UNIVERSITI MALAYA

WIE3007 DATA MINING AND WAREHOUSING

SESSION 2023/2024

SEMESTER 1

GROUP ASSIGNMENT

Chiew Kian Khoon	U2005256/1
Loi Yi Hang	U2005366/1
Sim Jia Hong	U2005316/1
Ooi Xie Gee	U2005379/1

1. Introduction

In urban environments, the impact of air quality on public health and the overall well-being of the community is a matter of paramount concern. This study revolves around a comprehensive dataset obtained from a gas multisensor device deployed in an Italian city. The dataset, spanning a year from March 2004 to February 2005, captures hourly responses from an array of 5 metal oxide chemical sensors embedded in an Air Quality Chemical Multi Sensor Device. The device was strategically positioned at road level in an area marked by significant pollution. This research represents a valuable opportunity to delve into the intricate interplay of air quality factors within the urban landscape.

2. Problem Statement

Urban areas often grapple with deteriorating air quality, posing a threat to the health and well-being of their inhabitants. Understanding the nuances of air quality dynamics, including cross-sensitivities, concept drifts, and sensor drifts, is crucial for accurate estimation of gas concentrations. This dataset, with its hourly averaged responses and ground truth concentrations from a certified analyzer, presents a unique opportunity to address these challenges. However, the presence of missing values and the potential impact of various drifts highlight the need for a sophisticated data mining approach to uncover hidden patterns and insights.

3. Objective

The primary objective of this study is to leverage data mining techniques to gain a deeper understanding of air quality dynamics in the specified Italian city. The specific goals include:

1. **Pattern Recognition**: Identify hidden patterns and trends within the hourly responses from the multisensor device, exploring correlations among the various gas concentrations and environmental factors.
2. **Sensor Drift Analysis**: Investigate and mitigate the impact of sensor drifts on concentration estimations, providing insights into the reliability and longevity of the deployed sensors.
3. **Concept Drift Exploration**: Examine instances of concept drifts and their implications on the accuracy of concentration estimations, contributing to the refinement of air quality modeling.
4. **Imputation of Missing Values**: Develop robust imputation strategies for handling missing values, ensuring the integrity of the dataset for accurate analysis.
5. **Predictive Modeling**: Build predictive models for gas concentrations based on the responses of the multisensor device, offering a tool for forecasting air quality variations in urban environments.

4. Dataset Description

Dataset: https://archive.ics.uci.edu/dataset/360/air+quality

AirQualityMeasurements (Fact Table)

FIELD NAME	DESCRIPTION
MeasurementID	Unique identifier for each measurement record
DateID	Foreign key linked to `Dim_Date` table. Represents the date on which the measurement was taken
TimeID	Foreign key linked to `Dim_Time` table. Represents the time at which the measurement was recorded
ReadingID	Foreign key linked to `Dim_SensorReading` table. Represents the sensor readings associated with this measurement
CO_Concentration	True hourly averaged concentration of Carbon Monoxide (CO) in mg/m^3. Obtained from reference analyzer
NMHC_Concentration	True hourly averaged overall Non-Methane HydroCarbons (NMHC) concentration in micro g/m^3. Obtained from reference analyzer
Benzene_Concentration	True hourly averaged Benzene concentration in micro g/m^3. Obtained from reference analyzer
NOx_Concentration	True hourly averaged Nitrogen Oxides (NOx) concentration in ppb. Obtained from reference analyzer
NO2_Concentration	True hourly averaged Nitrogen Dioxide (NO2) concentration in micro g/m^3. Obtained from reference analyzer
Temperature	Ambient temperature measured in degrees Celsius (°C)
Relative_Humidity	Relative Humidity measured in percentage (%)
Absolute_Humidity	Absolute Humidity, a measure of the amount of moisture in the air

Date (Dimension Table)

FIELD NAME	DESCRIPTION
DateID	Unique identifier for each date
Day	Day of the month (1-31)
Month	Month of the year (1-12)
Year	Year in four-digit format (YYYY)

Time (Dimension Table)

FIELD NAME	DESCRIPTION
TimeID	Unique identifier for each time record
Hour	Hour of the day in 24-hour format (0-23)
Minute	Minute of the hour (0-59)
Second	Second of the minute (0-59)

SensorReading (Dimension Table)

FIELD NAME	DESCRIPTION
ReadingID	Unique identifier for each sensor reading record
PT08_S1	Hourly averaged sensor response from PT08.S1 (tin oxide), nominally targeted for CO measurement
PT08_S2	Hourly averaged sensor response from PT08.S2 (titania), nominally targeted for NMHC measurement
PT08_S3	Hourly averaged sensor response from PT08.S3 (tungsten oxide), nominally targeted for NOx measurement
PT08_S4	Hourly averaged sensor response from PT08.S4 (tungsten oxide), nominally targeted for NO2 measurement

PT08_S5	Hourly averaged sensor response from PT08.S5 (indium oxide), nominally targeted for O3 (Ozone) measurement

5. Sampling

Google Colab Link for Sampling:
https://colab.research.google.com/drive/1PkBdJhorFGe00qjTDbFyQiv1PJVACKMv?usp=sharing

In this section, three distinct sampling methods are utilized to read data directly using the pandas library. The methods include random sampling based on proportion, grouping based on different categories of specified columns, and subsequently randomly sampling data within each group. Additionally, random sampling with weights is also employed. Diagram 5.1 below shows the raw data of our dataset.

	MeasurementID	DateID	TimeID	ReadingID	CO_Concentration	NMHC_Concentration	Benzene_Concentration	NOx_Concentration	NO2_Concentration
0	1	1	1	1	2.6	150.0	11.9	166.0	113.0
1	2	2	2	2	2.0	112.0	9.4	103.0	92.0
2	3	3	3	3	2.2	88.0	9.0	131.0	114.0
3	4	4	4	4	2.2	80.0	9.2	172.0	122.0
4	5	5	5	5	1.6	51.0	6.5	131.0	116.0
...	...	...	...	...	...	...	...	...	...
9466	9467	9467	9467	9467	NaN	NaN	NaN	NaN	NaN
9467	9468	9468	9468	9468	NaN	NaN	NaN	NaN	NaN
9468	9469	9469	9469	9469	NaN	NaN	NaN	NaN	NaN
9469	9470	9470	9470	9470	NaN	NaN	NaN	NaN	NaN
9470	9471	9471	9471	9471	NaN	NaN	NaN	NaN	NaN

9471 rows × 23 columns

Diagram 5.1: Raw Data

To implement random sampling, it's necessary to pre-specify the proportion of the sample, represented as a decimal between 0 and 1. This decimal signifies the portion of the original data intended for sampling. For example, in our code, we have entered 0.3 as the proportion, which means the code will sample 30% of the data. Additionally, we have included a seed number as a parameter when running the code. The reason for this is that by providing a fixed seed, we can obtain the same random sampling result every time the code is run. If we want to display different sampling results, we can simply change the seed. Below is Diagram 5.2, which shows the result of the sampling.

	MeasurementID	DateID	TimeID	ReadingID	CO_Concentration	NMHC_Concentration	Benzene_Concentration	NOx_Concentration	NO2_Concentration
7517	7518	7518	7518	7518	1.9	-200.0	7.4	298.0	143.0
3858	3859	3859	3859	3859	-200.0	-200.0	5.6	-200.0	-200.0
1336	1337	1337	1337	1337	2.9	-200.0	13.0	214.0	123.0
6259	6260	6260	6260	6260	4.9	-200.0	22.7	854.0	196.0
2501	2502	2502	2502	2502	2.4	-200.0	13.6	144.0	104.0
...	...	...	...	...	...	...	...	...	...
568	569	569	569	569	-200.0	-200.0	11.2	-200.0	-200.0
8707	8708	8708	8708	8708	1.8	-200.0	8.1	292.0	162.0
1888	1889	1889	1889	1889	-200.0	-200.0	9.6	-200.0	-200.0
3142	3143	3143	3143	3143	-200.0	-200.0	11.0	108.0	126.0
4202	4203	4203	4203	4203	3.0	-200.0	14.5	288.0	122.0

2841 rows × 23 columns

Diagram 5.2: Random Sampling

By utilizing the **groupby** and **apply** methods in Pandas, we can group and sample data based on different categories of a specified column. This method involves grouping the data within a DataFrame according to distinct categories of a specified column and subsequently applying random sampling to the data within each group. It's crucial to highlight that this sampling approach guarantees a balanced number of samples from each category, as the number of samples from each group is predefined and consistent. Diagram 5.3 below provides an example of sampling based on a specified category.

		MeasurementID	DateID	TimeID	ReadingID	CO_Concentration	NMHC_Concentration	Benzene_Concentration	NOx_Concentration	NO2_Conc
Month										
1.0	**7352**	7353	7353	7353	7353	1.1	-200.0	3.9	146.0	
	7209	7210	7210	7210	7210	2.0	-200.0	8.5	-200.0	
	7120	7121	7121	7121	7121	1.2	-200.0	4.7	190.0	
	7555	7556	7556	7556	7556	0.3	-200.0	4.6	193.0	
2.0	**8434**	8435	8435	8435	8435	1.0	-200.0	3.4	116.0	
	7964	7965	7965	7965	7965	1.1	-200.0	3.7	180.0	
	7873	7874	7874	7874	7874	4.1	-200.0	17.0	502.0	
	8145	8146	8146	8146	8146	2.7	-200.0	11.3	-200.0	
3.0	**8982**	8983	8983	8983	8983	1.9	-200.0	6.8	273.0	
	8597	8598	8598	8598	8598	2.1	-200.0	7.3	330.0	
	8829	8830	8830	8830	8830	0.1	-200.0	3.3	130.0	
	8536	8537	8537	8537	8537	1.4	-200.0	3.6	284.0	
4.0	**9346**	9347	9347	9347	9347	-200.0	-200.0	0.8	52.0	
	9335	9336	9336	9336	9336	1.4	-200.0	6.1	242.0	

Diagram 5.3: Sampling Based on Specific Category

Another sampling method that we have used is the weight sampling method. In traditional random sampling, each element has an equal chance of being chosen. However, in weighted sampling, the probability of selection is proportional to the assigned weight. The weights are usually positive values, and their magnitudes determine the likelihood of an element being included in the sample. In our provided code, the number of samples is determined by the **'num'** parameter, and the **'random_state'** parameter ensures that the sampling results can be consistently reproduced. Below is Diagram 5.4, showing the result of weight sampling.

	MeasurementID	DateID	TimeID	ReadingID	CO_Concentration	NMHC_Concentration	Benzene_Concentration	NOx_Concentration	NO2_Concentration
8034	8035	8035	8035	8035	1.8	-200.0	7.6	340.0	194.0
5359	5360	5360	5360	5360	-200.0	-200.0	10.0	-200.0	-200.0
4384	4385	4385	4385	4385	3.1	-200.0	19.9	590.0	171.0
3379	3380	3380	3380	3380	-200.0	-200.0	9.5	96.0	106.0
5580	5581	5581	5581	5581	1.3	-200.0	7.7	242.0	91.0
...	...	...	...	...	...	...	...	...	...
4117	4118	4118	4118	4118	-200.0	-200.0	2.8	-200.0	-200.0
9402	9403	9403	9403	9403	NaN	NaN	NaN	NaN	NaN
142	143	143	143	143	2.9	201.0	16.6	184.0	129.0
4094	4095	4095	4095	4095	1.2	-200.0	7.1	75.0	62.0
2677	2678	2678	2678	2678	2.2	-200.0	15.2	178.0	106.0

4735 rows × 23 columns

Diagram 5.4: Weight Sampling

6. Feature Tools and Star Schema Integration

6.1 Feature Tools

Jupyter Notebook Link for Feature Tools:

https://github.com/jiahongggg/Data-Mining-Warehousing-A2/blob/main/Tools/Feature%20Tools/Assignment%202.ipynb

In [1]:
```
import featuretools as ft
import pandas as pd
```

In [2]:
```
# Read the CSV file into a DataFrame
df = pd.read_csv('new_AirQualityUCI.csv')

# Create a new DataFrame "AirQualityMeasurements"
AirQuality = df[['MeasurementID','ReadingID','DateID', 'TimeID']]

# Create a new DataFrame "AirQualityMeasurements"
AirQualityMeasurements = df[['MeasurementID','CO_Concentration', 'NMHC_Concentration', 'Benzene_Concentration', '

# Create a new DataFrame "SensorReadings"
SensorReadings = df[['ReadingID', 'PT08_S1', 'PT08_S2', 'PT08_S3', 'PT08_S4', 'PT08_S5']]
```

In [3]:
```
AirQuality.head()
```

Out[3]:

	MeasurementID	ReadingID	DateID	TimeID
0	1	1	1	1
1	2	2	2	2
2	3	3	3	3
3	4	4	4	4
4	5	5	5	5

Diagram 6.1: Featuretools Code Repository (1)

Out[4]:

	MeasurementID	CO_Concentration	NMHC_Concentration	Benzene_Concentration	NOx_Concentration	NO2_Concentration	Te
0	1	2.6	150.0	11.9	166.0	113.0	
1	2	2.0	112.0	9.4	103.0	92.0	
2	3	2.2	88.0	9.0	131.0	114.0	
3	4	2.2	80.0	9.2	172.0	122.0	
4	5	1.6	51.0	6.5	131.0	116.0	

In [5]:
```
SensorReadings.head()
```

Out[5]:

	ReadingID	PT08_S1	PT08_S2	PT08_S3	PT08_S4	PT08_S5
0	1	1360.0	1046.0	1056.0	1692.0	1268.0
1	2	1292.0	955.0	1174.0	1559.0	972.0
2	3	1402.0	939.0	1140.0	1555.0	1074.0
3	4	1376.0	948.0	1092.0	1584.0	1203.0
4	5	1272.0	836.0	1205.0	1490.0	1110.0

Diagram 6.2: Featuretools Code Repository (2)

In [6]:

```
dataframes = {
    "AirQualityMeasurements": (AirQualityMeasurements, "MeasurementID"),
    "SensorReadings": (SensorReadings, "ReadingID"),
    "AirQuality": (AirQuality, "DateID"),
}
```

In [7]:

```
relationships = [
    ("SensorReadings", "ReadingID", "AirQuality", "ReadingID"),
    ("AirQualityMeasurements", "MeasurementID","AirQuality","MeasurementID")
]
```

In [8]:

```
feature_matrix_SensorReadings, features_defs = ft.dfs(
    dataframes=dataframes,
    relationships=relationships,
    target_dataframe_name="SensorReadings",
)
feature_matrix_SensorReadings
```

Out[8]:

	PT08_S1	PT08_S2	PT08_S3	PT08_S4	PT08_S5	COUNT(AirQuality)	MAX(AirQuality.TimeID)	MEAN(AirQuality.Time
ReadingID								
1	1360	1046	1056	1692	1268	1	1.0	
2	1292	955	1174	1559	972	1	2.0	
3	1402	939	1140	1555	1074	1	3.0	
4	1376	948	1092	1584	1203	1	4.0	
5	1272	836	1205	1490	1110	1	5.0	
...	...	...	...	...	...	...	...	
9467	<NA>	<NA>	<NA>	<NA>	<NA>	1	9467.0	946
9468	<NA>	<NA>	<NA>	<NA>	<NA>	1	9468.0	946
9469	<NA>	<NA>	<NA>	<NA>	<NA>	1	9469.0	946
9470	<NA>	<NA>	<NA>	<NA>	<NA>	1	9470.0	947
9471	<NA>	<NA>	<NA>	<NA>	<NA>	1	9471.0	947

9471 rows × 60 columns

Diagram 6.3: Featuretools Code Repository (3)

6.2 Star Schema

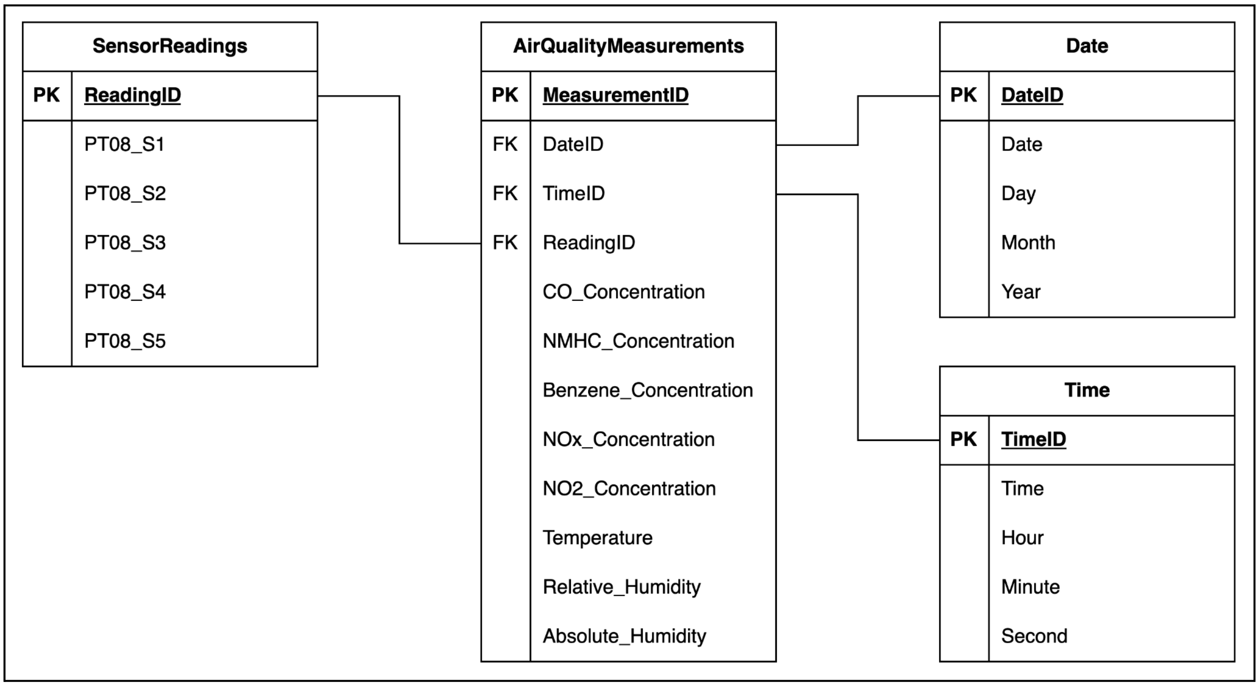


Diagram 6.4: Star Schema

7. Explore

Exploratory Data Analysis

Exploratory Data Analysis (EDA) employs data visualization to thoroughly inspect, analyze, and summarize essential aspects of datasets. It aids in efficiently manipulating data sources to extract desired insights, facilitating the identification of patterns, anomalies, hypothesis testing, and assumption verification. EDA plays a crucial role in assessing the suitability of chosen statistical techniques before delving into data analysis.

7.1 Average Temperature

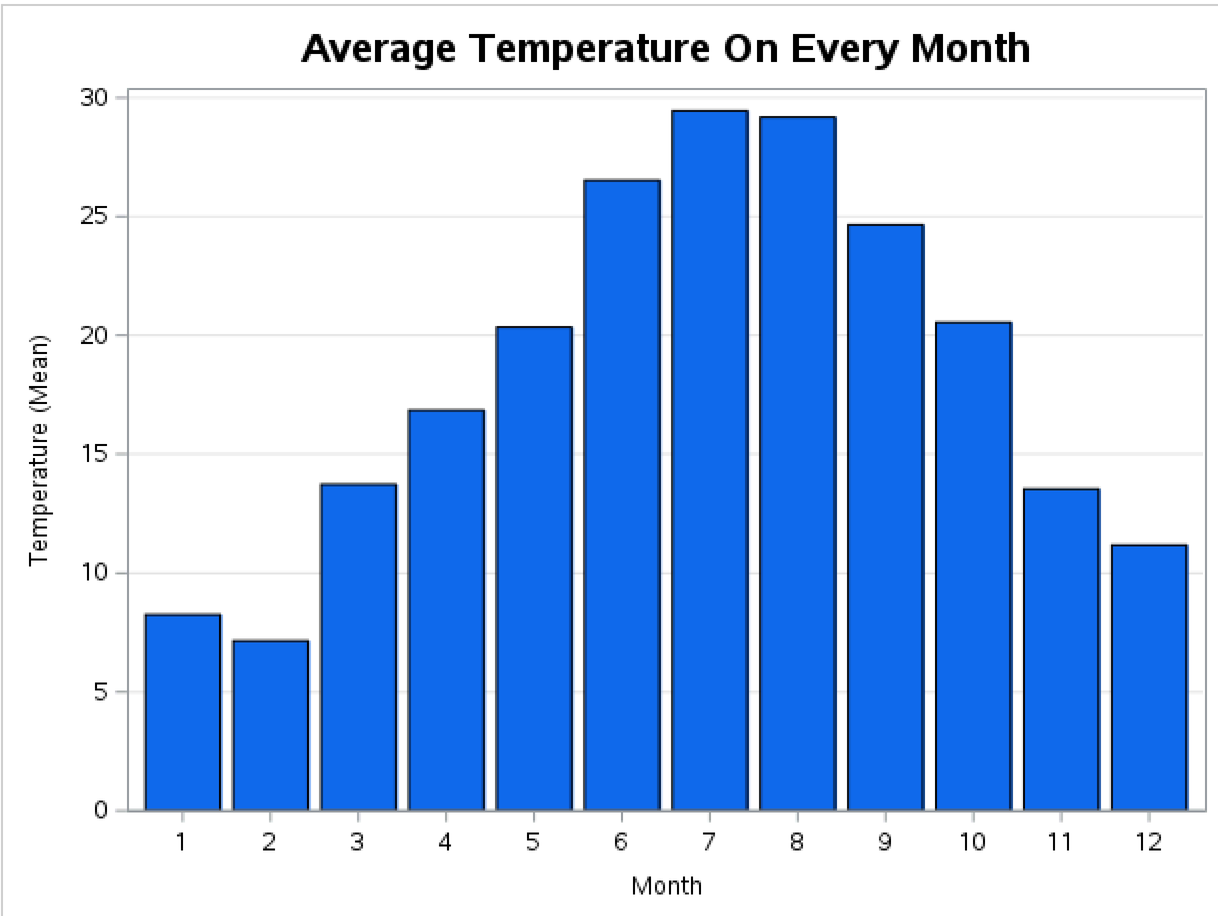


Diagram 7.1: Average Temperature Of Italy Base On Month

The diagram 7.1 vividly illustrates the monthly temperature variations in Italy through a bar chart. July stands out with the highest temperature, peaking at approximately 29 degrees

Celsius, while February records the lowest temperature at around 7 degrees Celsius. This clear distinction emphasizes the seasonal temperature fluctuations experienced in Italy, with summer months being notably warmer and winter months considerably cooler.

7.2 Correlation of Humidity & and Temperature

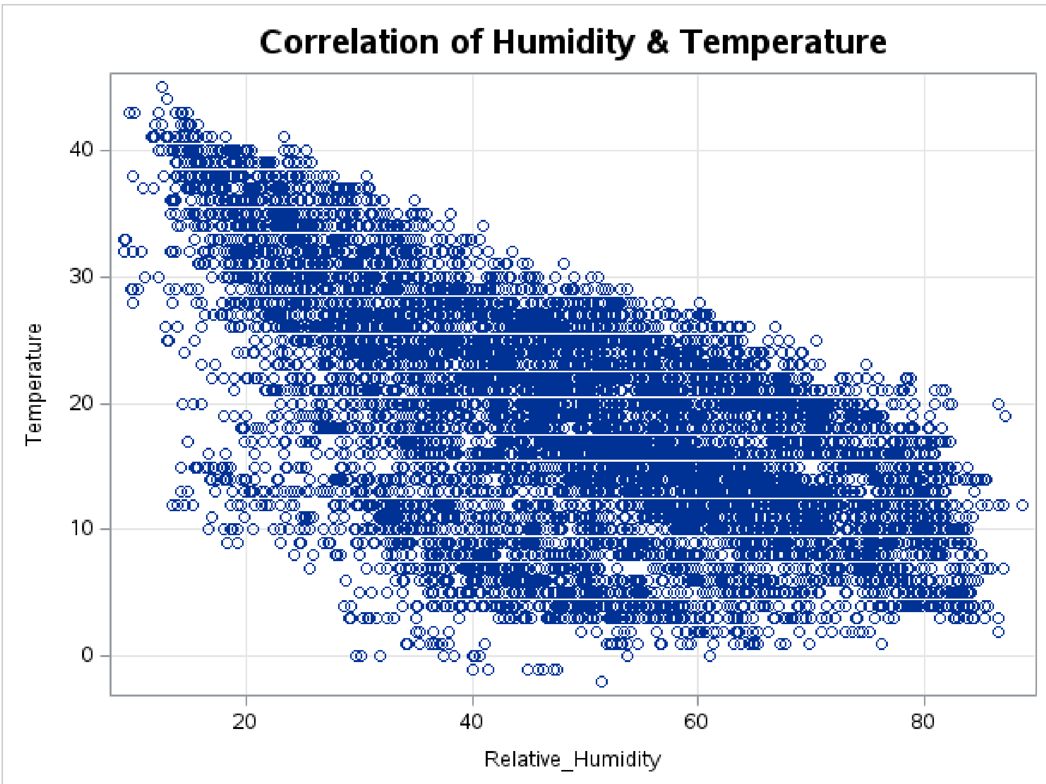


Diagram 7.2: Correlation of Humidity & Temperature

The diagram 7.2 depicting the correlation between humidity and temperature reveals a noticeable trend. It is evident that as relative humidity decreases, there is a corresponding drop in temperature. This observation indicates an inverse relationship between humidity and temperature, suggesting that lower humidity levels are associated with cooler temperatures. Hence, we may infer that temperature may be the factor that can affect humidity.

7.3 Correlation between NO2 and Benzene Concentration

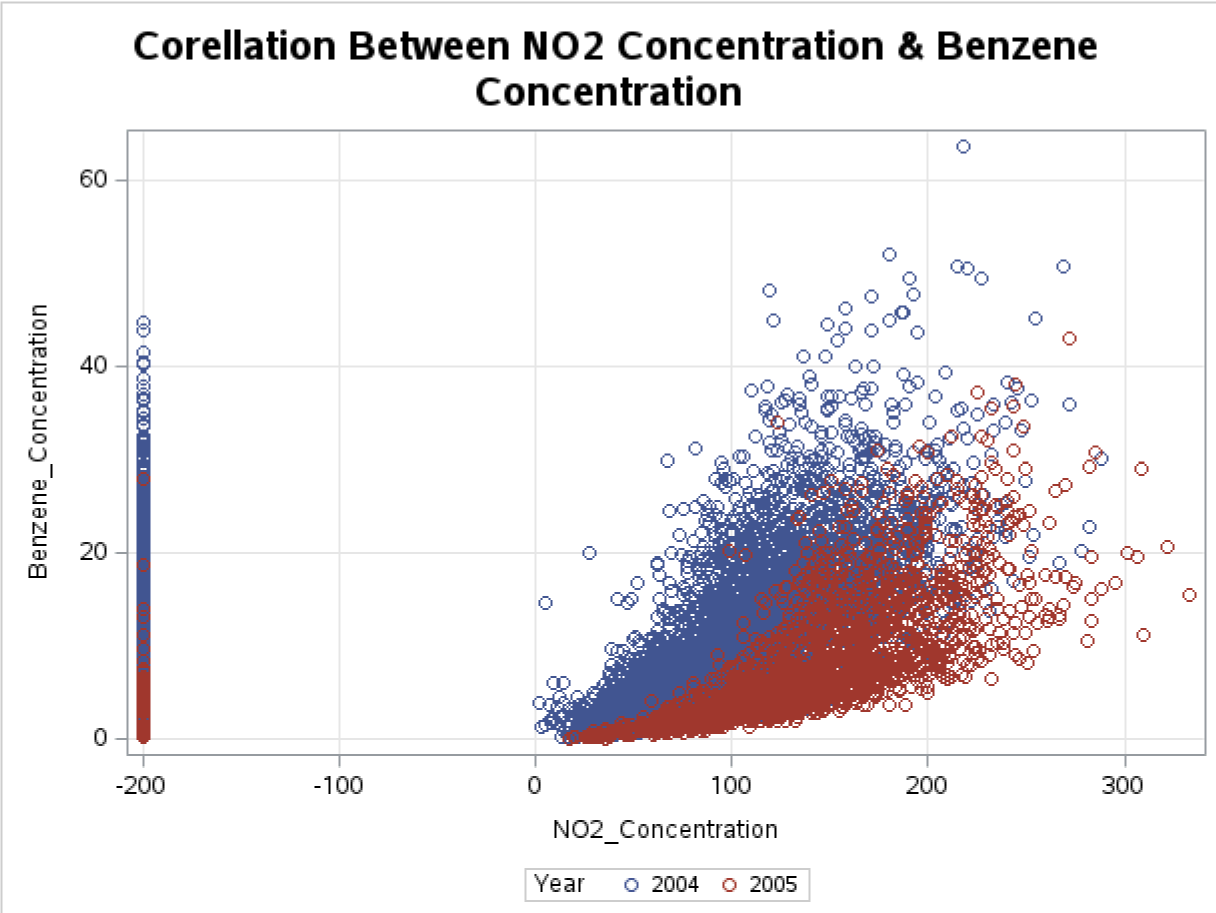


Diagram 7.3: Correlation Of NO2 & Benzene

The diagram 7.3, depicting the correlation between NO2 and benzene with different colors representing various years, reveals a notable trend. Upon examination, it becomes evident that the concentrations of both NO2 and benzene decreased in 2005 compared to 2004. This observation suggests an overall reduction in the levels of these pollutants between the two years, indicating a potential improvement in air quality and environmental conditions during that period. Furthermore, the graph illustrates a positive correlation between the concentrations of benzene and NO2. As the concentration of benzene increases, there is a corresponding increase in the concentration of NO2. This positive correlation suggests a potential relationship or co-occurrence between the two pollutants.

8. Modify

8.1 Talend

Data preprocessing is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. Data preprocessing helps ensure the quality of the data, correct errors, handle missing values, and prepare the dataset for further exploration. In this project, we have chosen to integrate Talend Data Preparation as the tool to perform the data preprocessing task on the selected dataset.

The air quality data was obtained from UC Irvine. After exploring the data, we discovered that in some rows, only MeasurementID, DateID, TimeID, and ReadingID have values, while the remaining columns are blank.

The dataset is loaded into Talend Data Preparation to perform data preprocessing.

Below are the steps we use to preprocess our dataset:

1. Remove negative values in the CO_Concentration column.
2. Delete all rows consisting of missing values.
3. Remove negative values such as -200 in the NMHC_Concentration column.
4. Remove negative values in the Benzene_Concentration column.
5. Remove negative values in the NOx_Concentration column.
6. Delete the unnecessary columns 'Minute' and 'Second.'

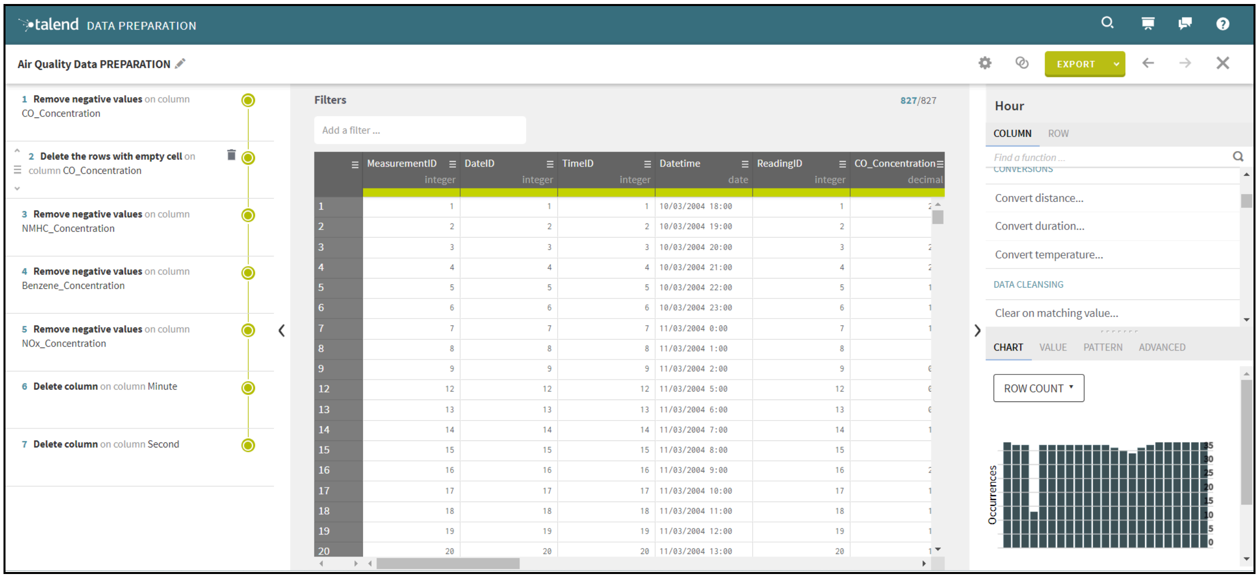


Diagram 8.1: Talend Data Preprocessing

Data Cleaning

- **Removing Negative Values:** We start by removing negative values in the CO_Concentration, NMHC_Concentration, Benzene_Concentration, and NOx_Concentration columns. Negative values in these columns might result from sensor errors or missing data, potentially distorting the analysis and leading to incorrect conclusions. By eliminating these values, we ensure that the data accurately represents air quality measurements.
- **Deleting Rows with Missing Values:** We also delete all rows that consist of missing values. Rows containing missing data can introduce complications during data analysis and modeling. By removing them, we ensure that the dataset is complete and ready for further processing.

Data Reduction

- **Removing Unnecessary Columns:** To streamline the dataset, we eliminate the unnecessary 'Minute' and 'Second' columns. These columns may not be relevant to the analysis. By removing extraneous data, we reduce the dataset's dimensionality, making it easier to manage and process. Additionally, this action helps us focus on the most pertinent data for our analysis.

8.2 SAS Enterprise Miner

Data Preparation and Modeling

- **Importing into SAS Enterprise Miner:** Following data cleaning and reduction, we import the preprocessed dataset into SAS Enterprise Miner using the File Import Node.
- **Data Partitioning:** The dataset is then connected to the Data Partitions Node, which helps us allocate the data according to the desired percentages. Using this node, we assign 70% of the data to our training dataset, 20% to the validation dataset, and 10% to our test dataset.
- **Variable Transformation:** Before submitting the data to the regression and neural networking modeling nodes, we perform variable transformation. Transforming the data can enhance model responsiveness by stabilizing variance, removing nonlinearity, improving additivity, and mitigating non-normality. These transformations contribute to better model fits.

Transformations Statistics

Source	Method	Variable Name	Formula	Number of Levels	Non Missing	Missing	Minimum	Maximum	Mean	Standard Deviation	Skewness	Kurtosis	Label
Input	Original	Absolute Humidity		.	579	0	0.4023	1.4852	0.832248	0.182761	0.760412	0.958562	
Input	Original	Benzene Concent...		.	579	0	0.5	38.4	10.6468	7.31004	0.955839	0.577603	
Input	Original	CO Concentration		.	579	0	0.3	8.1	2.339724	1.382382	0.979696	0.705202	
Input	Original	NMHC Concentra...		.	579	0	7	1084	228.6598	207.3925	1.445405	1.547807	
Input	Original	NO2 Concentration		.	579	0	20	196	100.0363	31.48148	0.112398	-0.12285	
Input	Original	NOx Concentration		.	579	0	12	478	143.342	81.31736	0.851289	0.433524	
Input	Original	PT08 S1		.	579	0	753	1975	1204.801	240.7075	0.515333	-0.25945	
Input	Original	PT08 S2		.	579	0	448	1737	962.0691	264.1464	0.345637	-0.49448	
Input	Original	PT08 S3		.	579	0	494	1935	965.9119	264.5125	0.81433	0.652152	
Input	Original	PT08 S4		.	579	0	955	2665	1595.301	300.9163	0.606134	0.032453	
Input	Original	PT08 S5		.	579	0	263	2359	1036.238	403.8245	0.316521	-0.40933	
Input	Original	Relative Humidity		.	579	0	14.9	83.2	49.13126	15.48082	0.014247	-0.9003	
Output	Computed	LG10 PT08 S1	log10(PT08 S1 +...	.	579	0	2.877371	3.295787	3.072823	0.085508	0.125483	-0.72195	Transformed PT0...
Output	Computed	LG10 PT08 S2	log10(PT08 S2 +...	.	579	0	2.652246	3.24005	2.966926	0.122097	-0.21199	-0.60217	Transformed PT0...
Output	Computed	LG10 PT08 S3	log10(PT08 S3 +...	.	579	0	2.694605	3.286905	2.96992	0.115352	0.160465	-0.45519	Transformed PT0...
Output	Computed	LG10 PT08 S4	log10(PT08 S4 +...	.	579	0	2.980458	3.42586	3.195644	0.080238	0.180058	-0.46249	Transformed PT0...
Output	Computed	LG10 PT08 S5	log10(PT08 S5 +...	.	579	0	2.421604	3.372912	2.979031	0.186829	-0.55778	-0.26754	Transformed PT0...
Output	Computed	OPT Absolute Hu...	Optimal Binning(4)	3	.	0	.	.	.	.	.	.	Transformed Abso...
Output	Computed	OPT Benzene C...	Optimal Binning(4)	3	.	0	.	.	.	.	.	.	Transformed Ben...
Output	Computed	OPT CO Concen...	Optimal Binning(4)	3	.	0	.	.	.	.	.	.	Transformed CO ...
Output	Computed	OPT NMHC Con...	Optimal Binning(4)	3	.	0	.	.	.	.	.	.	Transformed NMH...
Output	Computed	OPT NO2 Conce...	Optimal Binning(4)	3	.	0	.	.	.	.	.	.	Transformed NO2...
Output	Computed	OPT NOx Conce...	Optimal Binning(4)	2	.	0	.	.	.	.	.	.	Transformed NOx...
Output	Computed	OPT Relative Hu...	Optimal Binning(4)	4	.	0	.	.	.	.	.	.	Transformed Rela...

Diagram 8.2: Transformations Statistics

9. Techniques and Algorithms

9.1 Association Rule

Association Rule is a data mining method used to reveal connections between different variables within a dataset. It is often utilized in market basket analysis to uncover patterns in consumer purchasing behavior. For this project, our association rule is carried out using the day of the date as the ID, and our target variable is the temperature category. Our objective is to identify the relationships between the day and temperature category. This allows us to find rules that may predict the occurrence of a temperature category based on the occurrences of other temperature categories.

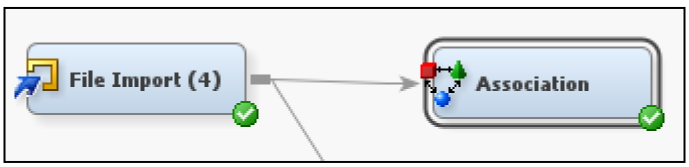


Diagram 9.1: Association Rule Diagram

Rule Description

Map	Rule
RULE1	High ==> Low
RULE2	Mid & High ==> Low
RULE3	High ==> Mid & Low
RULE4	Low ==> High
RULE5	Mid & Low ==> High
RULE6	Low ==> Mid & High
RULE7	Mid ==> High
RULE8	Mid ==> Low
RULE9	Mid ==> Low & High
RULE10	High ==> Mid
RULE11	Low ==> Mid
RULE12	Low & High ==> Mid

Diagram 9.2: Association Rule Description

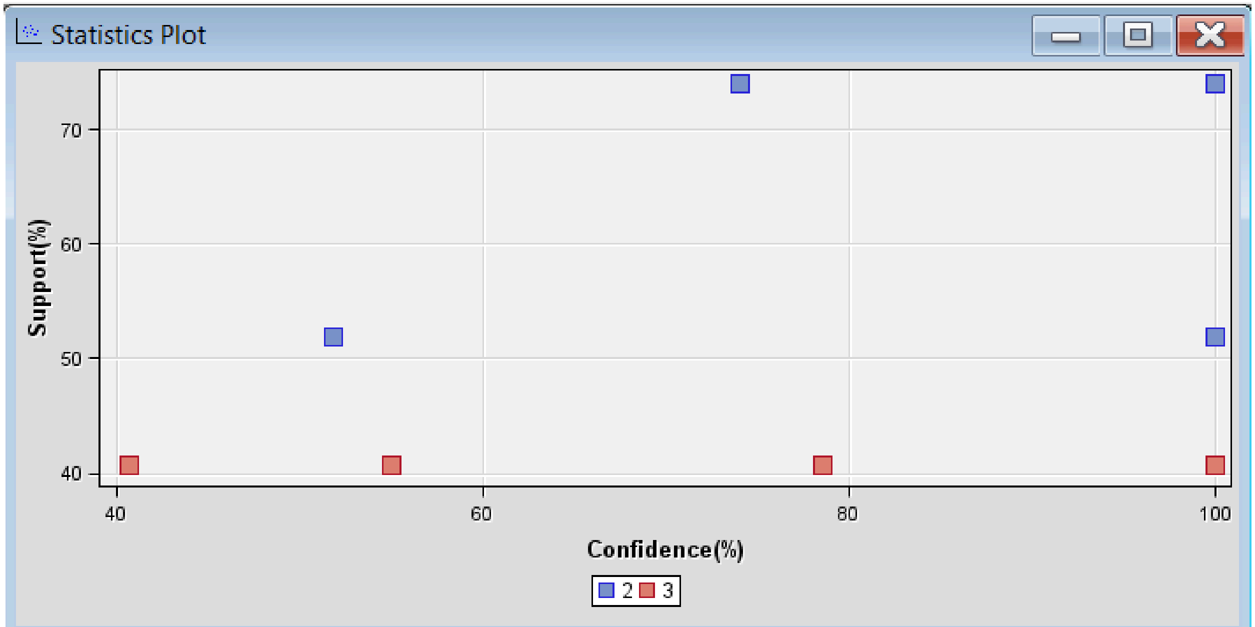


Diagram 9.3: Association Rule Statistic

Diagram 9.3 above displays the association rules statistics plot. The chart represents association rules, with the X-axis representing rule confidence and the Y-axis representing rule support. Each point on the graph corresponds to a specific association rule. Notably, a standout rule appears in the top right corner, symbolizing a robust rule with both high support and confidence. This rule is generally regarded as the most trustworthy. Conversely, the rule found in the bottom-left corner has the lowest support and confidence, suggesting a less reliable association rule.

Diagram 9.4: Association Rule Matrix

The illustration in Diagram 9.4 displays a rule matrix where the y-axis corresponds to the left side of association rules, and the x-axis corresponds to the right side of rules. This matrix serves the purpose of elucidating and examining the conditions that give rise to associations and their corresponding outcomes. Utilizing this rule matrix facilitates a thorough comprehension of the patterns and connections between the components on the left-hand side and the results on the right-hand side within association rules. By referring to the results, we can identify that the right-hand side of the rule consists of 'Mid,' 'Low,' 'Mid & Low,' 'High,' 'Mid & High,' and 'Low & High,' while the left-hand side of the rule consists of 'Mid,' 'Mid & High,' 'High,' 'Mid & Low,' and 'Low & High.'

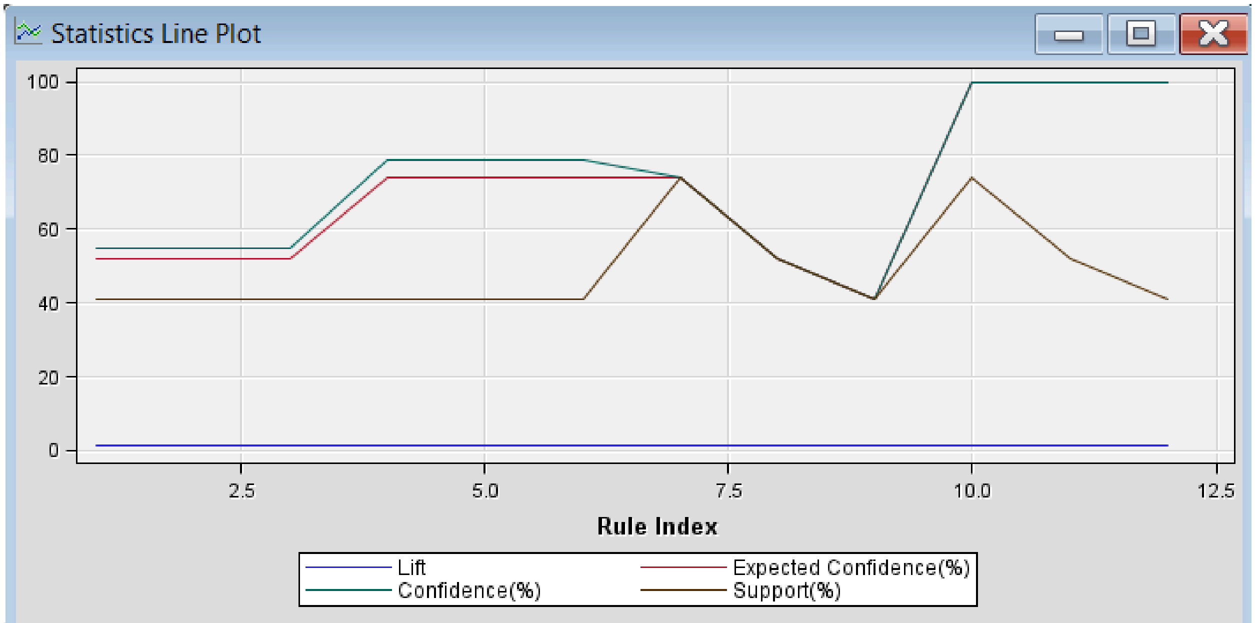


Diagram 9.5: Association Rule Line Plot

Diagram 9.5 above presents the statistics line plot. From the graph, we can observe that the blue line (lift) maintains a constant value of 1.06. This suggests that the antecedent and consequent are unrelated, and the rule does not offer any supplementary information. Next, the green line (confidence) exhibits a sudden drop from 80% to 40% but eventually achieves consistency at 100%. This indicates that, in the end, the presence of the antecedent guarantees the presence of the consequent. As for the brown line (support), it also displays an up and down trend, but overall, the percentage remains below 50%. This might indicate that the rules are only present in a small portion of the transactions.

Rules Table

Relations	Expected Confidence(%)	Confidence(%)	Support(%)	Lift	Transaction Count	Rule	Left Hand of Rule	Right Hand of Rule	Rule Item 1	Rule Item 2	Rule Item 3	Rule Item 4	Rule Item 5	Rule Index	Transpose Rule
2	51.85	55.00	40.74	1.06	11.00	High ==>...	High	Low	High	===>	Low			1	1
3	51.85	55.00	40.74	1.06	11.00	Mid & Hig...	Mid & High	Low	Mid	High	===>	Low		2	1
3	51.85	55.00	40.74	1.06	11.00	High ==>...	High	Mid & Low	High	===>	Mid	Low		3	1
2	74.07	78.57	40.74	1.06	11.00	Low ==> ...	Low	High	Low	===>	High			4	1
3	74.07	78.57	40.74	1.06	11.00	Mid & Lo...	Mid & Low	High	Mid	Low	===>	High		5	1
3	74.07	78.57	40.74	1.06	11.00	Low ==> ...	Low	Mid & High	Low	===>	Mid	High		6	1
2	74.07	74.07	74.07	1.00	20.00	Mid ==> ...	Mid	High	Mid	===>	High			7	1
2	51.85	51.85	51.85	1.00	14.00	Mid ==> ...	Mid	Low	Mid	===>	Low			8	1
3	40.74	40.74	40.74	1.00	11.00	Mid ==> ...	Mid	Low & High	Mid	===>	Low	High		9	1
2	100.00	100.00	74.07	1.00	20.00	High ==>...	High	Mid	High	===>	Mid			10	1
2	100.00	100.00	51.85	1.00	14.00	Low ==> ...	Low	Mid	Low	===>	Mid			11	1
3	100.00	100.00	40.74	1.00	11.00	Low & Hi...	Low & High	Mid	Low	High	===>	Mid		12	1

Diagram 9.6: Association Rule Table

9.2 Sequence Analysis

In this context, sequence analysis plays a crucial role in converting unprocessed data into practical and meaningful insights. For conducting sequence analysis on our dataset, the initial step involves redefining the dataset role to "Transaction" and designating "Day" as the ID, "Category" as the Target, and "Hour" as the Sequence. Subsequently, an Association node is introduced to the diagram workspace, linked to the dataset node. This Association node is then renamed as "Sequence Analysis."

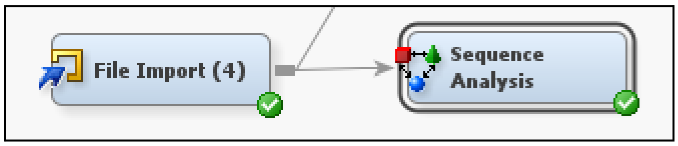


Diagram 9.7:Sequence Analysis Diagram

Sequence Report

Chain Length	Transaction Count	Support (%)	Confidence (%)	Pseudo Lift	Rule	Chain Item 1	Chain Item 2	Chain Item 3	Rule Index	Left Hand of Rule	Right Hand of Rule
2	26	96.30	96.30	0.96	Mid ==> Mid	Mid	Mid		1	Mid	Mid
3	26	96.30	100.00	1.00	Mid ==> Mid ==> Mid	Mid	Mid	Mid	2	Mid ==> Mid	Mid
2	20	74.07	100.00	1.00	High ==> Mid	High	Mid		3	High	Mid
3	20	74.07	100.00	1.00	High ==> Mid ==> Mid	High	Mid	Mid	4	High ==> Mid	Mid
2	19	70.37	95.00	1.28	High ==> High	High	High		5	High	High
2	19	70.37	70.37	0.95	Mid ==> High	Mid	High		6	Mid	High
3	19	70.37	73.08	0.99	Mid ==> Mid ==> High	Mid	Mid	High	7	Mid ==> Mid	High
3	19	70.37	100.00	1.00	High ==> High ==> Mid	High	High	Mid	8	High ==> High	Mid
3	19	70.37	100.00	1.00	Mid ==> High ==> Mid	Mid	High	Mid	9	Mid ==> High	Mid
3	18	66.67	94.74	1.28	High ==> High ==> High	High	High	High	10	High ==> High	High
3	18	66.67	94.74	1.28	Mid ==> High ==> High	Mid	High	High	11	Mid ==> High	High
2	13	48.15	48.15	0.93	Mid ==> Low	Mid	Low		12	Mid	Low
2	13	48.15	92.86	0.93	Low ==> Mid	Low	Mid		13	Low	Mid
3	13	48.15	50.00	0.96	Mid ==> Mid ==> Low	Mid	Mid	Low	14	Mid ==> Mid	Low
3	13	48.15	100.00	1.00	Low ==> Mid ==> Mid	Low	Mid	Mid	15	Low ==> Mid	Mid
2	12	44.44	85.71	1.65	Low ==> Low	Low	Low		16	Low	Low
2	12	44.44	100.00	1.00	Mid & High ==> Mid	Mid & High	Mid		17	Mid & High	Mid
2	12	44.44	44.44	1.00	Mid ==> Mid & High	Mid	Mid & High		18	Mid	Mid & High
3	12	44.44	92.31	0.92	Mid ==> Low ==> Mid	Mid	Low	Mid	19	Mid ==> Low	Mid
3	12	44.44	100.00	1.00	Mid & High ==> Mid ==> Mid	Mid & High	Mid	Mid	20	Mid & High ==> Mid	Mid
3	12	44.44	100.00	1.00	Mid ==> Mid & High ==> Mid	Mid	Mid & High	Mid	21	Mid ==> Mid & High	Mid
3	12	44.44	46.15	1.04	Mid ==> Mid ==> Mid & High	Mid	Mid	Mid & High	22	Mid ==> Mid	Mid & High
2	11	40.74	78.57	1.06	Low ==> High	Low	High		23	Low	High
2	11	40.74	91.67	1.24	Mid & High ==> High	Mid & High	High		24	Mid & High	High
3	11	40.74	100.00	1.35	Mid & High ==> High ==> High	Mid & High	High	High	25	Mid & High ==> High	High

Diagram 9.8:Sequence Analysis Report

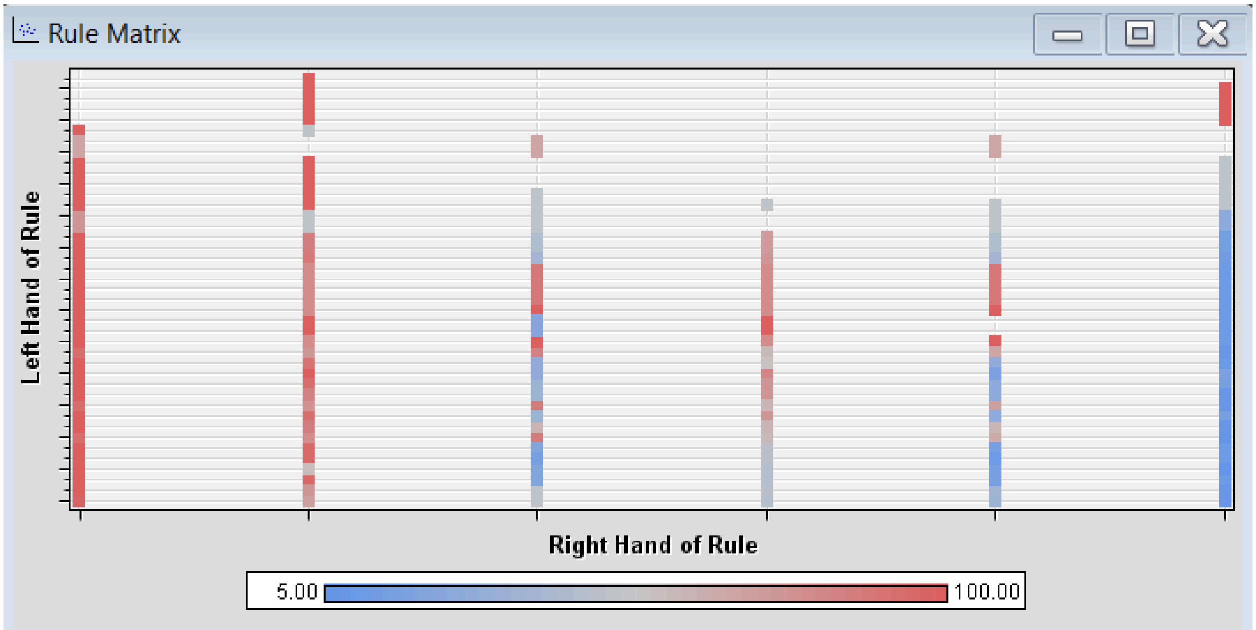


Diagram 9.9:Rule Matrix

Diagram 9.9 depicts a Rule Matrix that employs a scatter plot to create a graph illustrating the correlation between the Left Hand of Rule and the Right Hand of Rule.

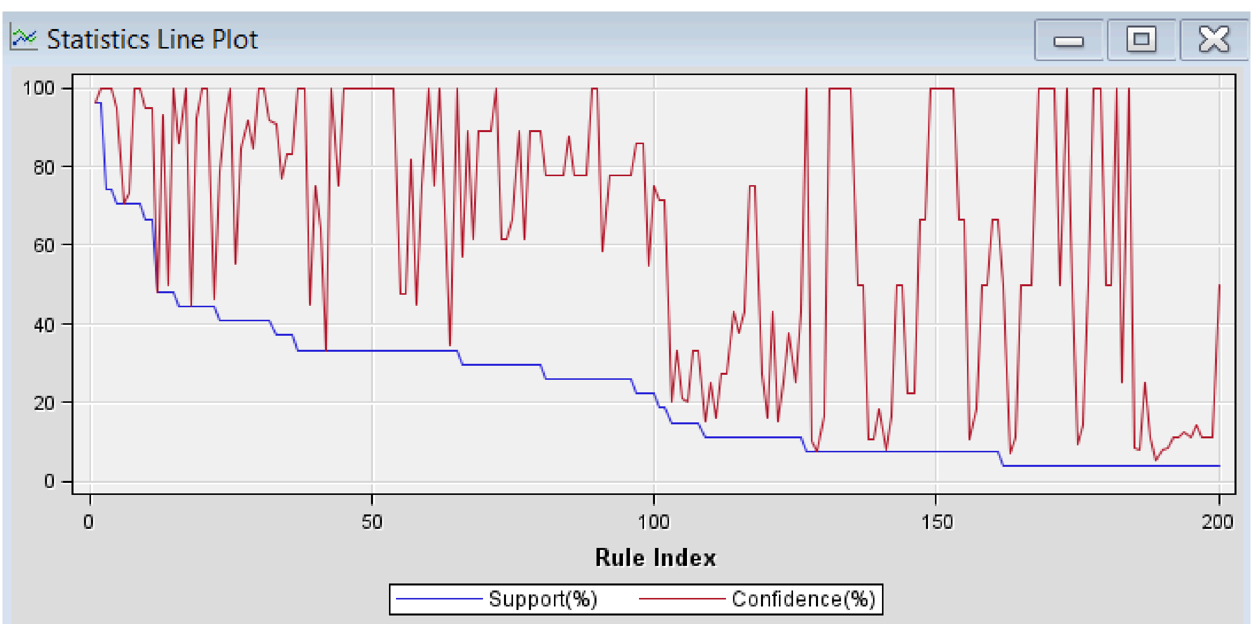


Diagram 9.10: Statistics Line Plot

```
Rule Statistics

The MEANS Procedure

Variable    Label                    Minimum        Maximum           Mean
--------------------------------------------------------------------------------
NITEMS      Chain Length           2.0000000      3.0000000      2.8200000
COUNT       Transaction Count      1.0000000     26.0000000      6.1150000
SUPPORT     Support(%)             3.7037037     96.2962963     22.6481481
CONF        Confidence(%)          5.0000000    100.0000000     63.7409009
--------------------------------------------------------------------------------
```

Diagram 9.11: Rule Statistics

Diagrams 9.10 and 9.11 above show an overview of the association rule performance within the dataset. From the diagrams, we can observe that the average confidence of 63.74% indicates a relatively strong association between the antecedent and consequent parts of the rules. However, the lower support metric, averaging at 22.64%, suggests that these rules might not be widely applicable across the entire dataset. In short, the high confidence but low support imply that the rules are moderately reliable but may not cover a large portion of the dataset.

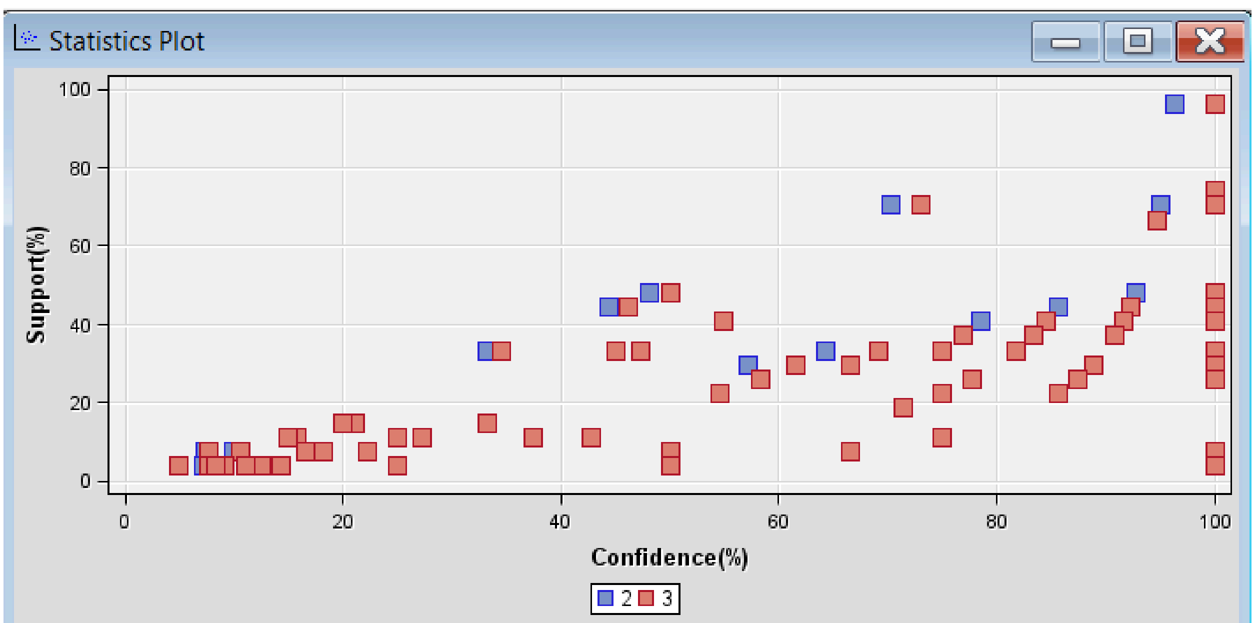


Diagram 9.12: Statistics Plot

```
Sequence Report

The FREQ Procedure

                         Chain Length

                                          Cumulative    Cumulative
NITEMS    Frequency     Percent     Frequency       Percent
-------------------------------------------------------------
     2           36       18.00            36         18.00
     3          164       82.00           200        100.00
```

Diagram 9.13: Sequence Report

According to Diagrams 9.12 and 9.13, there are two distinct n-items: 2 and 3. The n-items of size 2 account for 18% in both frequency and percentage of itemsets. In contrast, n-items of size 3 have a frequency and percentage of 82%.

9.3 Time Series Clustering

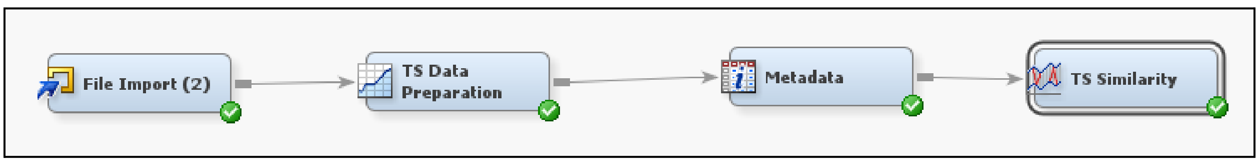


Diagram 9.14: Time Clustering Diagram

Report	
Similarity Plot Maximum	5
Preference of Similarity Plot	Most Similar
Output Data Set	Clustering Segment

Diagram 9.15: TS Similarity Node Properties

```
Exported Attributes for TRAIN Port
(maximum 500 observations printed)

Variable              Measurement
  Name     Role          Level       Creator    Label

 Day       TIMEID     INTERVAL
 _TS_01    TARGET     INTERVAL        TSDP      Temperature 1
 _TS_02    INPUT      INTERVAL        TSDP      Temperature 2
 _TS_03    INPUT      INTERVAL        TSDP      Temperature 3
 _TS_04    INPUT      INTERVAL        TSDP      Temperature 4
 _TS_05    INPUT      INTERVAL        TSDP      Temperature 5
 _TS_06    INPUT      INTERVAL        TSDP      Temperature 6
 _TS_07    INPUT      INTERVAL        TSDP      Temperature 7
 _TS_08    INPUT      INTERVAL        TSDP      Temperature 8
 _TS_09    INPUT      INTERVAL        TSDP      Temperature 9
 _TS_10    INPUT      INTERVAL        TSDP      Temperature 10
 _TS_11    INPUT      INTERVAL        TSDP      Temperature 11
 _TS_12    INPUT      INTERVAL        TSDP      Temperature 12
 _TS_13    INPUT      INTERVAL        TSDP      Temperature 13
 _TS_14    INPUT      INTERVAL        TSDP      Temperature 14
 _TS_15    INPUT      INTERVAL        TSDP      Temperature 15
 _TS_16    INPUT      INTERVAL        TSDP      Temperature 16
 _TS_17    INPUT      INTERVAL        TSDP      Temperature 17
 _TS_18    INPUT      INTERVAL        TSDP      Temperature 18
 _TS_19    INPUT      INTERVAL        TSDP      Temperature 19
 _TS_20    INPUT      INTERVAL        TSDP      Temperature 20
 _TS_21    INPUT      INTERVAL        TSDP      Temperature 21
 _TS_22    INPUT      INTERVAL        TSDP      Temperature 22
 _TS_23    INPUT      INTERVAL        TSDP      Temperature 23
```

Diagram 9.16: Time Series Data Preparation

Diagram 9.16 shows the results of the **TS Data Preparation Node** result. From the node we can see that we have 23 different time variables.

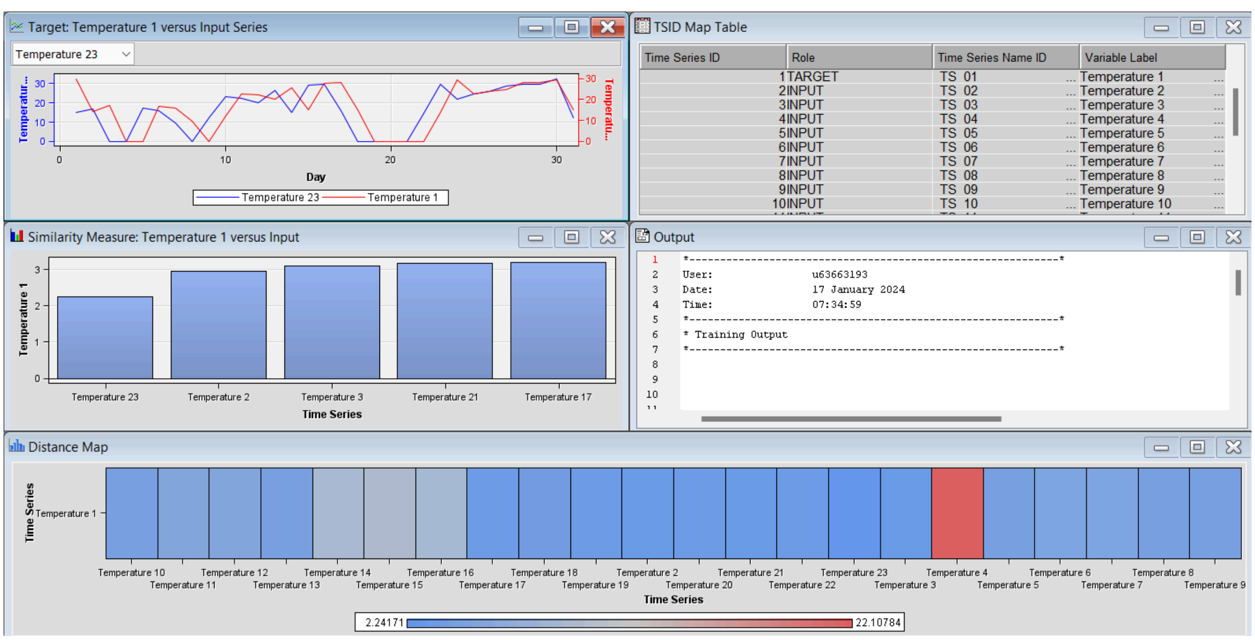


Diagram 9.17: Time Series Similarity Node

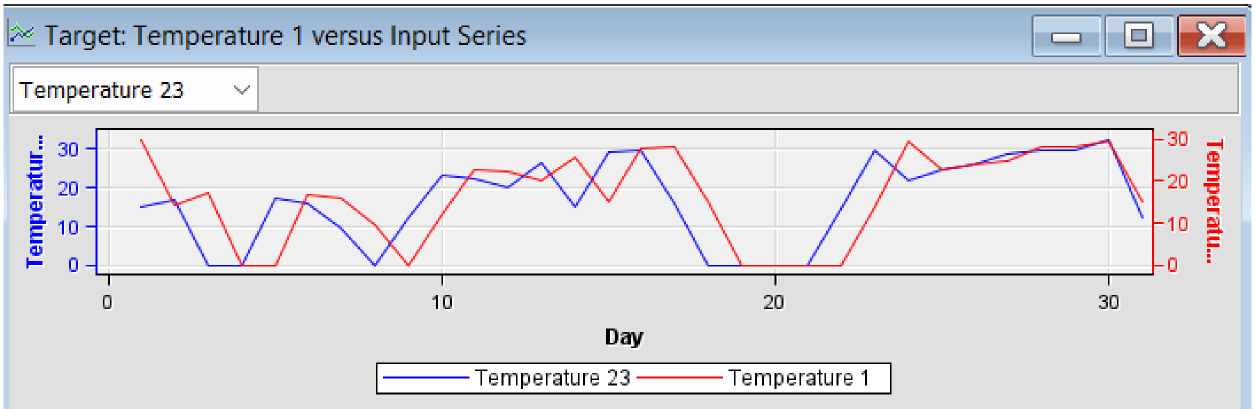


Diagram 9.18: Most Similar Time

10. Model

10.1 Classification

Classification is a supervised learning technique that characterizes and separates data classes. The models are developed by analyzing the training data. The model is then used to predict the label or class of unlabeled objects. Learning is supervised by the labeled examples in training data sets.

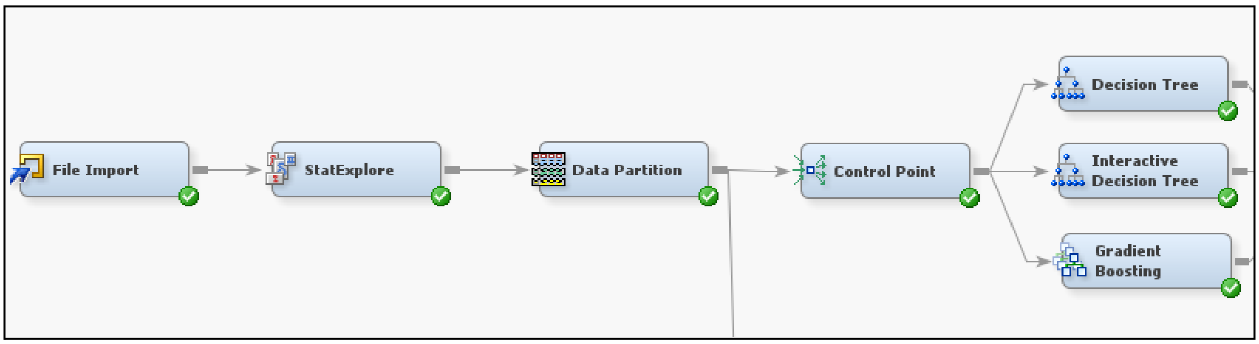


Diagram 10.1: Model Diagram

10.2 Decision tree

This model has been proven to be effective in supporting water quality assessment due to its easy visualization and automatic feature selection (Nasir, Nida, et al., 2022). Decision tree models are widely used because they simplify understanding various factors. The objective of employing decision trees is to create an initial model that can predict the class or category by learning a straightforward decision process from the original dataset or training data.

For the purpose of assessing air quality data in Italy, a classification model utilizing a decision tree is presented in a paper by Gakii, C., & Jepkoech, J. (2019). This paper focuses on classifying temperature using a decision tree.

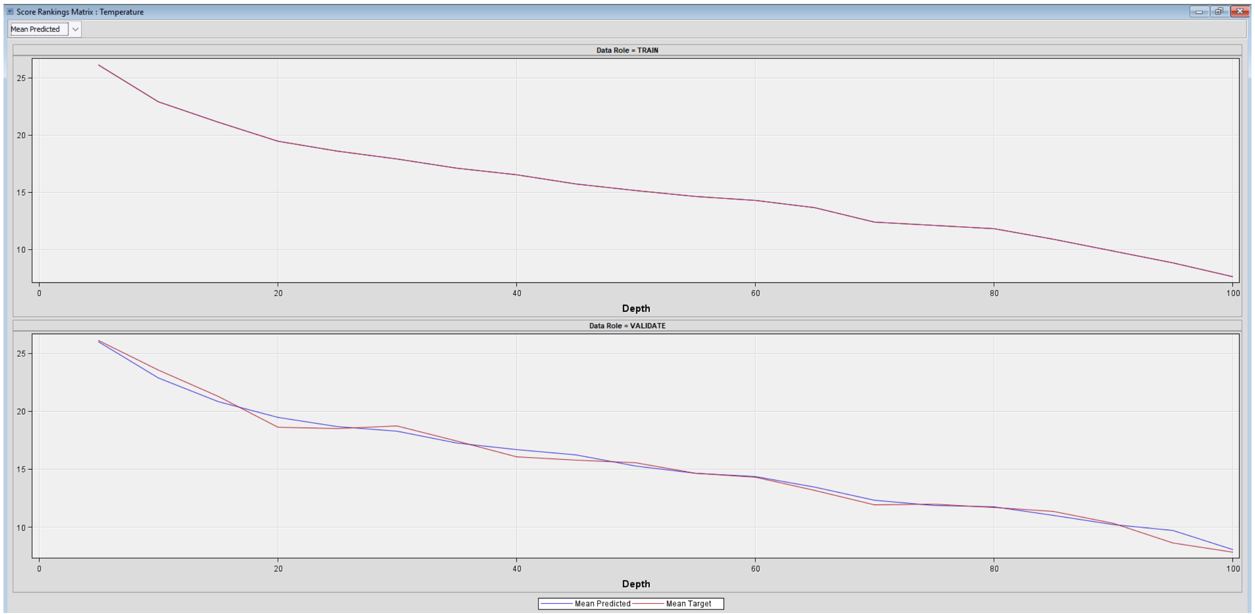


Diagram 10.2: Score Ranking Matrix (Decision Tree)

Diagram 10.2 reveals that, in the TRAIN data model, the trends of the Score Ranking Matrix for Mean Predicted and Mean Target are overlapping. However, in the VALIDATE train data model, there is a slight disparity between Mean Predicted and Mean Target, suggesting that the model may not be as accurate.

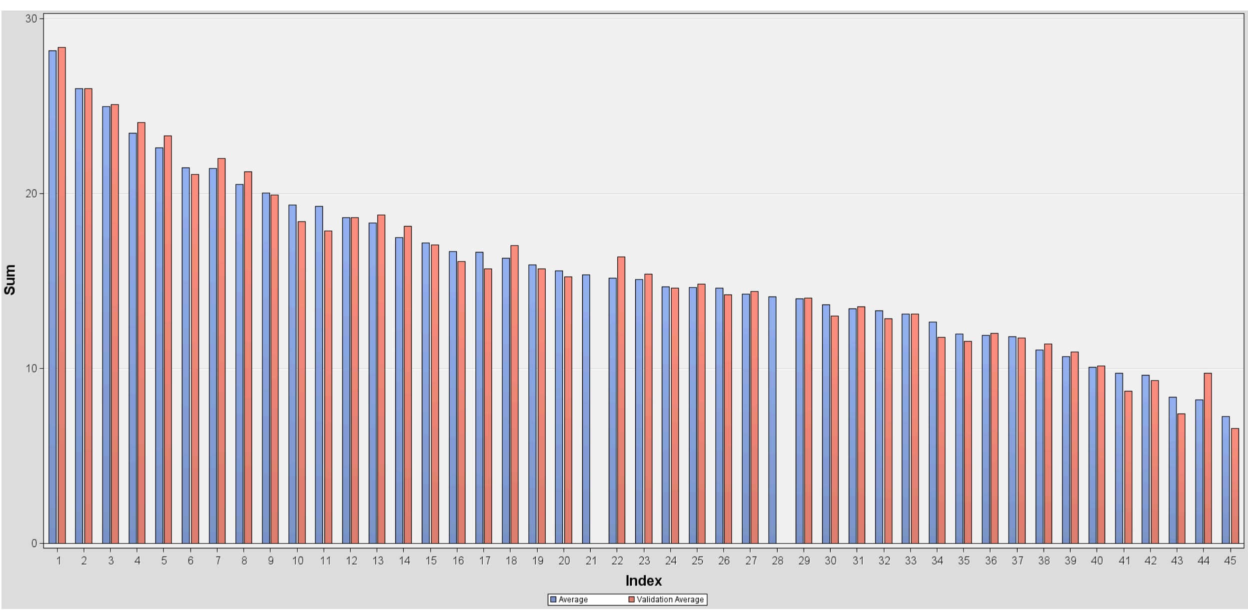


Diagram 10.3: Leaf Statistic (Decision Tree)

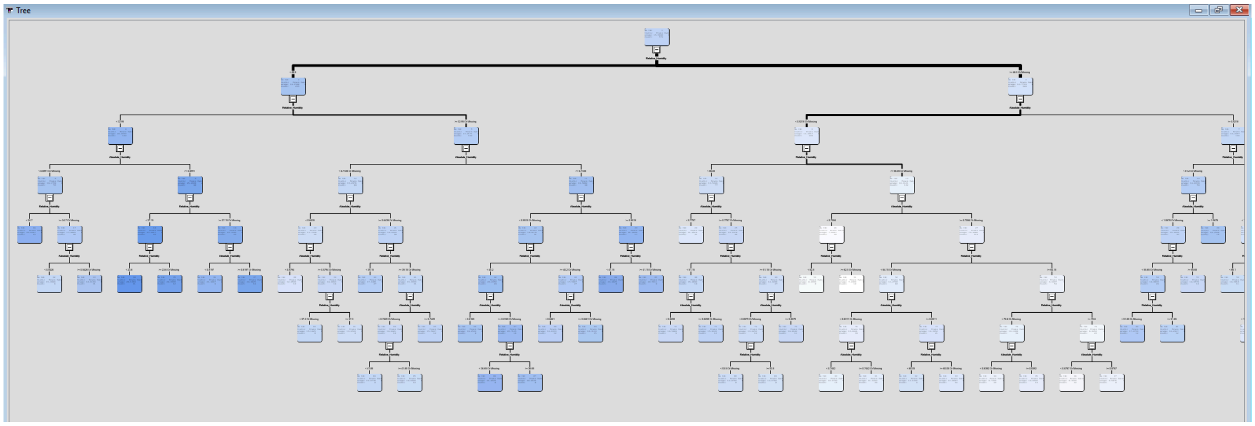


Diagram 10.4: Tree (Decision Tree)

Fit Statistics

Target	Fit Statistics	Statistics Label	Train	Validation	Test
Temperature	NOBS	Sum of Frequencies	579	165	83
Temperature	MAX	Maximum Absolute Error	5.893333	6.693333	2.4375
Temperature	SSE	Sum of Squared Errors	535.577	245.6968	71.99258
Temperature	ASE	Average Squared Error	0.925003	1.489071	0.86738
Temperature	RASE	Root Average Squared Error	0.961771	1.220275	0.931333
Temperature	DIV	Divisor for ASE	579	165	83
Temperature	DFT	Total Degrees of Freedom	579	.	.

Diagram 10.5: Fit Statistic (Decision Tree)

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

Depth	Number of Observations	Mean Target	Mean Predicted
5	39	26.1769	26.1769
10	23	22.9043	22.9043
15	26	21.1192	21.1192
20	32	19.4875	19.4875
25	31	18.6194	18.6194
30	27	17.9407	17.9407
35	27	17.1481	17.1481
40	38	16.5421	16.5421
45	22	15.7227	15.7227
50	26	15.1962	15.1962
55	30	14.6433	14.6433
60	30	14.2900	14.2900
65	26	13.6577	13.6577
70	58	12.4328	12.4328
80	33	11.8273	11.8273
85	38	10.8921	10.8921
90	35	9.8771	9.8771
95	20	8.8450	8.8450
100	18	7.6667	7.6667

Diagram 10.6: Assessment Score Ranking - Train (Decision Tree)

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

Depth	Number of Observations	Mean Target	Mean Predicted
5	11	26.1364	26.0167
10	6	23.5500	22.8919
15	12	21.2750	20.8251
20	6	18.6500	19.4552
25	14	18.4929	18.7094
30	2	18.7500	18.3133
35	13	17.4538	17.2739
40	3	16.1000	16.6786
45	9	15.8222	16.2739
50	8	15.5625	15.2823
55	9	14.6778	14.6402
60	6	14.3333	14.3630
65	10	13.2000	13.4980
70	12	11.9583	12.3551
75	4	12.0000	11.8667
80	7	11.7286	11.8048
85	13	11.3692	11.0333
90	11	10.3545	10.2306
95	4	8.6750	9.7300
100	5	7.9000	8.1235

Diagram 10.7: Assessment Score Ranking - Validate (Decision Tree)

10.3 Interactive decision tree

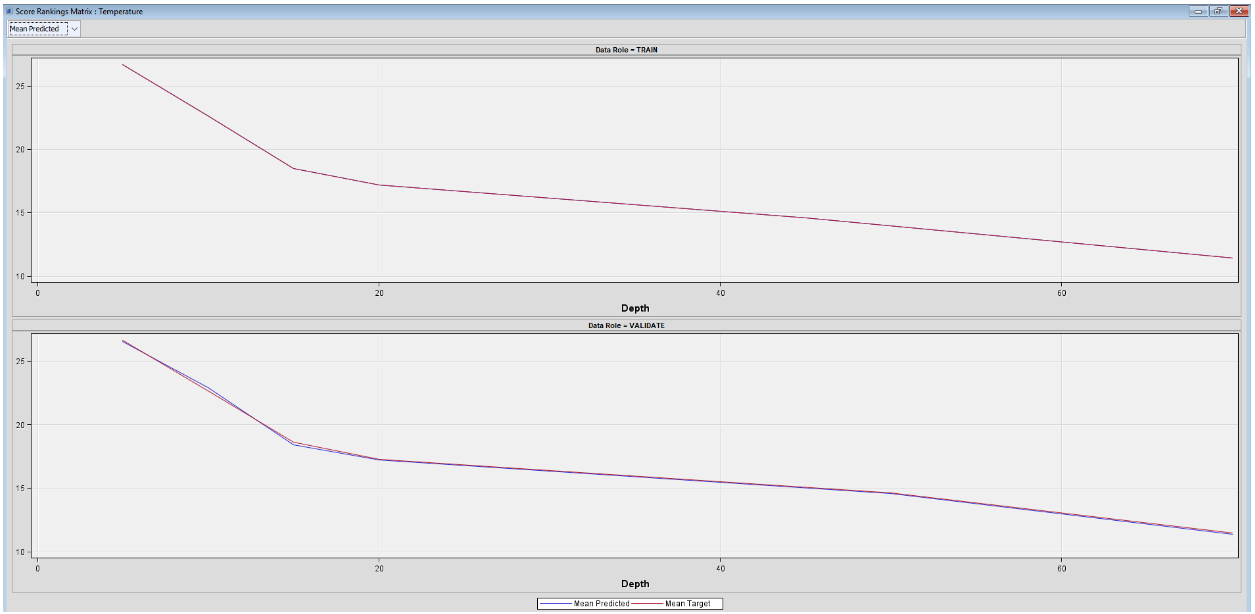


Diagram 10.8: Score Ranking Matrix (Interactive Decision Tree)

Diagram 10.8 illustrates that in both the TRAIN data model and the VALIDATE train data model, the trends of the Score Ranking Matrix for Mean Predicted and Mean Target are overlapping. This alignment suggests that the model is accurate and reliable.

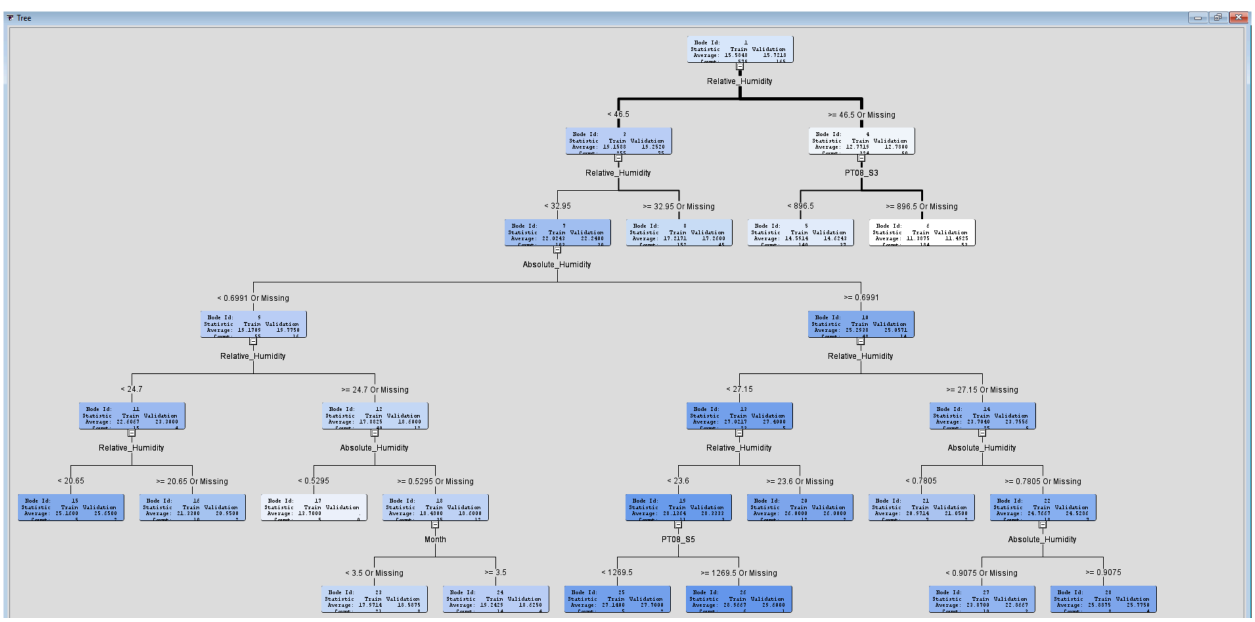


Diagram 10.9 Tree (Interactive Decision Tree)

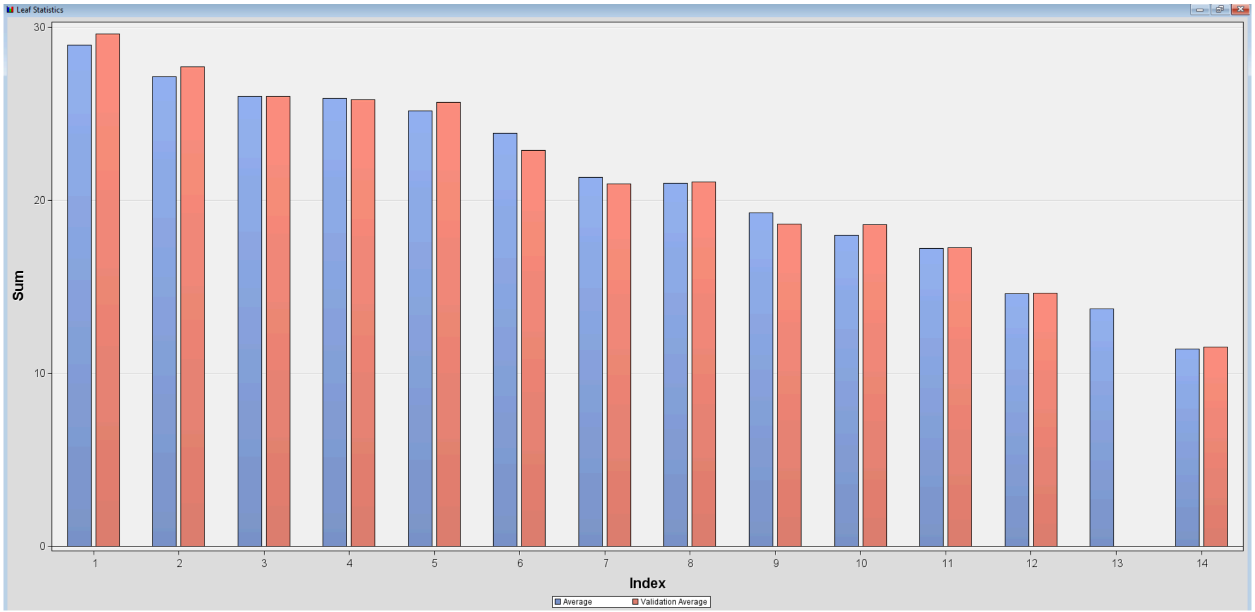


Diagram 10.10: Leaf Statistic (Interactive Decision Tree)

Fit Statistics

Target	Fit Statistics	Statistics Label	Train	Validation	Test
Temperature	NOBS	Sum of Frequencies	579	165	83
Temperature	MAX	Maximum Absolute Error	8.382895	7.082895	6.491429
Temperature	SSE	Sum of Squared Errors	3635.586	1144.089	483.8932
Temperature	ASE	Average Squared Error	6.279078	6.933872	5.830039
Temperature	RASE	Root Average Squared Error	2.505809	2.633225	2.414547
Temperature	DIV	Divisor for ASE	579	165	83
Temperature	DFT	Total Degrees of Freedom	579	.	.

Diagram 10.11: Fit Statistic (Interactive Decision Tree)

10.4 Gradient boosting

Gradient boosting is an effective method for making efficient and accurate predictions, especially when dealing with large and complex datasets. It operates by identifying the optimal way to partition the data based on a single variable. The primary objective is to create segments where the target variable exhibits a greater degree of similarity within each segment. This iterative process continues, further subdividing each segment until an optimal partitioning scheme is attained. Ultimately, these partitions are amalgamated to construct a predictive model.

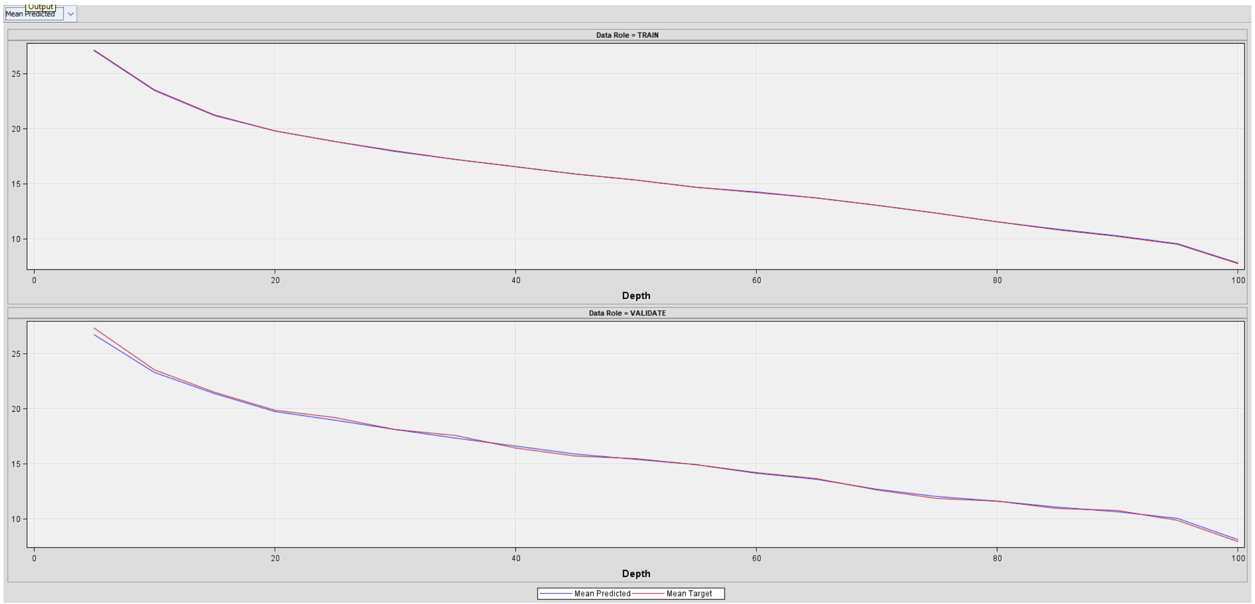


Diagram 10.12: Score Ranking Matrix (Gradient Boosting)

In Diagram 10.12, we observe that the trends of the Score Ranking Matrix for both the TRAIN data model (Mean Predicted and Mean Target) and the VALIDATE train data model (Mean Predicted and Mean Target) exhibit overlapping patterns. This consistency suggests that the model is characterized by accuracy and reliability.

10.5 Regression

Regression is a statistical technique used to predict a numeric target variable based on one or more predictor variables. It can accommodate both continuous and discrete inputs. A regression model is employed to assess whether changes in the dependent variable are correlated with changes in one or more of the explanatory variables.

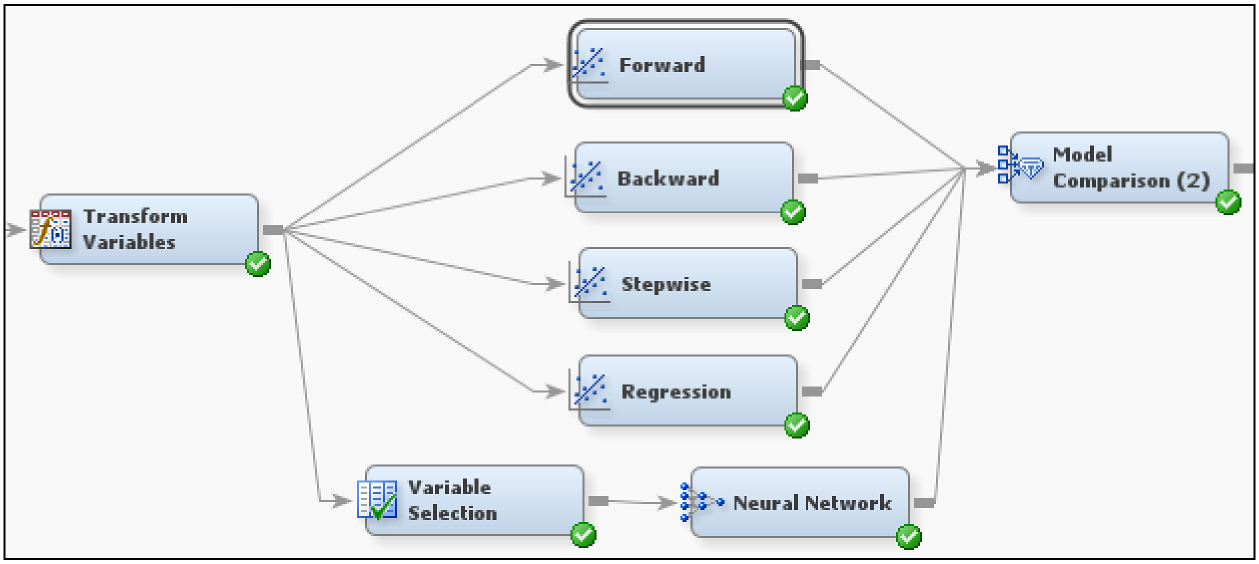


Diagram 10.13: Regression Node

First, we will execute the Regression node with its default settings. SAS Enterprise Miner employs the Logistic Regression with a Logit link function as the default regression type. Logistic regression, often associated with the sigmoid function, is utilized for binary classification tasks, where the outcome variable has only two possible results (typically coded as 0 and 1).

Additionally, we will employ a sequential selection approach within the Regression node by configuring the Selection Model to **Forward, Backward, or Stepwise**. This sequential selection method enhances the model's performance and identifies a subset of variables that offer the most effective explanation for the variations in the target variable.

Regression

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	22	11403	518.318166	150.64	<.0001
Error	556	1913.026593	3.440695		
Corrected Total	578	13316			

Model Fit Statistics

R-Square	0.8563	Adj R-Sq	0.8507
AIC	737.9857	BIC	741.8852
SBC	838.2957	C(p)	23.0000

Type 3 Analysis of Effects

Effect	DF	Sum of Squares	F Value	Pr > F
DateID	1	324.2996	94.25	<.0001
Day	1	6.3414	1.84	0.1751
Hour	1	31.3069	9.10	0.0027
LG10_PT08_S1	1	52.5490	15.27	0.0001
LG10_PT08_S2	1	25.5728	7.43	0.0066
LG10_PT08_S3	1	12.1799	3.54	0.0604
LG10_PT08_S4	1	17.0758	4.96	0.0263
LG10_PT08_S5	1	48.8527	14.20	0.0002
MeasurementID	0	.	.	.
Month	0	.	.	.
OPT_Absolute_Humidity	2	302.9028	44.02	<.0001
OPT_Benzene_Concentration	2	27.0032	3.92	0.0203
OPT_CO_Concentration	2	0.7071	0.10	0.9024
OPT_NMHC_Concentration	2	11.4225	1.66	0.1911
OPT_NO2_Concentration	2	7.3641	1.07	0.3437
OPT_NOx_Concentration	1	2.1225	0.62	0.4325
OPT_Relative_Humidity	3	4364.9374	422.87	<.0001
ReadingID	0	.	.	.

Diagram 10.14: Output Result (Default Regression)

In our regression model, we've obtained a Mean Square Error (MSE) value of 3.440695. During our Type 3 Analysis of Effects, we identified the input variables with the highest F values, revealing that Relative Humidity and Absolute Humidity exhibit the strongest correlations with temperature. These insights provide valuable information about the influential factors impacting our temperature predictions in the model.

Forward Regression

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	10	11325	1132.454237	322.99	<.0001
Error	568	1991.483883	3.506134		
Corrected Total	578	13316			

Model Fit Statistics

R-Square	0.8504	Adj R-Sq	0.8478
AIC	737.2577	BIC	739.2652
SBC	785.2320	C(p)	21.8027

Type 3 Analysis of Effects

Effect	DF	Sum of Squares	F Value	Pr > F
DateID	1	427.3215	121.88	<.0001
Hour	1	21.1499	6.03	0.0143
LG10_PT08_S1	1	112.0258	31.95	<.0001
LG10_PT08_S3	1	69.1998	19.74	<.0001
LG10_PT08_S5	1	44.2073	12.61	0.0004
OPT_Absolute_Humidity	2	621.4317	88.62	<.0001
OPT_Relative_Humidity	3	6893.3231	655.36	<.0001

Diagram 10.15: Output Result (Forward Regression)

We use Forward Regression with model selection set to Forward, and while the outcomes closely resemble those of other regression models, slight variations in the results arise due to distinct techniques in model training. In this case, Forward Regression yields an MSE of 3.506134.

Backward Regression

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	14	11379	812.789992	236.67	<.0001
Error	564	1936.966366	3.434338		
Corrected Total	578	13316			

Model Fit Statistics

R-Square	0.8545	Adj R-Sq	0.8509
AIC	729.1864	BIC	732.0397
SBC	794.6059	C(p)	13.9578

Type 3 Analysis of Effects

Effect	DF	Sum of Squares	F Value	Pr > F
DateID	1	328.0336	95.52	<.0001
Hour	1	34.9978	10.19	0.0015
LG10_PT08_S1	1	56.6001	16.48	<.0001
LG10_PT08_S2	1	31.1009	9.06	0.0027
LG10_PT08_S3	1	13.5299	3.94	0.0476
LG10_PT08_S4	1	17.3899	5.06	0.0248
LG10_PT08_S5	1	52.5520	15.30	0.0001
OPT_Absolute_Humidity	2	361.7818	52.67	<.0001
OPT_Benzene_Concentration	2	31.1051	4.53	0.0112
OPT_Relative_Humidity	3	4835.8558	469.36	<.0001

Diagram 10.16: Output Result (Backward Regression)

In Backward Regression, we utilize the Regression node with a model selection set to Backward. Although the outcomes closely resemble those of other regression models, slight variations occur due to distinct techniques in model training. The mean square error of the Backward Regression model is 3.434338.

Stepwise Regression

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	10	11325	1132.454237	322.99	<.0001
Error	568	1991.483883	3.506134		
Corrected Total	578	13316			

Model Fit Statistics

R-Square	0.8504	Adj R-Sq	0.8478
AIC	737.2577	BIC	739.2652
SBC	785.2320	C(p)	21.8027

Type 3 Analysis of Effects

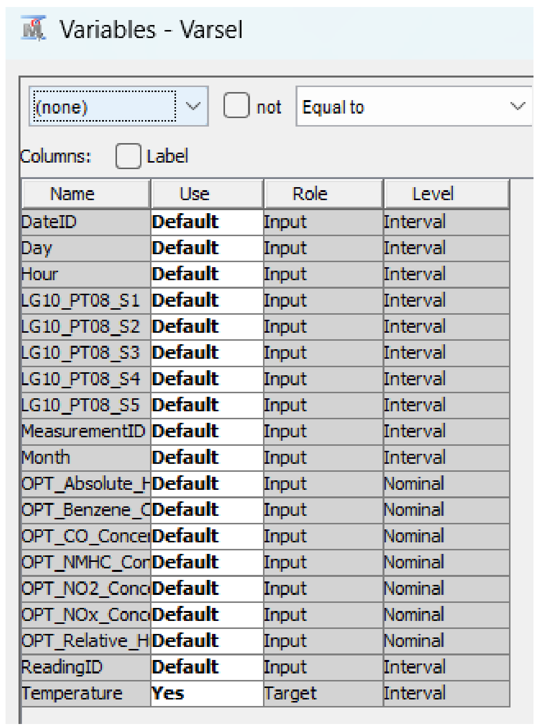
Effect	DF	Sum of Squares	F Value	Pr > F
DateID	1	427.3215	121.88	<.0001
Hour	1	21.1499	6.03	0.0143
LG10_PT08_S1	1	112.0258	31.95	<.0001
LG10_PT08_S3	1	69.1998	19.74	<.0001
LG10_PT08_S5	1	44.2073	12.61	0.0004
OPT_Absolute_Humidity	2	621.4317	88.62	<.0001
OPT_Relative_Humidity	3	6893.3231	655.36	<.0001

Diagram 10.17: Output Result (Stepwise Regression)

In Stepwise Regression, we utilize the Regression node with a model selection set to Stepwise. Although the outcomes closely resemble those of other regression models, The relative slight variations occur due to distinct techniques in model training. The mean square error of the Backward Regression model is 3.506134.

10.6 Neural Network

Neural networks are a class of parametric models that can handle a wider variety of nonlinear relationships between a set of predictors and a target variable



Name	Use	Role	Level
DateID	**Default**	Input	Interval
Day	**Default**	Input	Interval
Hour	**Default**	Input	Interval
LG10_PT08_S1	**Default**	Input	Interval
LG10_PT08_S2	**Default**	Input	Interval
LG10_PT08_S3	**Default**	Input	Interval
LG10_PT08_S4	**Default**	Input	Interval
LG10_PT08_S5	**Default**	Input	Interval
MeasurementID	**Default**	Input	Interval
Month	**Default**	Input	Interval
OPT_Absolute_H	**Default**	Input	Nominal
OPT_Benzene_C	**Default**	Input	Nominal
OPT_CO_Concei	**Default**	Input	Nominal
OPT_NMHC_Cor	**Default**	Input	Nominal
OPT_NO2_Conc	**Default**	Input	Nominal
OPT_NOx_Conc	**Default**	Input	Nominal
OPT_Relative_H	**Default**	Input	Nominal
ReadingID	**Default**	Input	Interval
Temperature	**Yes**	Target	Interval

Diagram 10.18: Variable Selection (Neutral Network)

We have used variable selection to reduce the number of input variables submitted to neural network models. We have employed the default settings from the variable selection node. As configured, variables with low R-square values are rejected.

Iter	Restarts	Function Calls	Active Constraints	Objective Function	Objective Function Change	Max Abs Gradient Element	Step Size	Slope of Search Direction
1	0	6	0	3.34922	0.0309	0.2709	0.00543	-11.528
2	0	10	0	3.26807	0.0812	0.2528	1.528	-0.106
3	0	12	0	3.16432	0.1037	0.1679	1.000	-0.176
4	0	16	0	3.09773	0.0666	0.3045	2.174	-0.0613
5	0	18	0	3.03020	0.0675	0.1813	1.000	-0.119
6	0	20	0	2.95697	0.0732	0.1919	1.391	-0.121
7	0	23	0	2.91497	0.0420	0.1287	1.508	-0.0549
8	0	25	0	2.87198	0.0430	0.1799	2.847	-0.0369
9	0	27	0	2.84807	0.0239	0.2544	2.742	-0.0392
10	0	29	0	2.81013	0.0379	0.1052	0.975	-0.0614
11	0	31	0	2.76792	0.0422	0.2118	2.500	-0.0379
12	0	33	0	2.71086	0.0571	0.1616	2.289	-0.0461
13	0	36	0	2.68391	0.0269	0.1139	1.179	-0.0451
14	0	38	0	2.66655	0.0174	0.1709	3.225	-0.0209
15	0	42	0	2.61517	0.0514	0.0850	2.049	-0.0501
16	0	44	0	2.58166	0.0335	0.2016	3.756	-0.0342
17	0	46	0	2.54081	0.0408	0.0757	1.500	-0.0559
18	0	48	0	2.48690	0.0539	0.1517	3.458	-0.0295
19	0	51	0	2.46299	0.0239	0.1228	1.147	-0.0434
20	0	53	0	2.42921	0.0338	0.1146	1.507	-0.0397
21	0	55	0	2.38376	0.0455	0.1414	1.828	-0.0462
22	0	57	0	2.37174	0.0120	0.2618	2.029	-0.0560
23	0	61	0	2.31469	0.0570	0.1033	1.199	-0.0973
24	0	64	0	2.29211	0.0226	0.0907	1.237	-0.0372
25	0	66	0	2.26056	0.0316	0.0795	1.353	-0.0417
26	0	69	0	2.23947	0.0211	0.0911	1.342	-0.0313
27	0	71	0	2.21386	0.0256	0.1179	2.387	-0.0221
28	0	73	0	2.19967	0.0142	0.2386	2.550	-0.0251
29	0	75	0	2.17496	0.0247	0.0727	0.825	-0.0430
30	0	77	0	2.15706	0.0179	0.2163	2.241	-0.0276
31	0	79	0	2.13094	0.0261	0.0815	1.591	-0.0281
32	0	81	0	2.09830	0.0326	0.1617	2.309	-0.0283
33	0	83	0	2.08664	0.0117	0.2186	2.340	-0.0349
34	0	87	0	2.05825	0.0284	0.0645	1.195	-0.0488
35	0	89	0	2.03486	0.0234	0.2208	4.971	-0.0143
36	0	91	0	2.00266	0.0322	0.1518	0.972	-0.0601
37	0	94	0	1.98797	0.0147	0.0854	0.974	-0.0295
38	0	96	0	1.96846	0.0195	0.0811	2.030	-0.0181
39	0	98	0	1.94669	0.0218	0.1778	1.820	-0.0268
40	0	100	0	1.93085	0.0158	0.1038	1.964	-0.0277
41	0	102	0	1.91123	0.0196	0.0535	1.618	-0.0245
42	0	104	0	1.89588	0.0153	0.1603	3.463	-0.0142
43	0	106	0	1.87914	0.0167	0.0828	1.387	-0.0277
44	0	109	0	1.86765	0.0115	0.0771	1.976	-0.0113
45	0	111	0	1.85374	0.0139	0.1001	1.775	-0.0162
46	0	113	0	1.84900	0.00474	0.1036	2.476	-0.0140
47	0	117	0	1.83645	0.0126	0.0407	1.225	-0.0205
48	0	120	0	1.82791	0.00854	0.0711	2.873	-0.0058
49	0	122	0	1.81625	0.0117	0.0408	1.496	-0.0143
50	0	124	0	1.81327	0.00298	0.1053	3.110	-0.0094

Diagram 10.19: Optimization Results Neural Network

The objective function value within our neural network optimization results serves as a critical performance metric that we endeavor to enhance throughout the process. It initiates at a value of 3.34922 and progressively diminishes as our optimization algorithm iteratively refines the neural network model. Ultimately, after a series of iterations, it converges to a final value of 1.81327.

11. Assess

The assessment step in SEMMA is of paramount importance because it involves the evaluation and comparison of various models. This critical phase aims to pinpoint the model that is most effective in tackling a particular problem or producing accurate predictions.

11.1 Decision Tree & Gradient Boosting

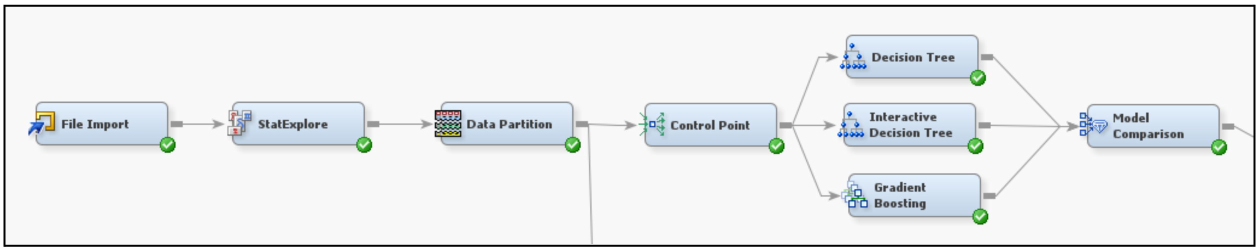


Diagram 11.1: Model Comparison Decision Tree & Gradient Boosting

To assess all the models and make comparisons between them, we utilized the Model Comparison Node in SAS Enterprise Miner for this purpose.

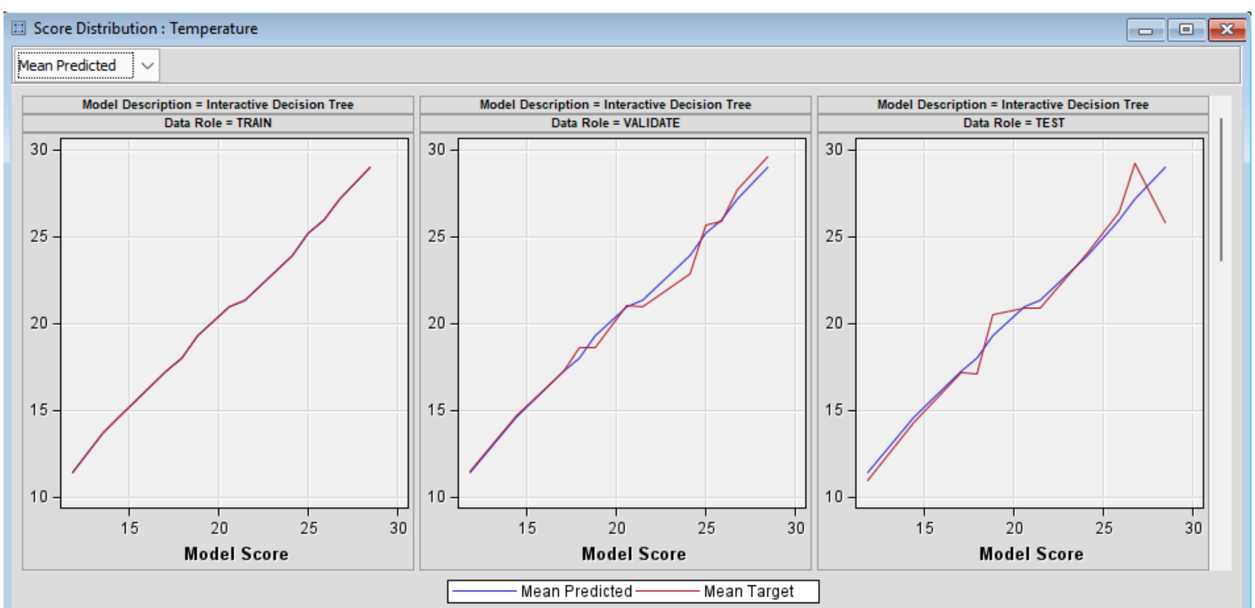


Diagram 11.2: Score Distribution Target Variable Interactive Decision tree

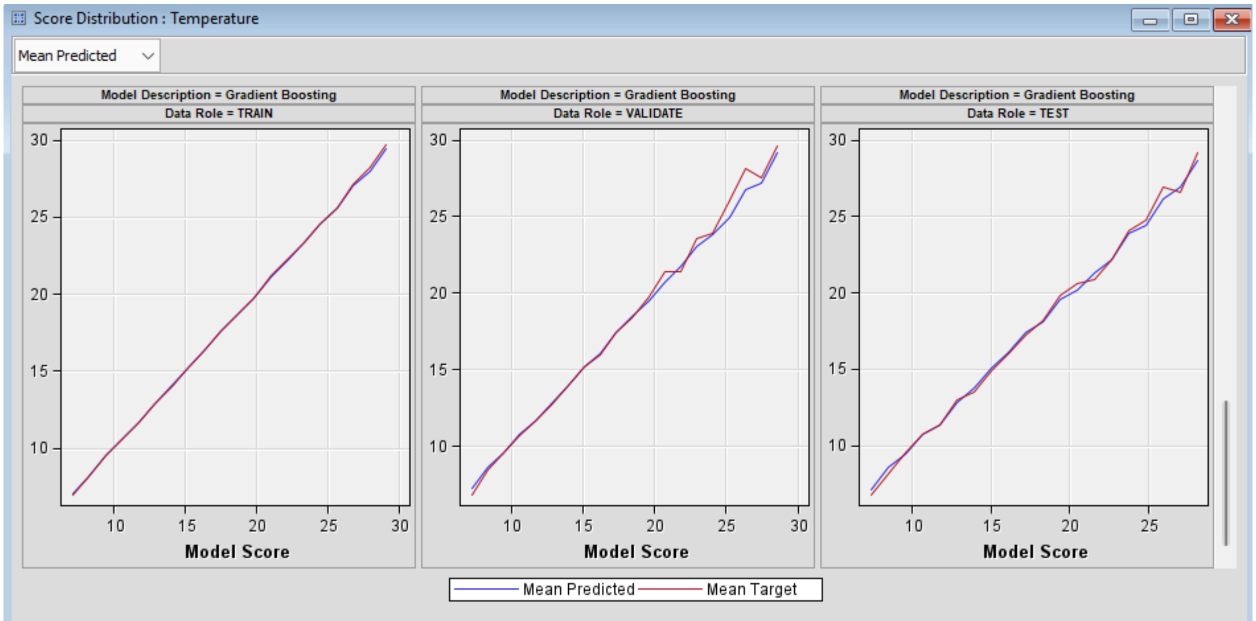


Diagram 11.3: Score Distribution Target Variable Gradient Boosting

Based on Diagrams 11.2 and 11.3, it is evident that the mean prediction line closely aligns with the mean target line. Consequently, we can conclude that, on average, the model's predictions are in agreement with the actual target values. This alignment between the mean prediction and mean target indicates that the model effectively captures the underlying data patterns.

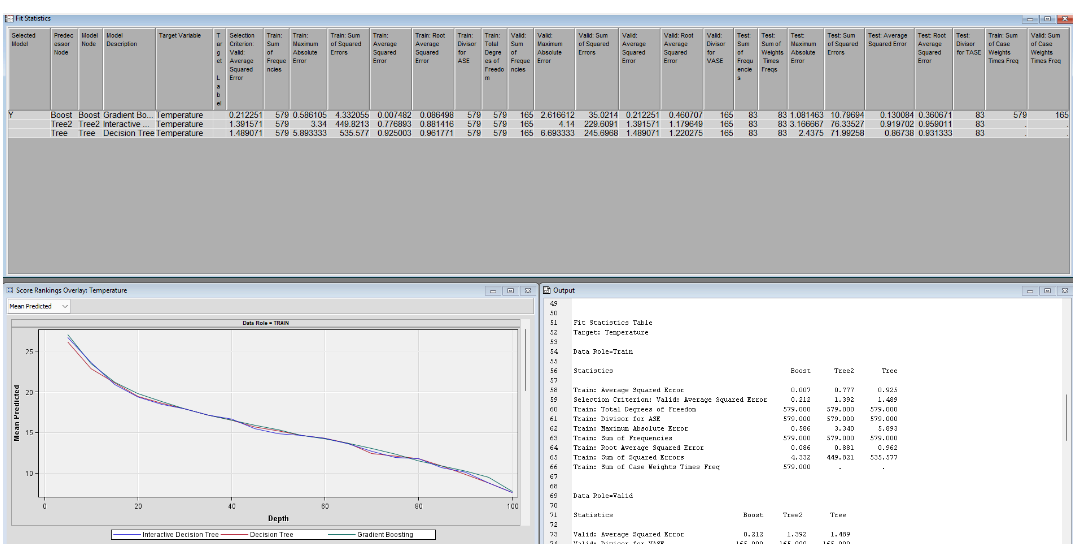


Diagram 11.4: Result of Model Comparison Node Decision Tree & Gradient Boosting

Diagram 11.4 displays the results obtained from the Model Comparison Node, which includes distinct tabs such as the Fit Statistics Table, Score Ranking Overlay Chart, and Output Result. A detailed explanation of each component follows.

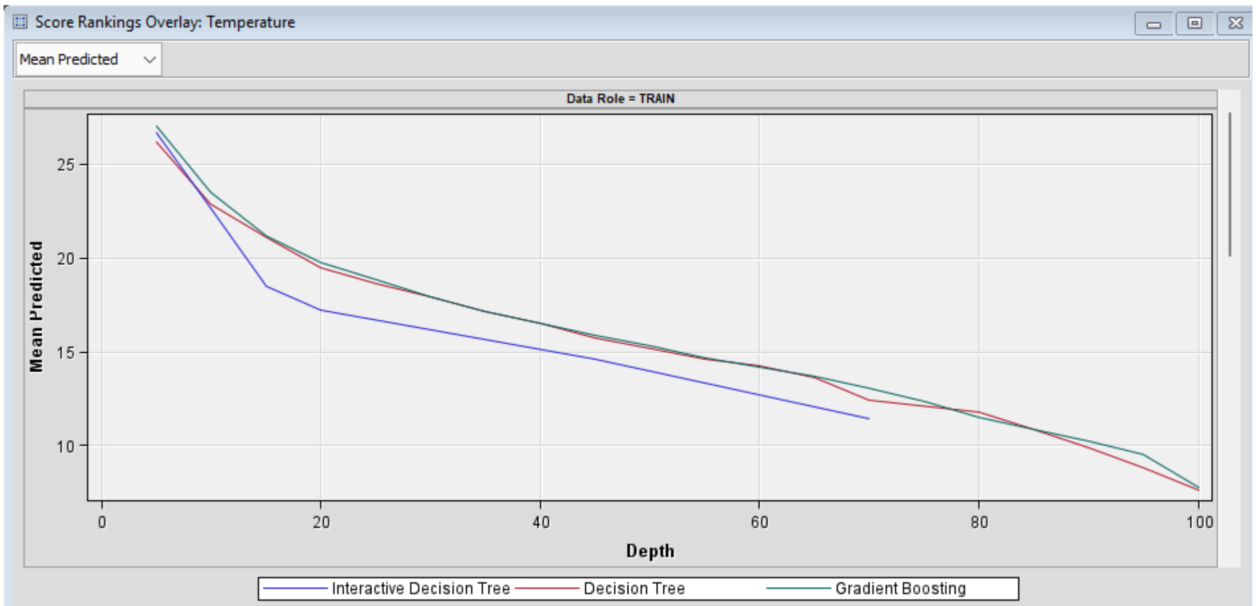


Diagram 11.5: Score Ranking Overlay Decision Tree & Gradient Boosting

Diagram 11.5 illustrates the cumulative lift graph generated by our trained model. A superior model will exhibit a larger area under the curve. In this graph, we observed that the Interactive Decision Tree halted midway. This occurred because, during the setup of the interactive decision tree, we constrained the tree from including all nodes, leading to the issue of underfitting, as demonstrated above. The graph also reveals that the decision tree and gradient boosting produced fairly similar results, with the line chart for gradient boosting showing a slightly higher curve.

Fit Statistics

Selected Model	Predecesso r Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Average Squared Error	Train: Sum of Frequencie s	Train: Maximum Absolute Error	Train: Sum of Squared Errors	Train: Average Squared Error	Train: Root Average Squared Error	Trai Divis ASE
Y	Boost	Boost	Gradient ...	Tempera...		0.212251	579	0.586105	4.332055	0.007482	0.086498	
	Tree	Tree	Decision ...	Tempera...		1.489071	579	5.893333	535.577	0.925003	0.961771	
	Tree2	Tree2	Interactiv...	Tempera...		6.933872	579	8.382895	3635.586	6.279078	2.505809	

Diagram 11.6: Fit Statistics Decision Tree & Gradient Boosting

Fit Statistics
Model Selection based on Valid: Average Squared Error (_VASE_)

Selected Model	Model Node	Model Description	Valid: Average Squared Error	Train: Average Squared Error
Y	Boost	Gradient Boosting	0.21225	0.00748
	Tree	Decision Tree	1.48907	0.92500
	Tree2	Interactive Decision Tree	6.93387	6.27908

Diagram 11.7: Fit Statistics Decision Tree & Gradient Boosting

Considering Diagrams 11.6 and 11.7, it becomes evident that among the models, including Decision Tree, Interactive Decision Tree, and Gradient Boosting, Gradient Boosting stands out as the superior choice. This decision is primarily rooted in the fact that Gradient Boosting exhibits the lowest Average Squared Error when compared to the other models.

11.2 Neural Network and regression

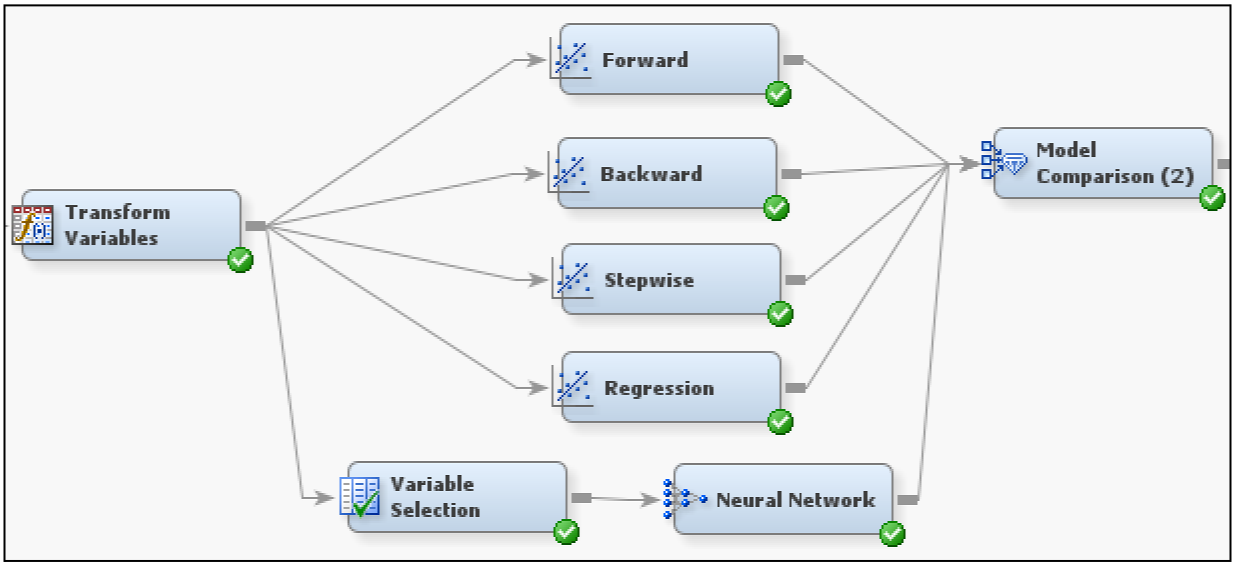


Diagram 11.8: Model Comparison Regression & Neural Network

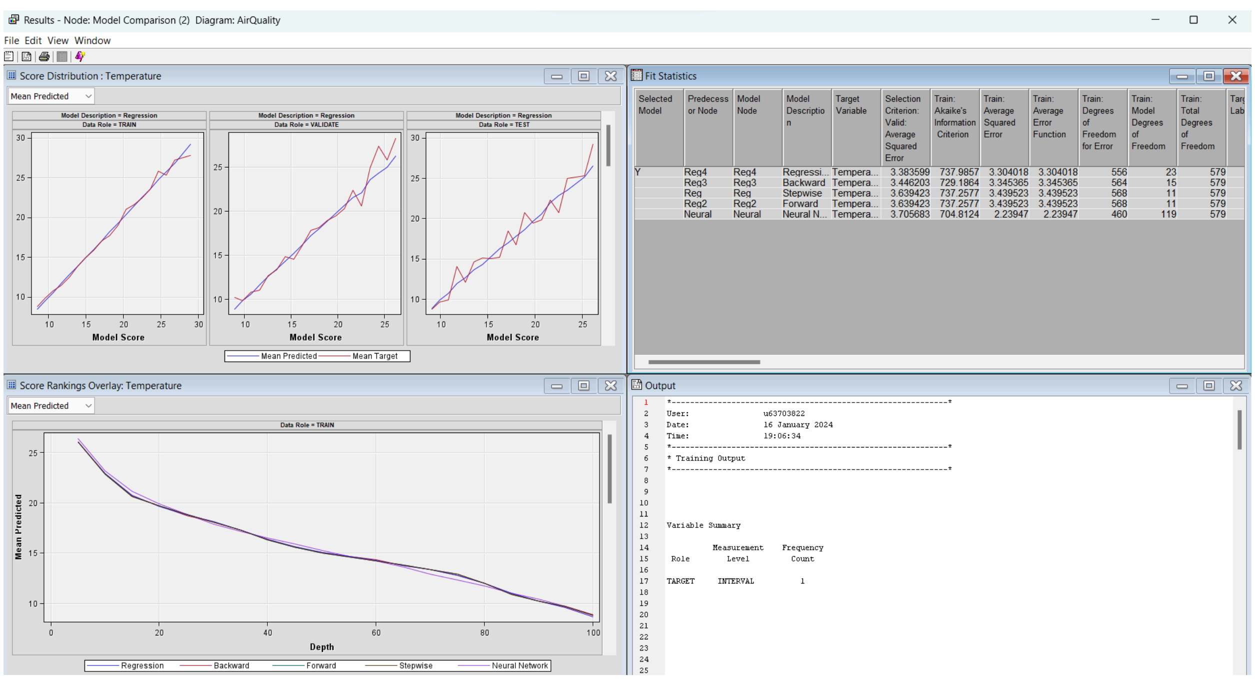


Diagram 11.9: Result of Model Comparison Node Regression & Neural Network

Diagram 11.9 displays the results obtained from the Model Comparison Node, which comprises various tabs, including the Fit Statistics Table, Score Ranking Overlay Chart, and Output Result. Further elaboration on each of these components will follow.

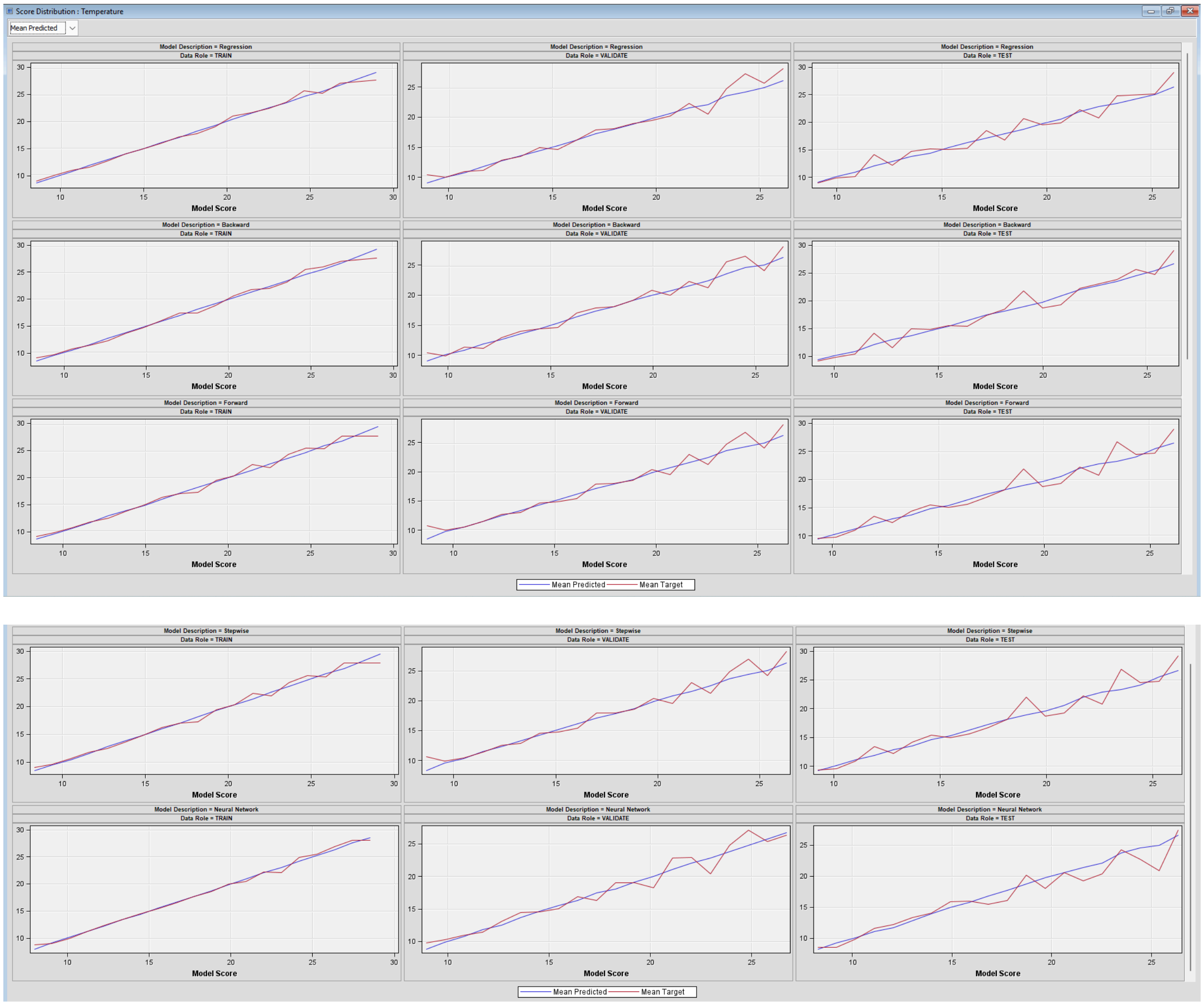


Diagram 11.10: Score Distribution Target Variable Regression & Neural Network

Diagram 11.10 displays the mean prediction line and mean target line. We can observe variations in trend lines among the models. Upon examination, it is evident that the Mean Predicted and Mean Target of the neural network exhibit the highest inconsistency, suggesting that this model may not be suitable for our project.

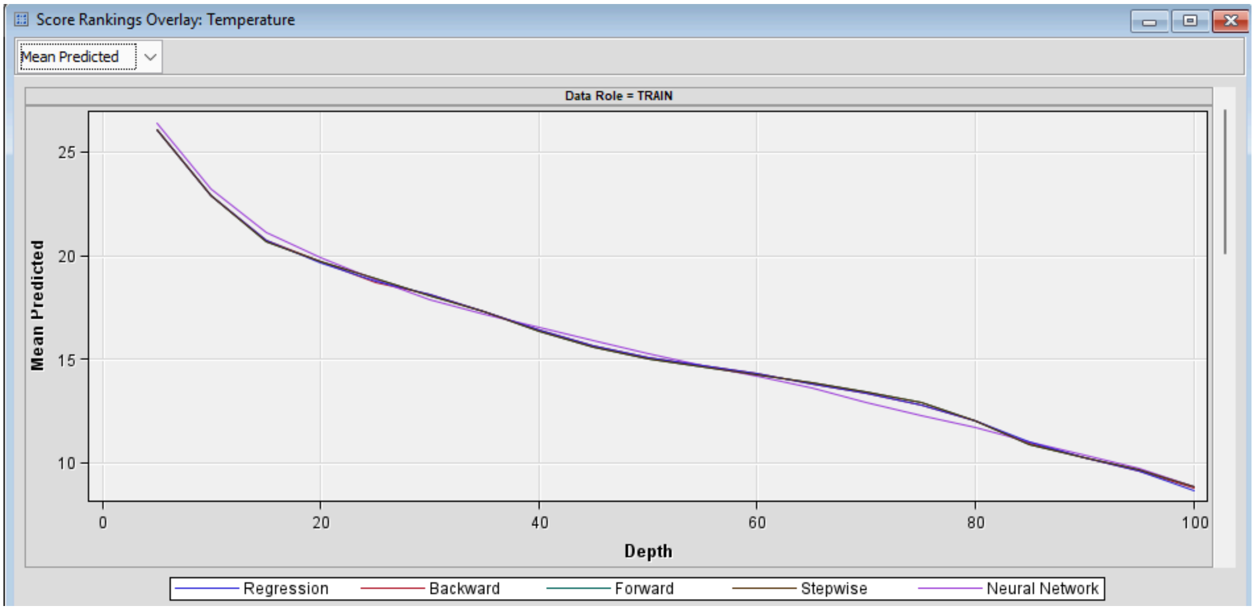


Diagram 11.11: Score Ranking Overlay Regression & Neural Network

Diagram 11.11 represents the mean predicted graph of our trained model. A larger area under the curve signifies a better-performing model. Upon examination, we note that all models exhibit similar results on the line chart. However, a closer inspection reveals that our standard regression model outperforms the others, making it the best choice among the regression and neural network models.

Fit Statistics

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Average Squared Error	Train: Akaike's Information Criterion	Train: Average Squared Error	Train: Average Error Function	Train: Degrees of Freedom for Error	Train: Model Degrees of Freedom	Trai Deg Free
Y	Reg4	Reg4	Regressi...	Tempera...		3.383599	737.9857	3.304018	3.304018	556	23	
	Reg3	Reg3	Backward	Tempera...		3.446203	729.1864	3.345365	3.345365	564	15	
	Reg	Reg	Stepwise	Tempera...		3.639423	737.2577	3.439523	3.439523	568	11	
	Reg2	Reg2	Forward	Tempera...		3.639423	737.2577	3.439523	3.439523	568	11	
	Neural	Neural	Neural N...	Tempera...		3.705683	704.8124	2.23947	2.23947	460	119	

Diagram 11.12: Fit Statistics Regression & Neural Network

Fit Statistics
Model Selection based on Valid: Average Squared Error (_VASE_)

Selected Model	Model Node	Model Description	Valid: Average Squared Error	Train: Average Squared Error	Train: Misclassification Rate
Y	Reg4	Regression	3.38360	3.30402	.
	Reg3	Backward	3.44620	3.34537	.
	Reg	Stepwise	3.63942	3.43952	.
	Reg2	Forward	3.63942	3.43952	.
	Neural	Neural Network	3.70568	2.23947	.

Diagram 11.13: Fit Statistics Regression & Neural Network

Considering Diagrams 11.12 and 11.13, Regression emerges as the best model among Backward, Stepwise, Forward, and Neural Network. We base this choice on Regression's lower Average Squared Error compared to the other models.

12. Conclusion

In summary, our investigation revolved around analyzing a dataset focused on air quality, which encompassed a variety of input variables, with air temperature as the target variable. We applied various models, including decision trees, interactive decision trees, gradient boosting, regression, as well as forward, backward, and stepwise regression, and neural networks.

For the classification tasks (decision tree and gradient boosting), we determined that gradient boosting is the most suitable model for our dataset. This choice was influenced by the potential presence of intricate relationships and dependencies among variables in our dataset. Gradient boosting excels at capturing such complexities by sequentially constructing weak learners, each designed to address specific aspects of the data.

In contrast, for the regression and neural network aspects, we found that the original regression model is the most appropriate for our dataset. This preference may be attributed to the possibility that forward, backward, and stepwise regression, as well as neural networks, introduced unnecessary complexity or overfit the training data, leading to reduced performance on new data. The original regression model, being simpler, may have avoided these issues.

In conclusion, each data mining model has its own strengths and limitations, which are influenced by the characteristics of the datasets they analyze. There is no universally superior or inferior model; the key lies in fine-tuning parameters and identifying settings that enhance accuracy for a specific dataset. Employing a variety of models in data mining enables the generation of predictive insights, facilitating well-informed decision-making and ultimately improving business outcomes.

13. References

Demir, F. (2022). Deep autoencoder-based automated brain tumor detection from MRI data. Artificial Intelligence-Based Brain-Computer Interface, 317–351. https://doi.org/10.1016/b978-0-323-91197-9.00013-8

Ghoson, A. M. A. (2011). Decision Tree Induction & Clustering Techniques In SAS Enterprise Miner, SPSS Clementine, And IBM Intelligent Miner A Comparative Analysis. International Journal of Management & Information Systems, 14(3). https://doi.org/10.19030/ijmis.v14i3.841

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. ACM computing surveys (CSUR), 31(3), 264-323.

Madhulatha, T. S. (2012). An overview on clustering methods. arXiv preprint arXiv:1205.1117, https://doi.org/10.48550/arXiv.1205.1117.

Rithika, S. (2023, January 18). Sequence data in Data Mining Simplified 101 - learn. Hevo. https://hevodata.com/learn/sequence-data-in-data-mining