wine.

There are plenty of options to modify the properties of each node, such as Impute node and Transform Variables node. Although modifying them may improve results, testing out each combination is time-consuming.

Furthermore, there are more high-performance models in High Performance Data Mining (HPDM), including HP SVM and HP GLM. All these modifications are believed to develop a better model in the future.

Finally, to ensure our models perform well across a range of circumstances, we should investigate which elements are most critical for forecasting wine quality and validate our models on many datasets.

## **GitHub Link**

<https://github.com/oscarhew/WIE3007Assignment>

## **Presentation Video and Slides Link**

[https://drive.google.com/drive/folders/130L6WR\\_VSmNclYJQ-kU4RspN7xOvTTpA?usp=sharing](https://drive.google.com/drive/folders/130L6WR_VSmNclYJQ-kU4RspN7xOvTTpA?usp=sharing)

# References

Wein Plus. (n.d.). What Factors Influence Wine Quality? Retrieved from <https://magazine.wein.plus/faq/wine-quality-and-sensor-technology/what-factors-influence-wine-quality>

SPC for Excel. (n.d.). Are Skewness and Kurtosis Useful Statistics? Retrieved from <https://www.spcforexcel.com/knowledge/basic-statistics/are-skewness-and-kurtosis-useful-statistics#:~:text=The%20rule%20of%20thumb%20seems,the%20data%20are%20highly%20skewed>

Câmara, J.S., Alves, M., Marques, J.C., & Marques, M. (2021). Wines' Sensory and Chemical Characteristics: A Statistical Insight on Their Variability and Relationships. *Molecules*, 26(3), 718. <https://doi.org/10.3390/molecules26030718>

SAS Institute Inc. (n.d.). Example 18.7: Determining Outliers Using Skewness and Kurtosis. Retrieved from <https://documentation.sas.com/doc/en/emref/15.1/n0f3ix7imzm4xrn1773i0xuq47mj.htm>