UNIVERSITY OF MALAYA

MASTER OF DATA SCIENCE (SEMESTER 2 – 2022/2023)

FACULTY OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

WQD7005 DATA MINING

GROUP PROJECT

ANALYZING RENTAL MARKET TRENDS AND PREDICTING RENTAL PRICE AND RENTAL LEVEL IN KUALA LUMPUR AND SELANGOR

GROUP MEMBERS OF GROUP 8	
S2149768	ZHANG ZITENG
S2141978	ZHU DAMING
S2139007	XU YANZHEN
S2177535	XU XIAOFAN

INSTRUCTOR: PROF DR TEH YING WAH

DATE: 25th June 2023

Table of Contents

1. Introduction

1.1. Background

As far as the development of society, it can be clearly found that the rent pricing in many cities of Malaysia such as Kuala Lumpur and Selangor has shown an increase trend. Understanding the rental market trends is crucial for both tenants and property owners. Tenants can make informed decisions about rental affordability and negotiate rental contracts, while property owners can optimize rental pricing strategies and maximize their returns. By analyzing historical data and assessing the influence of different variables, the project aim to gain insights into the complex dynamics of the rental market in Kuala Lumpur and Selangor.

The purpose of this project is to analyze rental market trends and predict rental prices in the dynamic regions of Kuala Lumpur and Selangor. By utilizing the data mining software SAS Enterprise Miner (SAS EMiner), the project aim to delve into historical rental market data to identify patterns and fluctuations in rental prices over time. Additionally, the project will examine various factors such as location, property characteristics, and market conditions to understand their impact on rental prices in these areas.

Through rigorous data analysis using SAS EMiner, the system will explore the historical rental data, identify key trends, and uncover any underlying patterns. By evaluating the impact of factors such as location dimension, house property dimension, facilities dimension and time dimension, it can be better comprehend the determinants of rental prices in these regions. This analysis will provide valuable insights into the current state of the rental market and enable us to make predictions about future rental price trends.

Thus, this project aims to contribute to the understanding of rental market trends and provide valuable predictions for rental prices in Kuala Lumpur and Selangor by using SAS EMiner. The utilization of SAS EMiner as a powerful analytical tool will allow us to extract meaningful insights and make accurate forecasts, enabling stakeholders to make well-informed decisions in the rental market.

1.2. Business understanding

1.2.1. Analysis Objective:

- To analyze historical rental market trends in Kuala Lumpur and Selangor to identify

patterns and fluctuations in rental prices over time.

- To assess the impact of various factors, such as location, property characteristics, and market conditions, on rental prices in Kuala Lumpur and Selangor.
- To develop predictive model using historical rental market data & relevant variables to forecast future rental price and price level by using regression model and classification model.

1.2.2. Analysis Data:

Forecasting rental pricing can provide insight into the expected price of a rental property based on various factors such as location, amenities, property type and market trends. This information is valuable to landlords, property managers and real estate agents as it can help them make informed decisions in setting rental prices, advertising properties and negotiating leases. Not only that, but it is also valuable to tenants, who can use rent forecasts to help them get a more reasonable price for their desired property.

By analyzing historical data on rental prices and identifying patterns and trends, machine learning models can be trained to predict future rental prices. This can help companies optimize their pricing strategies to maximize profits and minimize vacancy rates.

2. Methodology

2.1. ESMMA Description

SEMMA (Sample, Explore, Modify, Model, and Assess) is a comprehensive data mining methodology created by the SAS Institute. It serves as a structured framework that guides data scientists and analysts throughout the entire data mining process, ensuring a systematic approach to problem-solving. By following the SEMMA methodology, the project can effectively address each stage of a data mining project, from initial data preparation and exploration to building models, evaluating their performance, and interpreting the results to derive meaningful insights.

2.2. Importance

In data mining projects, a systematic methodology like SEMMA is essential for ensuring precise, reliable, and actionable outcomes. This comprehensive approach minimizes the risk of neglecting crucial steps or considerations, contributing to the overall quality and success of the project.

The structured framework of SEMMA assists in uncovering patterns and relationships within the dataset, reducing the likelihood of making erroneous inferences, and ensuring potential issues and biases are addressed at every stage. By adhering to the SEMMA methodology, it can showcase a thorough understanding of data mining principles, make well-informed decisions based on a deep understanding of the data, and enhance the chances of obtaining high-quality results.

SEMMA facilitates a thorough understanding of the underlying data, encourages the use of best practices, and helps to identify any potential issues or biases that may affect the analysis. By promoting a systematic approach, SEMMA ensures that each step of the project is executed with care and attention to detail, ultimately leading to more reliable and accurate models.

In conclusion, the SEMMA methodology is a valuable tool for guiding data mining projects from inception to completion. Its structured approach enables us to extract valuable insights from complex datasets, while also ensuring that results are accurate, reliable, and reproducible. By using the SEMMA methodology, the project can maximize the potential of their data mining efforts, contributing valuable insights to our project.

2.3. Process

In this group assignment, the first two methodologies, Sample and Explore, were focused to kickstart the analysis. Figure 2.3.1 showed the process of SEMMA.

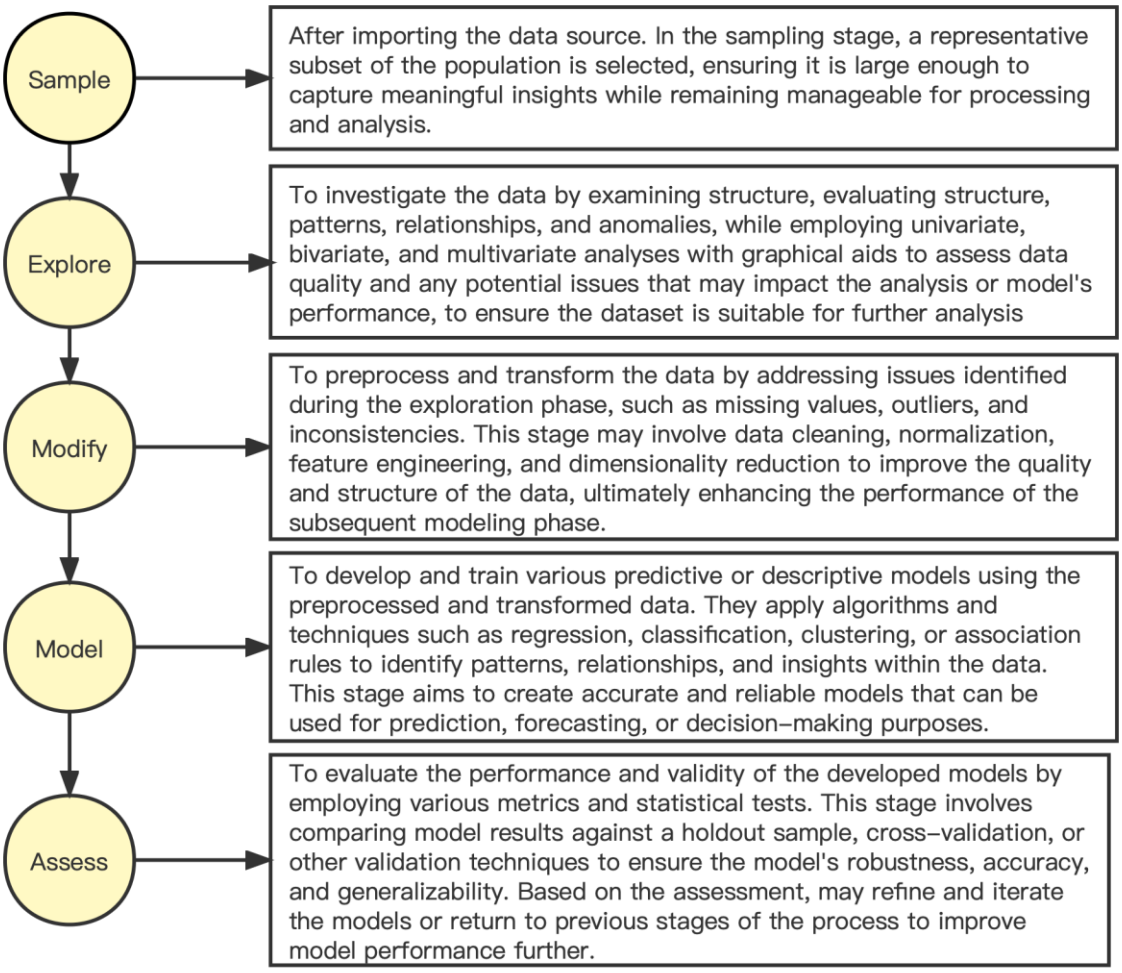


Figure 2.3.1 complete process of SEMMA

3. Raw Data

3.1. Dataset description

The dataset was collected from Kaggle website "https://www.kaggle.com/datasets/ariewijaya/rent-pricing-kuala-lumpur-malaysi". The rental dataset used in this project includes historical data from 1977 to 2025, obtained through web scraping from mudah.my, a popular online marketplace in Malaysia. This raw dataset consists of 17 features, including a unique identifier (ads_id) and a target variable (monthly_rent), which indicates the monthly rental price for each property listing. The dataset is focused on rental properties located in the Kuala Lumpur and Selangor regions of Malaysia. The dataset

may contain some noisy data due to the nature of web scraping, but we will address this issue during data preprocessing.

3.2. Variables description:

Attributes	Description
ads_id	The listing ids (unique)
prop_name	Name of the properties
completion_year	Completion/ established year of the property
completion_month	Completion/ established month of the property
completion_date	Completion/ established date of the property
monthly_rent	Monthly rent in ringgit malaysia (RM)
deposit	Deposit for the rental house
location	Property location in Kuala Lumpur region
property_type	Property type such as apartment, condominium, flat, duplex, studio, etc
rooms	Number of rooms in the unit
parking	Number of parking space for the unit
bathrooms	Number of bathrooms in the unit
size	Total area of the unit in square feet
furnished	The furnishing status of the unit, indicated by the numbers 1 to 3, indicates the degree of decoration.
no_of_facilities	Number of furnishing status of the unit
no_of_additional_facilities	Number of additional facilities

Table 3.2.1 dataset general description

Table 3.2.1 shows a detailed description of the variables included in the dataset. By analyzing this dataset, the project aim to identify trends and patterns in rental prices over time and assess the impact of various factors, such as location and property characteristics, on rental prices in these regions. The project will also develop a predictive model that utilizes historical rental market data and relevant variables to forecast future rental prices with reasonable accuracy.

3.3. Sample dataset

In SAS Enterprise Miner, there are basic and advanced settings available for specifying the data type of a variable. Figure 3.3.1 shows the metadata of the column and Figure 3.3.2 shows

the data type under the most basic configuration. Figure 3.3.3 shows the metadata of the column and Figure 3.3.4 shows the data type under its default advanced settings.

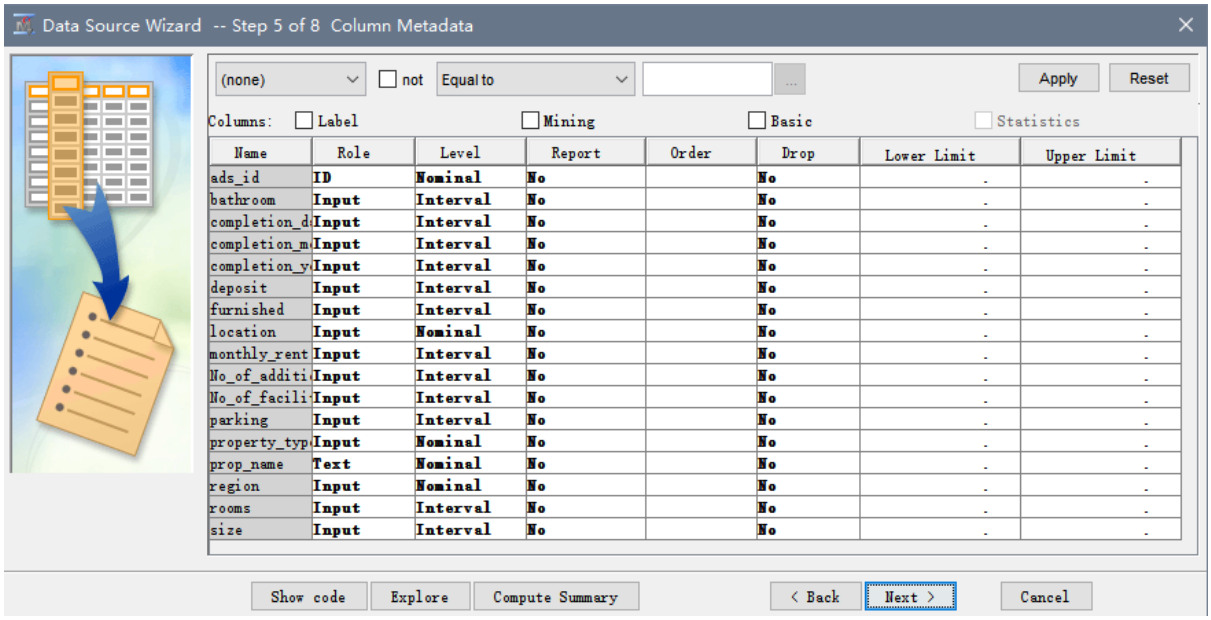


Figure 3.3.1 Column metadata (basic setting)

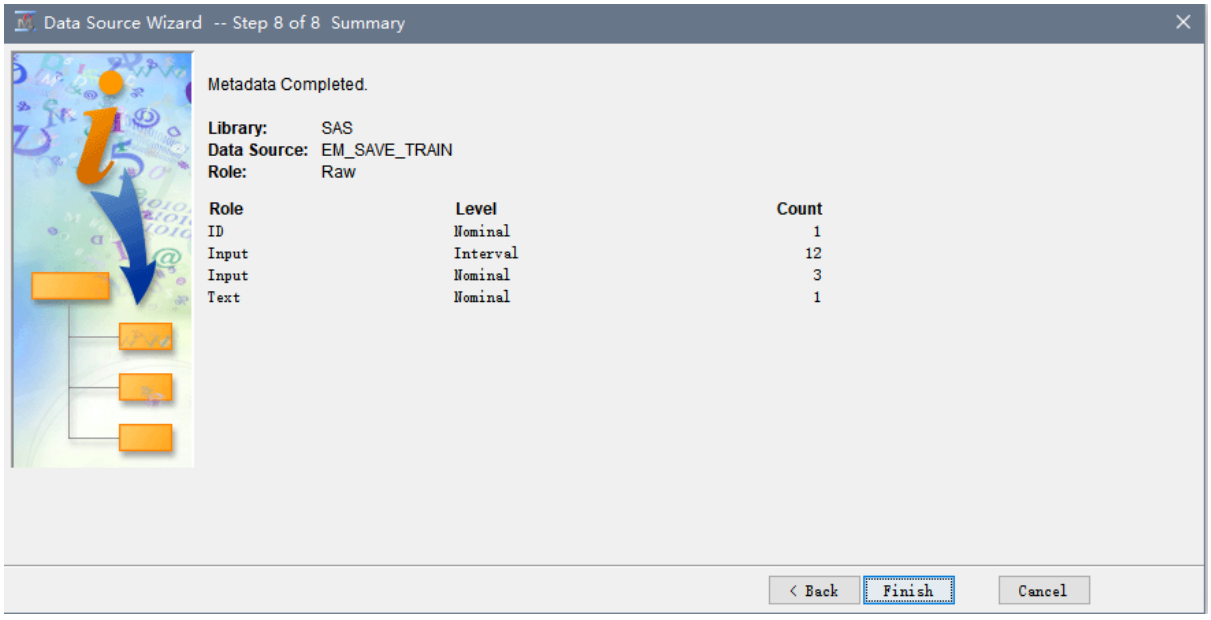


Figure 3.3.2 Data type summary (Basic setting)

Figures 3.3.1 and 3.3.2 show that with the default basic settings of the system, the data types are divided into interval and nominal as inputs, respectively. The measurement level is determined automatically by the system based on the value of the variable. By default, character data is considered as nominal type, while numeric data is considered as interval type. Most of the data types are incorrect in the CSV state, so the basic settings are not applicable to our task.

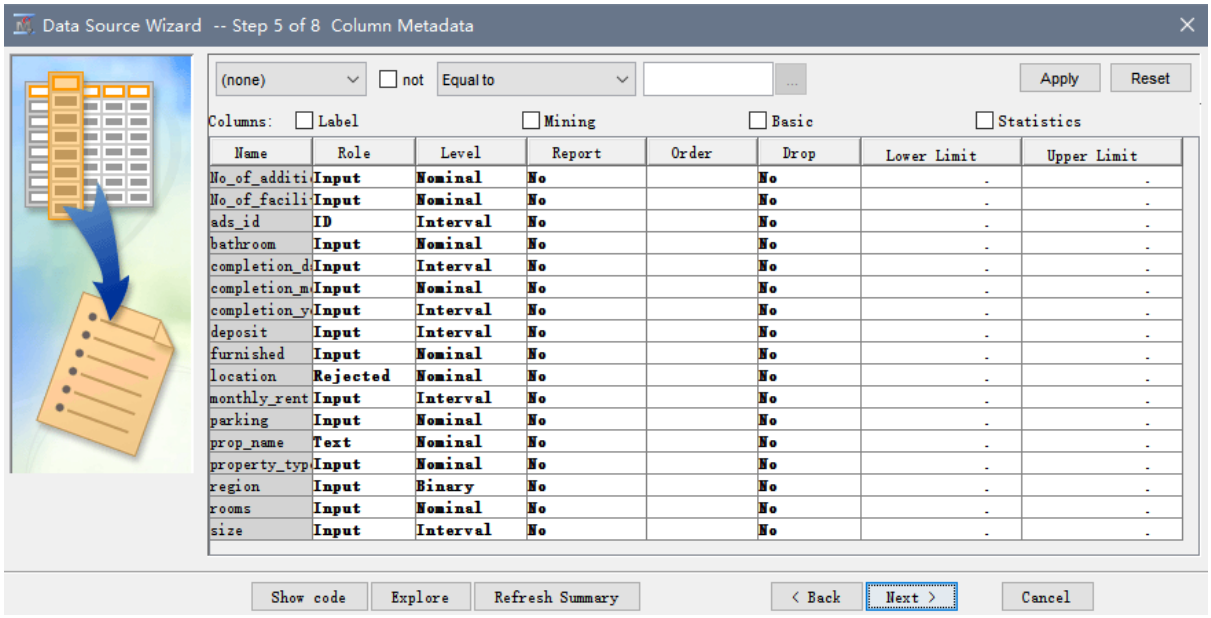


Figure 3.3.3 Column metadata (advanced setting)

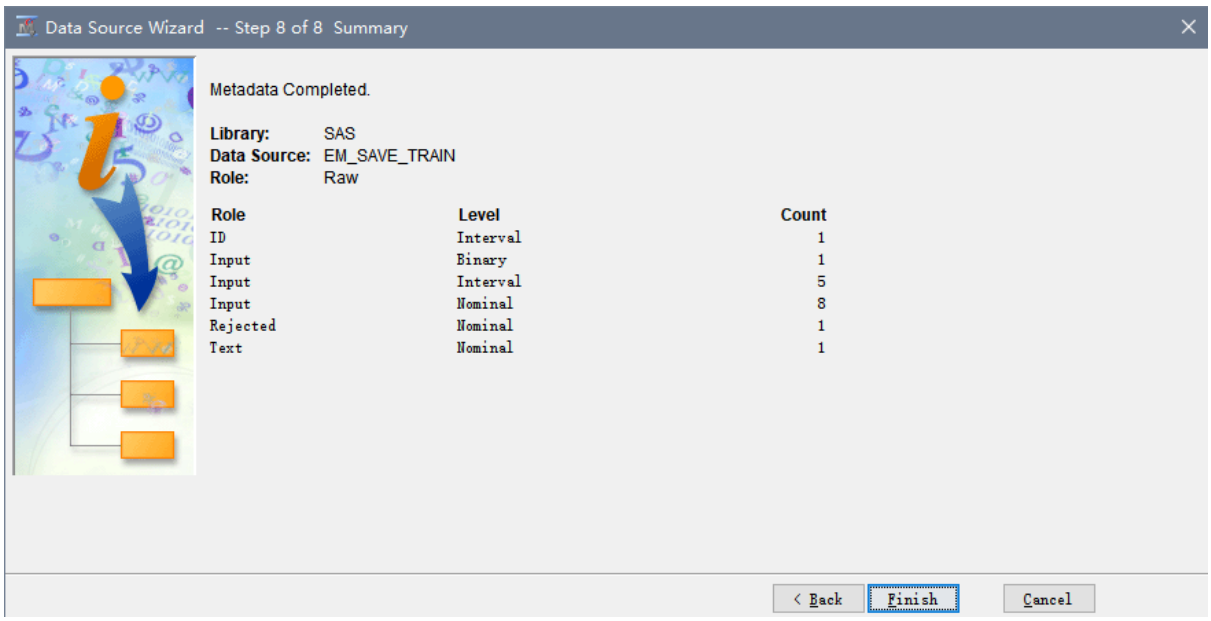


Figure 3.3.4 Data type summary (Advanced setting)

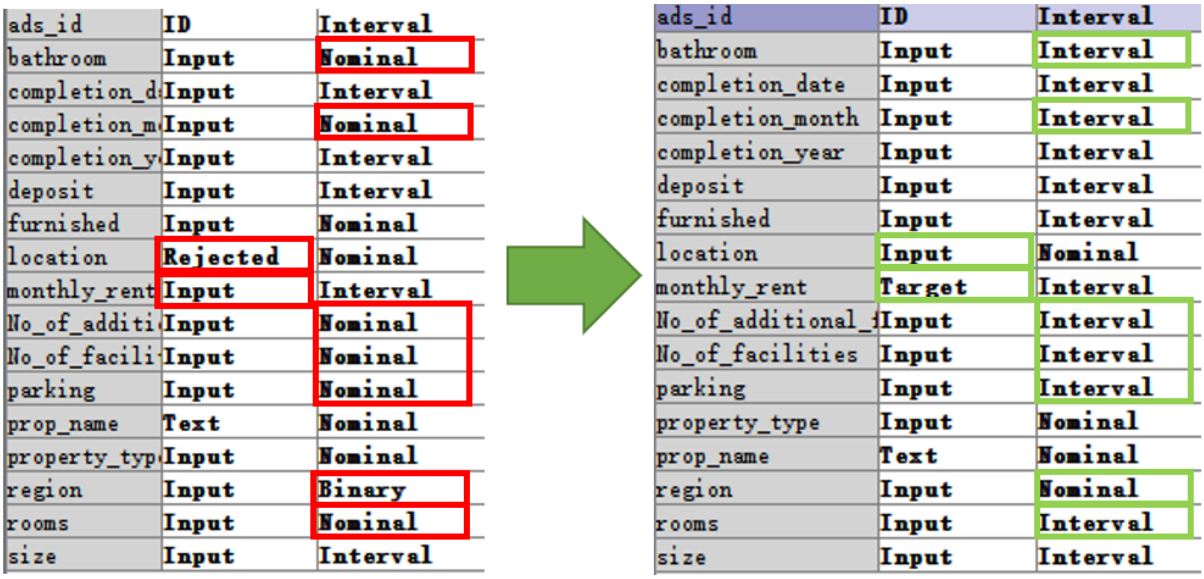
As shown in Figure *3.3*.3 and *3.3*.4, with the advanced settings selected, the data types are binary, nominal, and unary. The system can automatically recognize the role of variables to execute input variables and reject variables.

In our task, the data type does not correspond to the actual data type of the dataset. Therefore, a manual correction is required to modify the data type before entering the exploration phase.

4. Results

4.1. Reclassification of the Role and Variables Exploration

In order to be able to display the graphs correctly, the types of some variables were reclassified. The left side of Figure 4.1.1 depicts the manually corrected data type summary, while the right side of Figure 4.1.2 represents the finalized data type summary.



ads_id	ID	Interval
bathroom	Input	Nominal
completion_d	Input	Interval
completion_m	Input	Nominal
completion_y	Input	Interval
deposit	Input	Interval
furnished	Input	Nominal
location	Rejected	Nominal
monthly_rent	Input	Interval
No_of_additi	Input	Nominal
No_of_facili	Input	Nominal
parking	Input	Nominal
prop_name	Text	Nominal
property_typ	Input	Nominal
region	Input	Binary
rooms	Input	Nominal
size	Input	Interval

ads_id	ID	Interval
bathroom	Input	Interval
completion_date	Input	Interval
completion_month	Input	Interval
completion_year	Input	Interval
deposit	Input	Interval
furnished	Input	Interval
location	Input	Nominal
monthly_rent	Target	Interval
No_of_additional_f	Input	Interval
No_of_facilities	Input	Interval
parking	Input	Interval
property_type	Input	Nominal
prop_name	Text	Nominal
region	Input	Nominal
rooms	Input	Interval
size	Input	Interval

Figure 4.1.1 Comparison between role and measurement level between advance settings and manual reclassification

All variables were changed to input variables because all data had to go into SASEM before any analysis could be performed, as shown in Figure 4.1.1. The target variables were manually corrected. region was converted from a binary variable to nominal, and all other variables were changed to int input except ads_id. The rest of the variables are left unchanged.

As shown in Figure 4.1.2, with the advanced settings selected, the data types are divided into nominal and Interval.

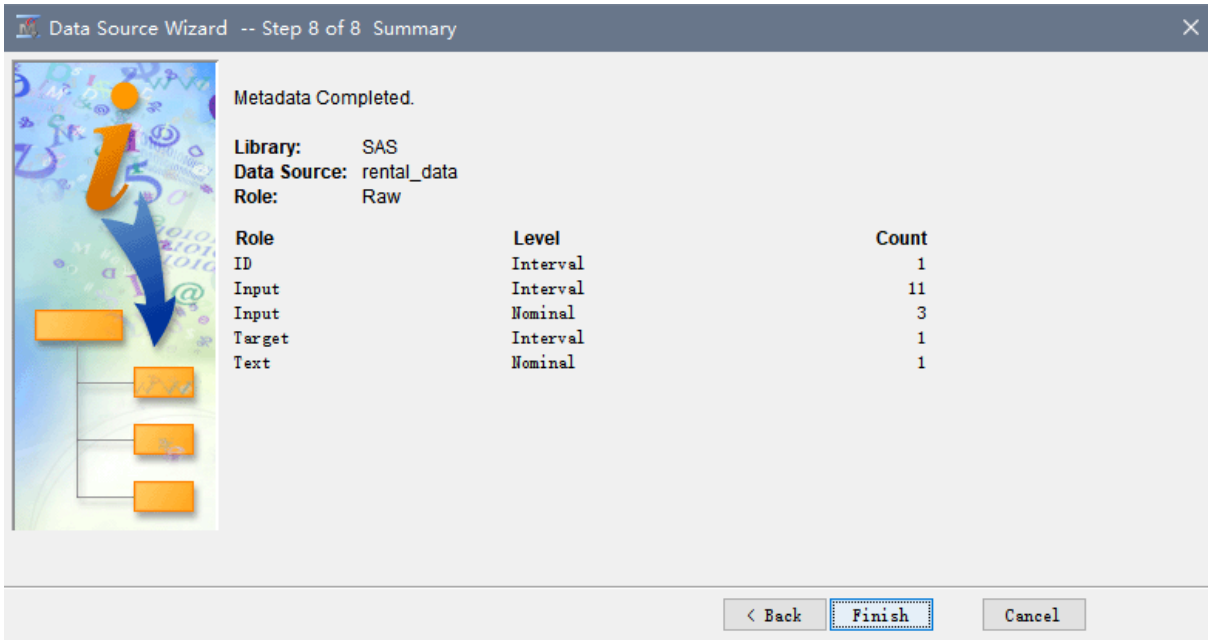


Figure 4.1.2 data type summary (manual reclassification)

After accessing and manually correcting the dataset, a summary statistic is produced. The purpose of the summary statistics is to provide a high-level overview of the data being analyzed. Since all variables are categorical variables, the summary statistics are shown in Figure 4.1.3.

Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
No_of_additional_facilities	INPUT	2.931995	1.252602	14043	5948	1	3	6	0.069955	-0.9825
No_of_facilities	INPUT	7.865932	3.381057	17782	2209	1	8	14	0.035349	-0.65842
bathroom	INPUT	1.891719	0.556266	19985	6	1	2	8	1.202556	9.399724
completion_date	INPUT	8.560502	10.20208	19991	0	0	3	31	0.784803	-0.86594
completion_month	INPUT	3.523886	4.116794	19991	0	0	1	12	0.743443	-0.92768
completion_year	INPUT	1089.099	1004.131	19991	0	0	2004	2025	-0.16265	-1.97365
deposit	INPUT	4017.463	2468.149	19989	2	140	3600	55500	4.773398	53.04783
furnished	INPUT	1.977084	0.90741	19986	5	1	2	3	0.045111	-1.78409
parking	INPUT	1.415284	0.548572	14289	5702	1	1	7	1.403437	6.196853
rooms	INPUT	2.680278	0.807838	19983	8	1	3	10	-0.49861	1.517796
size	INPUT	918.1943	354.528	19991	0	1	886	10000	5.975658	104.1953
monthly_rent	TARGET	1609.779	926.3814	19989	2	70	1400	18500	4.87562	51.10825

Figure 4.1.3 interval variable summary statistics

Figure 4.1.3 shows the number and level pattern of each category variable. In addition, it can be noted that some category variables, such as number of facilities, number of additional facilities, year of completion, and parking spaces, have a large number of missing values. In the SIZE column, the smallest variable is 1 the largest variable is 10000 which is not normal. In the year of completion column, the maximum variable is 2025, which is also abnormal Figure 4.1.4 Class variable summary statistics.

ads_id	prop_name	completion_year	completion_month	completion_date	monthly_rent	location	property_type	rooms	parking	bathroom	size	furnished	region	No_of_facilities	No_of_additional_facilities	deposit
100323185	The Hipster...	2022	5	4	4200	Taman Des...	Condomini...	5	2	6	1842	1	Kuala Lum...	10	3	12600
100203973	Segar Cour...	0	0	0	2300	Cheras ...	Condomini...	3	1	2	1170	3	Kuala Lum...	9	3	4600
100323128	Pangsapuri...	0	0	0	1000	Taman Des...	Apartment ...	3	.	2	650	1	Kuala Lum...	4	.	2000
100191767	Sentul Poin...	2020	5	14	1700	Sentul ...	Apartment ...	2	1	2	743	3	Kuala Lum...	8	3	5100
97022692	Arte Mont Ki...	0	0	0	1299	Mont Kiara ...	Service Re...	1	1	1	494	2	Kuala Lum...	11	1	2598
100322897	Residensi ...	0	0	0	1500	Setapak ...	Apartment ...	3	1	2	884	3	Kuala Lum...	6	2	4500
100322962	Sky Meridie...	0	0	0	2900	Sentul ...	Service Re...	3	2	2	982	1	Kuala Lum...	10	4	8700
100322885	Arte Plus Ja...	2018	4	22	1550	Ampang ...	Service Re...	1	1	1	700	1	Kuala Lum...	7	4	4650
100322866	Nova I ...	2014	2	3	1400	Segambut ...	Apartment ...	2	1	1	750	1	Kuala Lum...	5	4	2800
100322863	Sofiya Resi...	0	0	0	1350	Desa Park...	Condomini...	3	1	2	862	3	Kuala Lum...	6	3	4050
100322813	The Park S...	2019	5	28	2600	Bukit Jalil ...	Service Re...	2	.	2	868	3	Kuala Lum...	7	2	7800
100322809	PV9 Reside...	2022	10	6	2000	Setapak ...	Service Re...	4	2	2	1100	3	Kuala Lum...	9	4	6000
100322802	Arte Plus Ja...	2018	3	12	1500	Ampang ...	Service Re...	1	1	1	700	1	Kuala Lum...	13	4	4500
87950203	Maxim Citili...	2017	1	2	1300	Sentul ...	Service Re...	3	1	2	1009	1	Kuala Lum...	11	3	2600
100239196	Greenview ...	0	0	0	1000	Kepong ...	Apartment ...	5	.	2	855	3	Kuala Lum...	3	3	3000
100322320	Legasi Ka...	2020	1	28	3200	KL City ...	Apartment ...	3	1	2	950	1	Kuala Lum...	13	5	9600
100322311	Legasi Ka...	2020	1	9	3200	KL City ...	Apartment ...	3	1	2	950	1	Kuala Lum...	13	5	6400
100322212	Majestic Ma...	2021	10	2	1400	Cheras ...	Service Re...	2	2	2	650	2	Kuala Lum...	8	3	4200
100273500	Desa Villas...	0	0	0	2500	Wangsa Ma...	Condomini...	4	2	2	1784	2	Kuala Lum...	4	2	5000
100231953	East Side O...	0	0	0	1800	Ampang ...	Condomini...	3	1	2	1100	1	Kuala Lum...	8	.	5400
100322123	Majestic Ma...	2021	9	1	1099	Cheras ...	Service Re...	2	1	2	650	2	Kuala Lum...	8	3	3297
100310024	Majestic Ma...	2021	8	7	1099	Cheras ...	Service Re...	2	1	2	650	3	Kuala Lum...	8	4	2198
100310022	Majestic Ma...	2021	2	27	1199	Cheras ...	Service Re...	3	1	2	819	2	Kuala Lum...	8	4	3597
100290514	Majestic Ma...	2021	12	14	1100	Cheras ...	Service Re...	2	1	2	650	2	Kuala Lum...	8	2	2200
100267049	Sri Penara ...	0	0	0	1000	Cheras ...	Apartment ...	3	.	2	650	2	Kuala Lum...	3	.	3000
100201993	Prisma Ch...	0	0	0	1100	Cheras ...	Condomini...	1	1	1	900	3	Kuala Lum...	9	.	2200
100295196	Menara Alp...	0	0	0	1400	Wangsa Ma...	Condomini...	3	1	2	1140	2	Kuala Lum...	6	.	4200
100322088	Berlian Res...	0	0	0	1500	Setapak ...	Condomini...	3	1	2	900	2	Kuala Lum...	5	.	3000
10032208628	Dutama...	2017	3	21	2500	Solaris Dut...	Condomini...	4	2	3	1500	3	Kuala Lum...	6	2	5000
10032210028	Dutama...	2017	12	19	2900	Solaris Dut...	Condomini...	4	2	3	1719	1	Kuala Lum...	6	2	5800
100322078	Pangsapuri...	2006	11	29	1750	Sentul ...	Apartment ...	3	.	2	1065	3	Kuala Lum...	.	.	5250
100202252	Residensi ...	0	0	0	1800	Setapak ...	Apartment ...	3	1	2	885	1	Kuala Lum...	.	.	3600
100136931	Bayu Sentul...	2015	8	18	1900	Sentul ...	Condomini...	3	2	2	1300	3	Kuala Lum...	.	.	5700
100234716	Flexus Sign...	0	0	0	1500	Jalan Kuchi...	Studio	1	1	1	530	1	Kuala Lum...	6	3	4500
100322047	Menjalara 1...	0	0	0	2400	Bandar Me...	Condomini...	3	1	2	1316	1	Kuala Lum...	8	3	4800
93075352	G Residenc...	2015	3	19	4500	Desa Pand...	Service Re...	3	2	2	1539	1	Kuala Lum...	4	4	9000
99876261	Greenpark ...	1999	6	5	1400	Old Klang ...	Condomini...	3	1	2	1270	3	Kuala Lum...	10	3	2800
100322029	Majestic Ma...	2021	10	15	1800	Cheras ...	Service Re...	3	1	2	819	3	Kuala Lum...	10	4	3600
100322005	Baiduri Apa...	0	0	0	1500	Desa Pand...	Apartment ...	3	1	2	950	3	Kuala Lum...	2	.	4500
98292007	Bennington...	2019	10	3	3000	Setapak ...	Condomini...	3	2	2	1200	1	Kuala Lum...	11	4	9000
100321936	Sri Putrama...	2004	10	3	1600	Jalan Kuchi...	Condomini...	3	1	2	1100	1	Kuala Lum...	9	3	3200
100321976	Putra Villa ...	2007	8	9	1800	Setapak ...	Condomini...	3	.	2	1066	1	Kuala Lum...	.	.	5400
9962736238	Bidara ...	0	0	0	2100	KLCC ...	Condomini...	2	.	2	706	1	Kuala Lum...	9	.	4200
994923923	Towers ...	2015	9	28	2500	Ampang Hil...	Condomini...	2	1	1	890	1	Kuala Lum...	6	.	5000
100036659	One Maxim ...	2020	1	15	1300	Sentul ...	Service Re...	2	1	2	650	3	Kuala Lum...	9	.	2600
99528357	Casa Mutia...	2007	8	16	2000	Bukit Bintan...	Service Re...	2	1	2	742	1	Kuala Lum...	7	.	6000
100321885	Sofiya Resi...	0	0	0	2000	Desa Park...	Condomini...	3	1	2	862	1	Kuala Lum...	2	3	6000
100273689	Laman Tas...	2002	6	7	2100	Cheras ...	Condomini...	4	1	3	1300	1	Kuala Lum...	6	.	4200
100033999	Residensi ...	0	0	0	1300	KL City ...	Apartment ...	3	1	2	805	3	Kuala Lum...	9	.	2600
100033960	The Hamilt...	2020	5	3	2600	Wangsa Ma...	Condomini...	4	2	2	1000	1	Kuala Lum...	8	.	5200
998446673	Towers ...	2015	7	19	2000	Ampang Hil...	Service Re...	1	1	1	440	1	Kuala Lum...	6	.	4000
100321850	Menjalara 1...	0	0	0	2500	Bandar Me...	Condomini...	5	2	3	1600	1	Kuala Lum...	8	3	7500
100321800	The Havre ...	2020	9	25	2500	Bukit Jalil ...	Condomini...	3	2	2	1028	1	Kuala Lum...	14	4	7500
94858400	...	0	0	0	2000	Setapak ...	Condomini...	4	.	2	940	1	Kuala Lum...	10	4	4000
93420010	...	0	0	0	1800	Setapak ...	Condomini...	3	1	2	860	1	Kuala Lum...	10	4	5400

Figure 4.1.4 Overview of metadata (first 50 records)

Univariate analysis is the simplest form of statistical data analysis to explore each variable in the dataset. It does not deal with the causal or relationships but solely finds patterns from each of the variables. Graphs such as histogram, box plot and pie chart are best suited for conducting the univariate analysis to check pattern distribution, outliers, noisy data and any missing value.

4.2. Histogram findings

Firstly, histogram was used to visualize distribution and missing values for the 9 interval variables. Figure 4.2.1 showed the overview of all graphs and Table 4.2.1 summarized the patterns, analysis, and abnormalities.

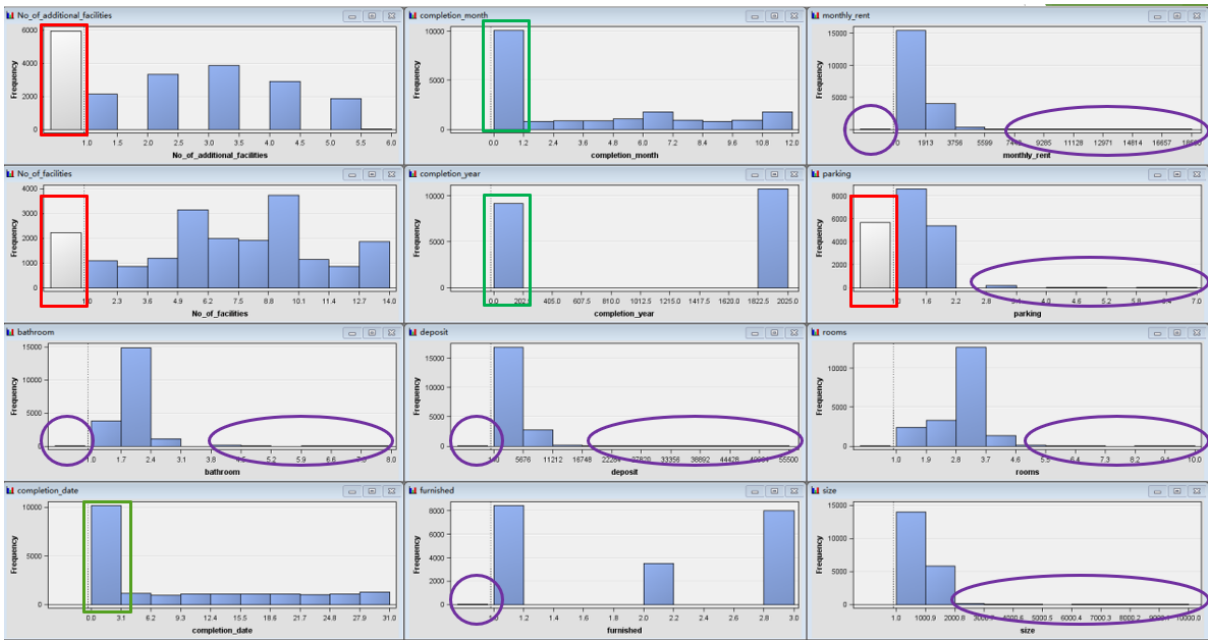
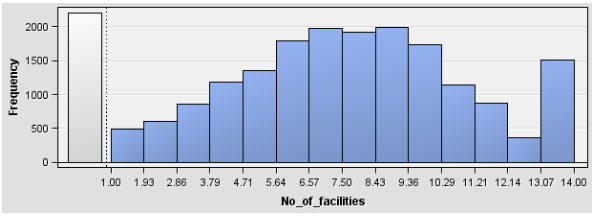
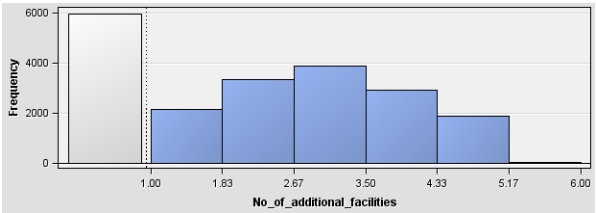
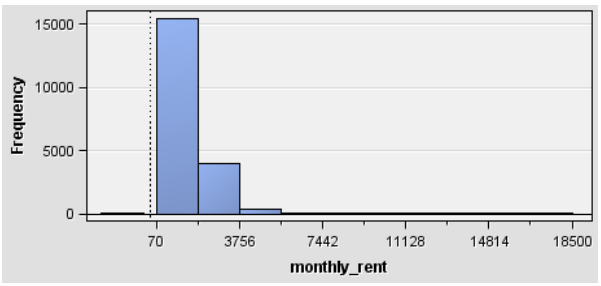
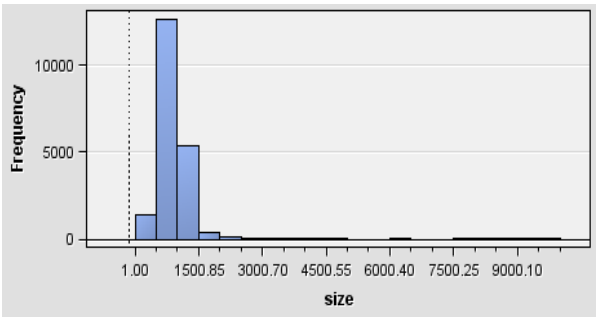
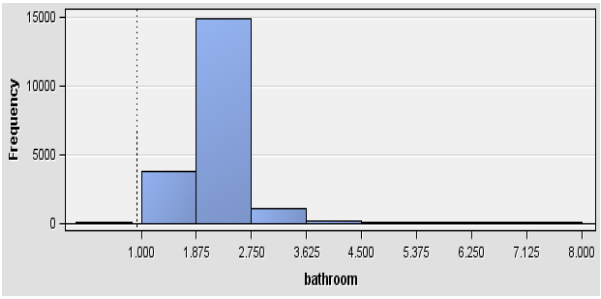
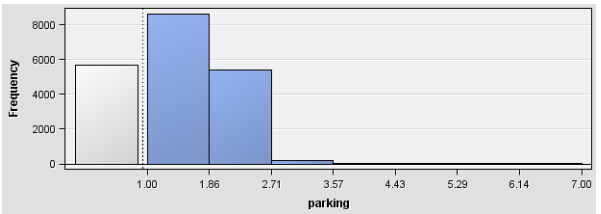


Figure 4.2.1 Histogram of interval variables

No.	Variable	Findings
1	 *Figure 4.2.2 no_of_facilities*  *Figure 4.2.3 no_of_additional_facilities*	There are some rental properties that have additional amenities, and this is usually a factor to increase the rent. **No outliers** detected. Over 2000 **missing values**. Small size of **outliers** detected. Over 5000 **missing values**.

2	 *Figure 4.2.4 monthly rent*	Right-skewed distribution. Most of the monthly rent is between 70-3756. **Noise** detected.
3	 *Figure 4.2.5 size*	Right-skewed distribution. Rental size within 1500sq.ft No **missing** values. **Noise** detected.
4	 *Figure 4.2.6 bathroom*	Right-skewed distribution. Most rental houses have only 1-2 bathrooms, which is common. Few **missing** values. **Outliers** detected.
5	 *Figure 4.2.7 parking*	Right-skewed distribution. Most rental houses have only 1-2 parking, which is common. Over 4000 **missing**. **Outliers** detected.

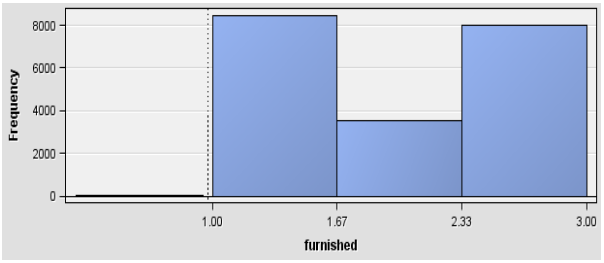
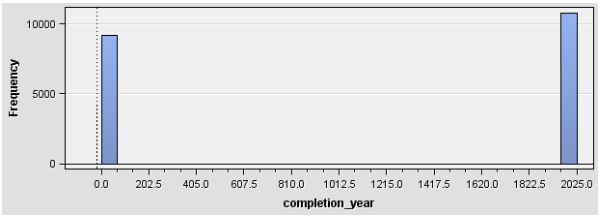
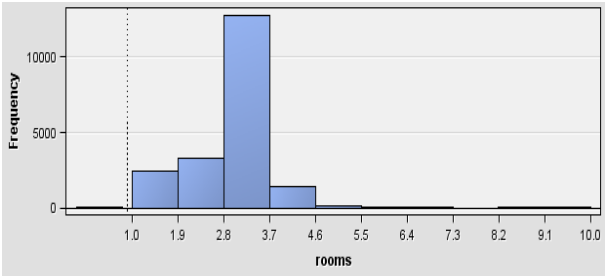
6	 *Figure 4.2.8 furnished*	Most of the rental houses have 1 set of furniture or 3 sets of furniture. Few **missing** values.
7	 *Figure 4.2.9 completion year*	**Inconsistent** data value was detected for unspecified year as 0. Normally, the year should represent a valid timestamp and should not be 0. If there are records in the dataset with a year of 0, this could be the result of a data entry error, missing data, or other data quality issues.
8	 *Figure 4.2.10 rooms*	Right-skewed distribution. A large number of rental houses have 4 rooms. Few missing values. **Outliers** detected.

Table 4.2.1 Histogram's findings

4.3. Box plot findings

Besides that, boxplot is recognized as one of the great tools to visualize outliers. Same to the findings from histogram, boxplots were created for the 12 interval variables to validate the presence of outliers as shown in Figure 4.3.1. Table 4.3.1 summarized the findings of outliers.

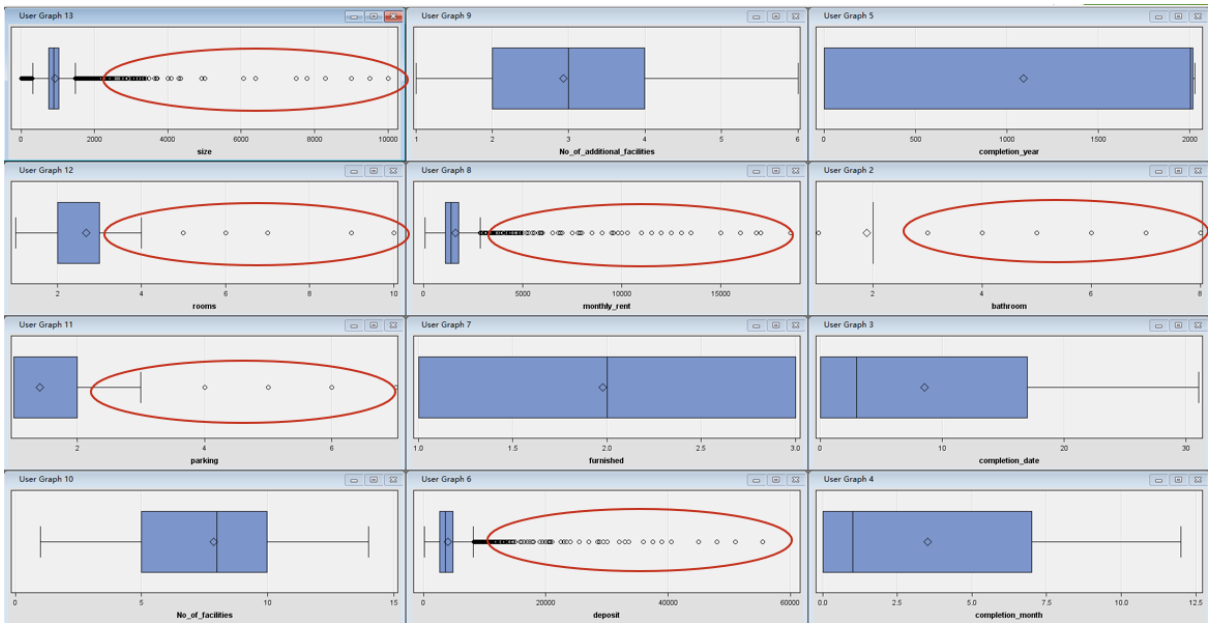


Figure 4.3.1 Box plots of interval variables

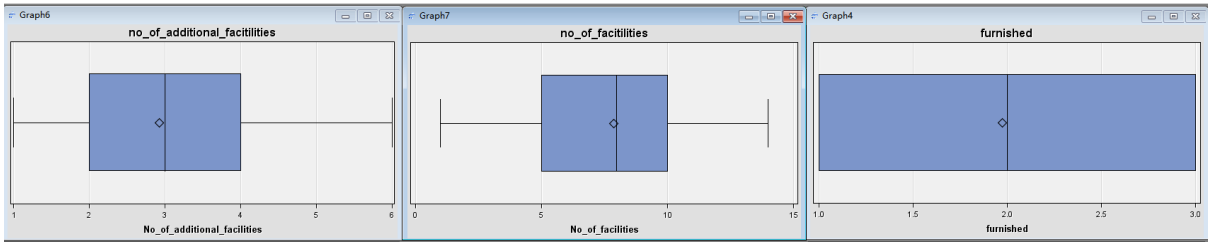


Figure 4.3.2 no outliers

No outliers detected.

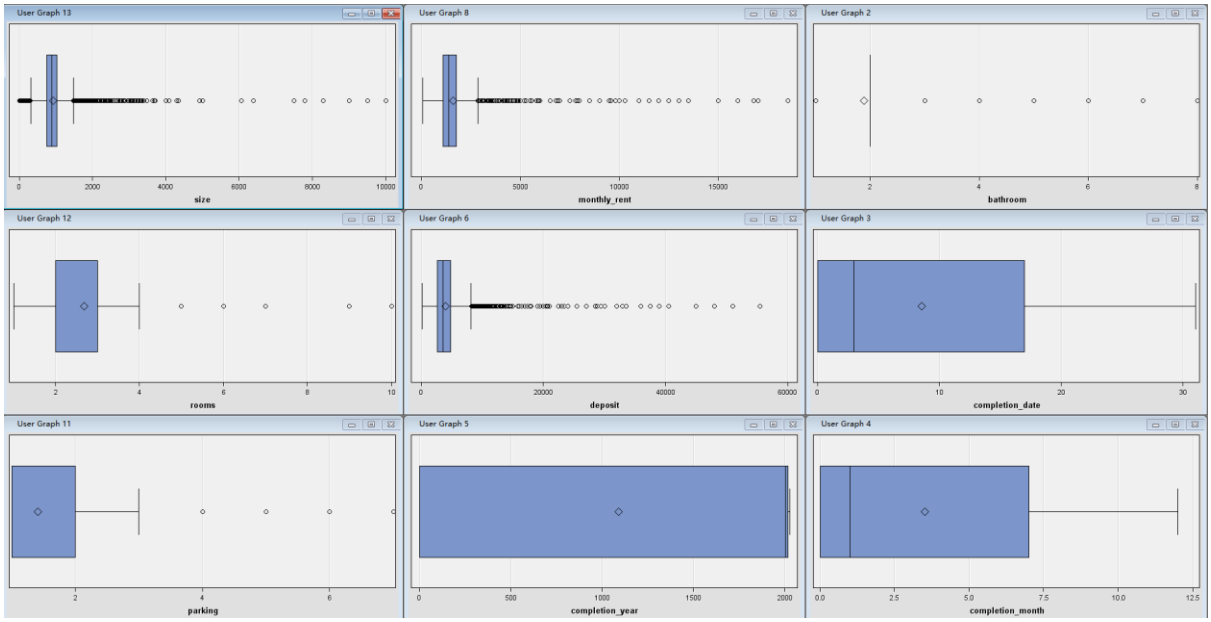


Figure 4.3.3 outliers exists

As shown in figure 4.3.3 The highest value of completion year reaches 2025, this year is 2023，it is possible that it is a term house. There are large number of value with 0 in completion year, month, and date. So, the box seems left-skewed distribution. These value can be processed as missing values.

Size, monthly_rent and deposit contain few very large noisy data which better be removed.

The nature of the remaining variables’ outliers needed to be examined further to conclude whether they were caused by error or omission or natural behavior prior to dispelling them from project.

4.4. Pie chart findings

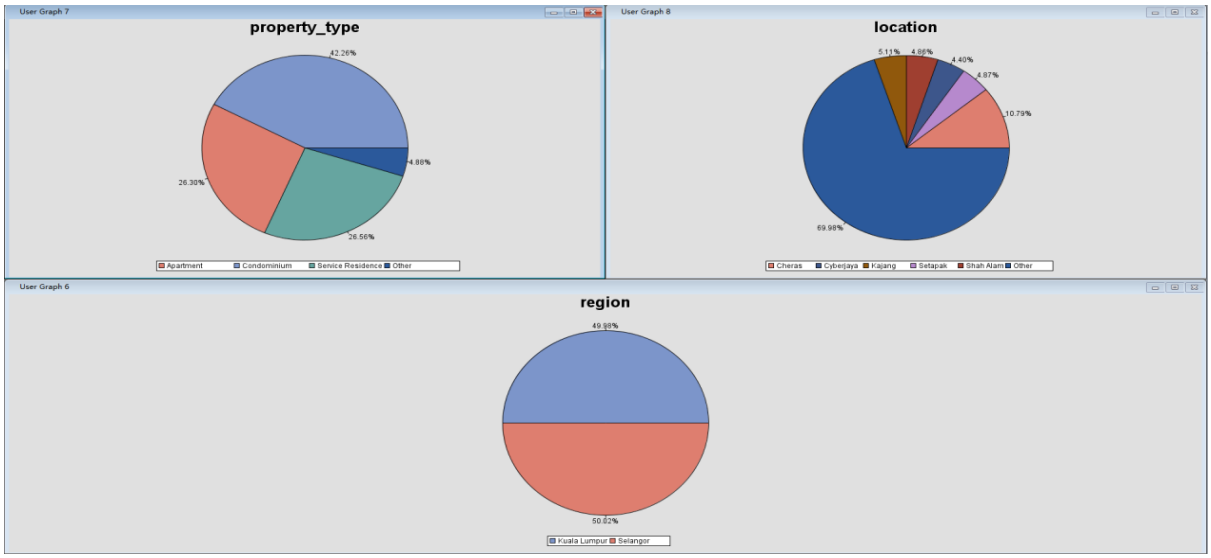


Figure 4.4.1 Pie charts of nominal variables

4.4.2 shows the nominal variables distribution using pie chart.

Figure 4.4.2 is interval but wrongly assigned to nominal in advanced setting

Figure 4.4.2 shows a significant number of blank values exists in these variables: parking，no_of_facillties, no_of_addtional_facillties.

4.5. Bar chart findings

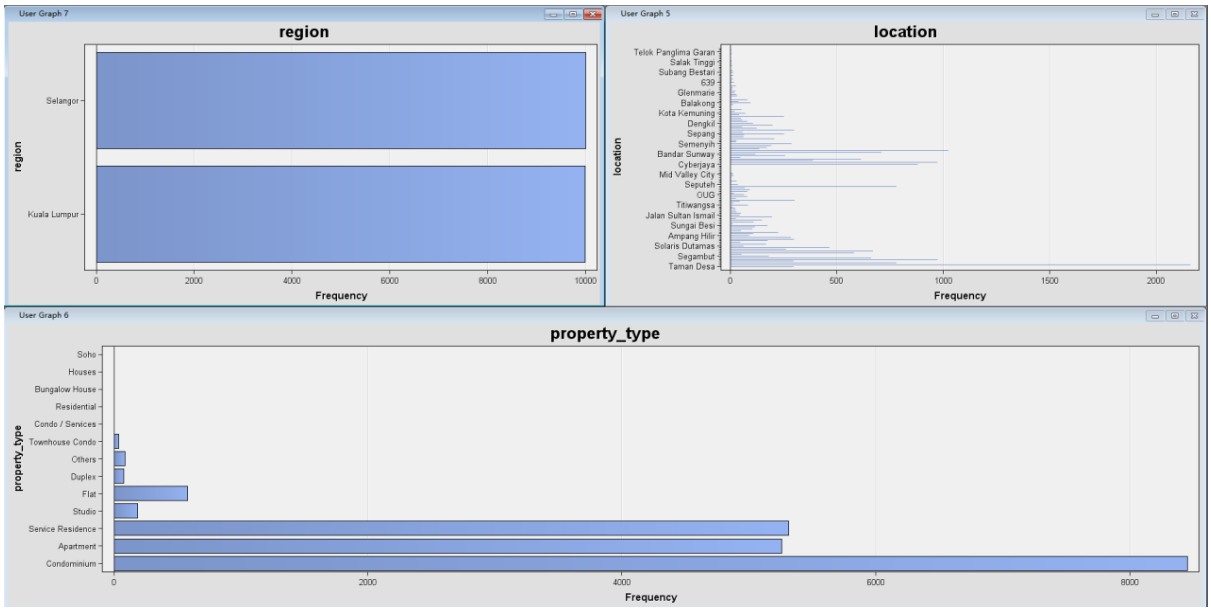


Figure 4.5.1 bar chart for nominal variables

In Figure 4.5.1, the rental houses are located in Kuala Lumpur and Selangor. More than 2000 houses of Location are in Tanman Desa. More than 8000 Property type is counominium.

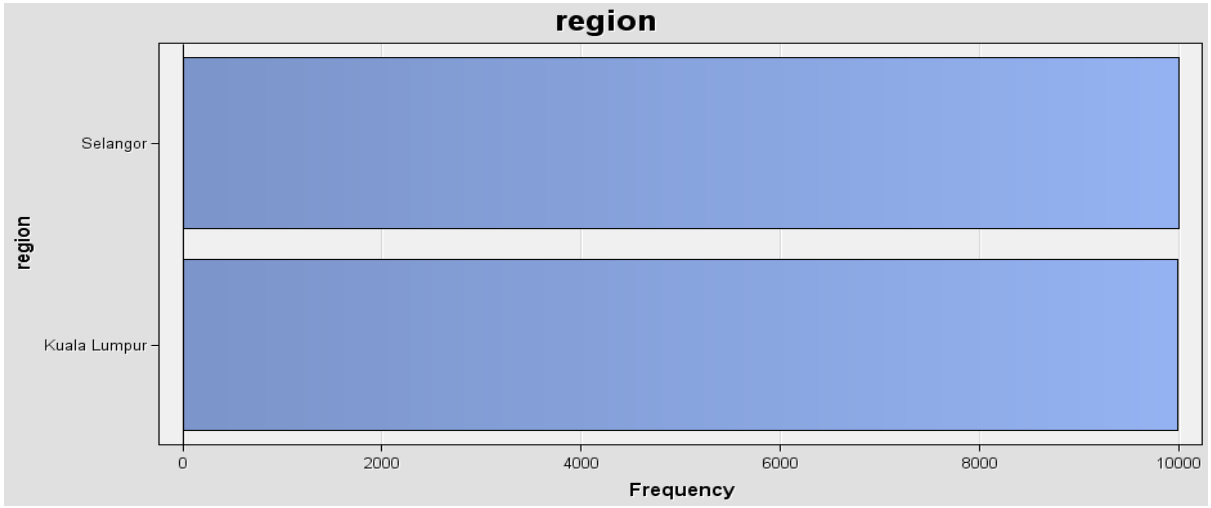


Figure 4.5.2 bar chart findings

From the figure 4.5.2 bar chart, it can be seen that the geographical distribution of the data set is very even, half of the data comes from Kuala Lumpur, and half of the data comes from Selangor.

4.6. bivariate analysis

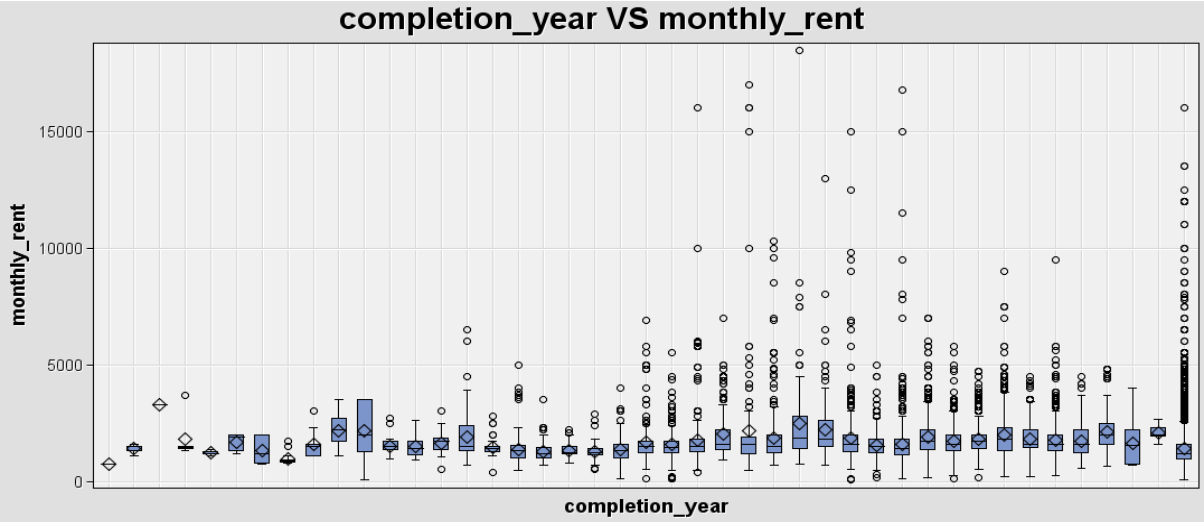


Figure 4.6.1 year vs rent

Figure 4.6.1 shows a box line plot of monthly rent vs. year of completion. It can see that the closer the year of completion is to the present, the higher the rent, but not too close to the present, a period of time is best. Most people will choose a house with 2-3 years of completion year from now for health reasons, so it also leads to higher rents.

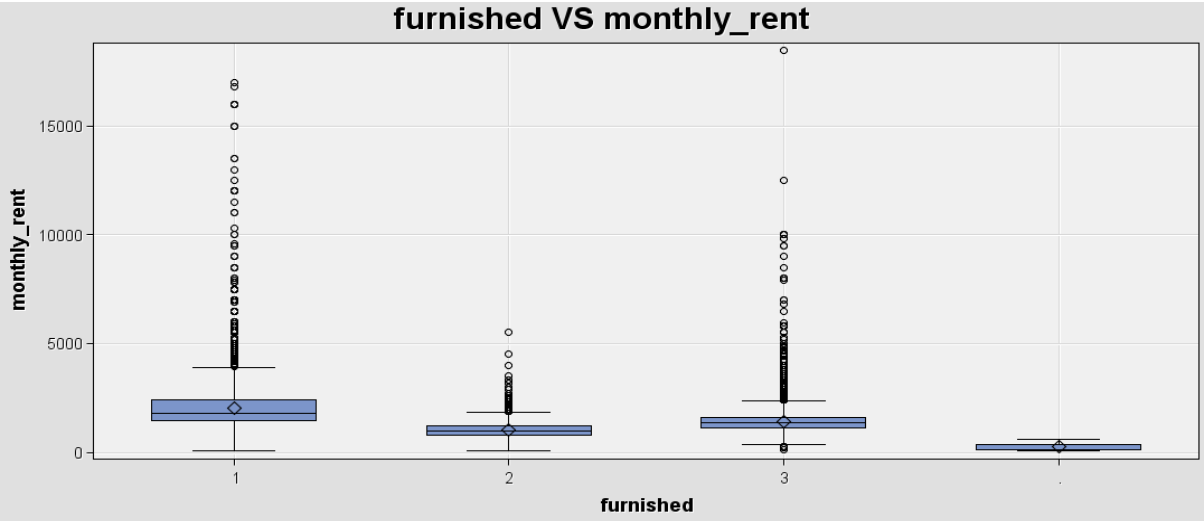


Figure 4.6.2 furnished vs rent

Figure 4.6.2 shows a box line plot of monthly rent vs. furniture. It can see that the price of monthly rent with 1 set of furniture and 3 sets of furniture is on the high side, and the maximum value of monthly rent with 2 sets of furniture will be low instead.

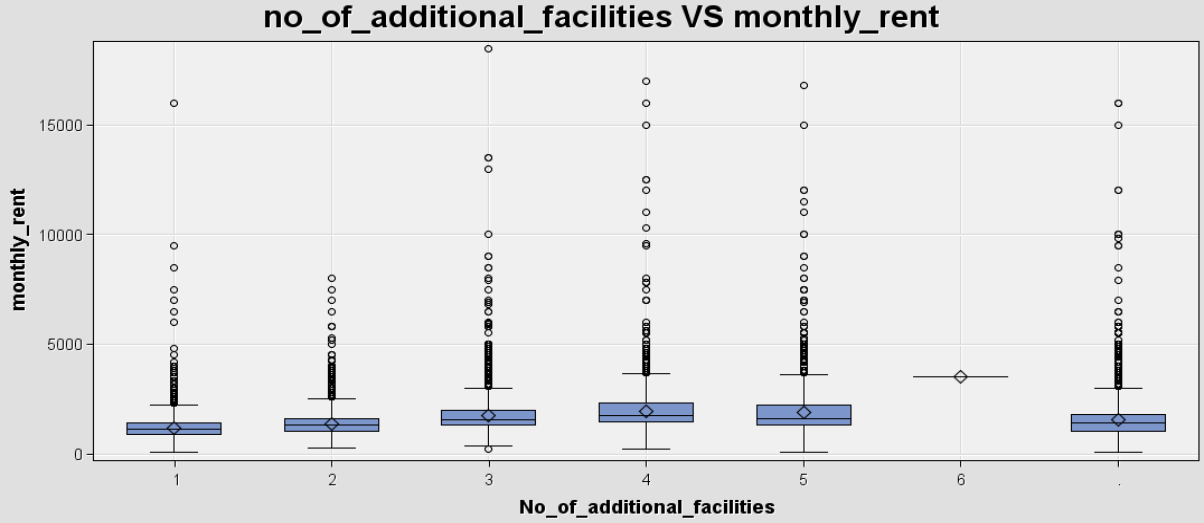


Figure 4.6.3 no of additional facilities vs monthly rent

Figure 4.6.3 shows a box line plot of monthly rent vs. additional facilities. It can see that there are 3 add-ons that have high maximum monthly rents, but their average values are within the normal range. Looking at the average values, the number of additional facilities is directly proportional to the monthly rent. There are a large number of missing values, and it is possible that these missing values are not entered or that there are rental houses without additional facilities.

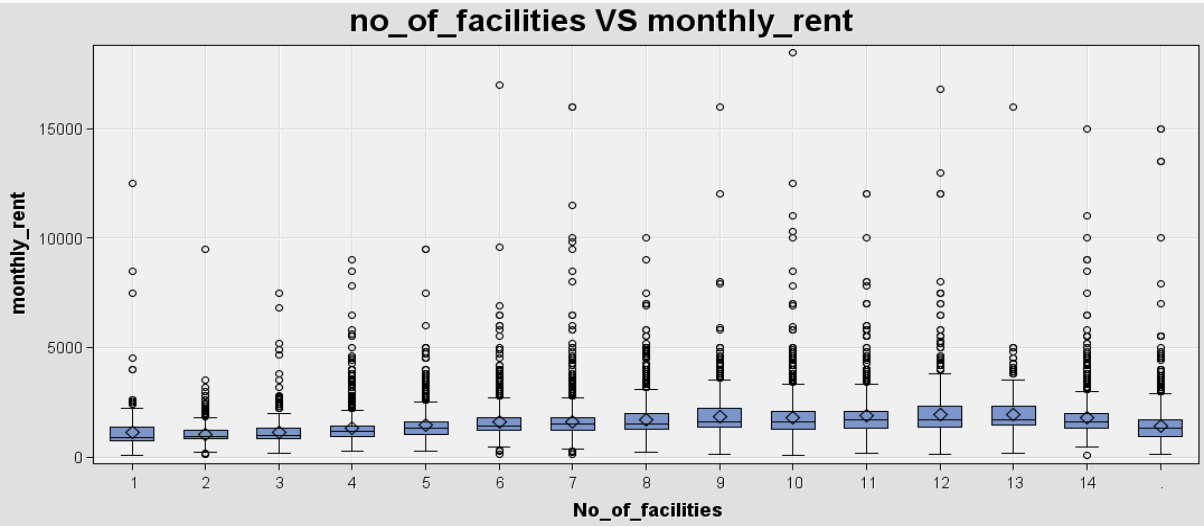


Figure 4.6.4 no of facilities vs monthly rent

Figure 4.6.6 shows a box line plot of monthly rent vs. amenities. It can see from the digit and average values that the more amenities the higher the monthly rent, and it is possible that these missing values were not entered or that there are rental homes without amenities.

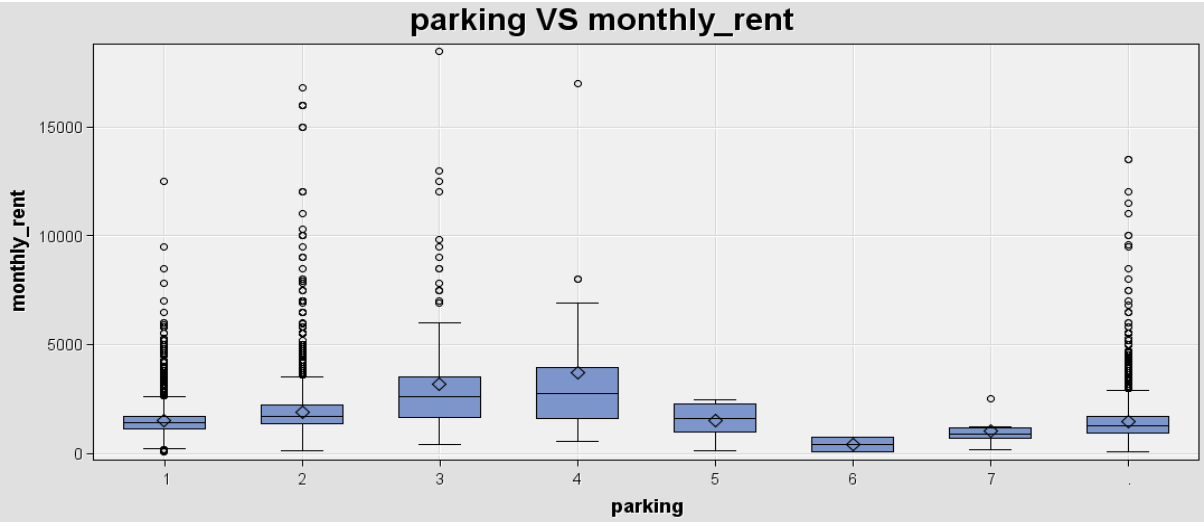


Figure 4.6.5 parking vs monthly rent

Figure 4.6.5 shows a box line plot of monthly rent vs. parking spaces. It can see from the median and mean that rentals with 1-3 parking spaces are more popular and people are usually willing to pay higher monthly rents because the average household has 1-3 cars. More parking spaces may not be popular. The other ones have a large number of missing values, and it is possible that these missing values were not entered, or that some of the rentals do not have parking.

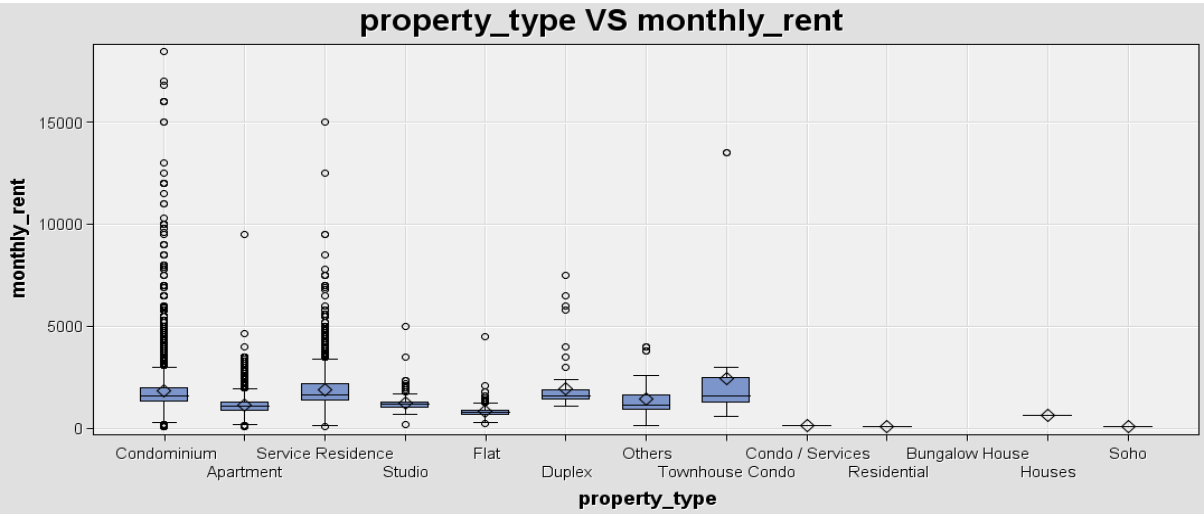


Figure 4.6.6 property type vs monthly rent

Figure 4.6.6 shows a box line plot of monthly rent vs. property. It can see that apartments and service residences are more popular and people are willing to afford high monthly rents.

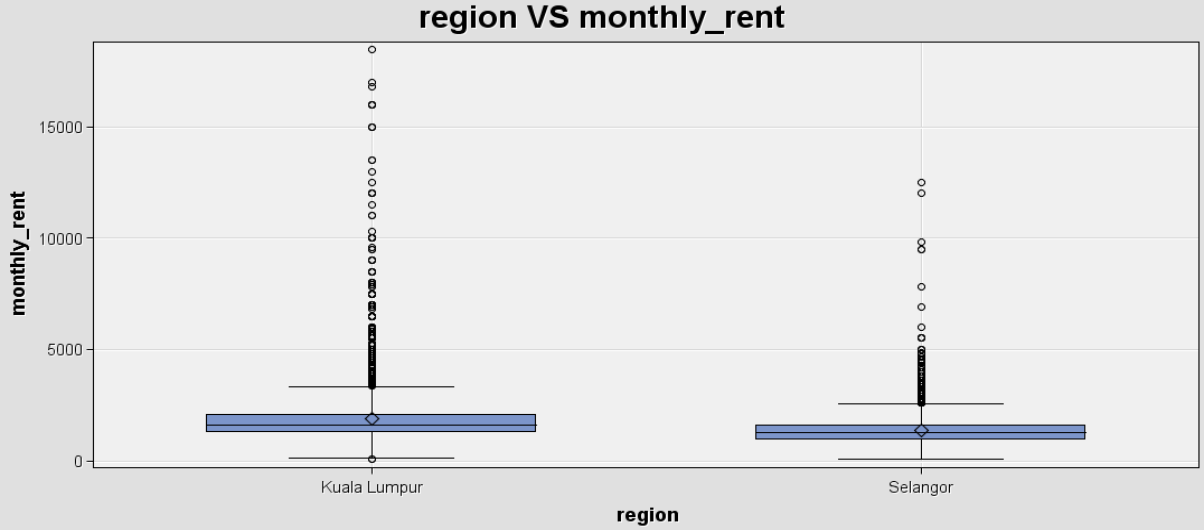


Figure 4.6.7 region vs monthly rent

Figure 4.6.7 shows a box line plot of monthly rent vs. region. It can be seen that rental houses in Kuala Lumpur are more popular. The few noises which monthly_rent include are more concentrated in KL.

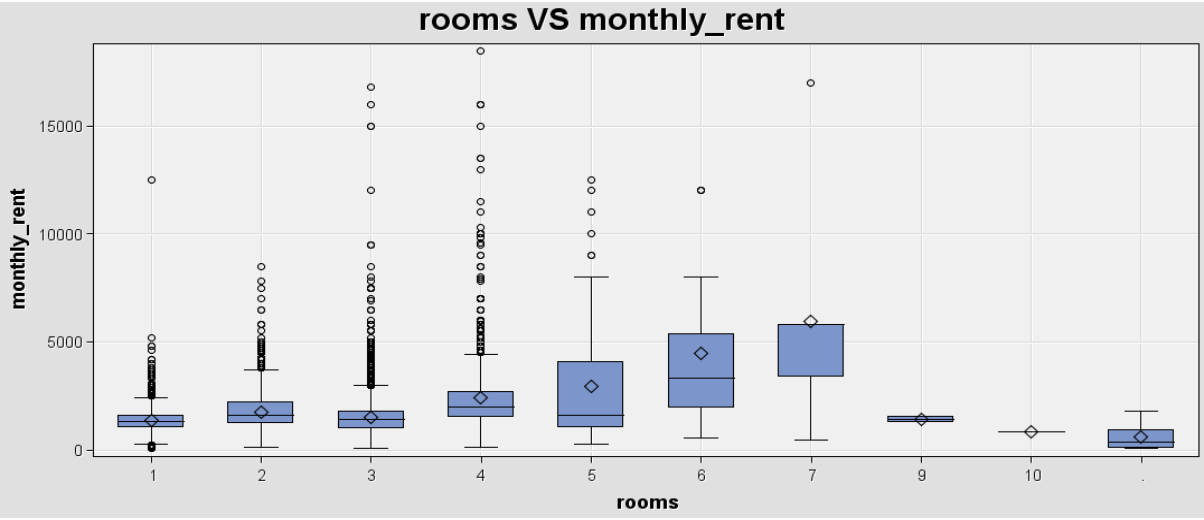


Figure 4.6.8 rooms vs monthly rent

Figure 4.6.8 shows a box line plot of monthly rent vs rooms. It can understand from the median and the mean that the monthly rent for a rental house within 7 rooms is gradually increasing, and in the case of 3-4 romms higher rents can also occur, which is in line with the demand for rental housing.

4.7. Multivariate analysis findings

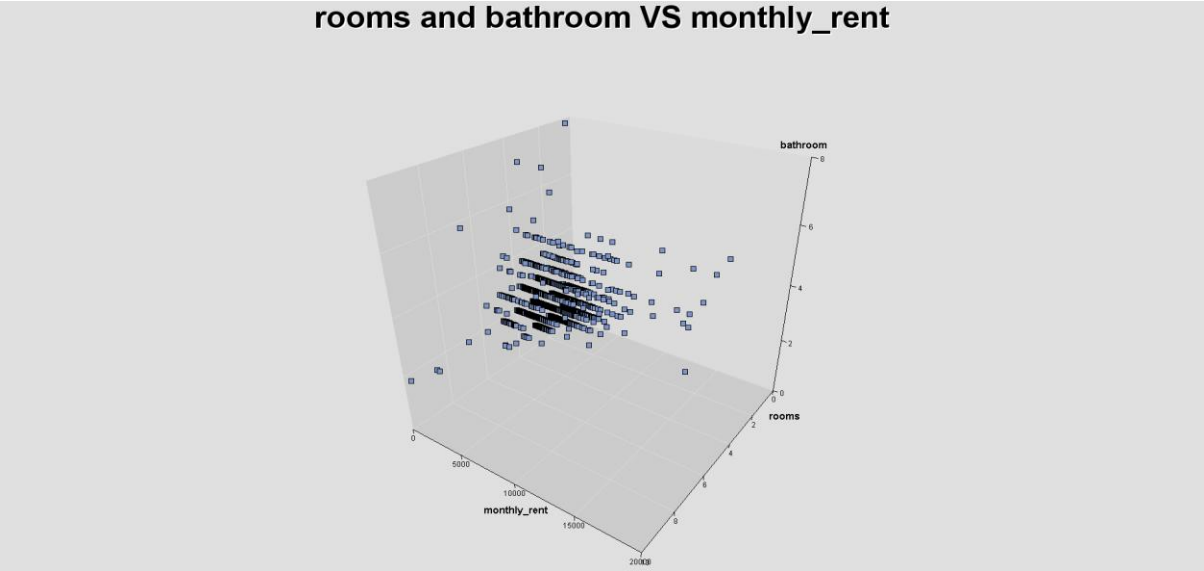


Figure 4.7.1 rooms vs bathrooms vs monthly rent

According to Figure 4.7.1, under bathroom, there are 2 arrows shaped as monthly rent (Y) and rooms (N) formed. It can see that the space points are around monthly rent 5000, rooms around 4, and bathroom is the most dense between 2 and 4.

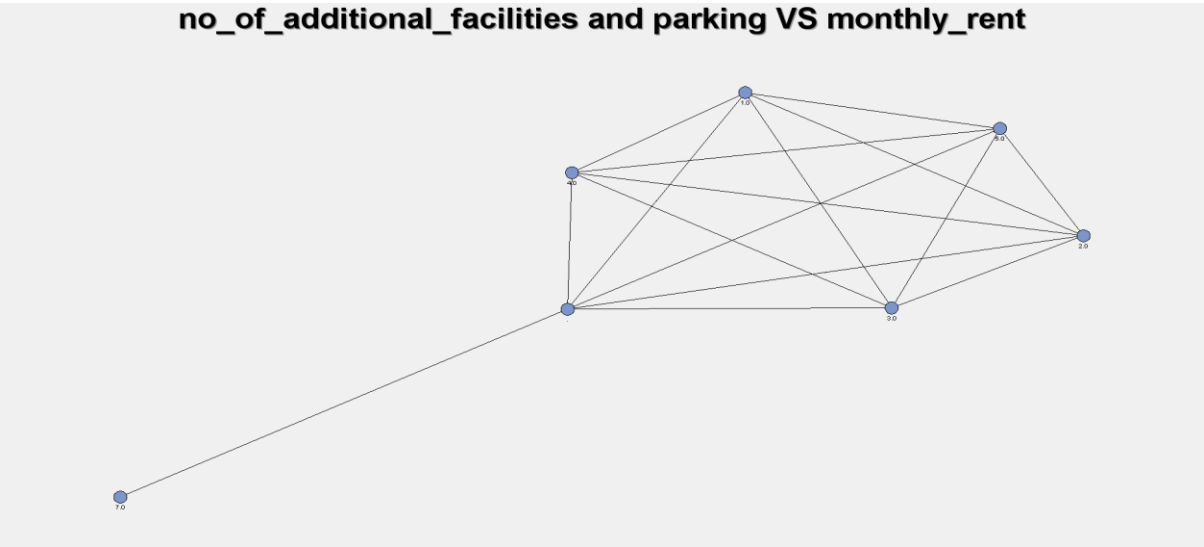


Figure 4.7.2 no of additional facilities and parking vs monthly rent

Figure 4.7.2 shows the link between the 4 add-ons and parking and monthly rent is clearer under normal circumstances. In the case of more than 4 add-ons, the relationship between monthly rent and add-ons and parking is not obvious.

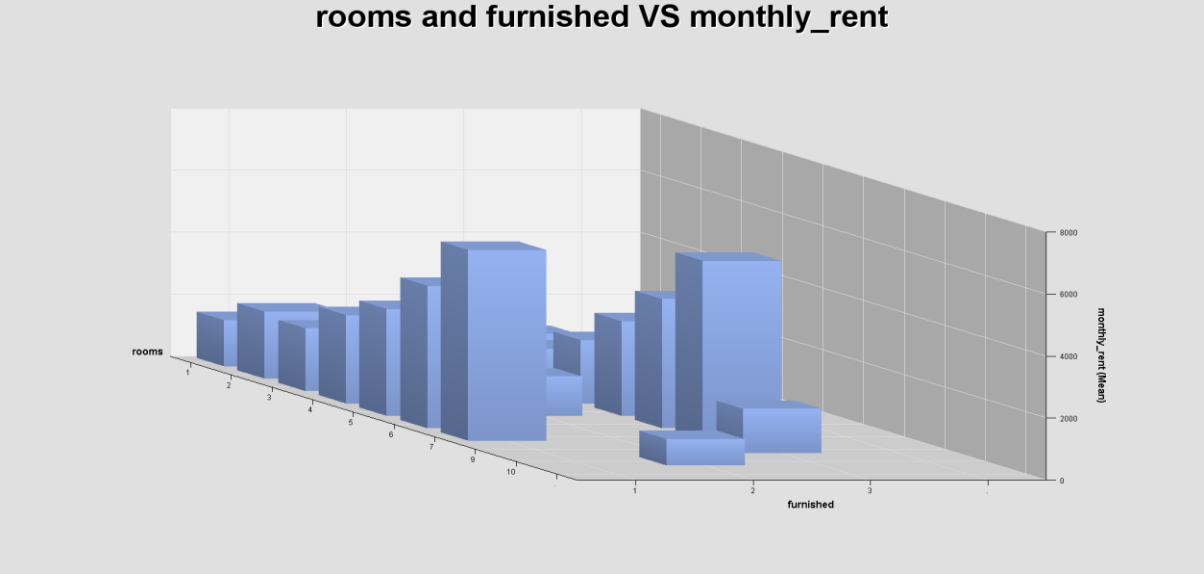


Figure 4.7.3 rooms and furnished vs monthly rent

Figure 4.7.3 shows the rental with 1 and 3 facilities are more popular, however not more rooms are better although the price of monthly rent increases with the number of rooms with some fluctuations in between. There is a decrease in monthly rent when there are more than 9 rooms.

4.8. Correlation analysis findings

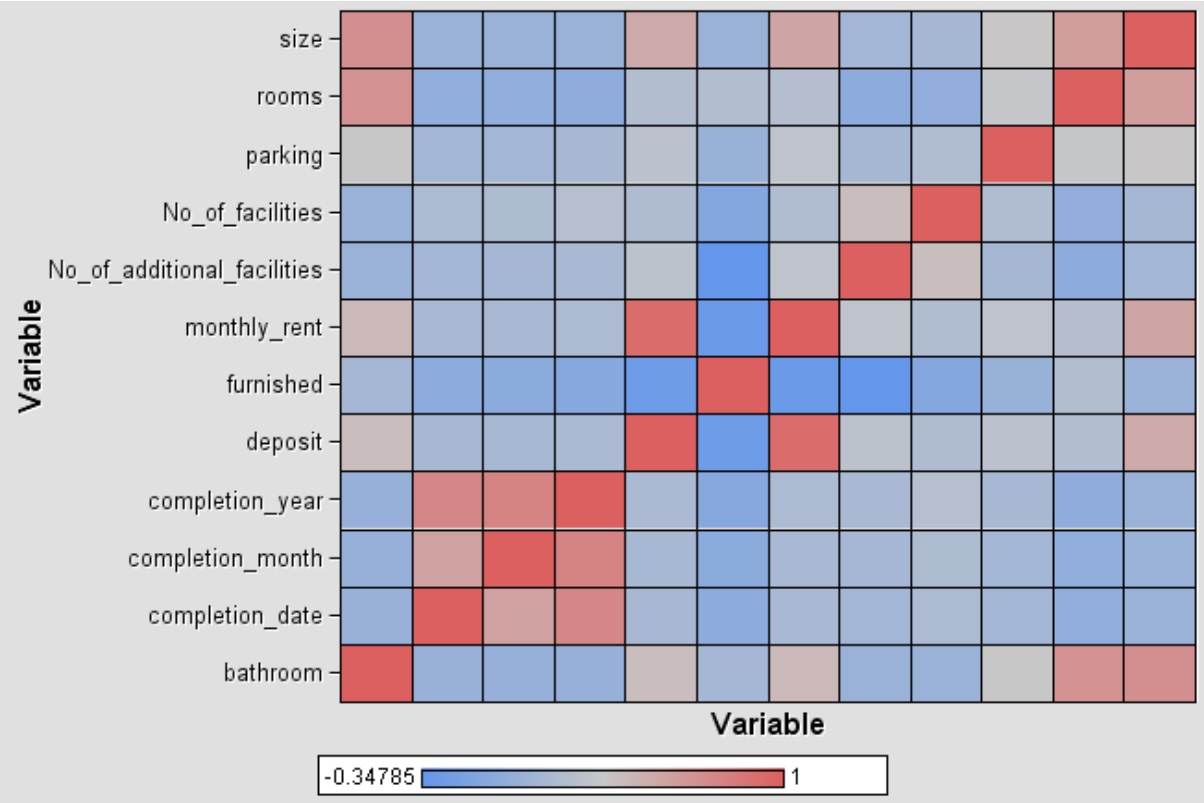


Figure 4.8.1 confusion matrix for correlation analysis

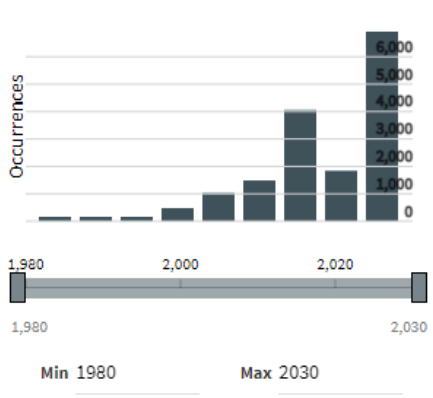
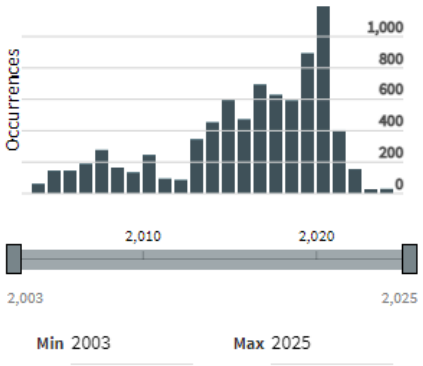
In addition to box plots and histograms, multivariate analysis matrices can be performed using correlation matrices. The strength of correlation is measured from -1 to 1, using different shades of blue and red. For blue and red, the darker the color, the higher the correlation. A pair of variables with a correlation value greater than 0.9 will be removed to avoid crosstalk problems. Avoid crosstalk problem. Figure 4.8.1 show the correlation matrix and correlation table.

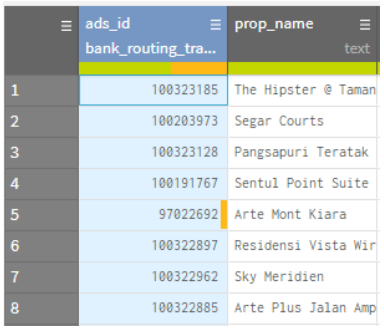
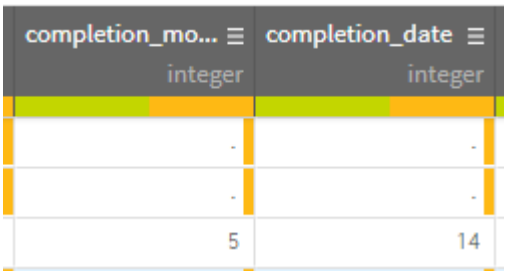
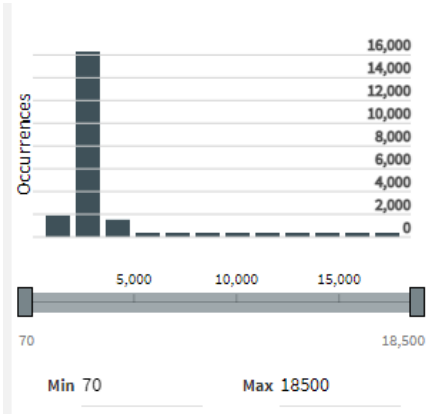
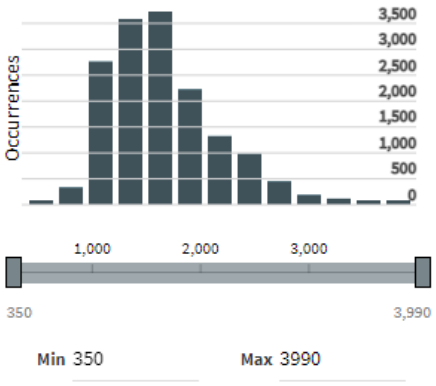
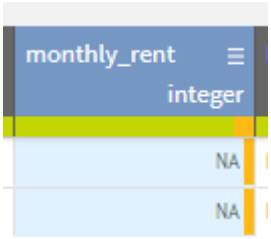
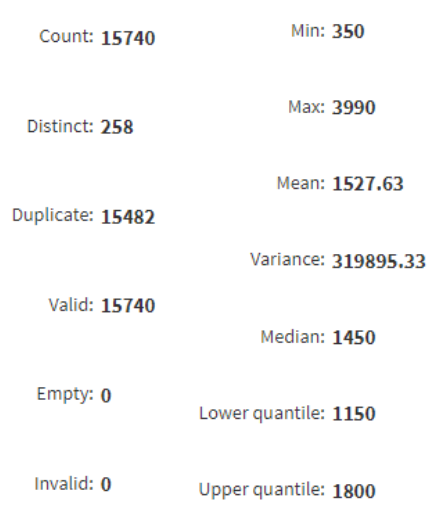
5. Modify

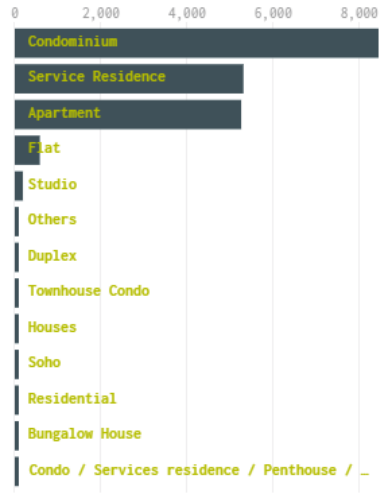
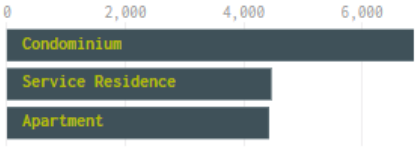
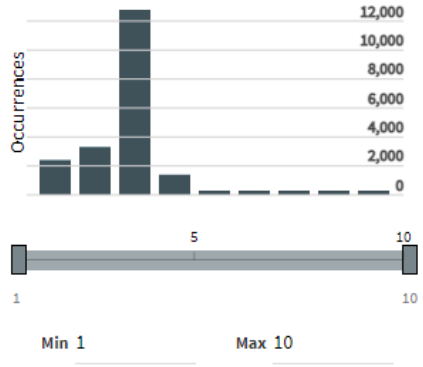
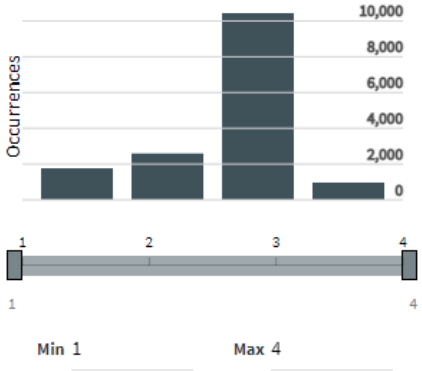
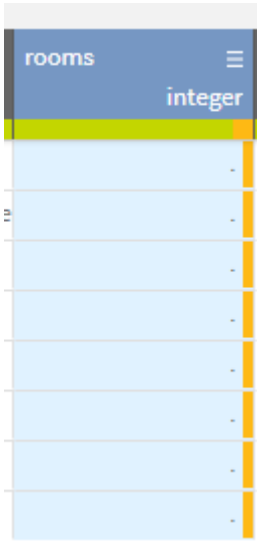
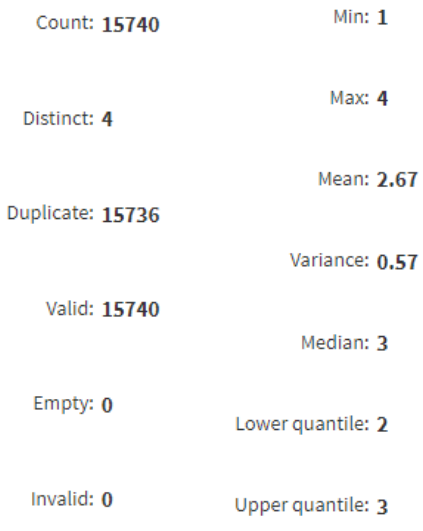
Upon conducting data exploration, it became evident that the dataset exhibits certain issues, including outliers, null values, and inconsistencies. To address these problems, this project will employ various data cleaning techniques such as replacement, imputation, variable transformation, and dropping of problematic data. The cleaning process will be performed using Talend and SAS Enterprise Miner tools.

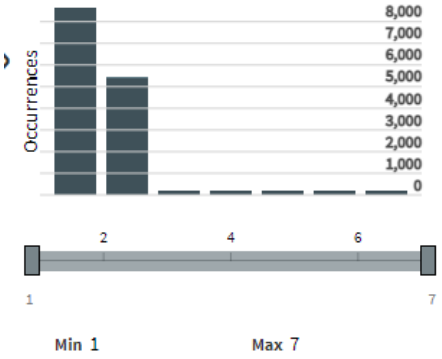
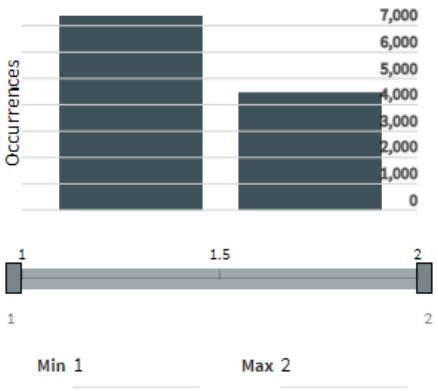
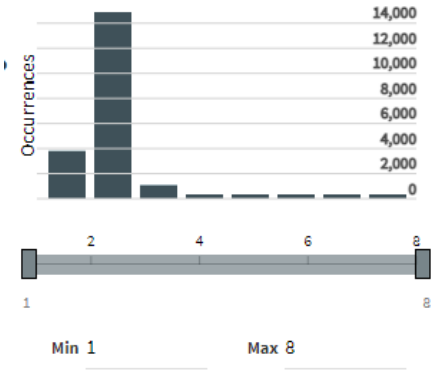
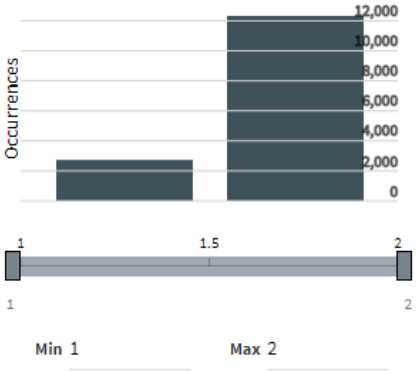
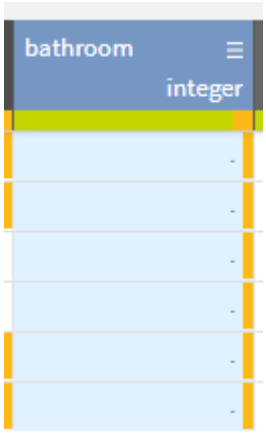
5.1. Outliers and inconsistencies processing

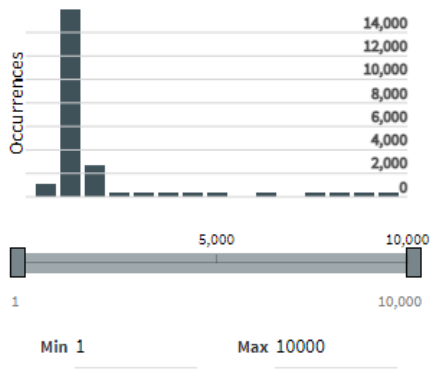
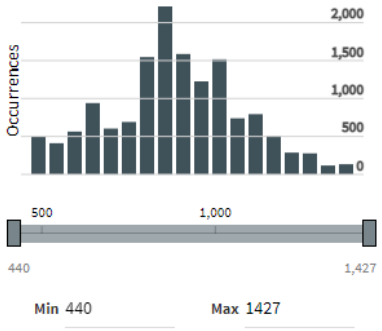
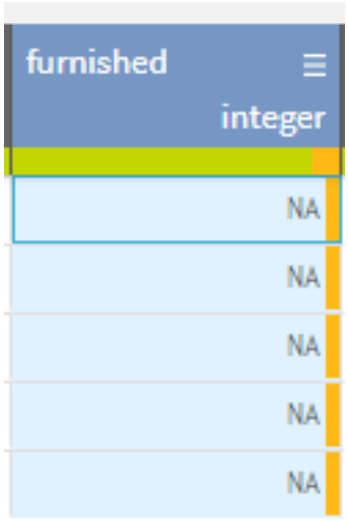
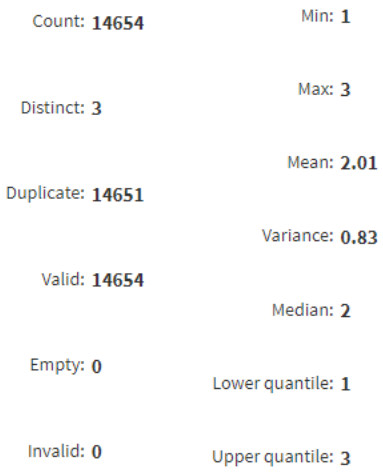
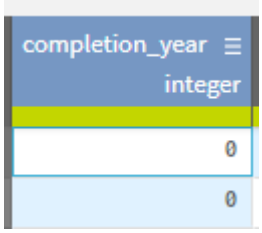
After data exploration, it can be understood that there are some problems in the data set, such as the inconsistency of the year, a large number of null values, and the outlier range of different variables. Table 5.1.1 shows the process of using Talend to solve some of these problems. However, some of these issues hampered the success rate of the treatment for couples of reasons. For example, when there are null values in the data, the interval filtering of variables cannot be successful. However, Talend itself cannot use complex algorithms for numerical filling to deal with null values. So, some parts of the problem will be further cleaned using SAS Enterprise Miner.

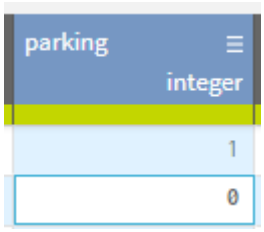
Before processing		After processing
	Change all the year value from 0 to 2030, and delete the rows in year range 1980 to 2003 which consider as outliers. And change 2030 to ".".	

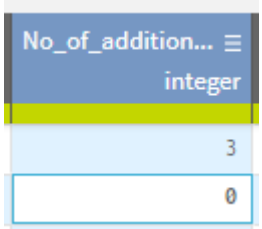
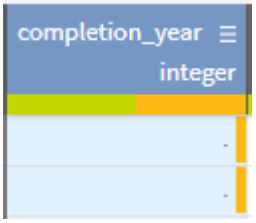
	Drop useless columns id and name.	/
	Drop useless columns month and date.	/
	Monthly rent range filter	
	Delete rows with NA value for monthly_rent	

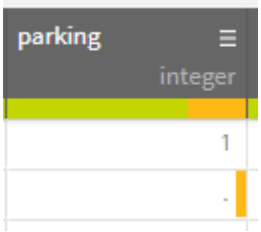
	Delete outliers for property_type . Only keep the three types.	
	Delete outliers for rooms. Filter range in 1 to 4.	
	Drop room NA value for all the rows	

	Delete outliers for parking	
	Delete outliers for bathroom	
	Drop rows with NA value for bathroom	

<table>
<tr>
<td> 
</td>
<td>Delete size range above 1427 and below 440.</td>
<td> 
</td>
</tr>
<tr>
<td> 
</td>
<td>Delete rows with NA value for furnished</td>
<td> 
</td>
</tr>
<tr>
<td> 




</td>
<td>Inconsistent processing. Change 0 to “” As NA value which can be identified by SAS Enterprise Miner.</td>
<td> 



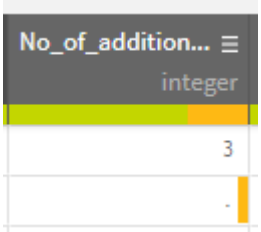
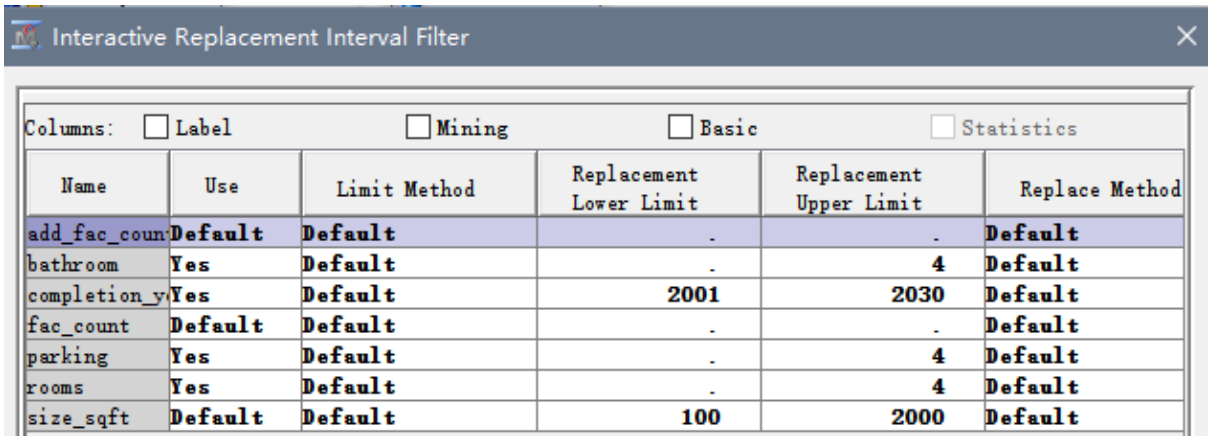

</td>
</tr>
</table>

Figure 5.1.1 outliers first processed by using talend

5.2. SAS further replacement



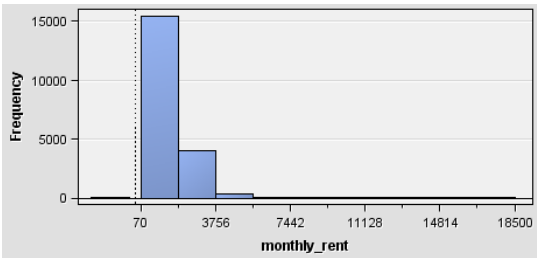
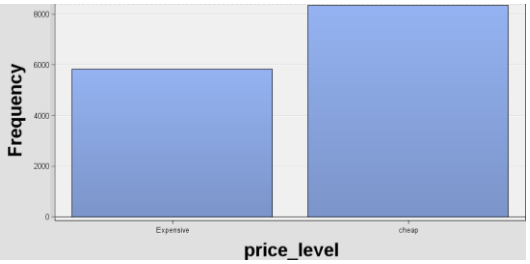
Name	Use	Limit Method	Replacement Lower Limit	Replacement Upper Limit	Replace Method
add_fac_coun	Default	Default	.	.	Default
bathroom	Yes	Default	.	4	Default
completion_y	Yes	Default	2001	2030	Default
fac_count	Default	Default	.	.	Default
parking	Yes	Default	.	4	Default
rooms	Yes	Default	.	4	Default
size_sqft	Default	Default	100	2000	Default

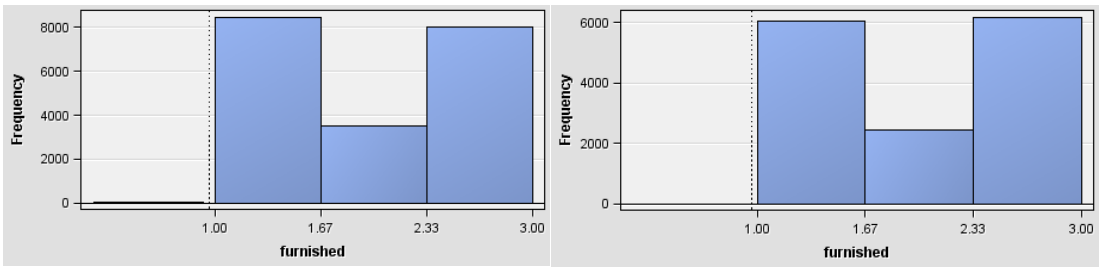
Figure 5.2.1 replacement

In terms of filtering data, one of the advantages of SAS Enterprise Minter is that when there are certain null values in the data, the system will not directly delete the null value data because of the set value range. Therefore, for parking and completion_year that are difficult to handle in Talend, replacement is used to re-filtering processing. In addition, some other data here are further filtered according to the difference explored to ensure that there are no outliers. The setting as shown in figure 5.2.1

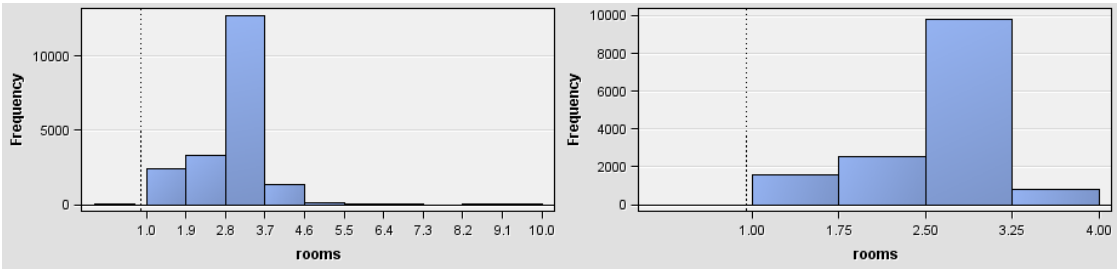
5.3. Histogram comparation before and after noise filtered

Table 5.3.1 Histogram comparison noise filtered

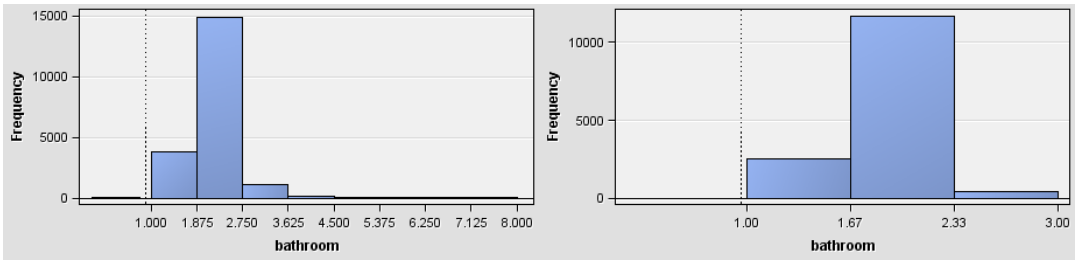
Histogram before Talend cleaning	Histogram after Talend cleaning
	
For monthly rent the system transform it to price level for build classification model purpose. The new target variable includes two type of variable which is expensive and cheap. The data distribution seems good.	



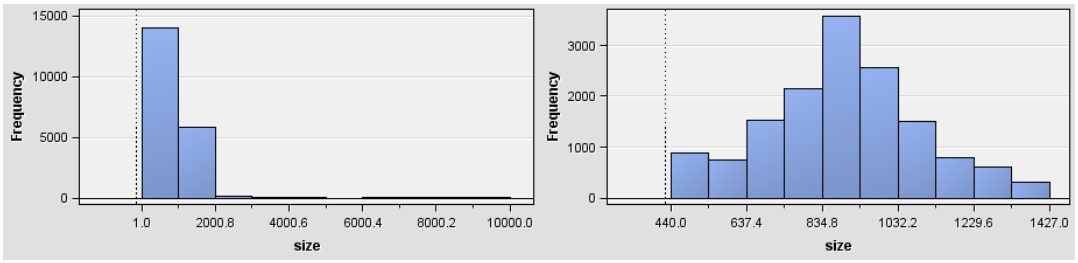
For furnished, the NA value is removed by using SAS EMiner.



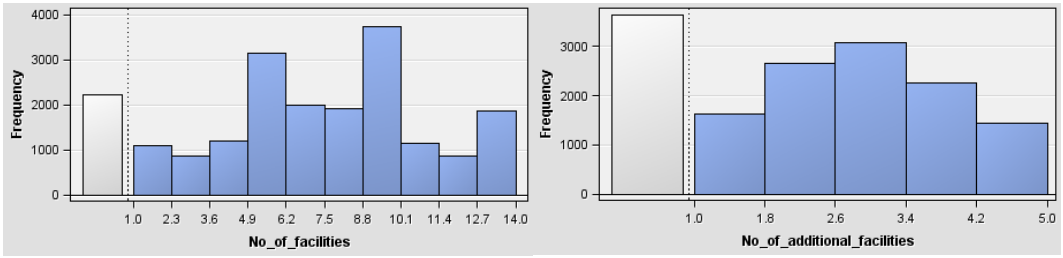
For rooms, the outliers exclude 1-4 is removed.



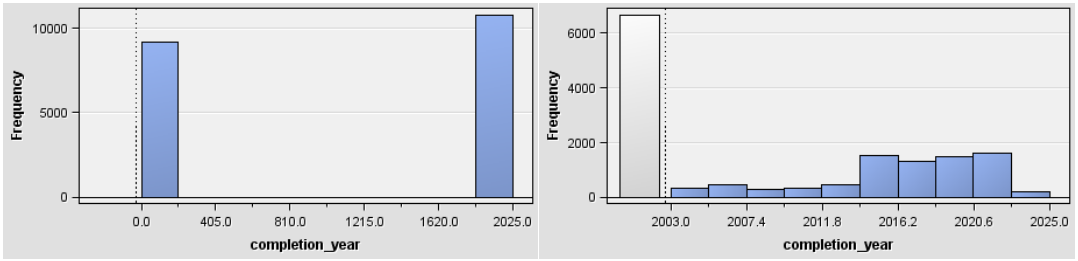
For bathroom, the data filtered in range 1 to 3.



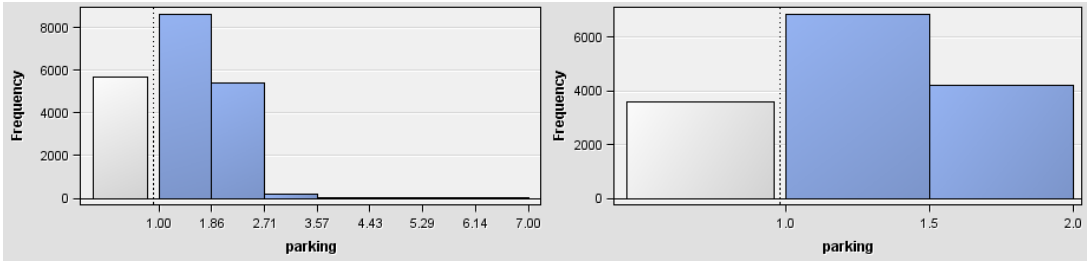
For noise, the data filtered in range 400 to 1500.



There is no change for no of facilities and no of additional facilities.



For completion year, the year is 0 marked as NA value and the year range only keep from 2001 to 2025.



For parking, the noise data bigger than 2 is removed.

According to the histogram in table 5.3.1, it can be viewed easily that there are still large numbers of NA value for 4 variables, which are additional_facilities, no_of_additional_facilities, completion_year and parking.

5.4. NA value imputation by SAS Enterprise Miner

Interval Variables	
Default Input Method	Median
Default Target Method	None

Figure 5.4.1 NA imputation

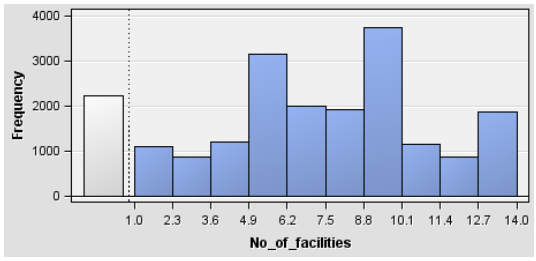
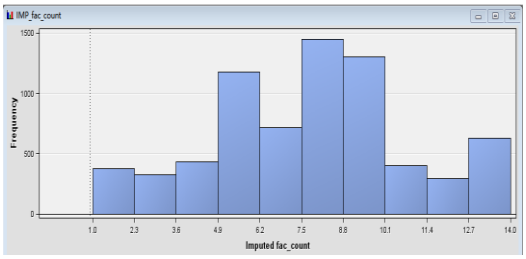
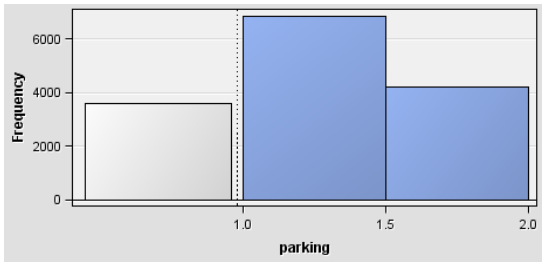
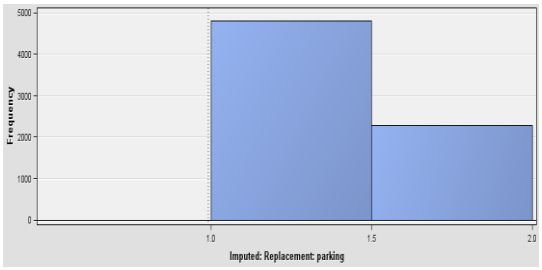
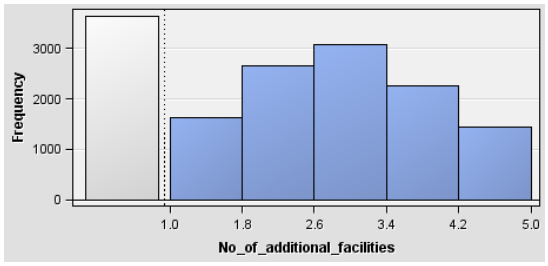
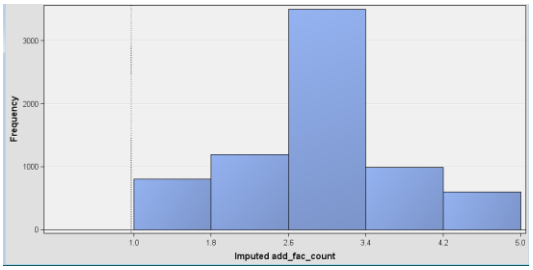
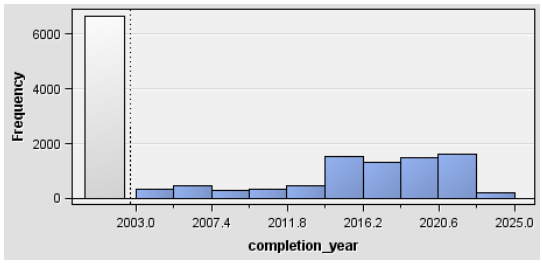
The system here using median to fill NA values as shown in figure 5.4.1. Since the data with NA value that the system needs to fill are number counted and year, the data filled will not be integer data type for these variables by use mean instead. So, mean is not suitable here.

Variable Name	Impute Method	Imputed Variable	Impute Value	Role	Measurement Level	Number of Missing for TRAIN
add_fac_count	MEDIAN	IMP_add_fac_count	3	INPUT	INTERVAL	2068
fac_count	MEDIAN	IMP_fac_count	8	INPUT	INTERVAL	753
parking	MEDIAN	IMP_parking	1	INPUT	INTERVAL	1212

Figure 5.4.2 results of imputation

As shown in figure 5.4.2, it the imputation results for NA value imputation. Here, it can be found that completion year is not imputed fill here because it include more than 50% NA values.

Table 5.4.1 histogram explore after imputation

	
	
	
	Since the Missing value is over than 50% NA values for completion_year,

As shown in table 5.4.1, it compares the results before and after the system impute the NA values.

5.5. Transform variables

Variables - Trans

(none) not Equal to

Columns: Label Mining Basic

Name	Method	Number of Bins	Role	Level
IMP_REP_park	Log 10	4	Input	Interval
IMP_add_fac_	Log 10	4	Input	Interval
IMP_fac_coun	Log 10	4	Input	Interval
REP_rooms	Log 10	4	Input	Interval
REP_bathroom	Log 10	4	Input	Interval
REP_size_sqf	Log 10	4	Input	Interval
property_typ	Dummy Indica	4	Input	Nominal
furnished	Dummy Indica	4	Input	Nominal
REP_completi	Default	4	Rejected	Interval
completion_y	Default	4	Rejected	Interval
bathroom	Default	4	Rejected	Interval
rooms	Default	4	Rejected	Interval
location	Default	4	Rejected	Nominal
parking	Default	4	Rejected	Interval
region	Default	4	Input	Nominal
REP_location	Default	4	Input	Nominal
size_sqft	Default	4	Rejected	Interval
price_level	Default	4	Target	Binary

Figure 5.5.1 transform setting

Figure 5.5.1 shows data transform processing setting. Most of processed interval variable transform to log 10 and for nominal variables here using dummy methods to transform it to binary data type.

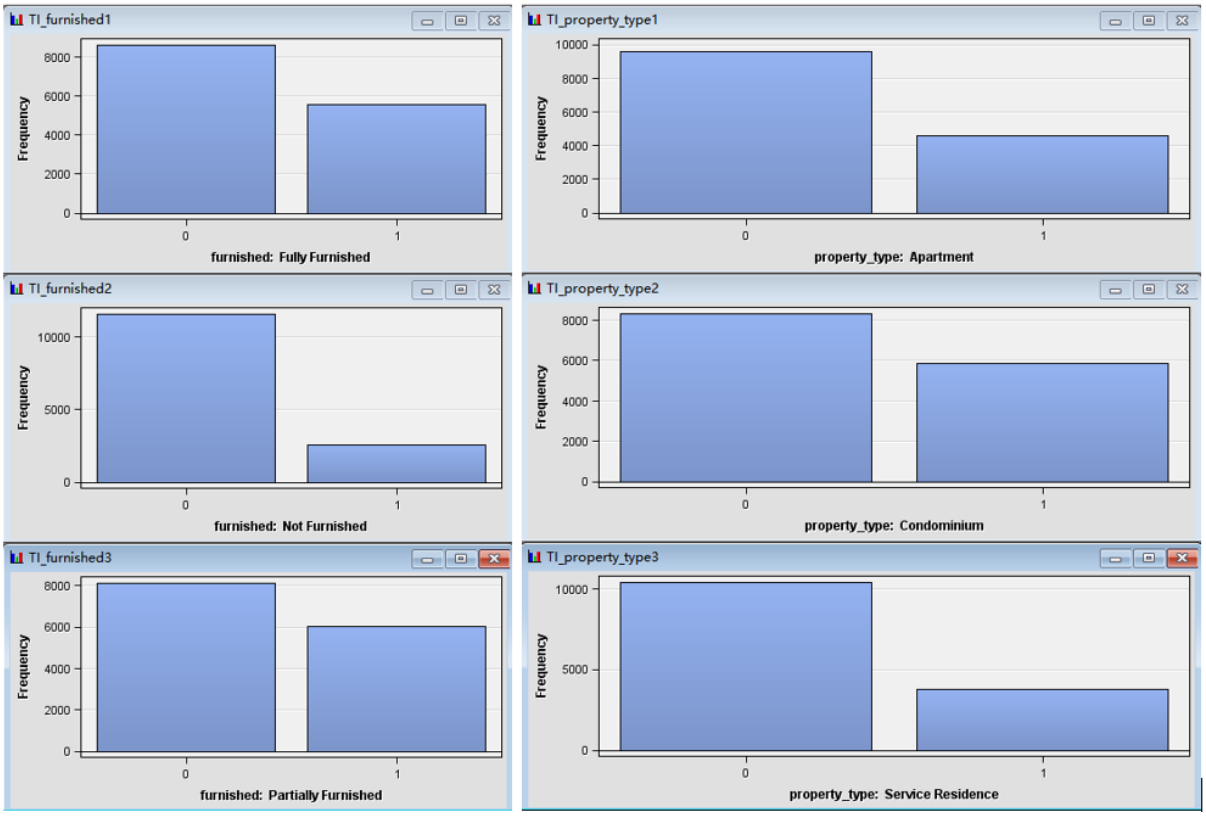


Figure 5.5.2 dummy processed for furnished and property_type

For nominal variable, the furnished and property_type transform to 3 variable which named by using the unique value of furnished. And the values is in binary which is 0 or 1.

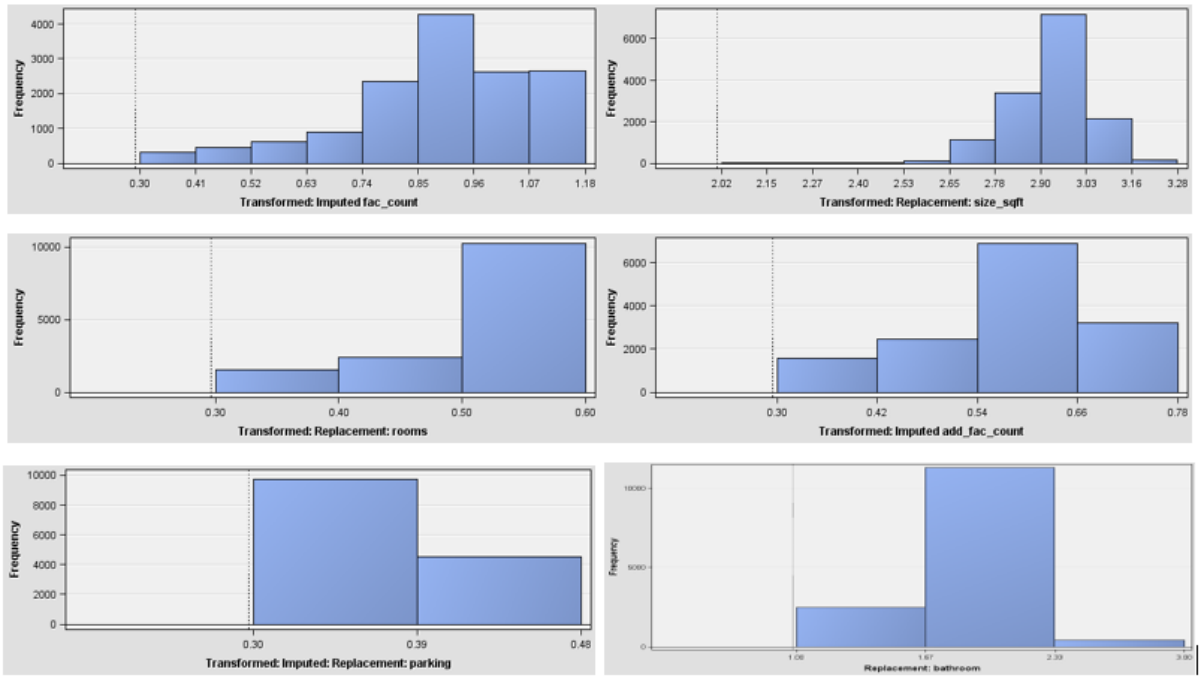


Figure 5.5.3 interval transform processed after log 10

Figure 5.5.3 shows the interval transformed after log 10. Log transformations used here for couple of reasons, it can make skewed distributions more symmetric, often mimicking a normal

distribution, which is beneficial since many statistical techniques assume normality. Log transformations also help reduce the scale of data, especially when it spans several orders of magnitude, making it more manageable for analysis. Moreover, they can stabilize variable variances, linearize relationships between variables, and convert multiplicative relationships into additive ones.

5.6. Train and valid set split

Data Set Allocations	
Training	50.0
Validation	50.0
Test	0.0

Figure 5.6.1 data allocation

As shown in figure 5.6.1, the data allocate into two set of data which are training set and validation set. Each one includes 50% of data.

Summary Statistics for Class Targets

Data=DATA

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
price_level	.	Expensive	5829	41.0869	
price_level	.	cheap	8358	58.9131	

Data=TRAIN

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
price_level	.	Expensive	2914	41.0886	
price_level	.	cheap	4178	58.9114	

Data=VALIDATE

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
price_level	.	Expensive	2915	41.0853	
price_level	.	cheap	4180	58.9147	

Figure 5.6.2 target variable distribution after allocation

Figure 5.6.2 shows the target variable distribution after data allocation. It can be found that the cheap include about 60% of data and expensive about 40% of data. And for test and validation, the data distribution almost same.

6. Modeling

This project aims to address two different research questions: predicting the monthly rent price of houses and predicting the rent price range. To achieve accurate predictions, the project will utilize two types of models: regression models and classification models. The regression model analysis will involve comparing and analyzing the results of linear regression, Gradient Boosting, and decision tree models to identify the most suitable model. Similarly, for the classification model, the project will compare and analyze the outcomes of logistic regression, Gradient Boosting, decision tree, and 16 neural networks with various parameter settings. By conducting predictive analyses for both objectives, this project aims to assist merchants in setting appropriate rent prices and provide customers with optimal solutions for their house purchases.

6.1. Decision Tree model

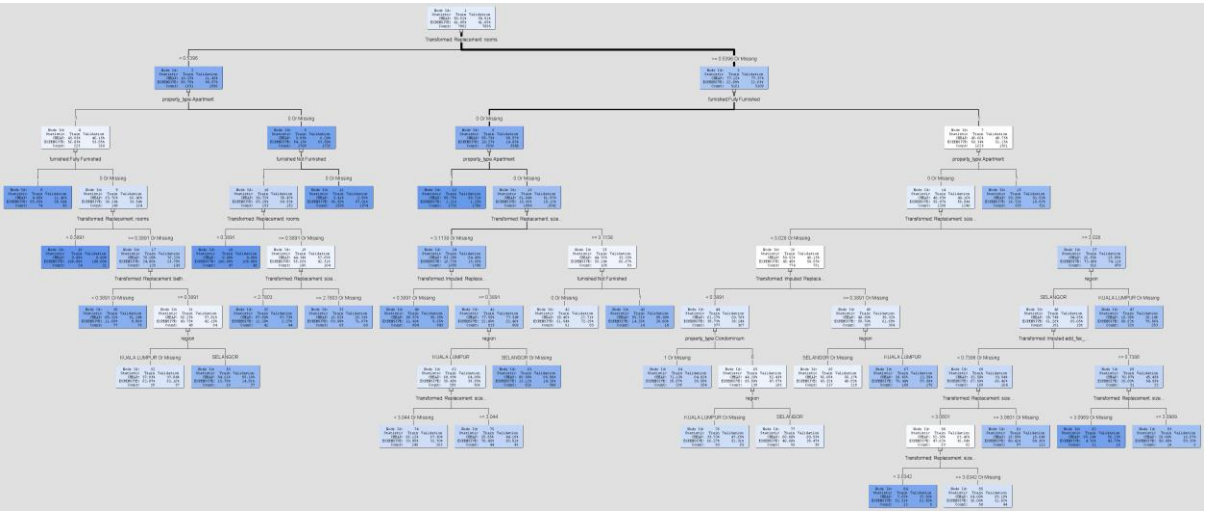


Figure 6.1.1 the tree for decision tree model

Figure 6.1.1 is the decision tree model the system built, as shown in the figure. According to the decision tree model, it can see the following information from it:

(1) Variable rooms has the highest information gain in the labels, so it is chosen as the root node and also appears several times in the leaf nodes, which means that it is able to classify the data effectively.

(2) The shortest split appears at the third level of the decision tree.

(3) In most cases, the higher the rank the higher the price. The decision rules are as follows:

● if property_type:Apartment IS ONE OF: 1, AND furnished:Fully Furnished IS ONE OF: 1, AND Transformed: Replacement: rooms < 0.53959, then Tree Node Identifier = 8, Number of Observations = 74, Predicted: price_level=cheap = 0.04, Predicted: price_level=Expensive = 0.96

●if property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Not Furnished IS ONE OF: 0 or MISSING. AND Transformed: Replacement: rooms < 0.53959, then Tree Node Identifier = 11, Number of Observations = 1555, Predicted: price_level=cheap = 0.03, Predicted: price_level=Expensive = 0.97

● if property_type:Apartment IS ONE OF: 1AND furnished:Fully Furnished IS ONE OF: 0 or MISSING, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, then Tree Node Identifier = 12, Number of Observations = 1732, Predicted: price_level=cheap = 0.99, Predicted: price_level=Expensive = 0.01

● if property_type:Apartment IS ONE OF: 1, AND furnished:Fully Furnished IS ONE OF: 1, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, then Tree Node Identifier = 15, Number of Observations = 335, Predicted: price_level=cheap = 0.83, Predicted: price_level=Expensive = 0.17

● if property_type:Apartment IS ONE OF: 1, AND furnished:Fully Furnished IS ONE OF: 0 or MISSING, AND Transformed: Replacement: rooms < 0.38908, then Tree Node Identifier = 16, Number of Observations = 24, Predicted: price_level=cheap = 0.00, Predicted: price_level=Expensive = 1.00

● if property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Not Furnished IS ONE OF: 1. AND Transformed: Replacement: rooms < 0.38908, then Tree Node Identifier = 18, Number of Observations = 47, Predicted: price_level=cheap = 0.00, Predicted: price_level=Expensive = 1.00

● if property_type:Apartment IS ONE OF: 1, AND furnished:Fully Furnished IS ONE OF: 0 or MISSING, AND Transformed: Replacement: rooms < 0.53959 AND Transformed: Replacement: rooms >= 0.38908 or MISSING, AND Transformed: Replacement: bathroom < 0.38908 or MISSING, then , Tree Node Identifier = 30, Number of Observations = 77, Predicted: price_level=cheap = 0.88, Predicted: price_level=Expensive = 0.12

● if property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Not Furnished IS ONE OF: 1, AND Transformed: Replacement: size_sqft < 2.7803, AND Transformed: Replacement: rooms < 0.53959 AND Transformed: Replacement: rooms >= 0.38908 or MISSING, then Tree Node Identifier = 32, Number of Observations = 41, Predicted: price_level=cheap = 0.88, Predicted: price_level=Expensive = 0.12

● if property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Not Furnished IS ONE OF: 1, AND Transformed: Replacement: size_sqft >= 2.78031 or MISSING, AND Transformed: Replacement: rooms < 0.53959 AND Transformed: Replacement: rooms >= 0.38908 or MISSING, then Tree Node Identifier = 33, Number of Observations = 65, Predicted: price_level=cheap = 0.17, Predicted: price_level=Expensive = 0.83

● if property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Fully Furnished IS ONE OF: 0 or MISSING, AND Transformed: Replacement: size_sqft < 3.11378 or MISSING, AND Transformed: Replacement:

rooms >= 0.53959 or MISSING, AND Transformed: Imputed: Replacement: parking < 0.38908 or MISSING, then Tree Node Identifier = 40, Number of Observations = 884, Predicted: price_level=cheap = 0.89, Predicted: price_level=Expensive = 0.11

● if property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Not Furnished IS ONE OF: 0 or MISSING, AND furnished:Fully Furnished IS ONE OF: 0 or MISSING, AND Transformed: Replacement: size_sqft >= 3.11378, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, then Tree Node Identifier = 42, Number of Observations = 91, Predicted: price_level=cheap = 0.38, Predicted: price_level=Expensive = 0.62

● if property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Not Furnished IS ONE OF: 1, AND furnished:Fully Furnished IS ONE OF: 0 or MISSING, AND Transformed: Replacement: size_sqft >= 3.11378, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, then Tree Node Identifier = 43, Number of Observations = 14, Predicted: price_level=cheap = 0.86, Predicted: price_level=Expensive = 0.14

●if region IS ONE OF: KUALA LUMPUR or MISSING, AND property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Fully Furnished IS ONE OF: 1, AND Transformed: Replacement: size_sqft >= 3.02796, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, then Tree Node Identifier = 47, Number of Observations = 325, Predicted: price_level=cheap = 0.19, Predicted: price_level=Expensive = 0.81

●if region IS ONE OF: KUALA LUMPUR or MISSING, AND property_type:Apartment IS ONE OF: 1, AND furnished:Fully Furnished IS ONE OF: 0 or MISSING, AND Transformed: Replacement: rooms < 0.53959 AND, Transformed: Replacement: rooms >= 0.38908 or MISSING, AND Transformed: Replacement: bathroom >= 0.38908, then Tree Node Identifier = 52, Number of Observations = 29, Predicted: price_level=cheap = 0.38, Predicted: price_level=Expensive = 0.62

● if region IS ONE OF: SELANGOR, AND property_type:Apartment IS ONE OF: 1

AND furnished:Fully Furnished IS ONE OF: 0 or MISSING, AND Transformed: Replacement: rooms < 0.53959 AND Transformed: Replacement: rooms >= 0.38908 or MISSING, AND Transformed: Replacement: bathroom >= 0.38908, then Tree Node Identifier = 53, Number of Observations = 19, Predicted: price_level=cheap = 0.84, Predicted: price_level=Expensive = 0.16

● if region IS ONE OF: SELANGOR or MISSING, AND property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Fully Furnished IS ONE OF: 0 or MISSING, AND Transformed: Replacement: size_sqft < 3.11378 or MISSING, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, AND Transformed: Imputed: Replacement: parking >= 0.38908, then Tree Node Identifier = 63, Number of Observations = 526, Predicted: price_level=cheap = 0.87, Predicted: price_level=Expensive = 0.13

●if property_type:Condominium IS ONE OF: 1 or MISSING, AND property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Fully Furnished IS ONE OF: 1, AND Transformed: Replacement: size_sqft < 3.02796 or MISSING, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, AND Transformed: Imputed:

Replacement: parking < 0.38908, then Tree Node Identifier = 64, Number of Observations = 239, Predicted: price_level=cheap = 0.71, Predicted: price_level=Expensive = 0.29

● if region IS ONE OF: SELANGOR or MISSING, AND property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Fully Furnished IS ONE OF: 1, AND Transformed: Replacement: size_sqft < 3.02796 or MISSING, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, AND Transformed: Imputed: Replacement: parking >= 0.38908 or MISSING, then Tree Node Identifier = 66, Number of Observations = 217, Predicted: price_level=cheap = 0.57, Predicted: price_level=Expensive = 0.43

● if region IS ONE OF: KUALA LUMPUR, AND property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Fully Furnished IS ONE OF: 1, AND Transformed: Replacement: size_sqft < 3.02796 or MISSING, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, AND Transformed: Imputed: Replacement: parking >= 0.38908 or MISSING, then Tree Node Identifier = 67, Number of Observations = 180, Predicted: price_level=cheap = 0.21, Predicted: price_level=Expensive = 0.79

● if region IS ONE OF: KUALA LUMPUR, AND property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Fully Furnished IS ONE OF: 0 or MISSING, AND Transformed: Replacement: size_sqft < 3.04395 or MISSING, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, AND Transformed: Imputed: Replacement: parking >= 0.38908, then Tree Node Identifier = 74, Number of Observations = 245, Predicted: price_level=cheap = 0.66, Predicted: price_level=Expensive = 0.34

● if region IS ONE OF: KUALA LUMPUR, AND property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Fully Furnished IS ONE OF: 0 or MISSING, AND Transformed: Replacement: size_sqft < 3.11378 AND Transformed: Replacement: size_sqft >= 3.04395, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, AND Transformed: Imputed: Replacement: parking >= 0.38908, then Tree Node Identifier = 75, Number of Observations = 44, Predicted: price_level=cheap = 0.30, Predicted: price_level=Expensive = 0.70

●if region IS ONE OF: KUALA LUMPUR or MISSING, AND property_type:Condominium IS ONE OF: 0, AND property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Fully Furnished IS ONE OF: 1, AND Transformed: Replacement: size_sqft < 3.02796 or MISSING, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, AND Transformed: Imputed: Replacement: parking < 0.38908, then Tree Node Identifier = 76, Number of Observations = 83, Predicted: price_level=cheap = 0.34, Predicted: price_level=Expensive = 0.66

● if region IS ONE OF: SELANGOR, AND property_type:Condominium IS ONE OF: 0, AND property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Fully Furnished IS ONE OF: 1, AND Transformed: Replacement: size_sqft < 3.02796 or MISSING, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, AND Transformed: Imputed: Replacement: parking < 0.38908, then Tree Node Identifier = 77, Number of Observations = 55, Predicted: price_level=cheap = 0.60, Predicted: price_level=Expensive = 0.40

- if region IS ONE OF: SELANGOR, AND property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Fully Furnished IS ONE OF: 1, AND Transformed: Replacement: size_sqft >= 3.06013 or MISSING, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, AND Transformed: Imputed add_fac_count < 0.73856 or MISSING, then Tree Node Identifier = 81, Number of Observations = 97, Predicted: price_level=cheap = 0.20, Predicted: price_level=Expensive = 0.80

- if region IS ONE OF: SELANGOR, AND property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Fully Furnished IS ONE OF: 1, AND Transformed: Replacement: size_sqft < 3.09691 AND Transformed: Replacement: size_sqft >= 3.02796 or MISSING, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, AND Transformed: Imputed add_fac_count >= 0.73856

  then Tree Node Identifier = 82, Number of Observations = 21, Predicted: price_level=cheap = 0.95, Predicted: price_level=Expensive = 0.05

- if region IS ONE OF: SELANGOR, AND property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Fully Furnished IS ONE OF: 1, AND Transformed: Replacement: size_sqft >= 3.09691, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, AND Transformed: Imputed add_fac_count >= 0.73856, then Tree Node Identifier = 83, Number of Observations = 10, Predicted: price_level=cheap = 0.20, Predicted: price_level=Expensive = 0.80

- if region IS ONE OF: SELANGOR, AND property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Fully Furnished IS ONE OF: 1, AND Transformed: Replacement: size_sqft < 3.03423 AND Transformed: Replacement: size_sqft >= 3.02796, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, AND Transformed: Imputed add_fac_count < 0.73856 or MISSING, then Tree Node Identifier = 84, Number of Observations = 13, Predicted: price_level=cheap = 0.08, Predicted: price_level=Expensive = 0.92

- if region IS ONE OF: SELANGOR, AND property_type:Apartment IS ONE OF: 0 or MISSING, AND furnished:Fully Furnished IS ONE OF: 1, AND Transformed: Replacement: size_sqft < 3.06013 AND Transformed: Replacement: size_sqft >= 3.03423 or MISSING, AND Transformed: Replacement: rooms >= 0.53959 or MISSING, AND Transformed: Imputed add_fac_count < 0.73856 or MISSING

  then Tree Node Identifier = 85, Number of Observations = 50, Predicted: price_level=cheap = 0.64, Predicted: price_level=Expensive = 0.36

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
price_level		_NOBS_	Sum of Frequencies	7092	7095	.
price_level		_MISC_	Misclassification Rate	0.116892	0.117125	.
price_level		_MAX_	Maximum Absolute Err...	0.987875	0.987875	.
price_level		_SSE_	Sum of Squared Errors	1267.12	1272.813	.
price_level		_ASE_	Average Squared Error	0.089334	0.089698	.
price_level		_RASE_	Root Average Squared...	0.298889	0.299496	.
price_level		_DIV_	Divisor for ASE	14184	14190	.
price_level		_DFT_	Total Degrees of Free...	7092	.	.

Figure 6.1.2 decision tree fit statistics

The statistical results obtained for the decision tree prediction model are shown in Figure 6.1.2. The results show that the decision tree model has a misclassification rate of 0.11712.

Variable Importance

Variable Name	Label	Number of Splitting Rules	Number of Surrogate Rules	Importance	Validation Importance	Ratio of Validation to Training Importance
LG10_REP_size_sqft	Transformed: Replacement: size_sqft	7	16	1.0000	1.0000	1.0000
LG10_REP_rooms	Transformed: Replacement: rooms	3	1	0.8812	0.8810	0.9999
LG10_REP_bathroom	Transformed: Replacement: bathroom	1	9	0.8572	0.8510	0.9928
TI_property_type3	property_type:Service Residence	0	4	0.7539	0.7460	0.9894
TI_furnished1	furnished:Fully Furnished	2	0	0.4811	0.4890	1.0163
LG10_IMP_add_fac_count	Transformed: Imputed add_fac_count	1	10	0.4627	0.4656	1.0063
TI_furnished3	furnished:Partially Furnished	0	3	0.4324	0.4393	1.0160
LG10_IMP_fac_count	Transformed: Imputed fac_count	0	15	0.3599	0.3727	1.0357
TI_property_type1	property_type:Apartment	3	0	0.3503	0.3539	1.0100
region		5	2	0.2268	0.2306	1.0169
TI_property_type2	property_type:Condominium	1	3	0.2246	0.1969	0.8768
LG10_IMP_REP_parking	Transformed: Imputed: Replacement: parking	2	2	0.1990	0.2048	1.0288
TI_furnished2	furnished:Not Furnished	2	0	0.1242	0.1537	1.2371

Figure 6.1.3 variable importance for decision tree model

The importance of the variables for the decision tree model is shown in figure 6.1.3. It was found that the three variables with the highest importance were size, rooms and bathroom. The importance was 1.000, 0.8812 and 0.8572 respectively.

6.2. Gradient Boosting

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
price_level		_NOBS_	Sum of Frequencies	7092	7095	.
price_level		_SUMW_	Sum of Case Weights ...	14184	14190	.
price_level		_MISC_	Misclassification Rate	0.13621	0.134884	.
price_level		_MAX_	Maximum Absolute Err...	0.987779	0.983339	.
price_level		_SSE_	Sum of Squared Errors	1378.015	1367.38	.
price_level		_ASE_	Average Squared Error	0.097153	0.096362	.
price_level		_RASE_	Root Average Squared...	0.311693	0.310423	.
price_level		_DIV_	Divisor for ASE	14184	14190	.
price_level		_DFT_	Total Degrees of Free...	7092	.	.

Figure 6.2.1 fit statistics results for Gradient Boosting Model

Figure 6.2.1 shows the fit statistics for gradient boosting model. The misclassification rate is 0.13621. The dataset is divided into training and validation sets, with 7092 and 7095 observations, respectively. The sum of case weights times frequencies is 14184 for the training set and 14190 for the validation set.

The misclassification rate, which is the ratio of the number of incorrect predictions to the total number of predictions, is 0.1362 for the training set and 0.1349 for the validation set. This indicates that the model has a similar performance on both the training and validation sets.

The maximum absolute error, which represents the largest difference between the predicted values and the true values, is 0.9878 for the training set and 0.9833 for the validation set.

The sum of squared errors (SSE) measures the overall difference between the predicted and true values, with the training set having an SSE of 1378.0155 and the validation set having an SSE of 1367.3796. The average squared error (ASE) is 0.0972 for the training set and 0.0964

for the validation set. The root average squared error (RASE), which is the square root of the ASE, is 0.3117 for the training set and 0.3104 for the validation set. These metrics provide a quantitative measure of the model's performance on both sets.

The divisor for ASE is 14184 for the training set and 14190 for the validation set. Finally, the total degrees of freedom for the training set is 7092.

Variable Importance

Variable Name	Label	Number of Splitting Rules	Number of Surrogate Rules	Importance	Validation Importance	Ratio of Validation to Training Importance
LG10_REP_rooms	Transformed: Replacement: rooms	30	12	1	1	1
LG10_REP_size_sqft	Transformed: Replacement: size_sqft	28	79	0.985172	0.984041	0.998852
LG10_REP_bathroom	Transformed: Replacement: bathroom	0	50	0.942574	0.944984	1.002556
TI_furnished1	furnished:Fully Furnished	23	4	0.568441	0.604039	1.062624
TI_furnished3	furnished:Partially Furnished	4	25	0.516996	0.549606	1.063076
LG10_IMP_add_fac_count	Transformed: Imputed add_fac_count	4	31	0.494578	0.522434	1.056321
TI_property_type1	property_type:Apartment	18	19	0.405467	0.394154	0.9721
TI_property_type2	property_type:Condominium	0	37	0.366245	0.357024	0.974823
LG10_IMP_fac_count	Transformed: Imputed fac_count	10	28	0.347232	0.334862	0.964376
LG10_IMP_REP_parking	Transformed: Imputed: Replacement: parking	13	22	0.210932	0.212653	1.008157
TI_property_type3	property_type:Service Residence	8	0	0.203255	0.17396	0.855871
region		23	3	0.195842	0.205011	1.046817
TI_furnished2	furnished:Not Furnished	7	6	0.131755	0.141784	1.076116

Figure 6.2.2 variable importance of Gradient Boosting Model

For variable correlation, include the following insights:

As shown in figure 6.2.2, the variable 'LG10_REP_rooms' (Transformed: Replacement: rooms) has the highest variable importance, both for the training and validation datasets, with a score of 1.0. It has been used in 30 splitting rules and 12 surrogate rules.

The variable 'LG10_REP_size_sqft' (Transformed: Replacement: size_sqft) is the second most important variable, with a slight decrease in importance in comparison to 'LG10_REP_rooms', having an importance of approximately 0.985 in both the training and validation datasets.

'LG10_REP_bathroom' (Transformed: Replacement: bathroom) doesn't contribute to any splitting rules, but is used in 50 surrogate rules. Despite this, it still retains high importance with scores close to 0.94.

For 'TI_furnished1' (furnished:Fully Furnished) and 'TI_furnished3' (furnished:Partially Furnished), although the importance score is relatively lower (around 0.57 and 0.52 respectively), it's notable that they have a higher ratio of validation to training importance, indicating that they might perform slightly better in the validation dataset.

The variables related to 'property_type' ('TI_property_type1', 'TI_property_type2', and 'TI_property_type3') demonstrate varying degrees of importance and show a slight decrease in the validation data compared to the training data.

The 'region' variable, despite having lower importance scores (around 0.196), shows a higher performance in the validation dataset compared to the training dataset, with a ratio of 1.047.

6.3. Logistic Regression

Logistic regression is a statistical model used for classification and predictive analytics, frequently employed to compute the probability of a particular event occurring. For instance, in the realm of real estate leasing, it may be used to predict the rental price for a specific property in Kuala Lumpur and Selangor. The mechanism of logistic regression involves establishing the log odds of an event as a linear combination of one or more independent variables, thereby computing the probability of the event occurring. This method is particularly well-suited to situations where the dependent variable is binary, i.e., ranges from 0 to 1.

At the core of logistic regression analysis is the estimation of the log odds of an event. This implies the seek to ascertain how the probability of an event occurring changes as the independent variables vary. It provides us with a means to comprehend the relationships between independent variables of all types (nominal, interval, ordinal, ratio) and binary dependent variables. For instance, we could utilize a logistic regression model to investigate the relationship between rental price (a binary variable, high or low) and independent variables such as geographic location, property type, and property size.

In this study, the project employed SAS Enterprise Miner to conduct model number regression and built our logistic regression model using a stepwise method. This model will assist us in revealing various factors influencing rental market trends in Kuala Lumpur and Selangor and predicting the rental price for a specific property. Through this approach, we can gain a deeper understanding of the various factors influencing rental prices. Moreover, it can provide valuable information for both tenants and landlords, aiding them in making more informed decisions in the rental market.

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
price_level		_AIC_	Akaike's Information C...	4085.54	.	.
price_level		_ASE_	Average Squared Error	0.089422	0.088459	.
price_level		_AVERR_	Average Error Function	0.286488	0.283087	.
price_level		_DFE_	Degrees of Freedom f...	7081	.	.
price_level		_DFM_	Model Degrees of Fre...	11	.	.
price_level		_DFT_	Total Degrees of Free...	7092	.	.
price_level		_DIV_	Divisor for ASE	14184	14190	.
price_level		_ERR_	Error Function	4063.54	4016.998	.
price_level		_FPE_	Final Prediction Error	0.0897	.	.
price_level		_MAX_	Maximum Absolute Err...	0.999793	0.999858	.
price_level		_MSE_	Mean Square Error	0.089561	0.088459	.
price_level		_NOBS_	Sum of Frequencies	7092	7095	.
price_level		_NW_	Number of Estimate ...	11	.	.
price_level		_RASE_	Root Average Sum of ...	0.299035	0.29742	.
price_level		_RFPE_	Root Final Prediction ...	0.299499	.	.
price_level		_RMSE_	Root Mean Squared E...	0.299267	0.29742	.
price_level		_SBC_	Schwarz's Bayesian C...	4161.074	.	.
price_level		_SSE_	Sum of Squared Errors	1268.357	1255.228	.
price_level		_SUMW_	Sum of Case Weights ...	14184	14190	.
price_level		_MISC_	Misclassification Rate	0.126904	0.128259	.

Figure 6.3.1 fit statistic for logistic regression

The project used the regression node to link with the data partition node, and selected logistic regression as the regression type. The figure demonstrates a misclassification rate of 0.126904, the accuracy of the model is 87.31%. For the validation dataset, the misclassification rate is 0.128259, resulting in an accuracy of 87.1741%. The results of the statistical test will be shown in the figure 6.3.1.

Odds Ratio Estimates

Effect		Point Estimate
LG10_IMP_REP_parking		0.007
LG10_IMP_fac_count		0.504
LG10_REP_bathroom		0.003
LG10_REP_rooms		999.000
LG10_REP_size_sqft		<0.001
TI_furnished1	0 vs 1	21.164
TI_furnished3	0 vs 1	3.421
TI_property_type1	0 vs 1	0.109
TI_property_type2	0 vs 1	0.438
region	Kuala Lumpur vs Selangor	0.308

Figure 6.3.2 fit Statistics for Logistic Regression Model

The concept of Odds Ratio Estimates is used as a measure of the strength of association between an event and its influencing factors or exposures. Essentially, it quantifies how strongly the presence or absence of a property of interest affects the occurrence of the event

under study. In the context of our research, the event we're investigating is the range of rental prices for properties in Kuala Lumpur and Selangor.

The exposure, or the influential factor, in this scenario, could be a variety of variables related to the properties, such as the number of rooms, size of the property, location, and the type of property, among others. The odds ratio essentially provides a measure of how each of these individual factors can influence, or are associated with, the rental price range of properties.

By analyzing the Odds Ratio Estimates, we can gain a clearer understanding of the dynamics that drive rental prices in these regions. It gives us insights into which variables have a stronger association with higher rental prices and which ones may not contribute as significantly. Such information is valuable for property owners, real estate developers, and renters alike, as it aids in making informed decisions in the rental market.

Based on the output of the Odds Ratio Estimates as shown in figure 6.3.2, it can be found that:

- The representative number of parking spots (LG10_IMP_REP_parking) has 0.007 times the odds of having a higher rental price range level than a lower rental price range level.
- The number of facilities (LG10_IMP_fac_count) has 0.504 times the odds of having a higher rental price range level than a lower rental price range level.
- The representative number of bathrooms (LG10_REP_bathroom) has 0.003 times the odds of having a higher rental price range level than a lower rental price range level.
- The representative number of rooms (LG10_REP_rooms) has 999 times the odds of having a higher rental price range level than a lower rental price range level.
- The representative property size (LGl0_REP_size_sqft) has less than 0.001 times the odds of having a higher rental price range level than a lower rental price range level.
- A change in furnishing status (TI_furnished1) from 0 to 1 results in 21.164 times the odds of having a higher rental price range level than a lower rental price range level.
- A second change in furnishing status (TI_furnished1) from 0 to 1 results in 3.421 times the odds of having a higher rental price range level than a lower rental price range level.
- A change in property type (TI_property_type1) from 0 to 1 results in 0.109 times the odds of having a higher rental price range level than a lower rental price range level.
- A change in property type (TI_property_type2) from 0 to 1 results in 0.438 times the odds of having a higher rental price range level than a lower rental price range level.

- A change in region from Kuala Lumpur to Selangor results in 0.308 times the odds of having a higher rental price range level than a lower rental price range level.

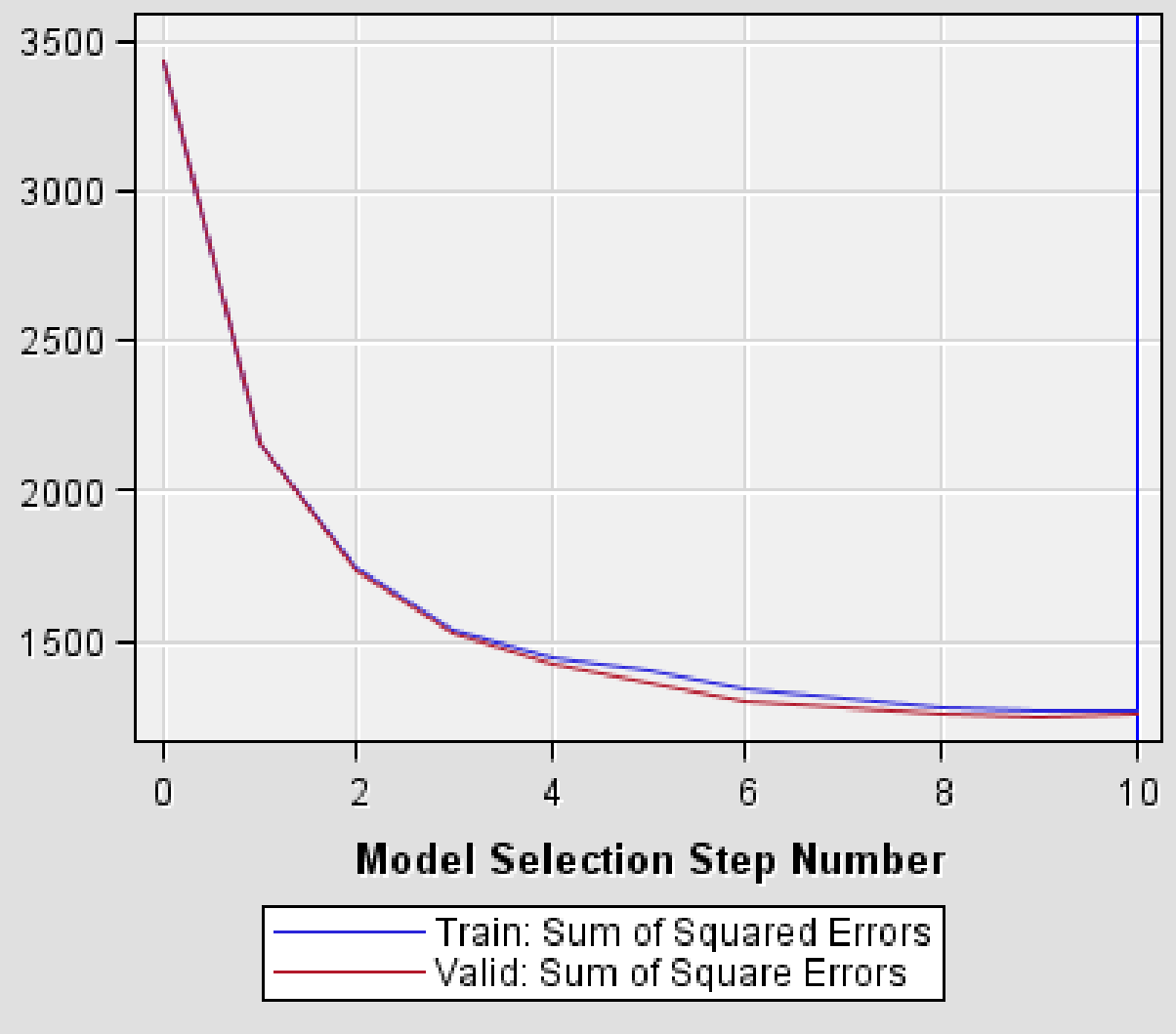


Figure 6.3.3 changes in training and validation error with model iteration steps

Figure 6.3.3 represents the change in the sum of squared errors (SSE) for both the training set and the validation set over the number of steps in the model selection process. The SSE for the training set is represented by a blue line, while that of the validation set is represented by a red line.

The near overlap of the two curves indicates that the training error and the validation error are very close, suggesting that the model neither overfits nor underfits. The highest point on the vertical axis is at 3400, indicating that the model had higher errors at the beginning of the steps.

As the number of steps increases, both curves gradually decrease, showing that the error is gradually decreasing as the model iterates. This is the model has very small possible error on both the training and validation data.

6.4. Auto Neural

The system will use misclassification as the score to evaluate the performance of the models. Different Parameter adjustment will affect the performance of the Neural network models. So, the report will compare the score for different parameters setting by using misclassification results. The misclassification as lower as better.

- Different Termination with default settings.

Overfitting

price_level		_MISC_	Misclassification Rate	0.164129	0.164341

Figure 6.4.1 termination set with overfitting

Time limit (winner)

price_level		_MISC_	Misclassification Rate	0.115482	0.117407

Figure 6.4.2 termination set with time limit

Training error returns error.

- Different Architecture with time limit, others default settings.

Single layer (winner)

price_level		_MISC_	Misclassification Rate	0.115482	0.117407

Figure 6.4.3 Architecture set with single layer

Block layers

price_level		_MISC_	Misclassification Rate	0.536943	0.545736

Figure 6.4.4 Architecture set with block layers

Funnel Layers

price_level		_MISC_	Misclassification Rate	0.237733	0.248767

Figure 6.4.5 Architecture set with funnel layers

Cascade

price_level		_MISC_	Misclassification Rate	0.427242	0.430162

Figure 6.4.6 Architecture set with cascade

- Different iteration and hidden units with time limit, single layer, others default settings.

Iteration 8, 2 hidden units (winner)

Figure 6.4.7 Iteration 8, 2 hidden units

Iteration 20, 2 hidden units

Figure 6.4.8 Iteration 20, 2 hidden units

Iteration 20, 4 hidden units

Figure 6.4.9 Iteration 20, 4 hidden units

Iteration 20, 8 hidden units

Figure 6.4.10 Iteration 20, 8 hidden units

- Different tolerance, 8 iteration, 2hidden layers, single layer, others default settings.

Tolerance low

Figure 6.4.11 tolerance set with low

Tolerance high (winner)

Figure 6.4.12 tolerance set with high

Tolerance median (winner)

Figure 6.4.13 tolerance set with median

- Different train action, tolerance median, 8 iteration, 2hidden layers, single layer, others default settings.

Search (winner)

Figure 6.4.14 train action set with search

Train

price_level		_MISC_	Misclassification Rate	0.331218	0.328118

Figure 6.4.15 train action set with train

Increment

price_level		_MISC_	Misclassification Rate	0.391286	0.390275

Figure 6.4.16 train action set with increment

- Different standardization, train action search, tolerance median, 8 iteration, 2hidden layers, single layer, others default settings.

Yes

price_level		_MISC_	Misclassification Rate	0.115482	0.117407

Figure 6.4.17 standardization - yes

No

price_level		_MISC_	Misclassification Rate	0.115482	0.117407

Figure 6.4.18 standardization - no

- Different total number of hidden units, standardization, train action search, tolerance median, 8 iteration, 2hidden layers, single layer, others default settings.

Total number of hidden Units = 50, final iteration = 5

price_level		_MISC_	Misclassification Rate	0.115482	0.117407

Figure 6.4.19 Total number of hidden Units = 50, final iteration = 5

Total number of hidden Units = 30, final iteration = 10

price_level		_MISC_	Misclassification Rate	0.115059	0.117125

Figure 6.4.20Total number of hidden Units = 30, final iteration = 10

Total number of hidden Units = 50, final iteration = 20

price_level		_MISC_	Misclassification Rate	0.115059	0.117125

Figure 6.4.21Total number of hidden Units = 50, final iteration = 20

Total number of hidden Units = 30, final iteration = 5 (winner) – model 17

price_level		_MISC_	Misclassification Rate	0.113931	0.114306

Figure 6.4.22Total number of hidden Units = 30, final iteration = 5 (winner)

Final selection is the last model, which parameters set as shown in figure 6.4.23:

General	
Node ID	AutoNeural17
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Model Options	
Architecture	Single Layer
Termination	Time Limit
Train Action	Search
Target Layer Error Function	Default
Maximum Iterations	8
Number of Hidden Units	2
Tolerance	Medium
Total Time	One Hour
Increment and Search Options	
Adjust Iterations	Yes
Freeze Connections	No
Total Number of Hidden Units	30
Final Training	Yes
Final Iterations	5
Activation Functions	
Direct	Yes
Exponential	No
Identity	No
Logistic	No
Normal	Yes
Reciprocal	No
Sine	Yes
Softmax	No
Square	No
Tanh	Yes
Score	
Hidden Units	No
Residuals	Yes
Standardization	Yes

Figure 6.4.23 Best network parameters setting

Data Role	Target Variable	Fit Statistics	Statistics Label ▼	AutoNeural15
Valid	price_level	_VSSE_	Valid: Sum of Squared Errors	1224.506
Valid	price_level	_VNOBS_	Valid: Sum of Frequencies	7095
Valid	price_level	_VSUMW_	Valid: Sum of Case Weights Times Freq	14190
Valid	price_level	_VRMSE_	Valid: Root Mean Squared Error	0.293758
Valid	price_level	_VRASE_	Valid: Root Average Squared Error	0.293758
Valid	price_level	_VAUR_	Valid: Roc Index	0.949
Valid	price_level	_VRESP_	Valid: Percent Response	99.43662
Valid	price_level	_VCAP_	Valid: Percent Captured Response	8.444976
Valid	price_level	_VWRONG_	Valid: Number of Wrong Classifications	831
Valid	price_level	_VMISC_	Valid: Misclassification Rate	0.117125
Valid	price_level	_VMSE_	Valid: Mean Squared Error	0.086294
Valid	price_level	_VMAX_	Valid: Maximum Absolute Error	0.999063
Valid	price_level	_VLIFT_	Valid: Lift	1.687806
Valid	price_level	VKS	Valid: Kolmogorov-Smirnov Statistic	0.752
Valid	price_level	_VKS_PRO...	Valid: Kolmogorov-Smirnov Probability Cutoff	0.53
Valid	price_level	_VGINI_	Valid: Gini Coefficient	0.899
Valid	price_level	_VGAIN_	Valid: Gain	69.25871
Valid	price_level	_VERR_	Valid: Error Function	3942.252
Valid	price_level	_VDIV_	Valid: Divisor for VASE	14190
Valid	price_level	_VRESPC_	Valid: Cumulative Percent Response	99.71831
Valid	price_level	_VCAPC_	Valid: Cumulative Percent Captured Response	16.9378
Valid	price_level	_VLIFTC_	Valid: Cumulative Lift	1.692587
Valid	price_level	_VKS_BIN_	Valid: Bin-Based Two-Way Kolmogorov-Smirn...	0.751
Valid	price_level	_VBINNED_...	Valid: Bin-Based Two-Way Kolmogorov-Smirn...	0.604
Valid	price_level	_VASE_	Valid: Average Squared Error	0.086294
Valid	price_level	_VAVERR_	Valid: Average Error Function	0.277819

Figure 6.4.24 results evaluation for the best network parameters

According to figure 6.4.24, it can be found the following information:

'Total Degrees of Freedom (DFT)': The total degrees of freedom for the training dataset is 7092. This indicates the total number of values in the data that are free to vary.

'Degrees of Freedom for Error (DFE)': The DFE for the training data is 7003. This reflects the number of data points that are free to vary after the model has been fit.

'Model Degrees of Freedom (DFM)': The DFM is 89.0, which suggests the number of independent ways by which your model can vary.

'Number of Estimated Weights (NW)': The NW is 89.0, which corresponds to the number of parameters estimated by the model.

'Akaike's Information Criterion (AIC)': The AIC for the training dataset is 3950.87. AIC balances model fit and complexity, with lower values generally indicating better models.

'Schwarz's Bayesian Criterion (SBC)': The SBC is 4562.01 for the training dataset. Like the AIC, the SBC balances model fit and complexity, with lower values being preferable.

'Average Squared Error (ASE)': The ASE is 0.083 for the training dataset and slightly higher for the validation dataset at 0.085. These values suggest a low average of squared differences between the actual and predicted values.

'Maximum Absolute Error (MAX)': The maximum absolute error is approximately 1.0 for both the training and validation datasets, indicating the largest individual prediction error.

'Divisor for ASE (DIV)': The divisor for ASE is 14184 for both the training and validation datasets.

'Sum of Frequencies (NOBS)': The total number of observations is 7092 for the training set and 7095 for the validation set.

'Root Average Squared Error (RASE)': The RASE is around 0.288 for the training and 0.291 for the validation datasets, suggesting an overall good fit of the model.

'Sum of Squared Errors (SSE)': The SSE is 1176.58 for the training set and 1202.85 for the validation set, indicating the total deviation of the response values from the fit of the model.

'Sum of Case Weights Times Freq (SUMW)': The SUMW is 14184 for both the training and validation datasets.

'Final Prediction Error (FPE)': The FPE for the training dataset is approximately 0.085, indicating a relatively low prediction error for the model.

'Mean Squared Error (MSE)': The MSE is 0.084 for the training dataset and 0.085 for the validation dataset, suggesting a small average squared difference between actual and predicted values.

'Root Final Prediction Error (RFPE)': The RFPE is approximately 0.292 for the training dataset, providing another measure of the prediction error of the model.

'Root Mean Squared Error (RMSE)': The RMSE is approximately 0.290 for the training set and 0.291 for the validation set, representing the standard deviation of the residuals.

'Average Error Function (AVERR)': The average error function is about 0.266 for the training set and 0.275 for the validation set, indicating a small average deviation from the actual values in both datasets.

'Error Function (ERR)': The error function is 3772.87 for the training dataset and slightly higher for the validation dataset at 3904.64, suggesting a generally good fit of the model.

'Misclassification Rate (MISC)': The misclassification rate is approximately 0.114 for both the training and validation datasets, indicating the proportion of instances where the predicted category does not match the actual category.

'Number of Wrong Classifications (WRONG)': The number of wrong classifications is 808 for the training set and 811 for the validation set, indicating the total number of instances where the model's predicted category did not match the actual category.

6.5. Assess

Fit Statistics

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Train: Misclassification Rate
Y	Tree3	Tree3	Decision Tree	price_level	0.116892
	Reg2	Reg2	Regression	price_level	0.126904
	Boost	Boost	Gradient Boosting	price_level	0.13621

Figure 6.5.1 Gradient Boosting VS Logistic Regression VS Decision Tree

In this subsection, the performance of the decision tree, regression and gradient boosting models is first compared with each other. As shown in the figure below (figure 6.5.1), the results show that the misclassification rates of the three models are not very different, with the decision tree model having the lowest misclassification rate of 0.117125, followed by the regression model with a misclassification rate of 0.128259, and finally the gradient boosting model with a misclassification rate of 0.134884.

Fit Statistics

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Selection Criterion: Valid: Misclassification Rate
Y	AutoNeural15	AutoNeural15	AutoNeural units 30-10	price_level	0.117125
	AutoNeural16	AutoNeural16	AutoNeural units 50-20	price_level	0.117125
	AutoNeural10	AutoNeural10	AutoNeural high	price_level	0.117407
	AutoNeural13	AutoNeural13	AutoNeural standardization	price_level	0.117407
	AutoNeural14	AutoNeural14	AutoNeural units 50	price_level	0.117407
	AutoNeural2	AutoNeural2	AutoNeural Time Limit	price_level	0.117407
	AutoNeural6	AutoNeural6	AutoNeural 20-2	price_level	0.127414
	AutoNeural8	AutoNeural8	AutoNeural 20-8	price_level	0.130796
	AutoNeural9	AutoNeural9	AutoNeural low	price_level	0.135025
	AutoNeural7	AutoNeural7	AutoNeural 20-4	price_level	0.138548
	AutoNeural	AutoNeural	AutoNeural Overfitting	price_level	0.164341
	AutoNeural4	AutoNeural4	AutoNeural Funnel Layers	price_level	0.248767
	AutoNeural11	AutoNeural11	AutoNeural train	price_level	0.328118
	AutoNeural12	AutoNeural12	AutoNeural increment	price_level	0.390275
	AutoNeural5	AutoNeural5	AutoNeural Cascade	price_level	0.430162
	AutoNeural3	AutoNeural3	AutoNeural Block Layers	price_level	0.545736

Figure 6.5.2 auto neural network comparation

Next, the performance was compared between Auto Neural. The results are known as shown in the figure (fugure 6.5.2), with the same and lowest misclassification rate of 0.117125 for both 15th and 16th of Auto Neural, thus achieving an accuracy rate of 88.29%.

Fit Statistics

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Selection Criterion: Valid: Misclassification Rate	Train: Misclassification Rate
Y	AutoNeural17	AutoNeural17	AutoNeural units 30-5	price_level	0.114306	0.113931
	Tree	Tree	Decision Tree	price_level	0.117125	0.116892

Figure 6.5.3 Best neural VS Decision Tree

Finally, the two best models mentioned above, namely the decision tree model and Auto Neural, were selected for comparison. It can be concluded as shown below (figure 6.5.3) that Auto Neural has a lower misclassification rate of 0.114306 and therefore has the best classification accuracy of 88.57%.

6.6. Regression models assess

Statistics Comparison

Data Role	Target Variable	Fit Statistics	Statistics Label ▲	Reg2	Boost	Tree	Target Label
Train	monthly_rent_rm	_SBC_	Train: Schwarz's Bayesian Criterion	82658.07	.	.	
Train	monthly_rent_rm	_SUMW_	Train: Sum of Case Weights Times Freq	7094	7094	.	
Train	monthly_rent_rm	_NOBS_	Train: Sum of Frequencies	7094	7094	7094	
Train	monthly_rent_rm	_SSE_	Train: Sum of Squared Errors	7.1754E8	8.0133E8	7.6946E8	
Train	monthly_rent_rm	_DFT_	Train: Total Degrees of Freedom	7094	7094	7094	
Valid	monthly_rent_rm	_VAVERR_	Valid: Average Error Function	101158.8	.	.	
Valid	monthly_rent_rm	_VASE_	Valid: Average Squared Error	101158.8	111034.8	117235.2	
Valid	monthly_rent_rm	_VDIV_	Valid: Divisor for VASE	7093	7093	7093	
Valid	monthly_rent_rm	_VERR_	Valid: Error Function	7.1752E8	.	.	
Valid	monthly_rent_rm	_VMAX_	Valid: Maximum Absolute Error	2328.088	1783.303	1974.633	
Valid	monthly_rent_rm	_VMSE_	Valid: Mean Square Error	101158.8	.	.	
Valid	monthly_rent_rm	_VRASE_	Valid: Root Average Squared Error	318.0546	333.2189	342.3963	
Valid	monthly_rent_rm	_VRMSE_	Valid: Root Mean Square Error	318.0546	.	.	
Valid	monthly_rent_rm	_VSUMW_	Valid: Sum of Case Weights Times Freq	7093	7093	.	
Valid	monthly_rent_rm	_VNOBS_	Valid: Sum of Frequencies	7093	7093	7093	
Valid	monthly_rent_rm	_VSSE_	Valid: Sum of Squared Errors	7.1752E8	7.8757E8	8.3155E8	

Figure 6.6.1 linear regression VS gradient boosting VS decision tree

Figure 6.1.1 shows the results of three different models statistics comparison which are linear regression (Reg2), gradient boosting (Boost), and decision tree (Tree). There are mainly three key score that can evaluate the performance of the model which are Average Squared Error (ASE), Maximum Absolute Error(MAE), and Root Average Squared Error(RASE).

The ASE score represents the average of the squared errors for each prediction. Lower values indicate better performance. The linear regression model has an average squared error of 101,158.76, the gradient boosting model has 111,034.81, and the decision tree model has 117,235.25.

The MAE shows the maximum absolute difference between the predicted and actual values. Lower values indicate better performance. The linear regression model has a maximum absolute error of 2,328.09, the gradient boosting model has 1,783.30, and the decision tree model has 1,974.63.

The RASE is the square root of the average squared error and provides a measure of the average prediction error. Lower values indicate better performance. The linear regression model has a root average squared error of 318.05, the gradient boosting model has 333.22, and the decision tree model has 342.40.

Based on these statistics, it can observe that the linear regression model generally performs better than the gradient boosting and decision tree models, as it has lower values for average squared error, maximum absolute error, and root average squared error. However, it is essential to consider other factors such as the dataset and specific requirements when determining the most suitable model for a given task.

7. Conclusion

Initially, the dataset contains 13 interval variables and 4 nominal variables. After manually revising the metadata, the output is shown in the Table 7.1.

Table 7.1 Revised metadata

Role	Level	Count
ID	Interval	1
Input	Interval	11
Input	Nominal	3
Target	Interval	1
Text	Nominal	1

During data exploration processing, there are 3 types of data error such as incomplete, noisy, inconsistent were found in the dataset. The findings are summarized as Table 7.2.

Table 7.2 data error type statistics

Variable	Data Error type
completion_year	Inconstant/incomplete
completion_month	
completion_date	
monthly_rent	Noisy
deposit	Noisy
property_type	Incomplete
rooms	Noisy, incomplete
parking	Noisy, incomplete
bathrooms	Noisy, incomplete
size	Noisy
furnished	Incomplete
no_of_facilities	Incomplete
no_of_additional_facilities	Noisy, incomplete

Based on the visualizations displayed in Chapter 3, the first two objectives are achieved. The objectives are proven with key findings as listed in Table 7.3.

Table 7.3 Key finding for all the objectives

Objectives	Key findings
To analyze historical rental market trends in Kuala Lumpur and Selangor to identify patterns and fluctuations in rental prices over time.	• Most of the monthly rent is between 70-3756. • The effective time dimension can be set from 2000 to 2030 for this dataset. • Rental house prices were higher in 2010. And rental house prices also started to show signs of increase in 2019, but suddenly dropped in 2021. • There will be no appreciable difference in rental prices depending on the month and date of the year.

	• The closer the year of completion is to the present, the higher the rent, but not immediately adjacent to the present. People often prefer rental houses completed 2-3 years from now for health reasons, leading to higher rents.
To assess the impact of various factors, such as location, property characteristics, and market conditions, on rental prices in Kuala Lumpur and Selangor	• Location-wise, rental houses in Kuala Lumpur are more popular compared to Selangor. • The monthly rent reaches its highest point for houses with 4-6 bathrooms, and having more bathrooms does not necessarily mean a higher monthly rent. • Rental properties with 1 set or 3 sets of furniture tend to have higher monthly rents, while the maximum rent for properties with 2 sets of furniture is relatively lower. • Certain add-ons or additional facilities have high maximum monthly rents, but their average values remain within the normal range. On average, the number of additional facilities is directly proportional to the monthly rent, and there are a significant number of missing values for these amenities. • Generally, the more amenities a rental property offers, the higher the monthly rent. However, there are also missing values for amenities, possibly due to incomplete data entry or rental properties without any additional facilities. • Rental properties with 1-3 parking spaces are more popular and tend to have higher monthly rents, as it aligns with the average household having 1-3 cars. Having more parking spaces may not necessarily be popular, and there are missing values indicating some rentals without parking. • Apartments and service residences are in high demand, and people are willing to pay higher monthly rents for these types of properties. • Monthly rent for rental houses with 7 rooms gradually increases, and higher rents can also be observed for houses with 3-4 rooms, which corresponds to the demand for rental housing.
To develop predictive model using historical rental market data & relevant variables to forecast future rental price and price level by using regression model and classification model.	• For the classification model predicting the price level, the neural network outperforms the other models in terms of suitability after conducting analysis and comparison. • In the regression model predicting the monthly rent price, linear regression is determined to be more suitable for this dataset based on analysis and comparison of the three models. • The parameter settings of the neural network significantly influence various aspects, including training time, model training results, scalability, and stability.

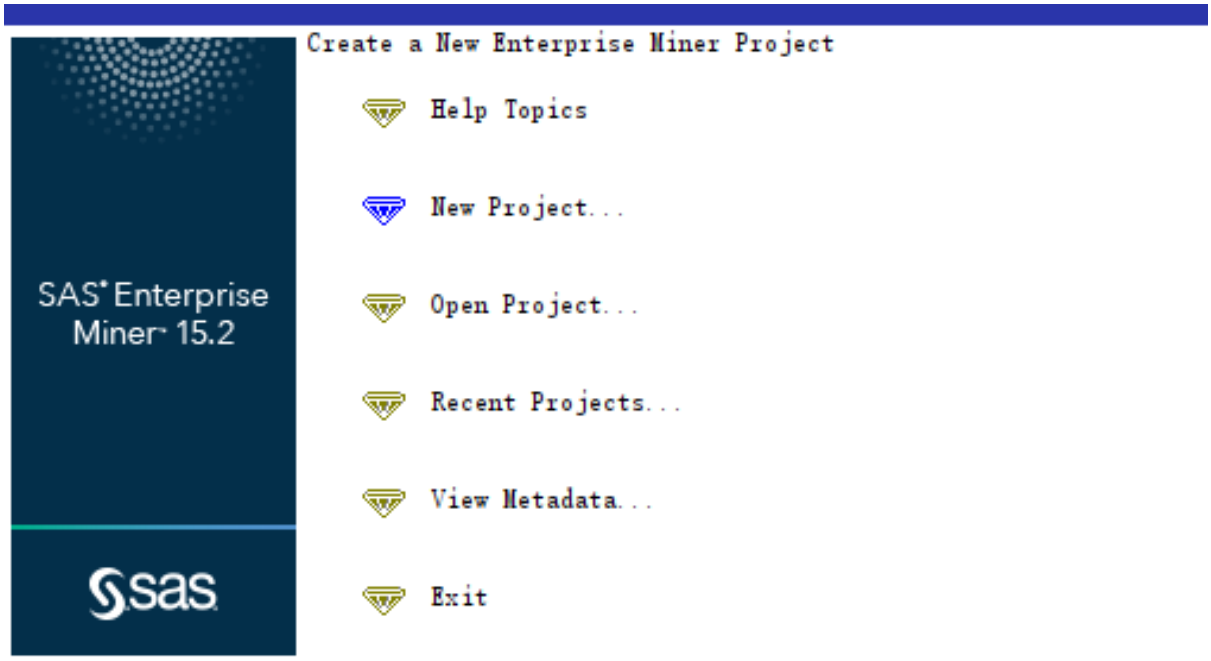
8. Appendix

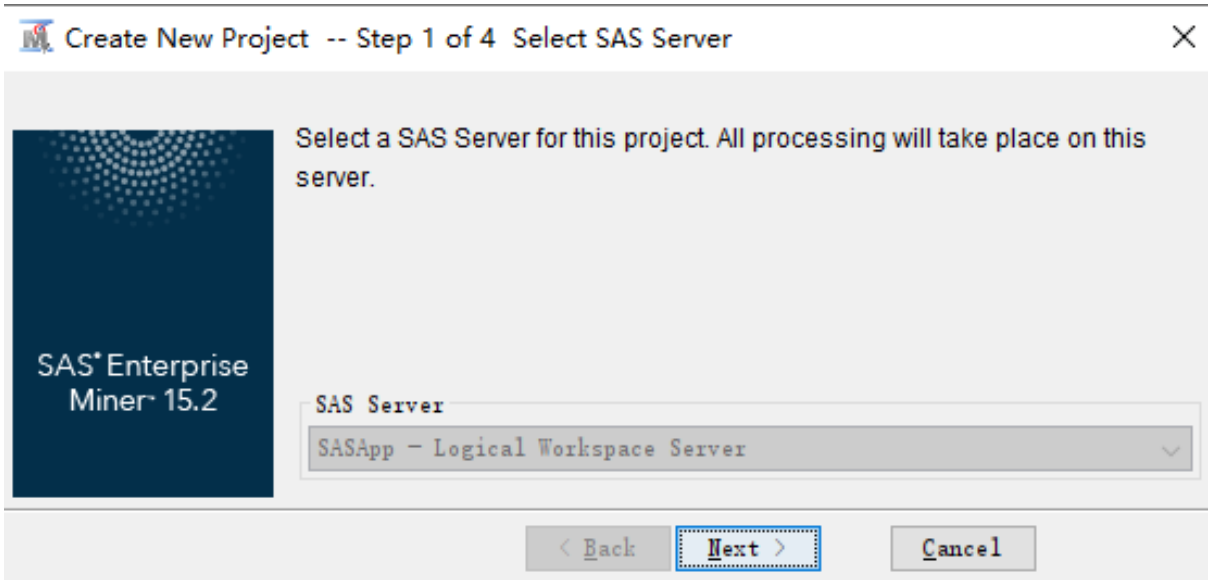
8.1. Presentation Link

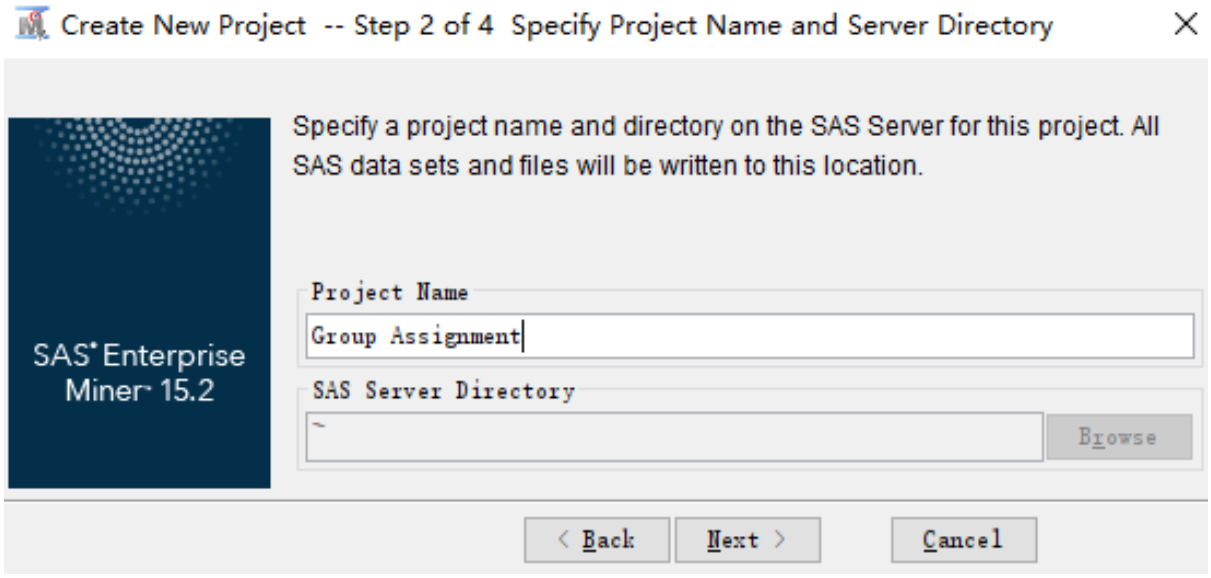
Link: https://youtu.be/63W1HhECfzQ

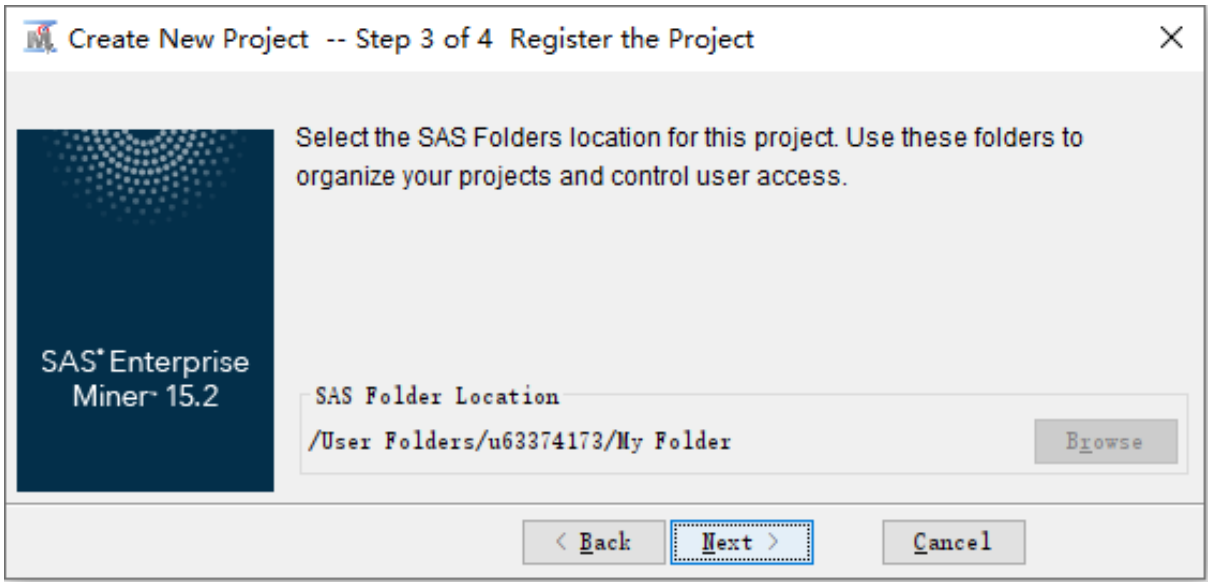
8.2. Create project/diagram/library/dataset/data source

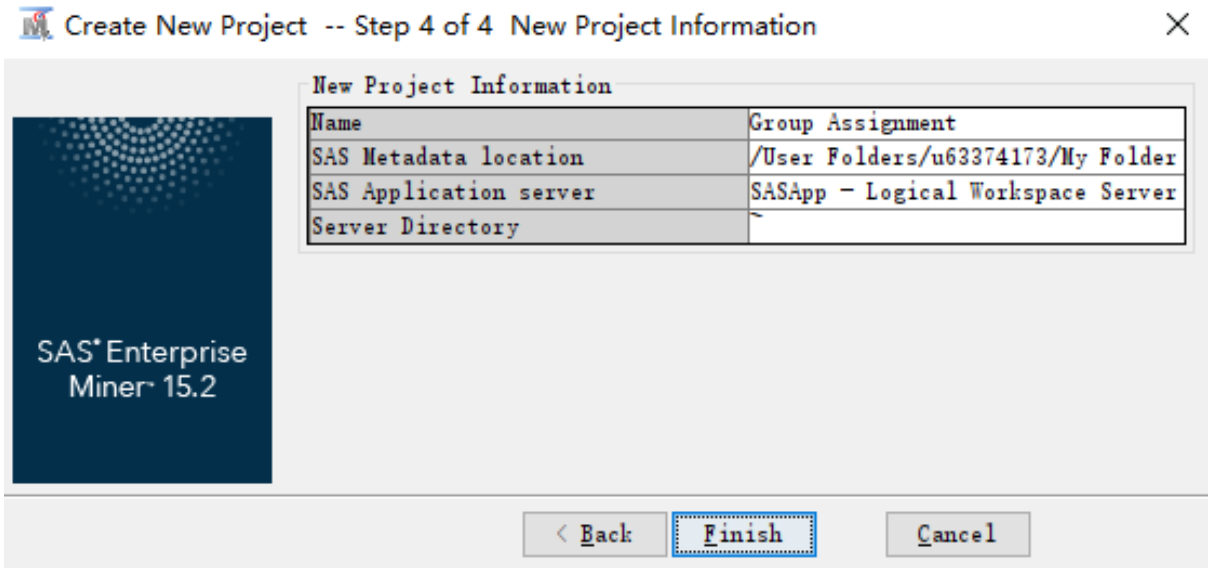
Create a new Enterprise Miner Project

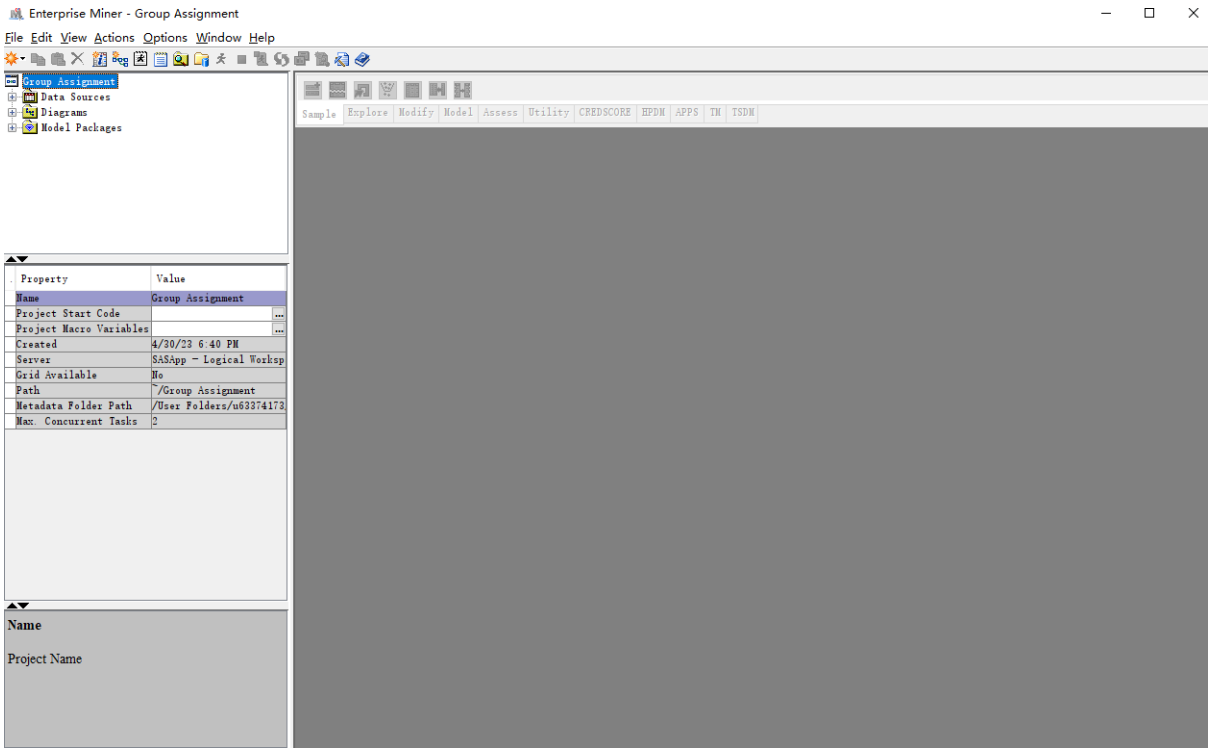




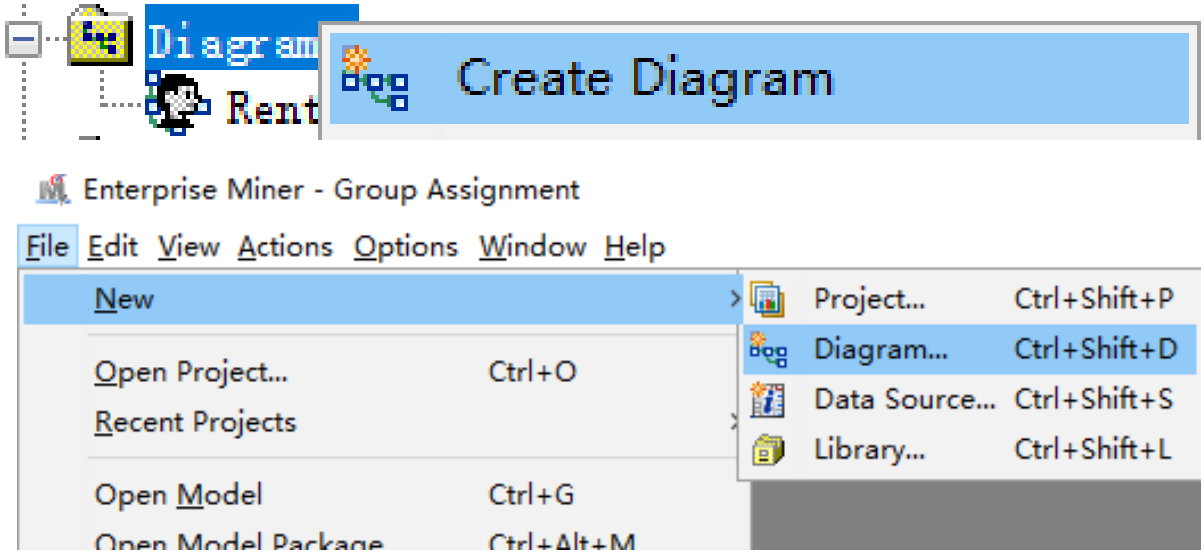

Create New Project -- Step 2 of 4 Specify Project Name and Server Directory
SAS Enterprise Miner 15.2
Specify a project name and directory on the SAS Server for this project. All SAS data sets and files will be written to this location.
Project Name
Group Assignment
SAS Server Directory
~
Browse
< Back
Next >
Cancel

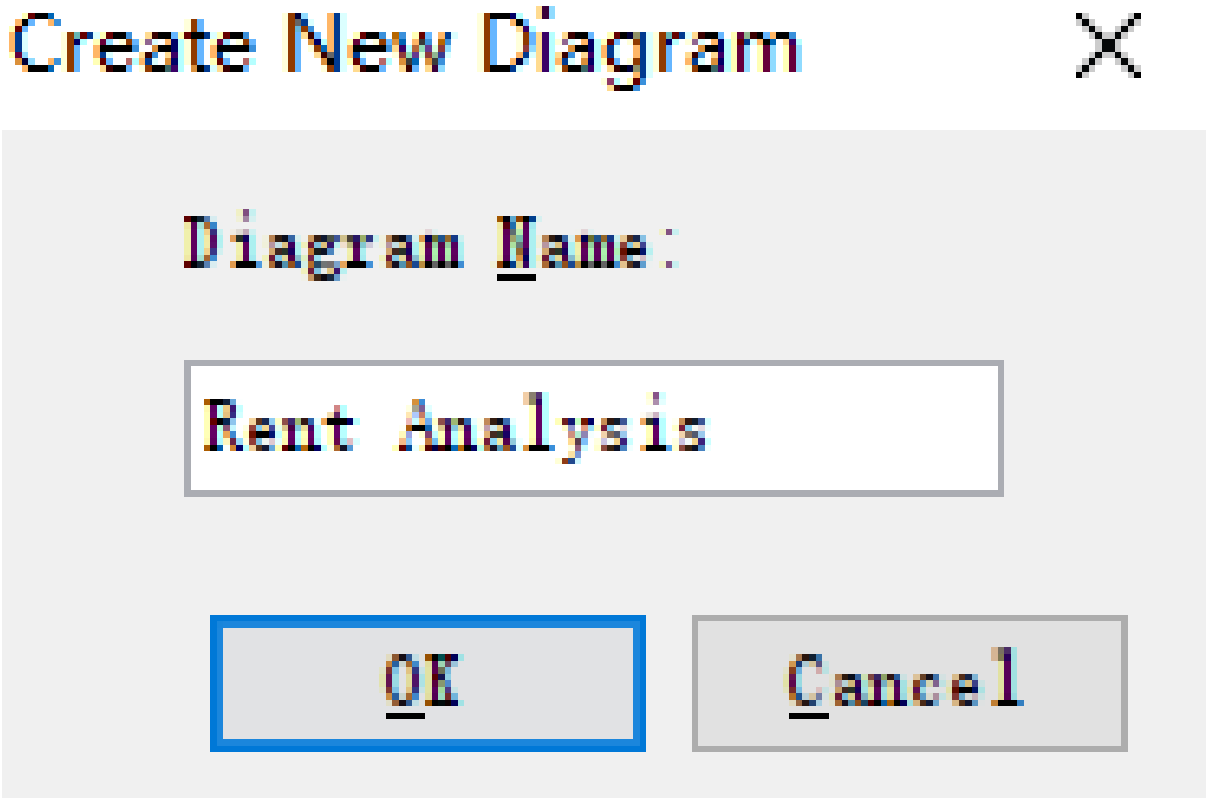

Create New Project -- Step 3 of 4 Register the Project
SAS Enterprise Miner 15.2
Select the SAS Folders location for this project. Use these folders to organize your projects and control user access.
SAS Folder Location
/User Folders/u63374173/My Folder
Browse
< Back
Next >
Cancel


Create New Project -- Step 4 of 4 New Project Information
SAS Enterprise Miner 15.2
New Project Information
Name
Group Assignment
SAS Metadata location
/User Folders/u63374173/My Folder
SAS Application server
SASApp - Logical Workspace Server
Server Directory
~
< Back
Finish
Cancel

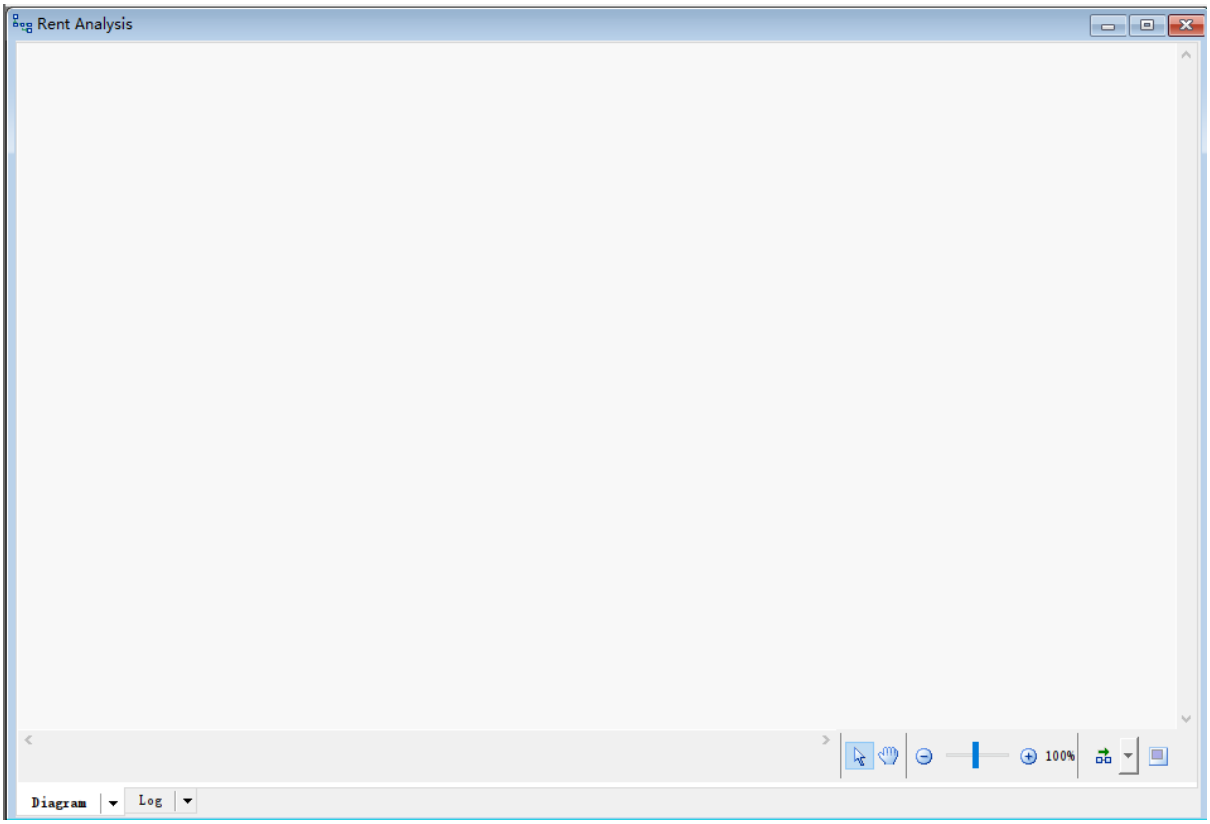


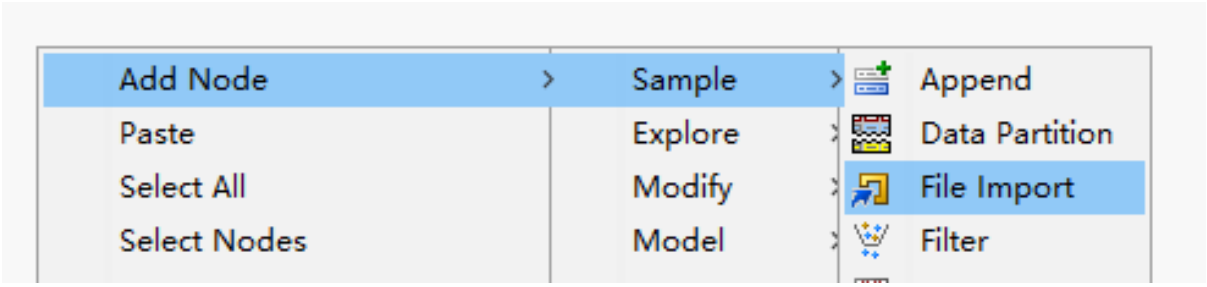
Create a new Diagram

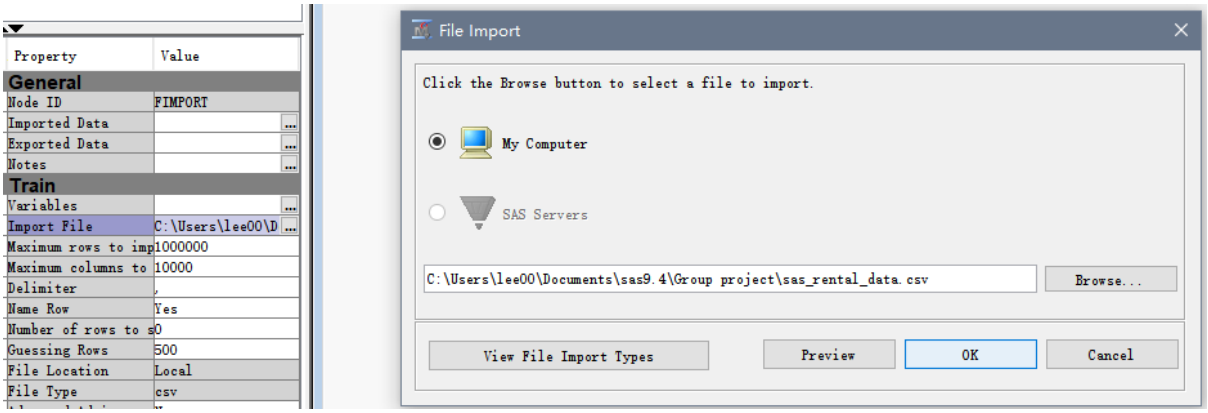


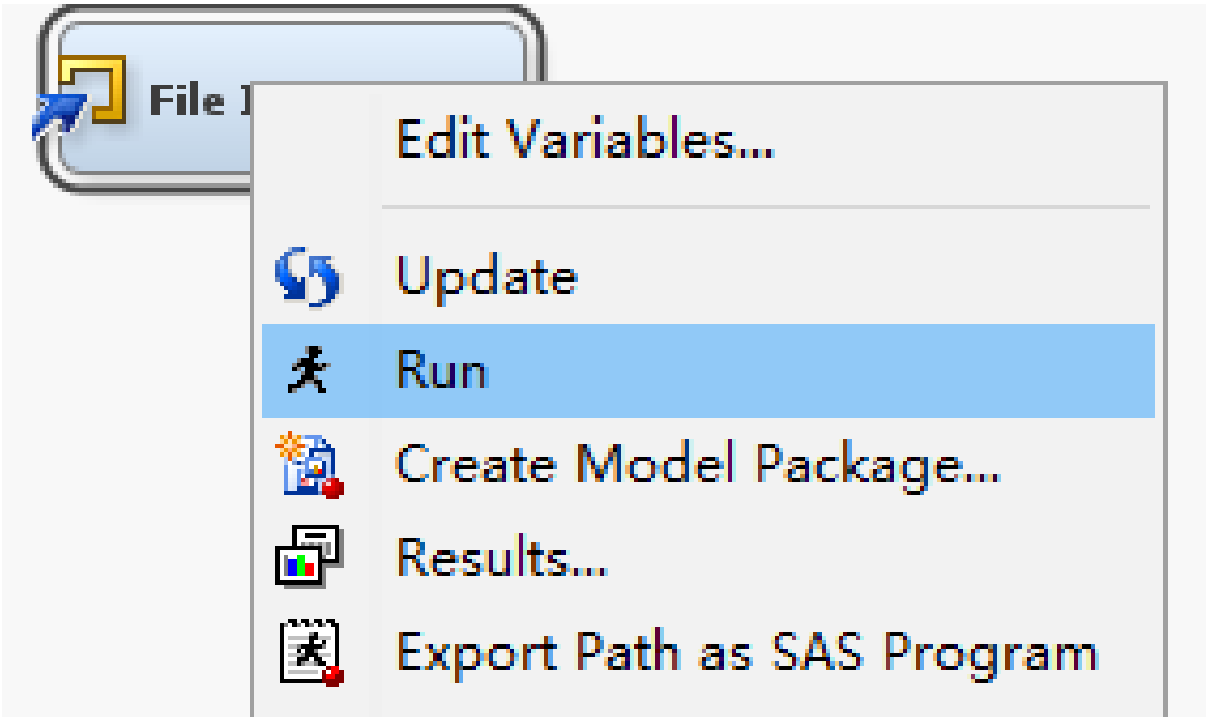


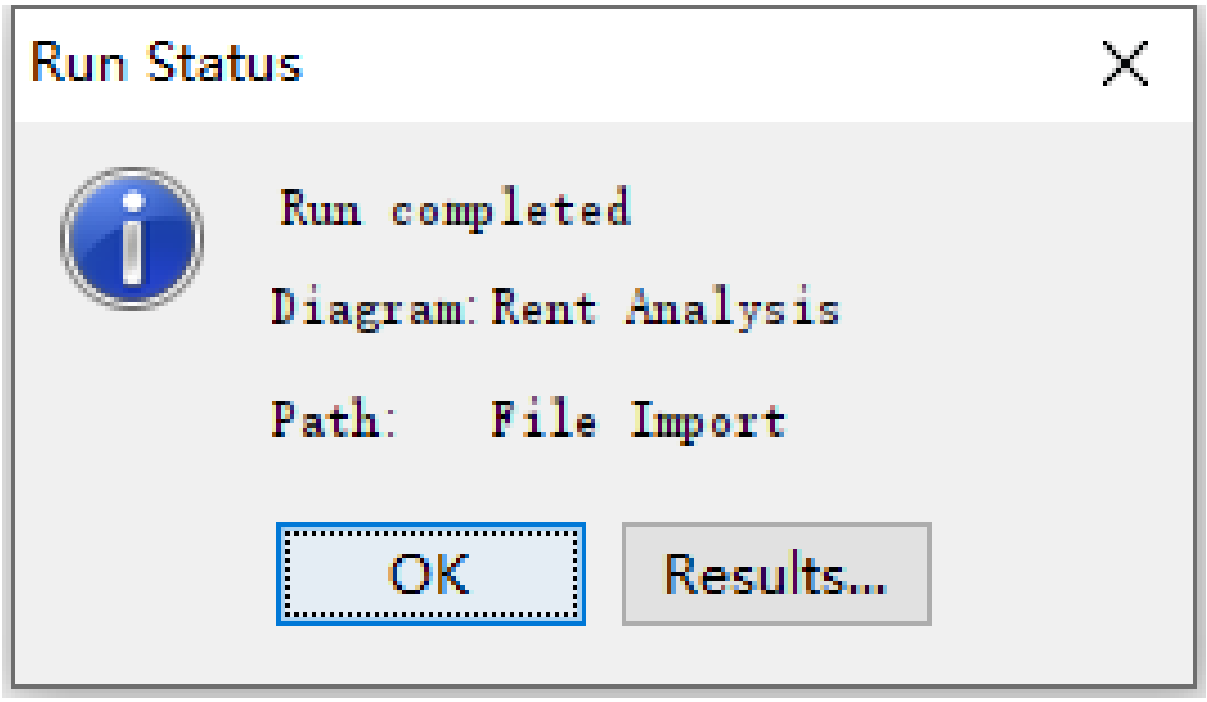
CSV Dataset Import



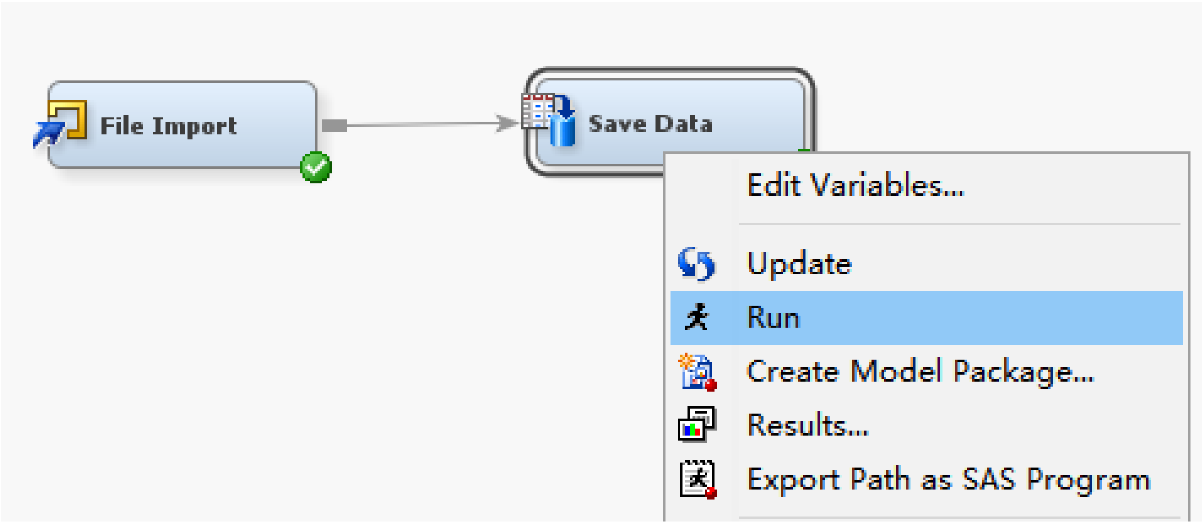

Add Node
Paste
Select All
Select Nodes
Sample
Explore
Modify
Model
Append
Data Partition
File Import
Filter

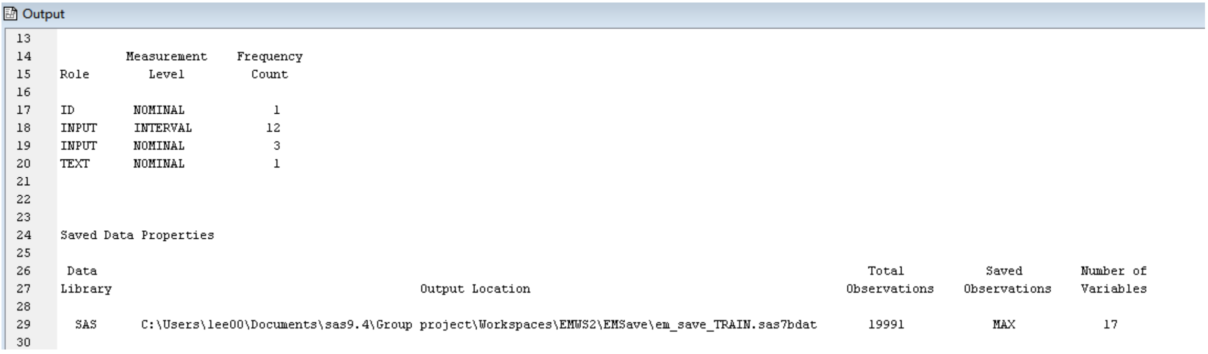

Property
Value
General
Node ID
FIMPORT
Imported Data
Exported Data
Notes
Train
Variables
Import File
C:\Users\lee00\D
Maximum rows to imp
1000000
Maximum columns to
10000
Delimiter
,
Name Row
Yes
Number of rows to s
0
Guessing Rows
500
File Location
Local
File Type
csv
File Import
Click the Browse button to select a file to import.
My Computer
SAS Servers
C:\Users\lee00\Documents\sas9.4\Group project\sas_rental_data.csv
Browse...
View File Import Types
Preview
OK
Cancel


File I
Edit Variables...
Update
Run
Create Model Package...
Results...
Export Path as SAS Program


Run Status
Run completed
Diagram: Rent Analysis
Path: File Import
OK
Results...

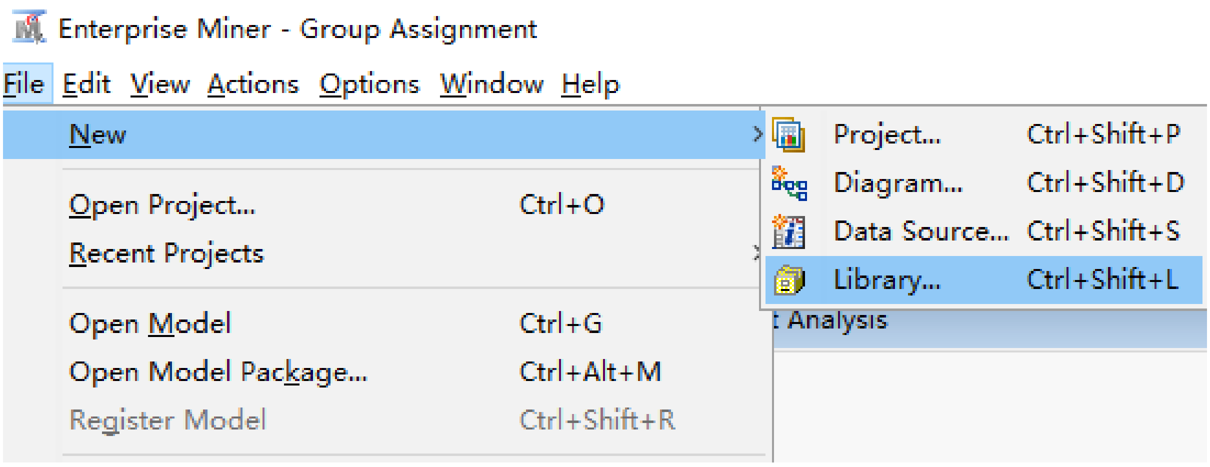
CSV transform to SAS

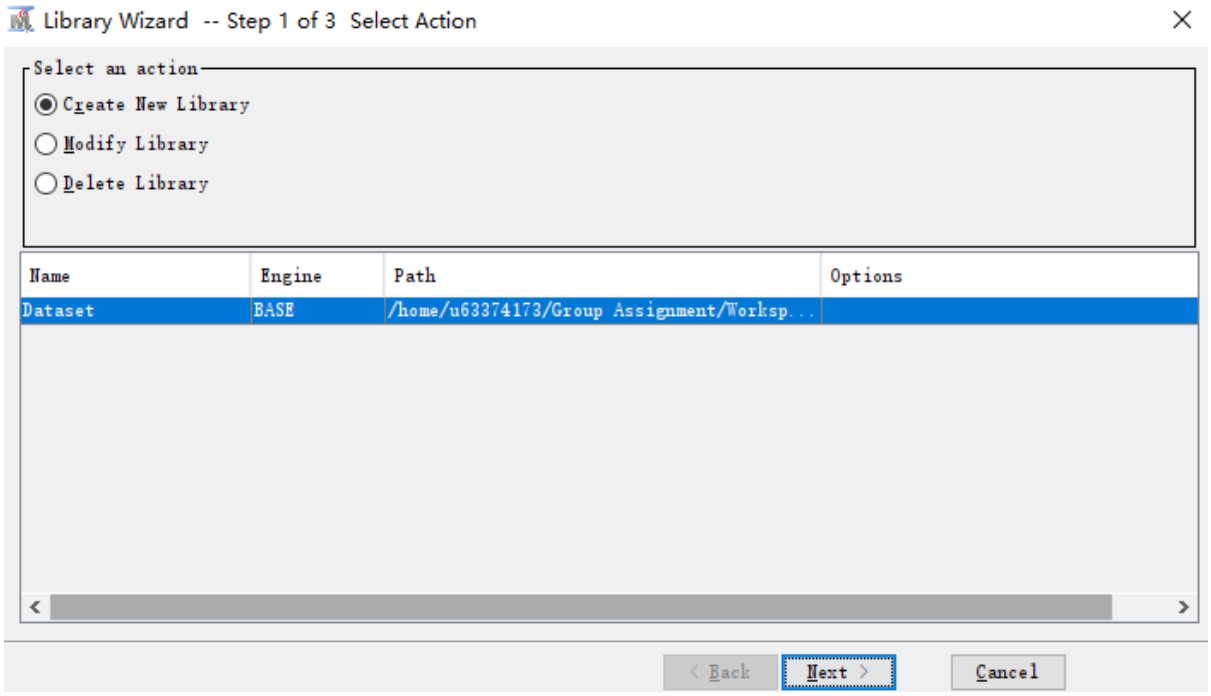


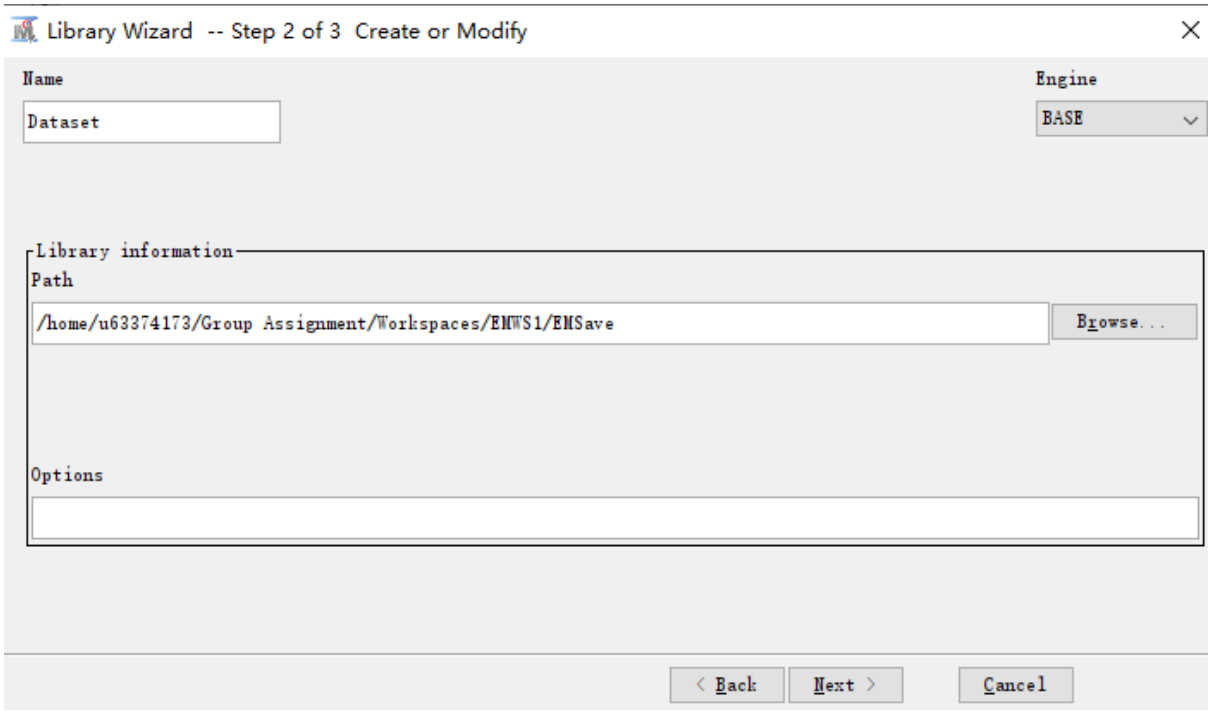


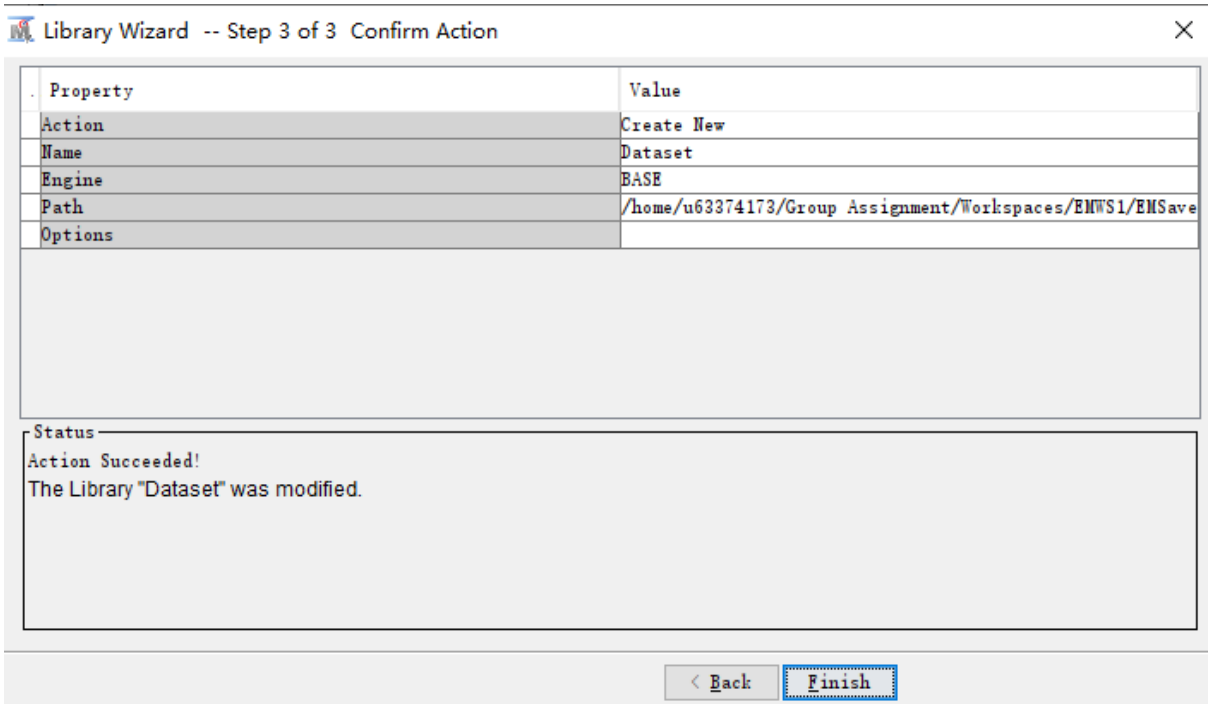
```
Output

                 Measurement    Frequency
Role                Level          Count

ID                NOMINAL           1
INPUT             INTERVAL         12
INPUT             NOMINAL           3
TEXT              NOMINAL           1


Saved Data Properties

 Data                                                                           Total          Saved        Number of
Library                          Output Location                             Observations   Observations   Variables

  SAS     C:\Users\lee00\Documents\sas9.4\Group project\Workspaces\EMWS2\EMSave\em_save_TRAIN.sas7bdat      19991          MAX            17
```

Create a new library

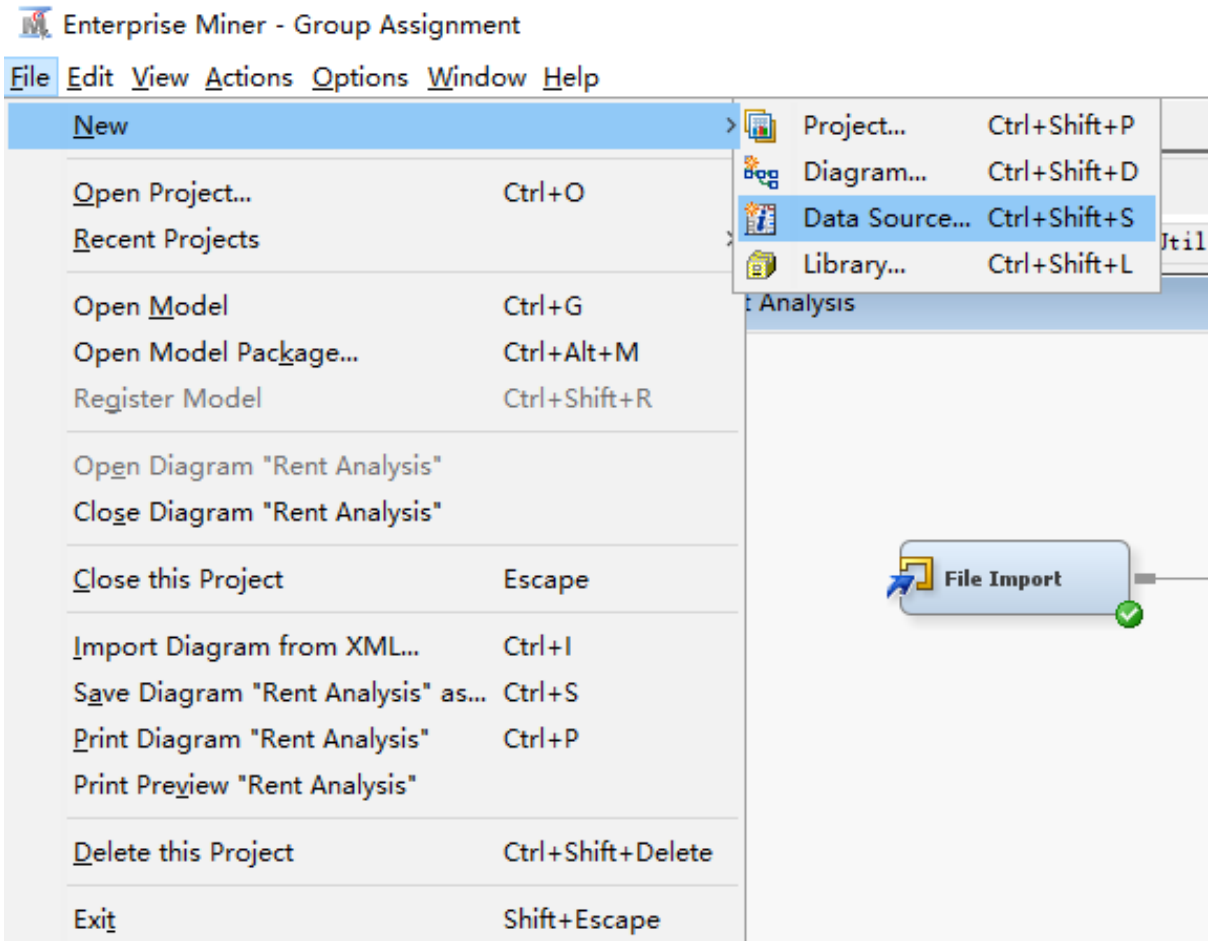



Library Wizard -- Step 1 of 3 Select Action
Select an action
Create New Library
Modify Library
Delete Library
Name
Engine
Path
Options
Dataset
BASE
/home/u63374173/Group Assignment/Worksp...
Back
Next
Cancel


Library Wizard -- Step 2 of 3 Create or Modify
Name
Dataset
Engine
BASE
Library information
Path
/home/u63374173/Group Assignment/Workspaces/EMWS1/EMSave
Browse...
Options
Back
Next
Cancel



Create data source



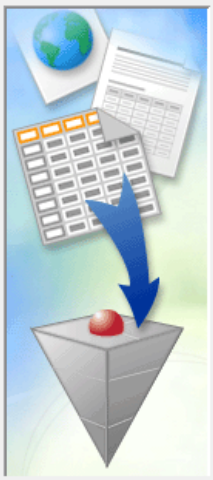
Data Source Wizard -- Step 1 of 8 Metadata Source

Select a metadata source

Source : SAS Table

< Back | Next > | Cancel

Data Source Wizard -- Step 2 of 8 Select a SAS Table



Select a SAS table

Table : DATASET.EM_SAVE_TRAIN Browse..

< Back Next > Cancel

Data Source Wizard -- Step 3 of 8 Table Information

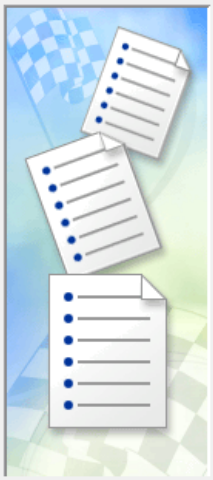


Table Properties

Property	Value
Table Name	DATASET.EM_SAVE_TRAIN
Description	
Member Type	DATA
Data Set Type	DATA
Engine	BASE
Number of Variables	16
Number of Observations	19991
Created Date	April 30, 2023 10:47:14 AM CST
Modified Date	April 30, 2023 10:47:14 AM CST

< Back Next > Cancel

Data Source Wizard -- Step 4 of 8 Metadata Advisor Options

Metadata Advisor Options

Use the basic setting to set the initial measurement levels and roles based on the variable attributes.

Use the advanced setting to set the initial measurement levels and roles based on both the variable attributes and distributions.

○ Basic ◉ Advanced Customize...

< Back Next > Cancel

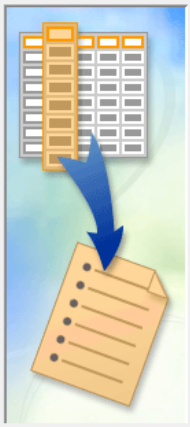
Data Source Wizard -- Step 5 of 8 Column Metadata

(none) not Equal to

Columns: Label Mining Basic Statistics

Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit	Type	Format	Informat	Length
No_of_additi	Input	Nominal	No		No	.	.	Character	$2.	$2.	2
No_of_facili	Input	Nominal	No		No	.	.	Character	$2.	$2.	2
additional_f	Rejected	Nominal	No		No	.	.	Character	$68.	$68.	68
ads_id	ID	Interval	No		No	.	.	Numeric	BEST12.0	BEST32.0	8
bathroom	Input	Nominal	No		No	.	.	Numeric	BEST12.0	BEST32.0	8
completion_y	Rejected	Nominal	No		No	.	.	Character	$4.	$4.	4
facilities	Text	Nominal	No		No	.	.	Character	$169.	$169.	169
furnished	Input	Nominal	No		No	.	.	Numeric	BEST12.0	BEST32.0	8
location	Rejected	Nominal	No		No	.	.	Character	$20.	$20.	20
monthly_rent	Input	Interval	No		No	.	.	Numeric	BEST12.0	BEST32.0	8
parking	Input	Nominal	No		No	.	.	Character	$2.	$2.	2
prop_name	Text	Nominal	No		No	.	.	Character	$84.	$84.	84
property_typ	Input	Nominal	No		No	.	.	Character	$17.	$17.	17
region	Input	Binary	No		No	.	.	Numeric	BEST12.0	BEST32.0	8
rooms	Input	Nominal	No		No	.	.	Numeric	BEST12.0	BEST32.0	8
size	Input	Interval	No		No	.	.	Numeric	BEST12.0	BEST32.0	8

Data Source Wizard -- Step 6 of 8 Create Sample



Do you wish to create a sample data set?

(•) No ( ) Yes

Table Info

Columns 16

Rows 19991

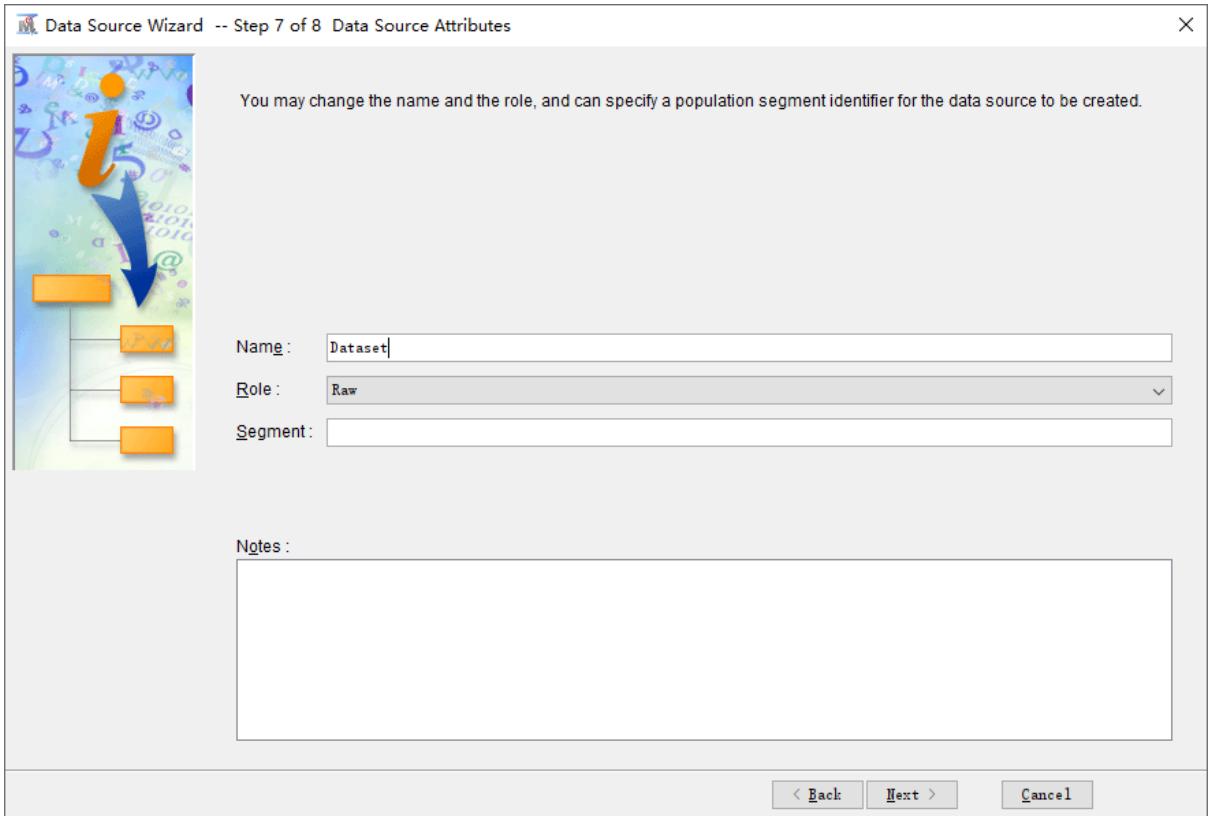
Sample Size

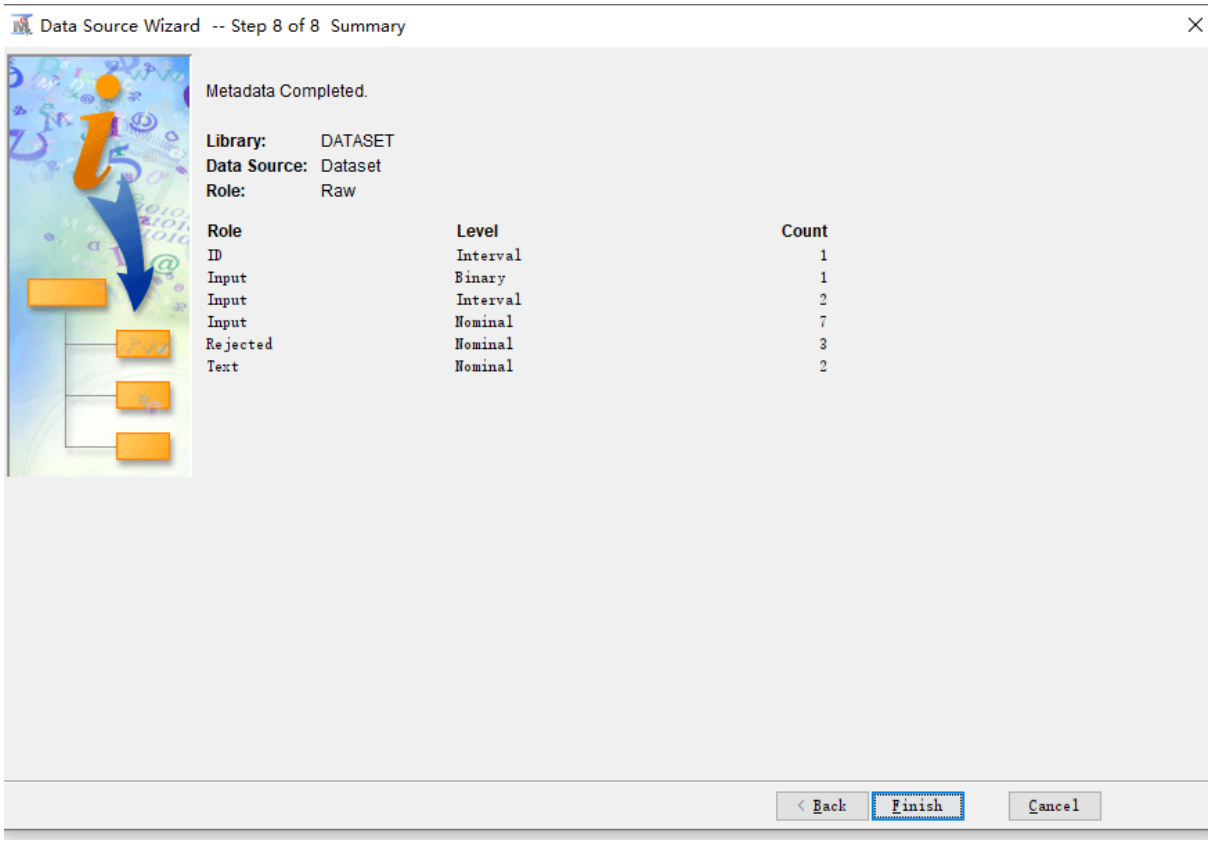
Type Percent

Percent 20

Rows

< Back Next > Cancel

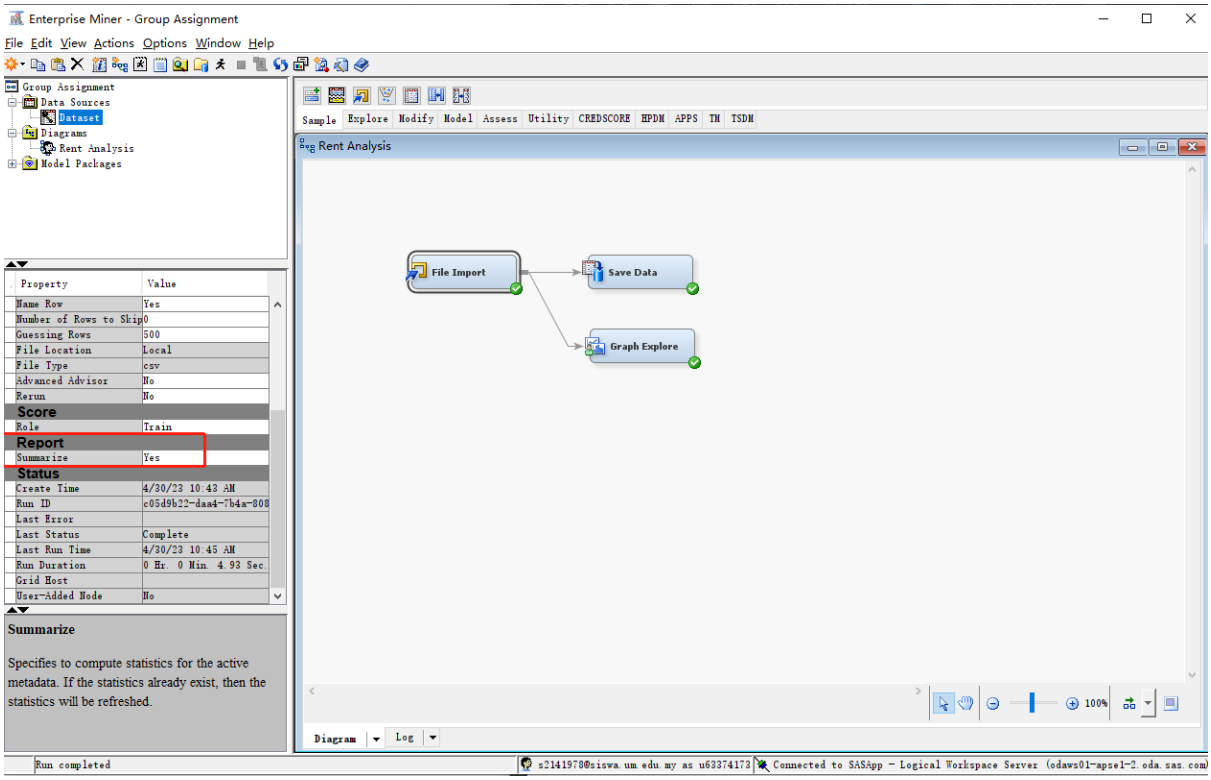

Data Source Wizard -- Step 7 of 8 Data Source Attributes
You may change the name and the role, and can specify a population segment identifier for the data source to be created.
Name :
Dataset
Role :
Raw
Segment :
Notes :
< Back
Next >
Cancel


Data Source Wizard -- Step 8 of 8 Summary
Metadata Completed.
Library: DATASET
Data Source: Dataset
Role: Raw
Role Level Count
ID Interval 1
Input Binary 1
Input Interval 2
Input Nominal 7
Rejected Nominal 3
Text Nominal 2
< Back
Finish
Cancel

8.3. Raw data Exploration

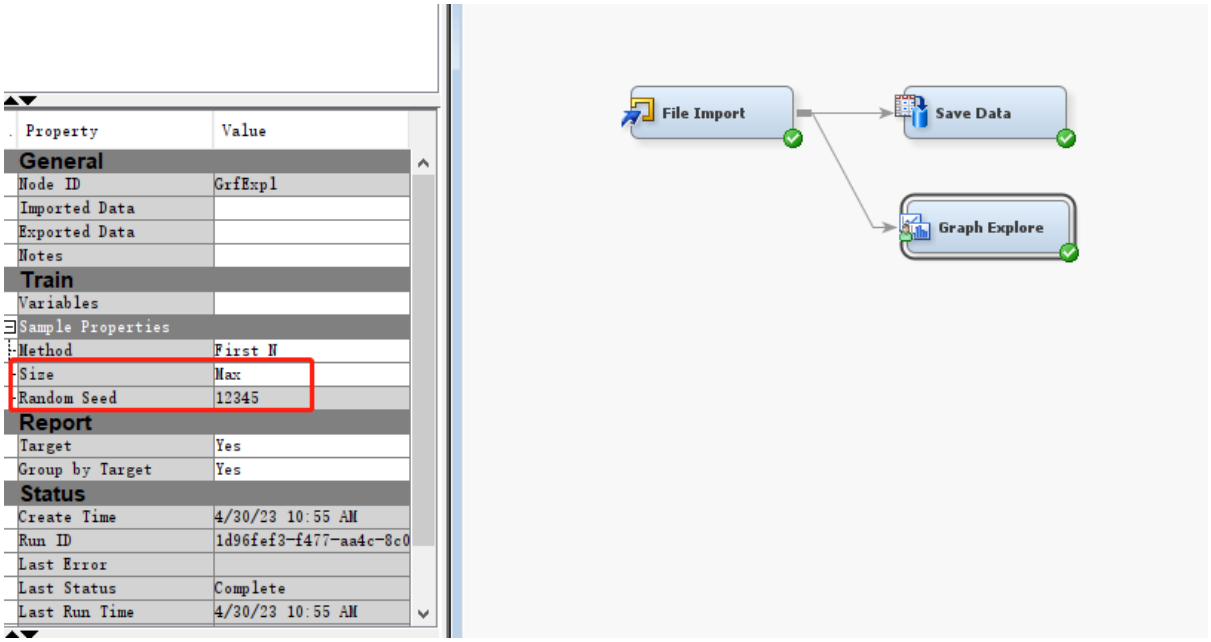
Generate a Statistics Table

Set the Summarize property at the property side bar to Yes before running the “File Import” node. Right click “File Import” node and select **Results...** to display the Statistics Table.



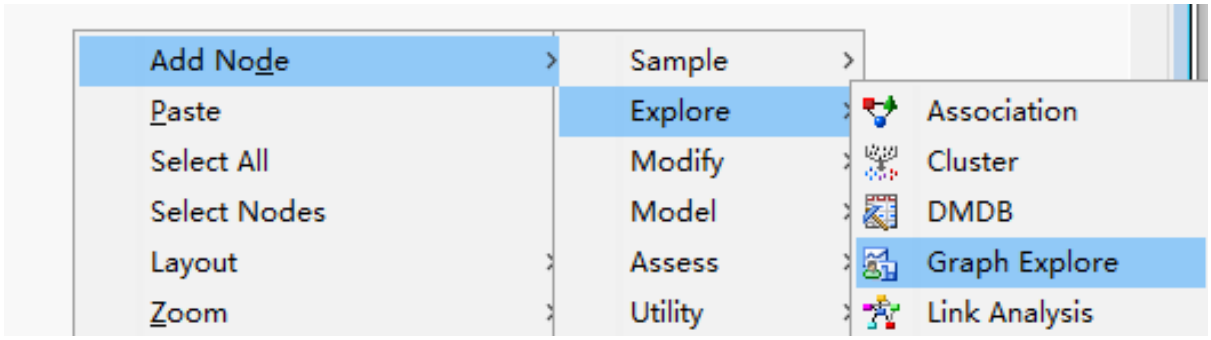
Set a sample size to be used for creating graphs

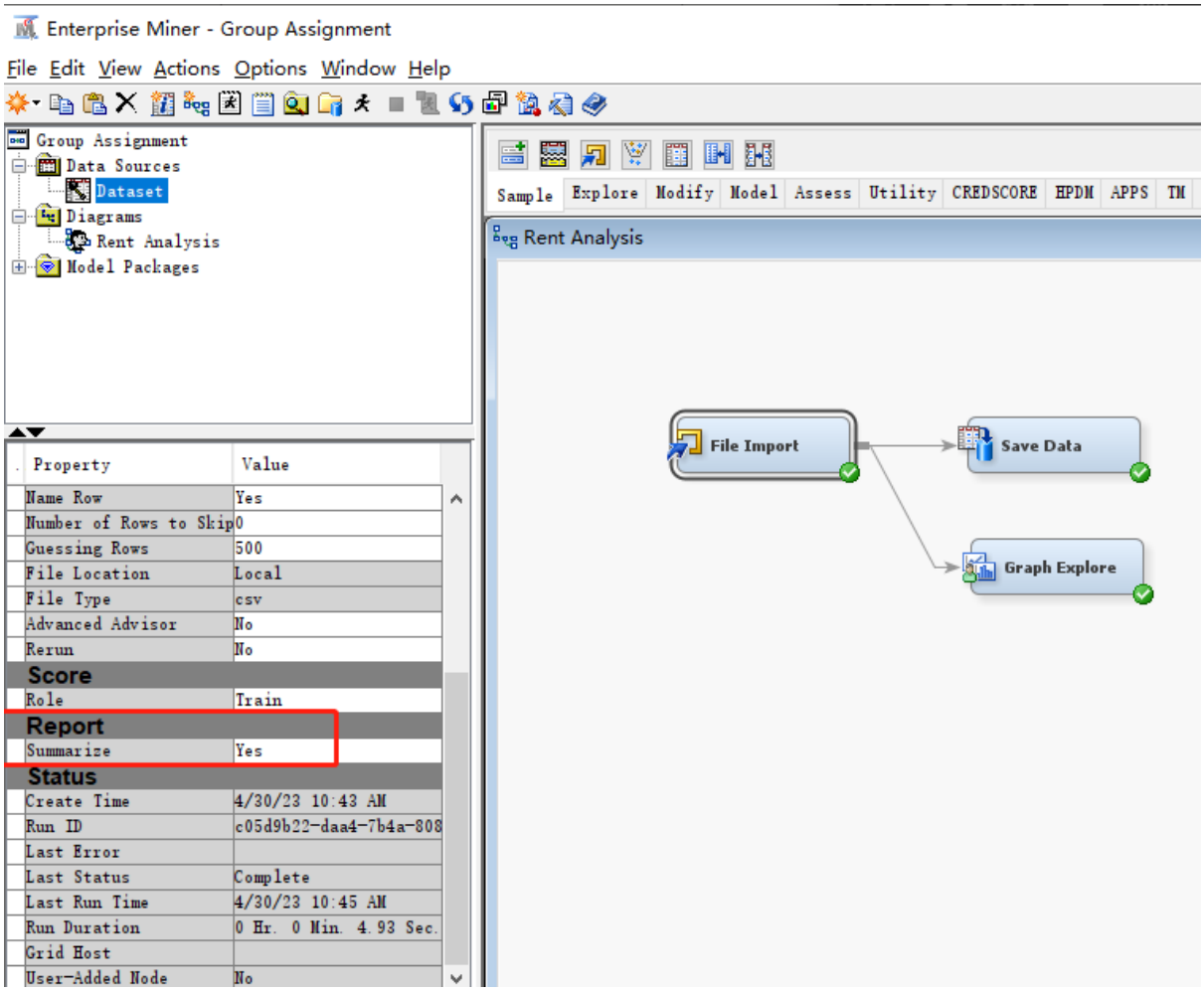
Click the “Graph Explore” node. Under the “Train” property at the property side bar, select “Max” for “Size” under the Sample Properties before running the node.

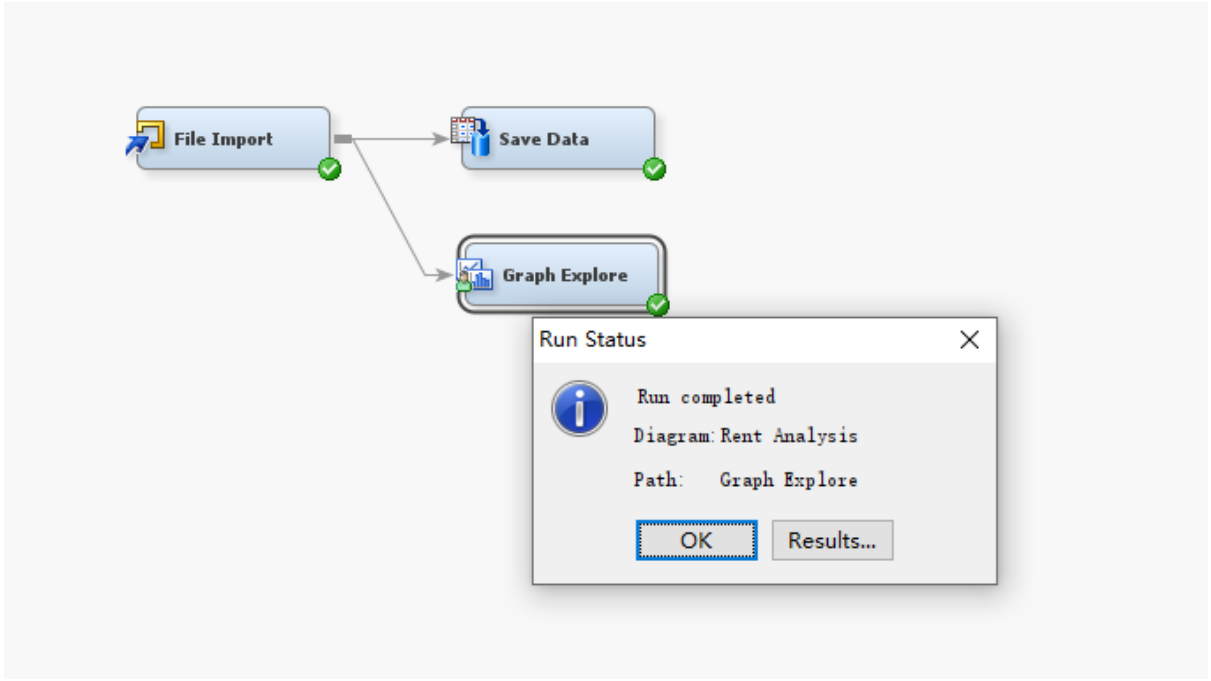


Create graphs

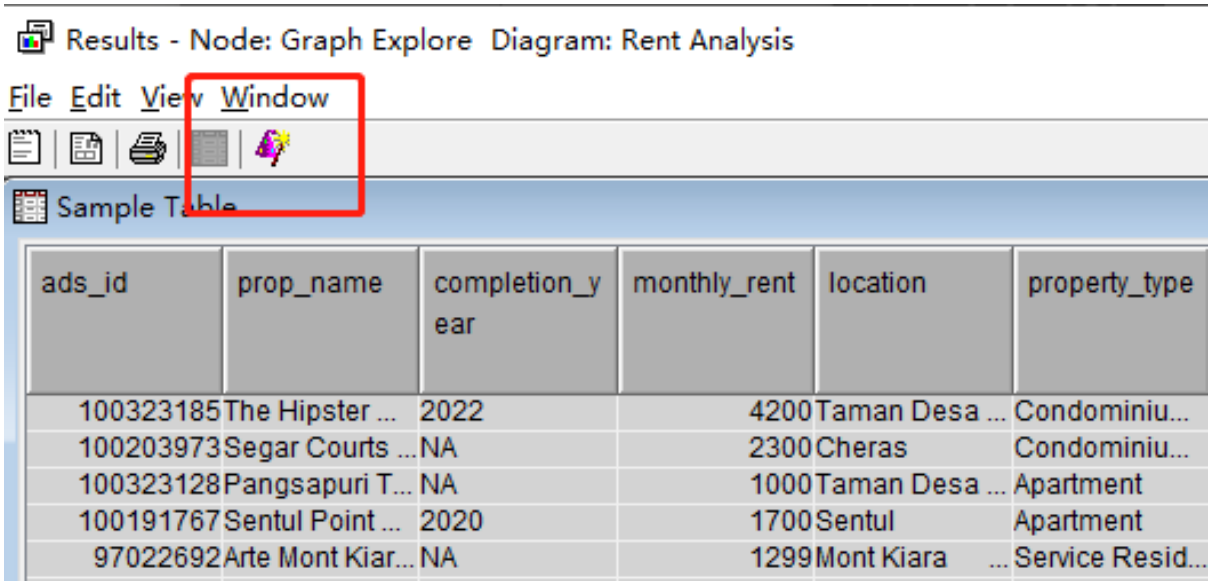
Click **Explore** and drag **Graph Explore** into the diagram.




Enterprise Miner - Group Assignment
File Edit View Actions Options Window Help
Group Assignment
Data Sources
Dataset
Diagrams
Rent Analysis
Model Packages
Sample Explore Modify Model Assess Utility CREDSCORE HPDM APPS TM
Rent Analysis
File Import
Save Data
Graph Explore
Property Value
Name Row Yes
Number of Rows to Skip 0
Guessing Rows 500
File Location Local
File Type csv
Advanced Advisor No
Rerun No
Score
Role Train
Report
Summarize Yes
Status
Create Time 4/30/23 10:43 AM
Run ID c05d9b22-daa4-7b4a-808
Last Error
Last Status Complete
Last Run Time 4/30/23 10:45 AM
Run Duration 0 Hr. 0 Min. 4.93 Sec.
Grid Host
User-Added Node No

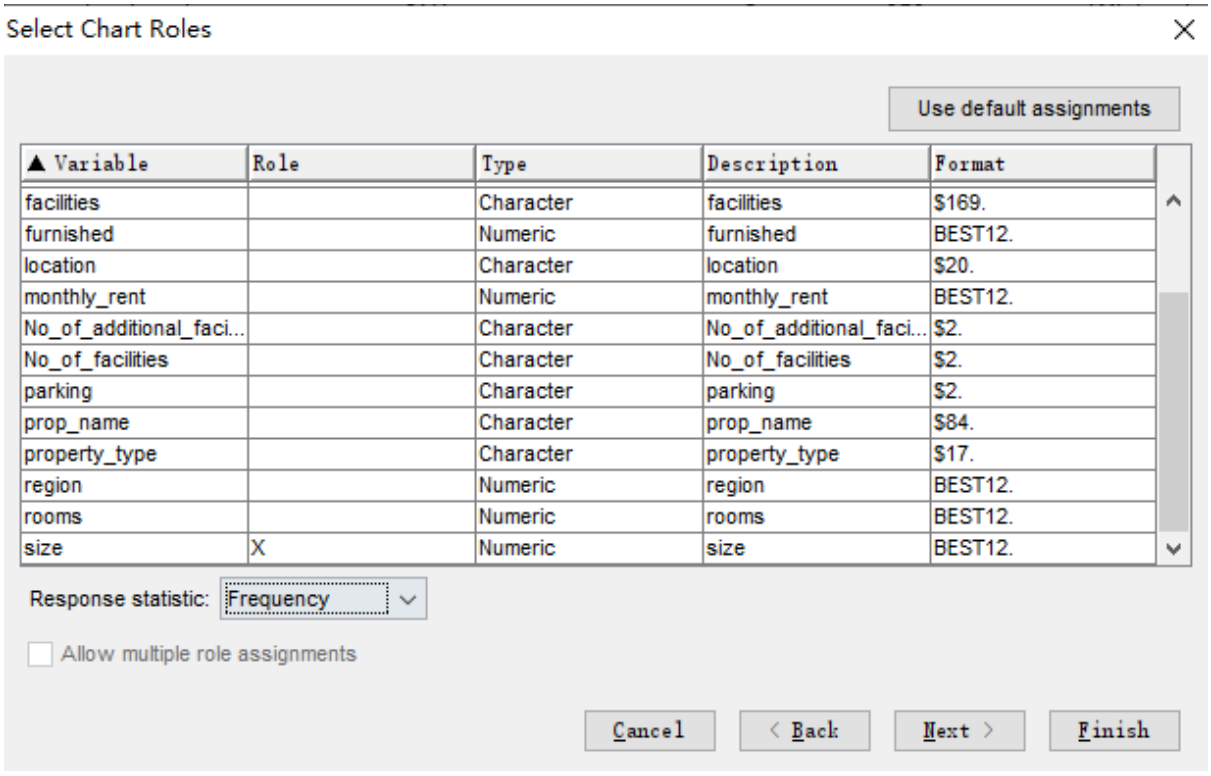

File Import
Save Data
Graph Explore
Run Status
Run completed
Diagram: Rent Analysis
Path: Graph Explore
OK
Results...

After the “Graph Explore” node has been successfully run, select **Plot…** icon as highlighted below and choose the desired graphs.



ads_id	prop_name	completion_year	monthly_rent	location	property_type
100323185	The Hipster ...	2022	4200	Taman Desa ...	Condominiu...
100203973	Segar Courts ...	NA	2300	Cheras	Condominiu...
100323128	Pangsapuri T...	NA	1000	Taman Desa ...	Apartment
100191767	Sentul Point ...	2020	1700	Sentul	Apartment
97022692	Arte Mont Kiar...	NA	1299	Mont Kiara ...	Service Resid...

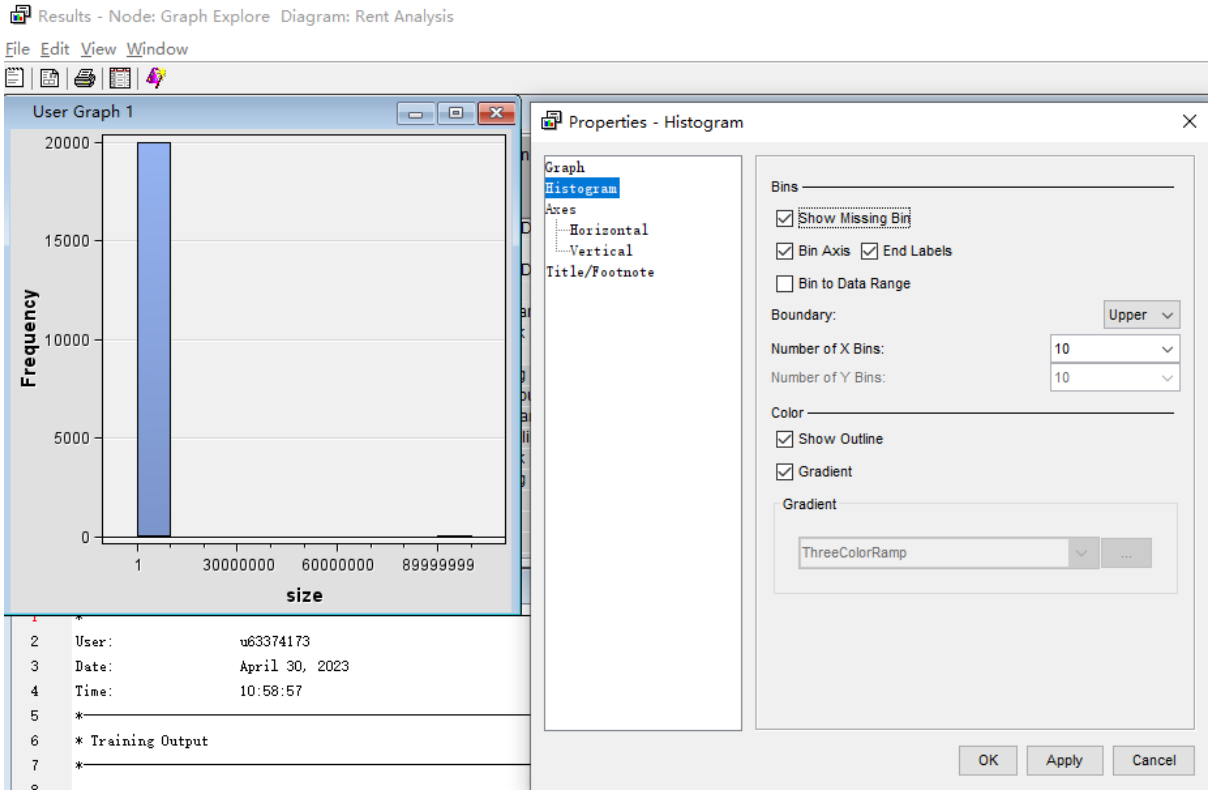
Specify the variable that the system wish to visualize and click **Finish**.

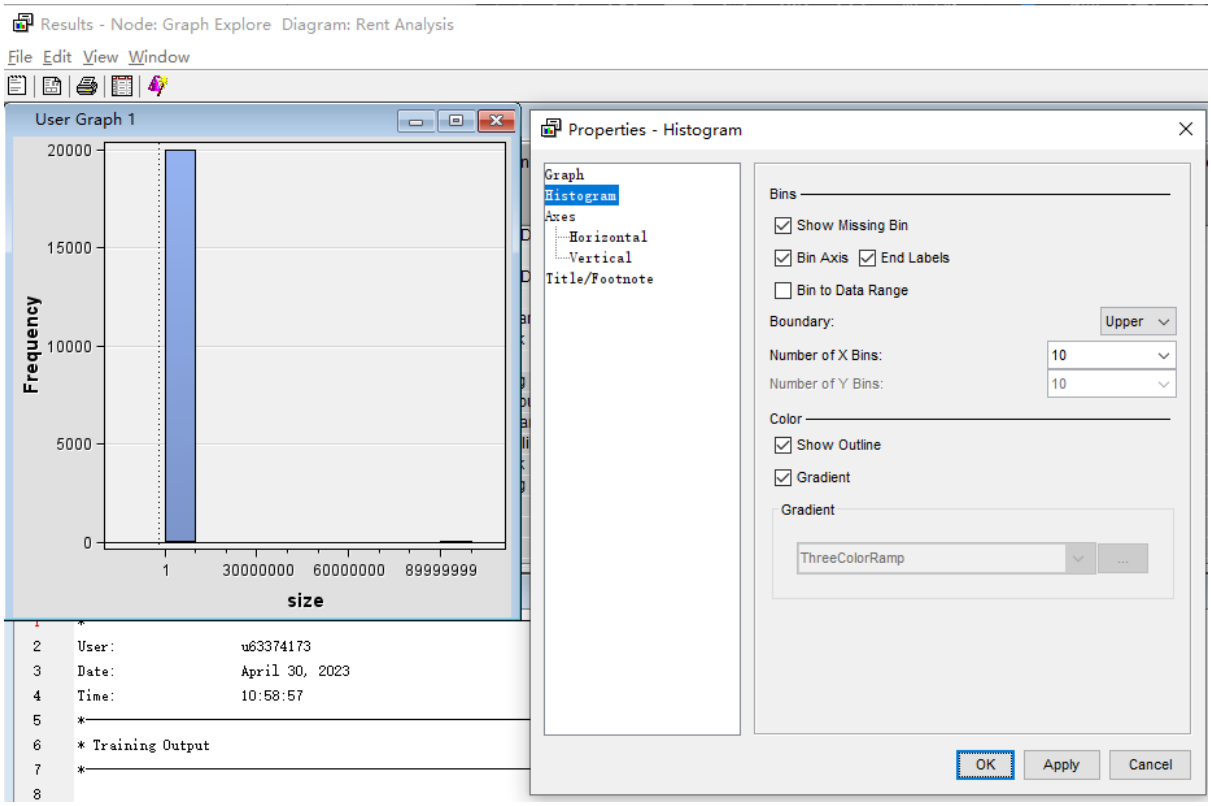


▲ Variable	Role	Type	Description	Format
facilities		Character	facilities	$169.
furnished		Numeric	furnished	BEST12.
location		Character	location	$20.
monthly_rent		Numeric	monthly_rent	BEST12.
No_of_additional_faci...		Character	No_of_additional_faci...	$2.
No_of_facilities		Character	No_of_facilities	$2.
parking		Character	parking	$2.
prop_name		Character	prop_name	$84.
property_type		Character	property_type	$17.
region		Numeric	region	BEST12.
rooms		Numeric	rooms	BEST12.
size	X	Numeric	size	BEST12.

To add a “missing” bin to the histogram

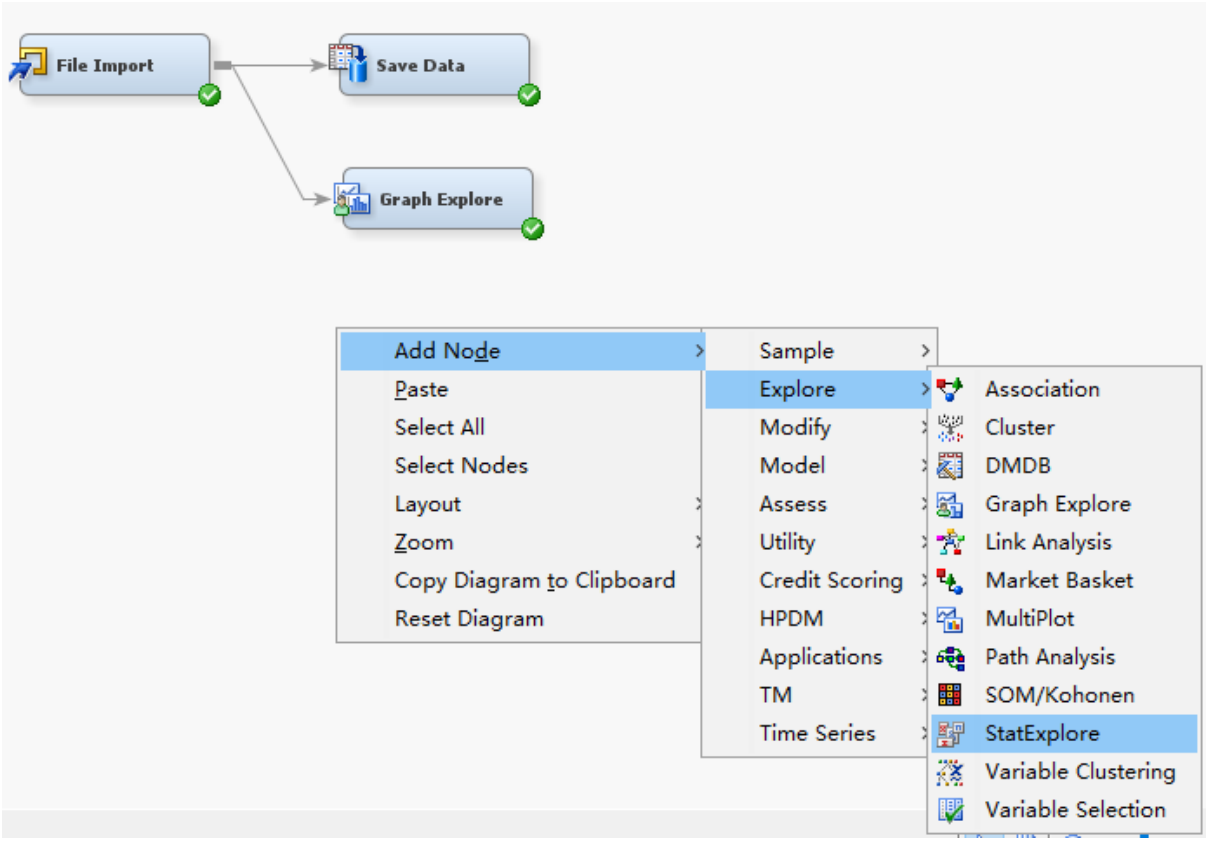
Right click on the created histogram and select Graph Properties. Tick “Show Missing Bin” and select OK.

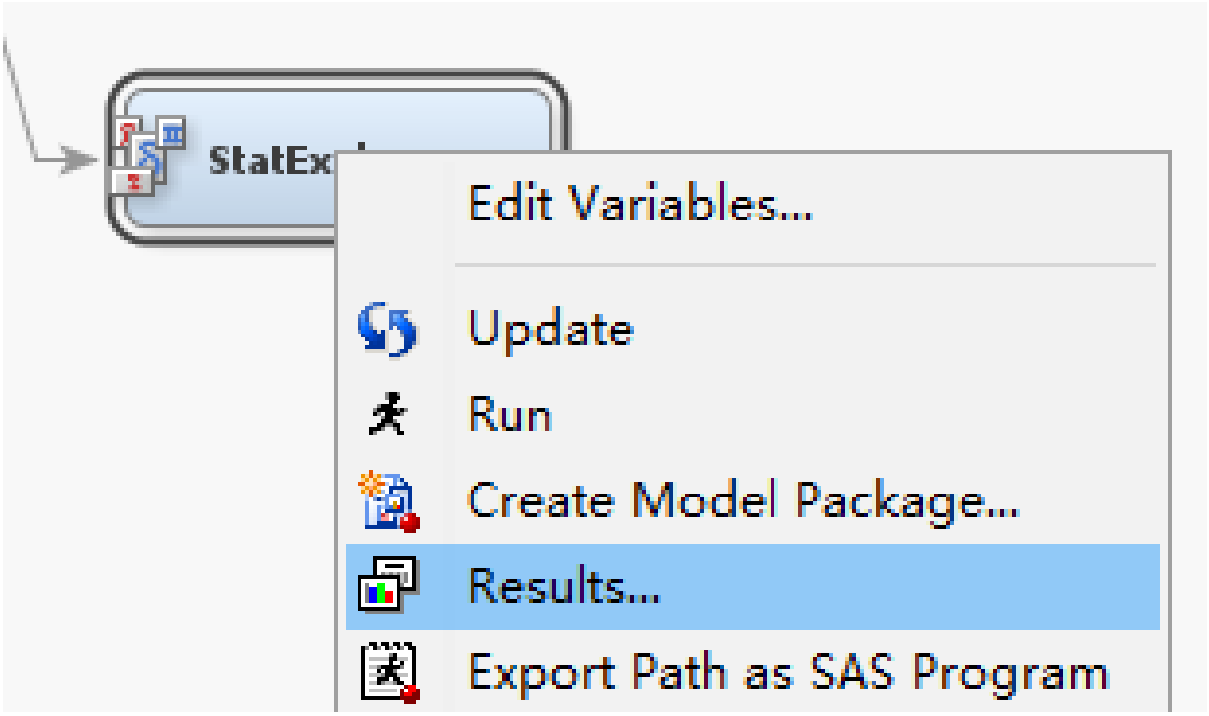




To obtain Cramer's V statistic, StatExplore node is added.

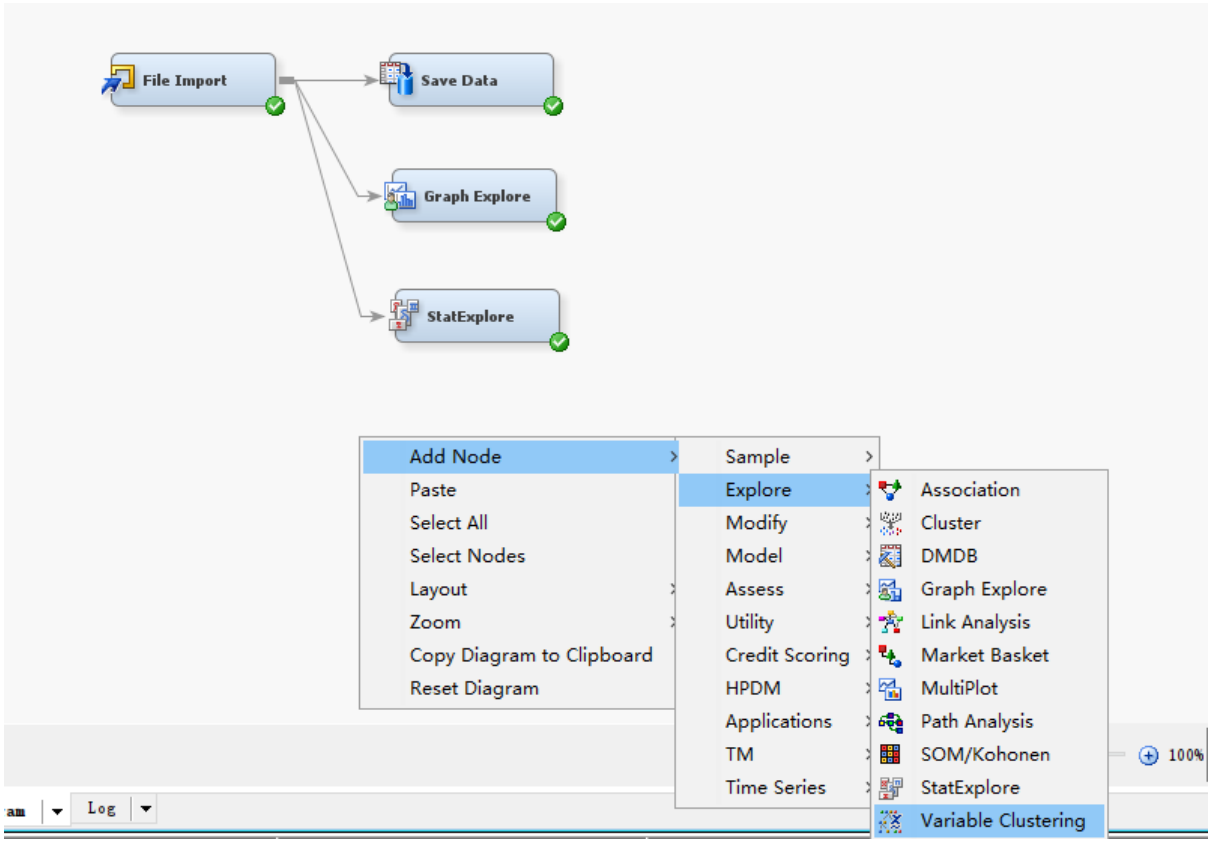
Select Run > Results... > View > Plots > Chi-Square Plot

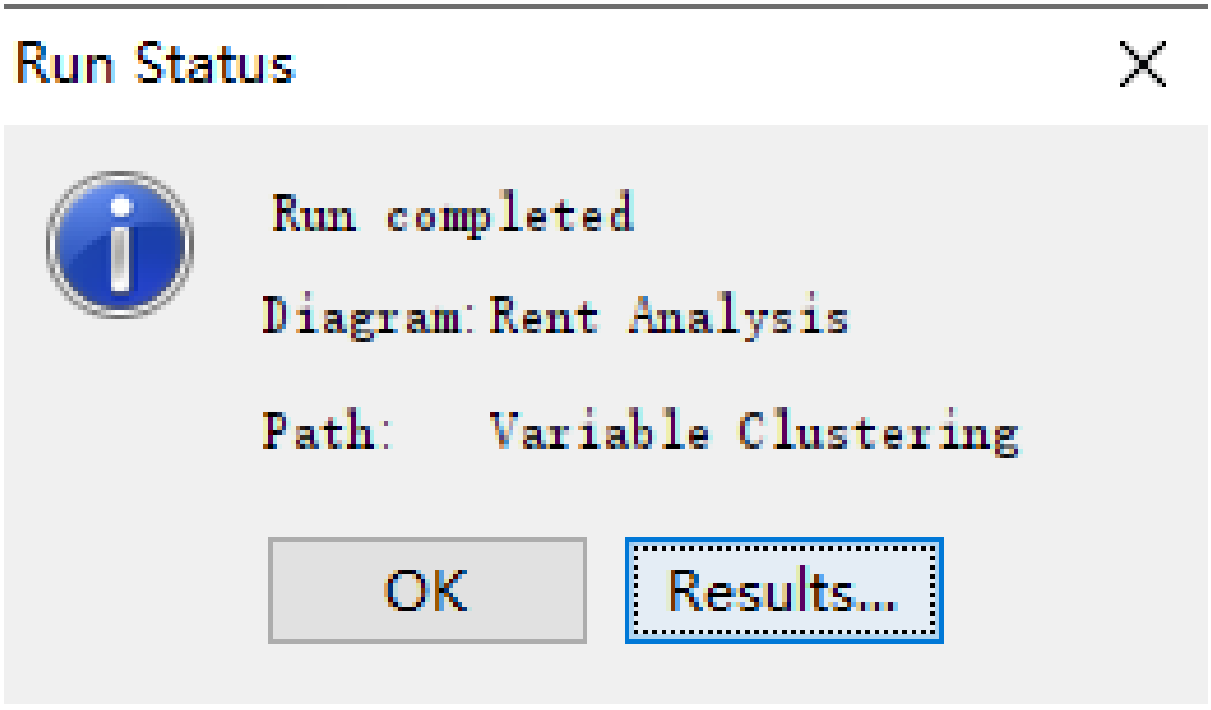




To understand the correlation among the variables, the Variable Clustering node is added.

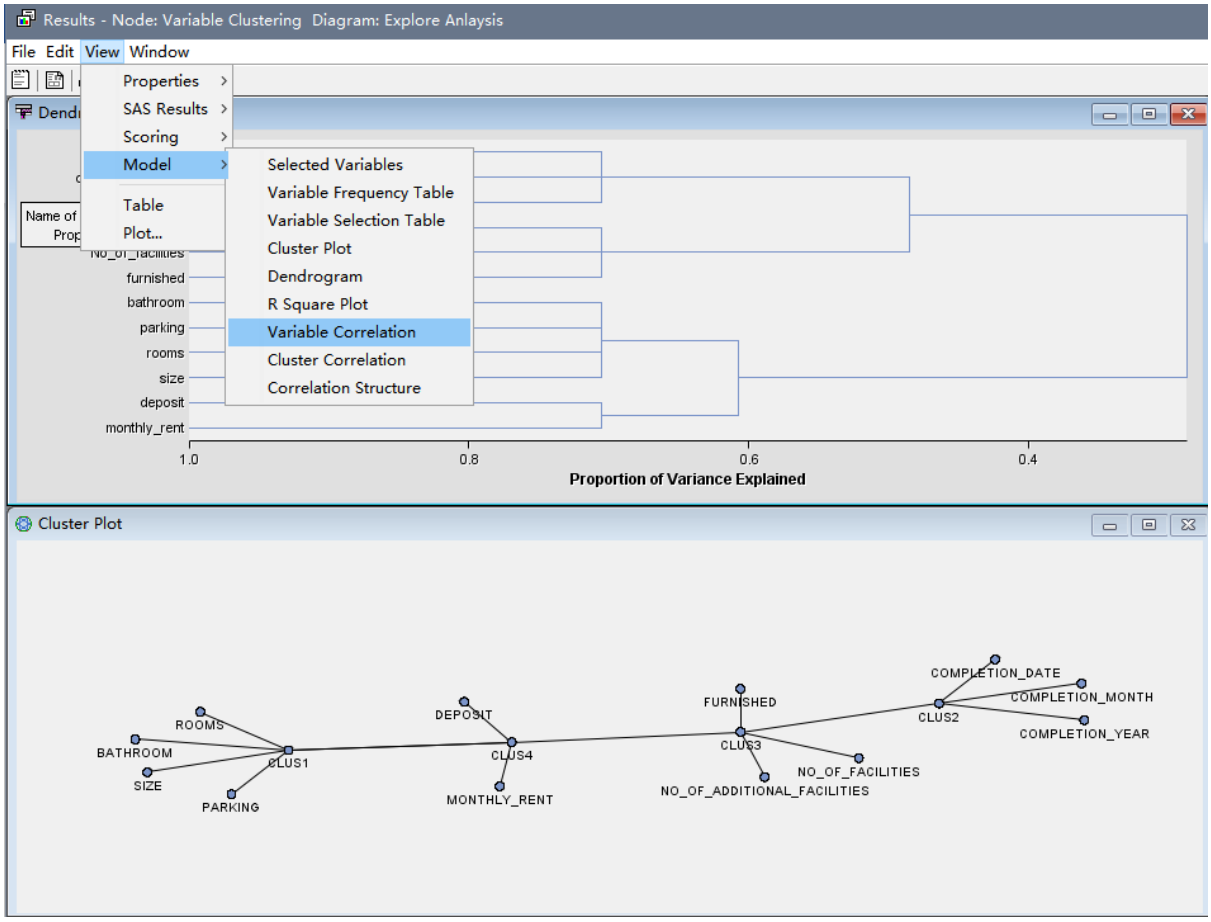
Select Add Node > Explore > Variable Clustering



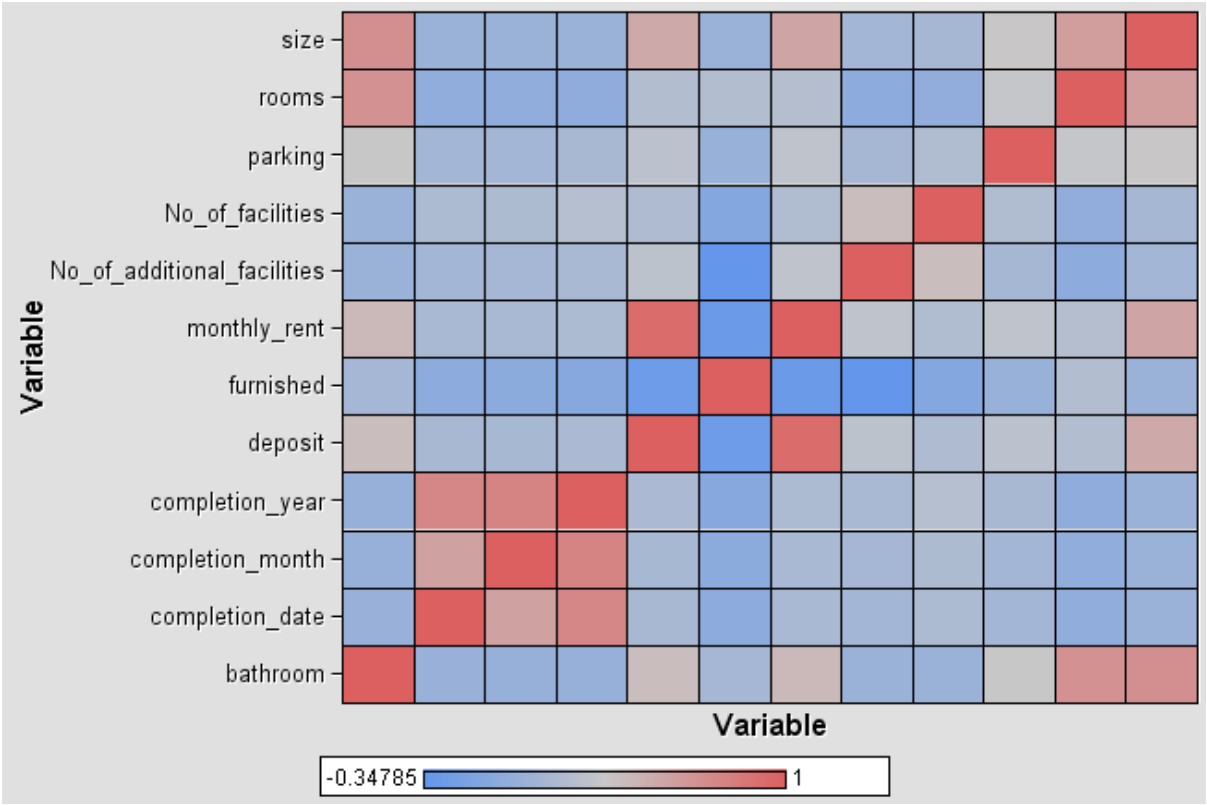


When viewing the results on variable correlation, a correlation matrix will be shown.

Select Run > Results... > View > Model > Variable Correlation

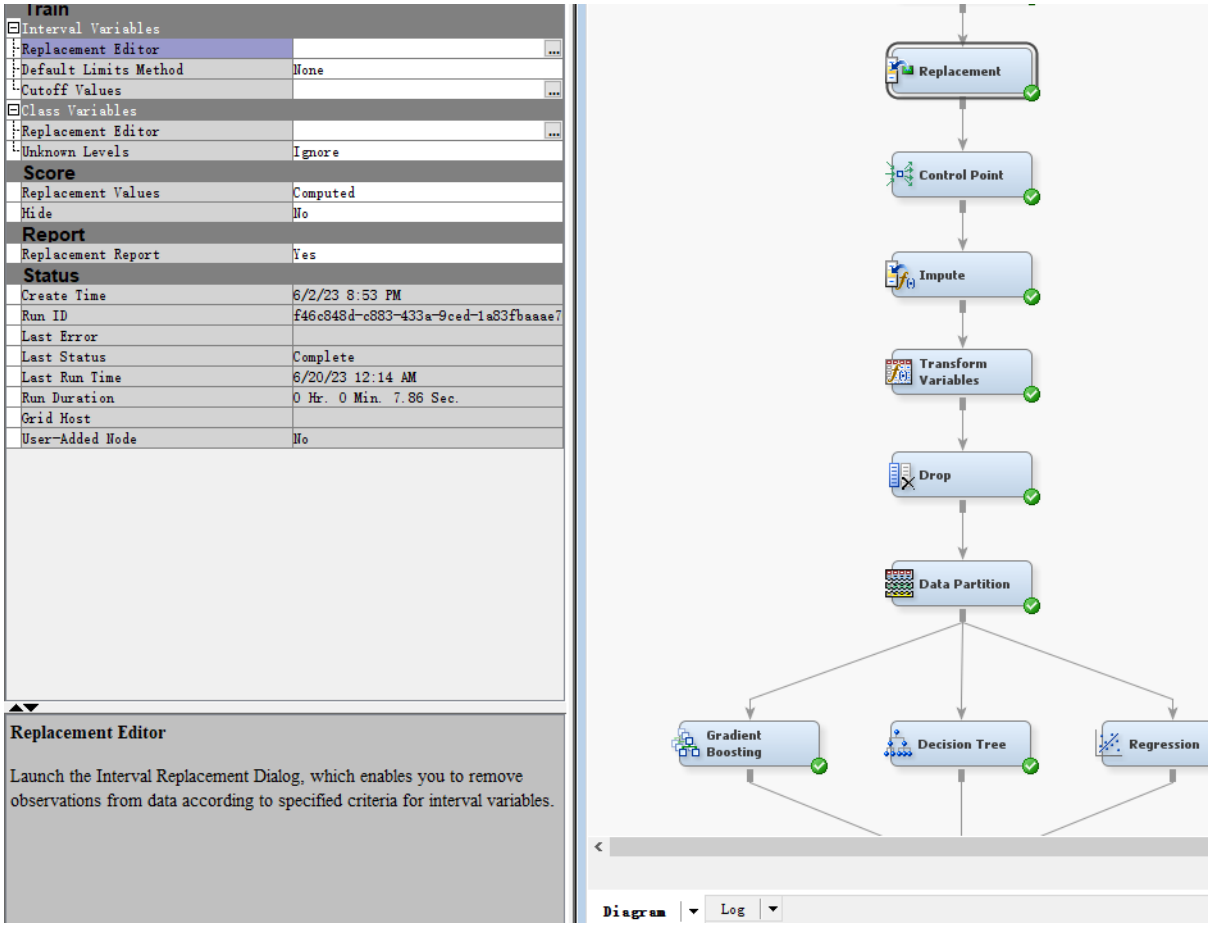


To view the values used in deriving the correlation matrix, click on the Table icon on the top left of the result screen. A table should be displayed showing the correlation values.

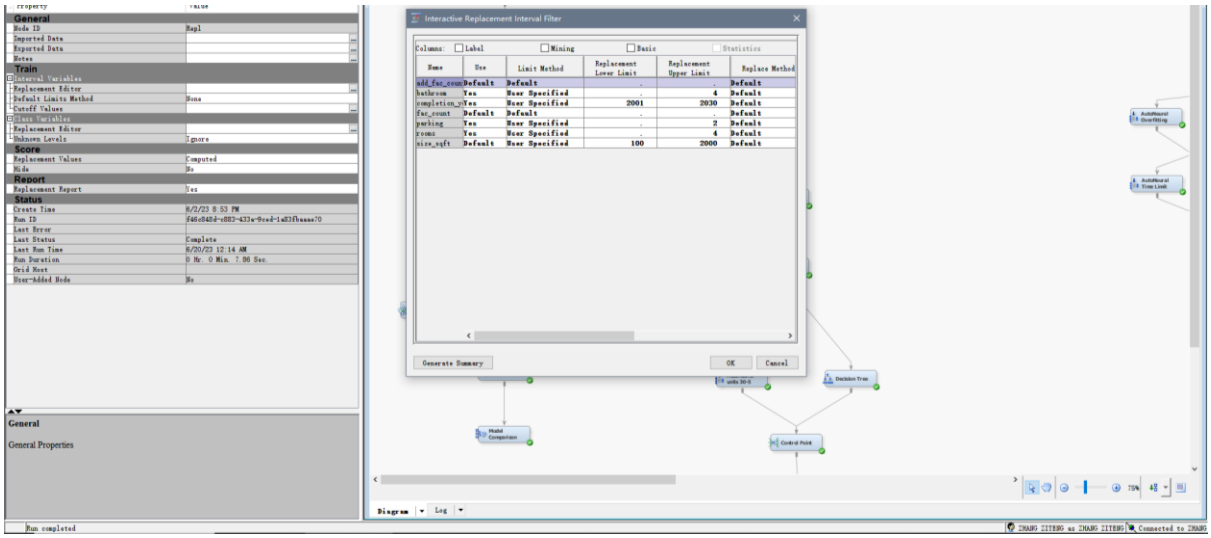

size
rooms
parking
No_of_facilities
No_of_additional_facilities
monthly_rent
furnished
deposit
completion_year
completion_month
completion_date
bathroom
Variable
Variable
-0.34785
1

8.4. Modify

Replacement

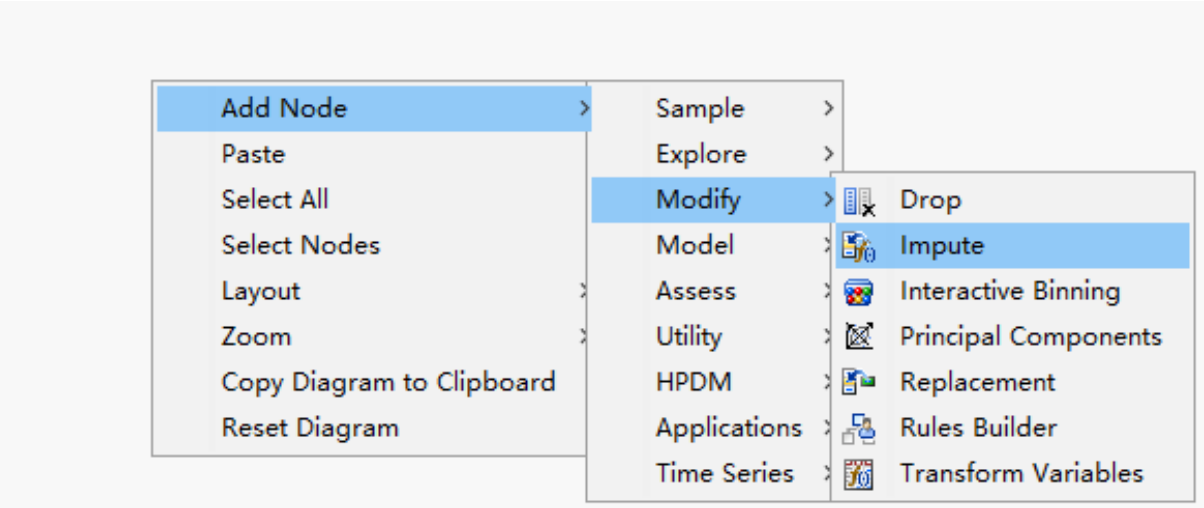


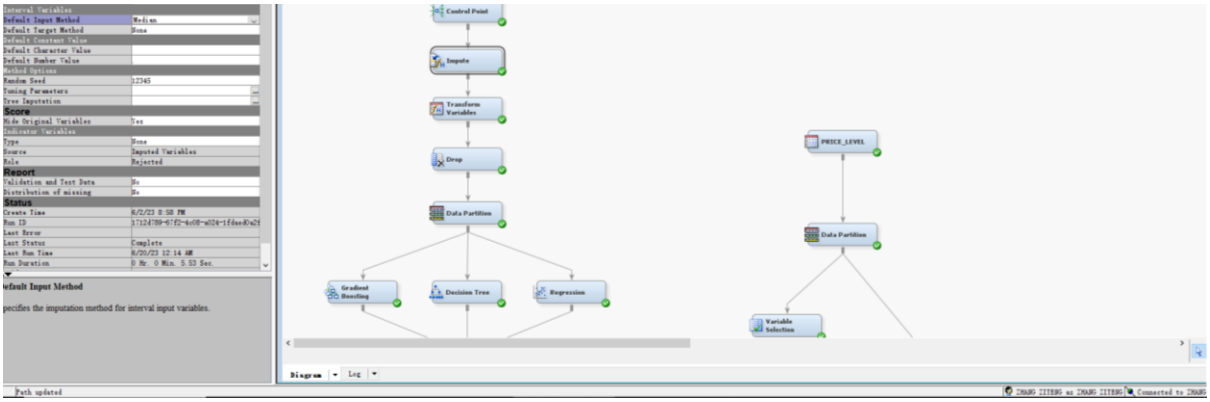
By clicking the interval replacement editor, the system can use replacement to filter the range of interval variables.



The graph shows the filter range setting.

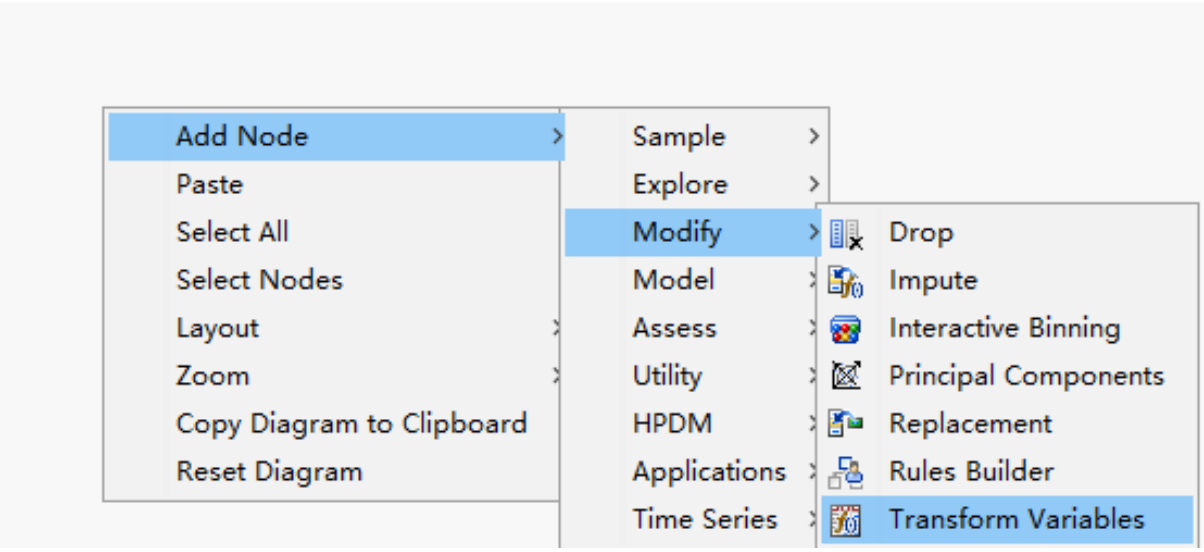
Impute

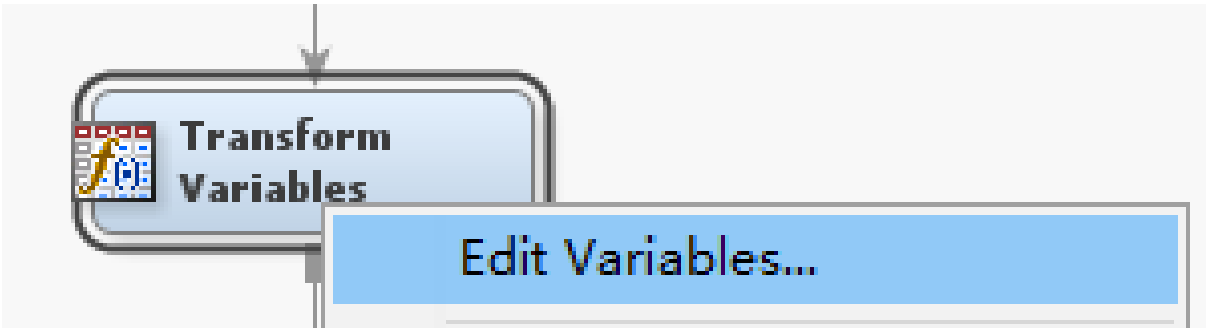


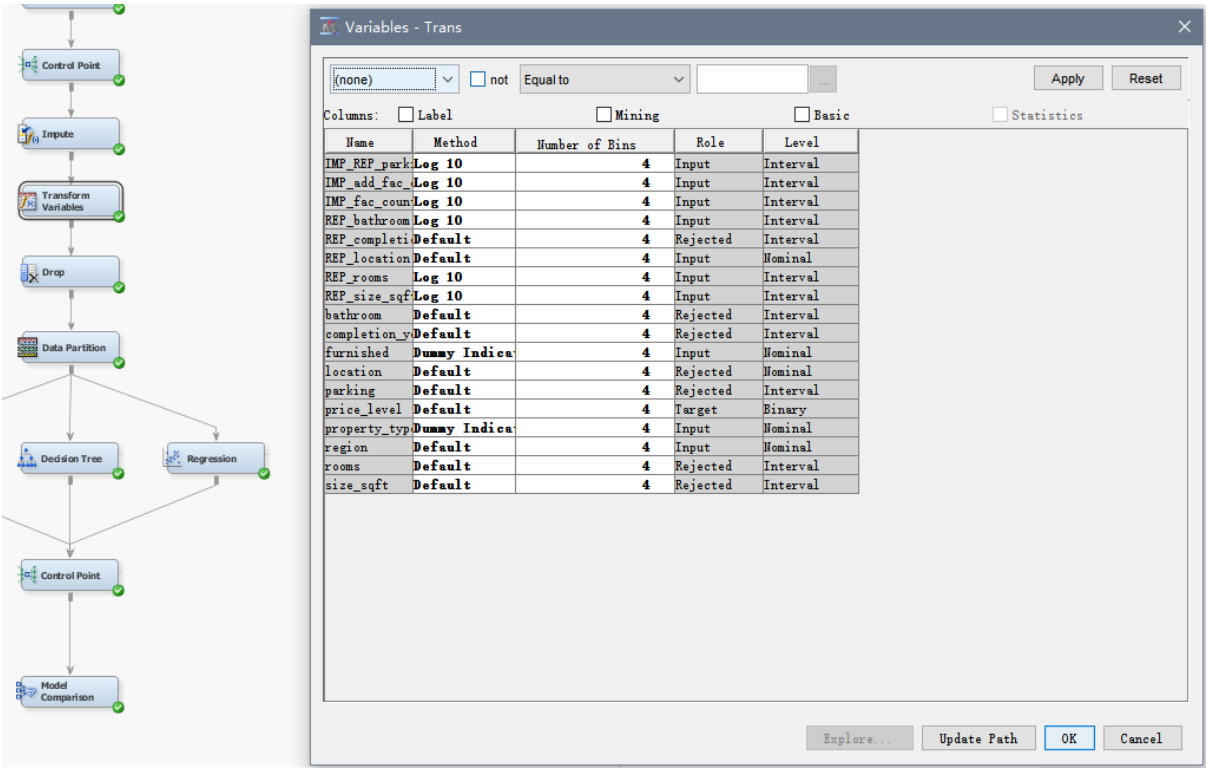


Using impute to process NA value. Here using median to fill the NA value.

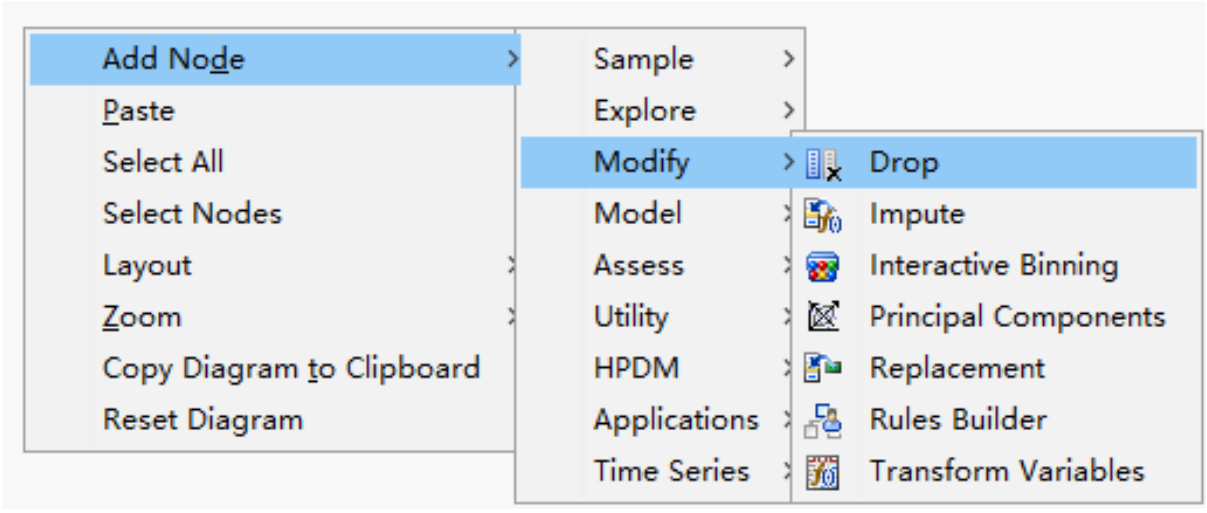
Transform

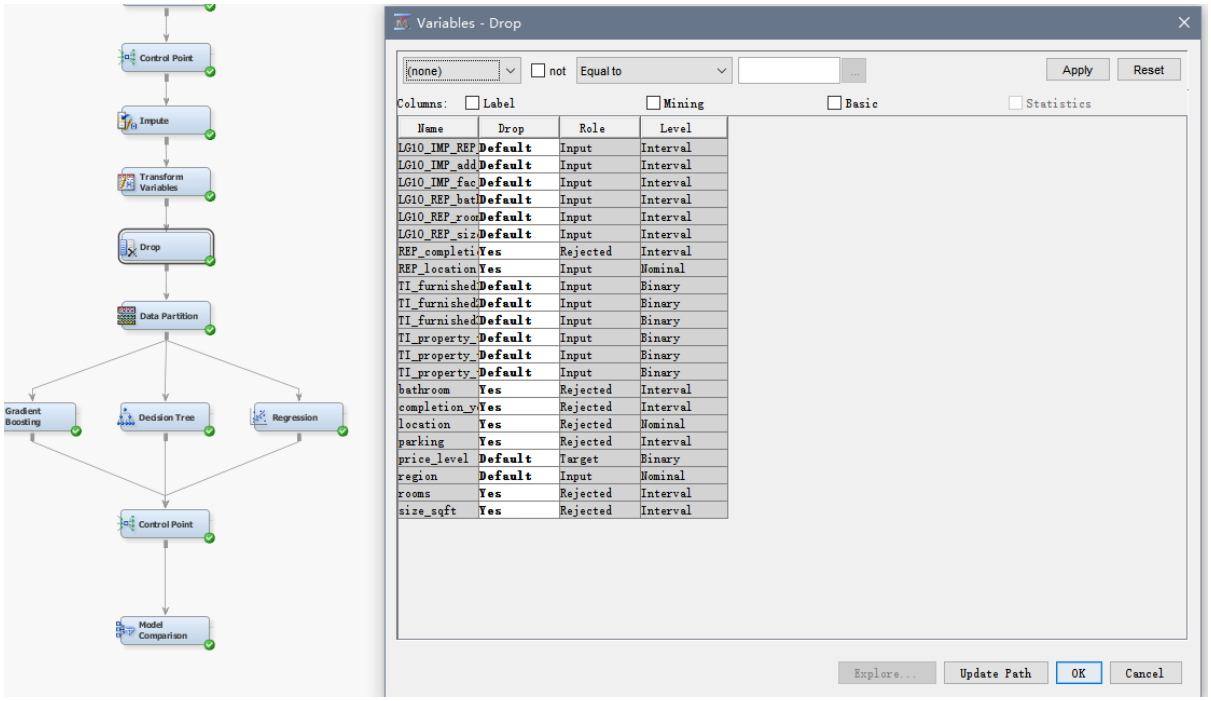



Transform Variables
Edit Variables...

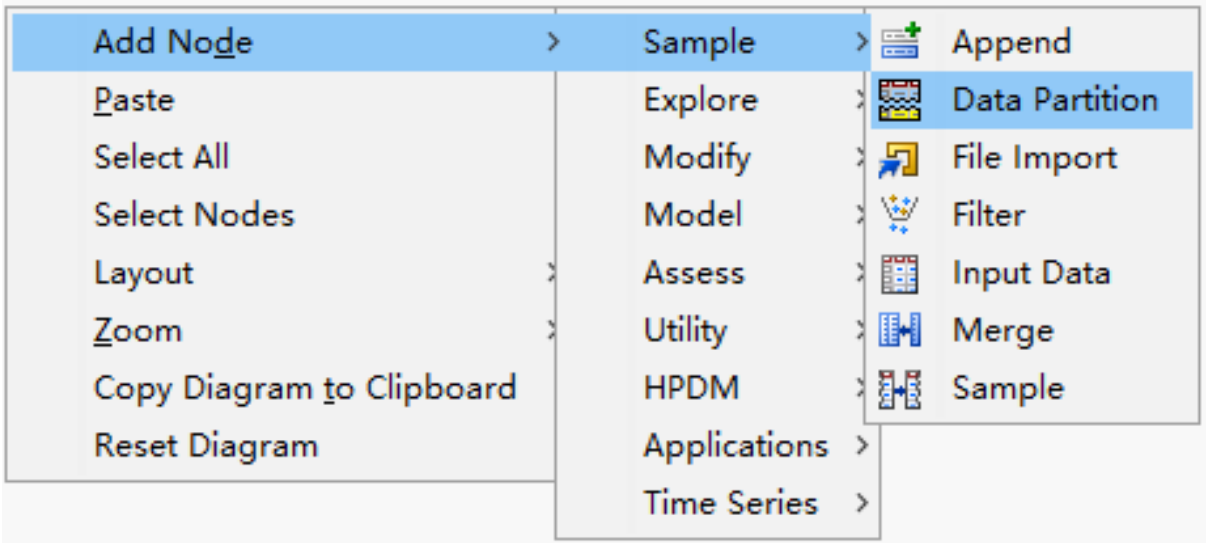

Control Point
Impute
Transform Variables
Drop
Data Partition
Decision Tree
Regression
Control Point
Model Comparison
Variables - Trans
(none)
not
Equal to
Apply
Reset
Columns:
Label
Mining
Basic
Statistics
Name
Method
Number of Bins
Role
Level
IMP_REP_park
Log 10
4
Input
Interval
IMP_add_fac_
Log 10
4
Input
Interval
IMP_fac_coun
Log 10
4
Input
Interval
REP_bathroom
Log 10
4
Input
Interval
REP_completi
Default
4
Rejected
Interval
REP_location
Default
4
Input
Nominal
REP_rooms
Log 10
4
Input
Interval
REP_size_sqf
Log 10
4
Input
Interval
bathroom
Default
4
Rejected
Interval
completion_y
Default
4
Rejected
Interval
furnished
Dummy Indica
4
Input
Nominal
location
Default
4
Rejected
Nominal
parking
Default
4
Rejected
Interval
price_level
Default
4
Target
Binary
property_typ
Dummy Indica
4
Input
Nominal
region
Default
4
Input
Nominal
rooms
Default
4
Rejected
Interval
size_sqft
Default
4
Rejected
Interval
Explore...
Update Path
OK
Cancel

Drop

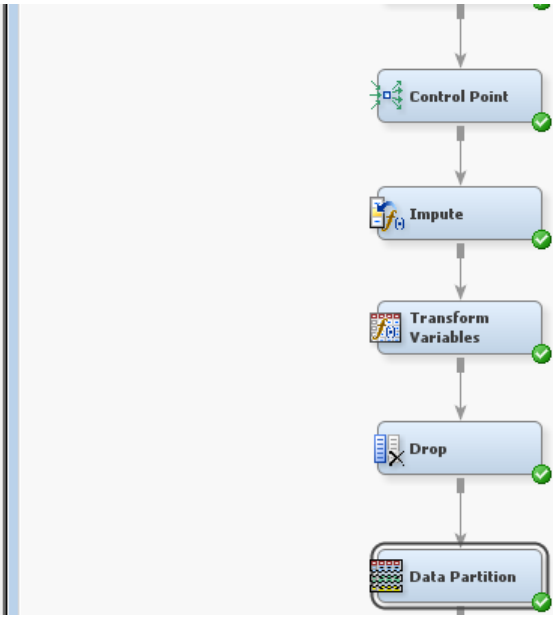

Add Node
Paste
Select All
Select Nodes
Layout
Zoom
Copy Diagram to Clipboard
Reset Diagram
Sample
Explore
Modify
Model
Assess
Utility
HPDM
Applications
Time Series
Drop
Impute
Interactive Binning
Principal Components
Replacement
Rules Builder
Transform Variables



Data partition

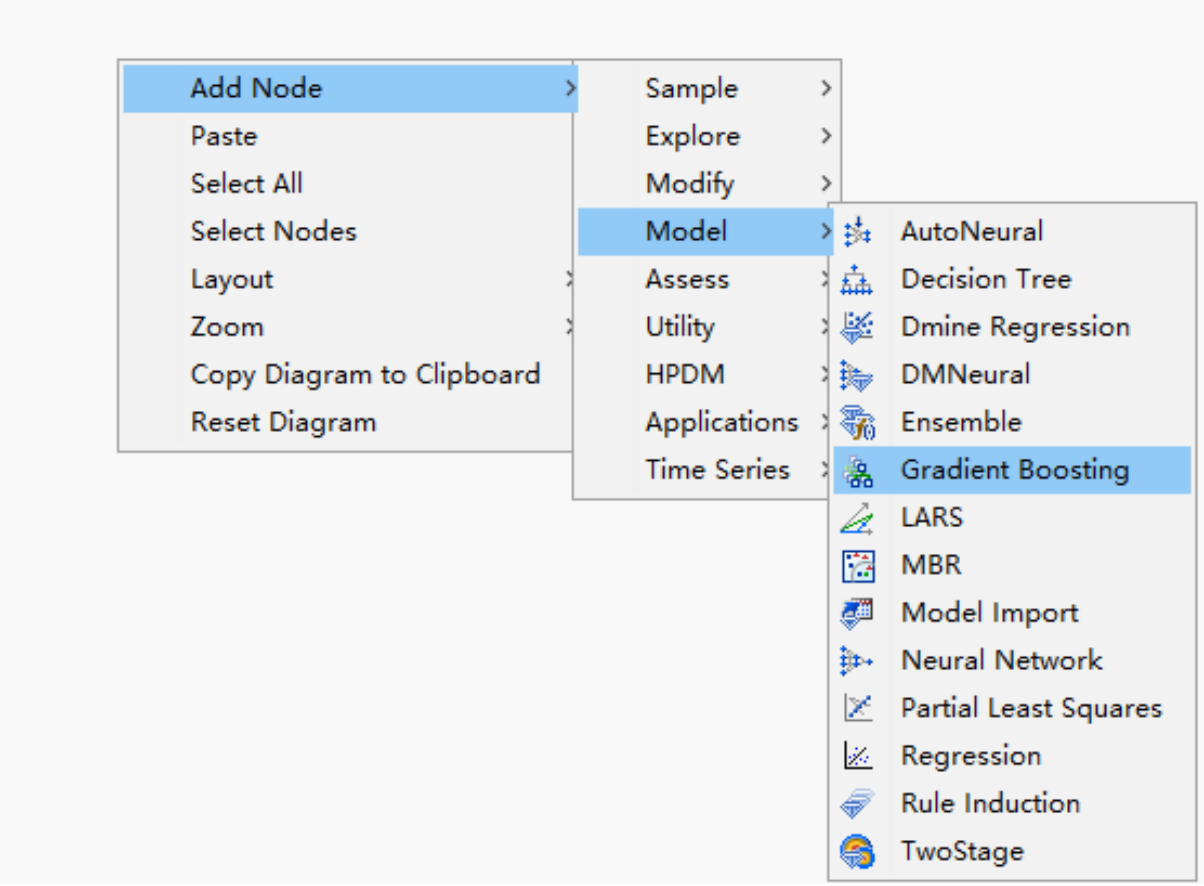


Data Set Allocations	
Training	50.0
Validation	50.0
Test	0.0
Report	
Interval Targets	Yes
Class Targets	Yes
Status	
Create Time	5/31/23 10:08 PM
Run ID	958c83d2-69c5-4b19-b8da-61571ec1799
Last Error	
Last Status	Complete
Last Run Time	6/20/23 1:20 AM
Run Duration	0 Hr. 0 Min. 5.28 Sec.
Grid Host	
User-Added Node	No



8.5. Model

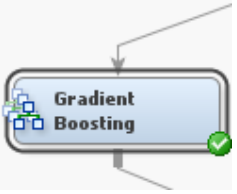
Gradient boosting

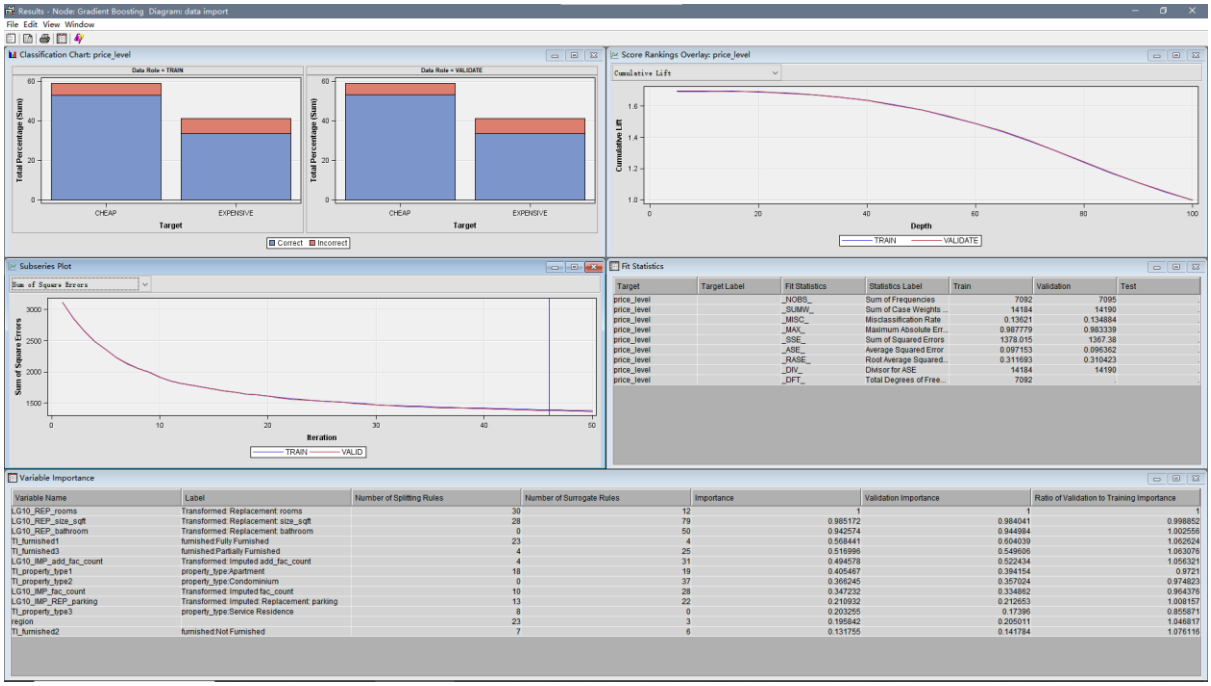


General	
Node ID	Boost
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Series Options	
N Iterations	50
Seed	12345
Shrinkage	0.1
Train Proportion	60
Splitting Rule	
Huber M-Regression	No
Maximum Branch	2
Maximum Depth	10
Minimum Categorical Size	5
Reuse Variable	1
Categorical Bins	30
Interval Bins	100
Missing Values	Use in search
Performance	Disk
Node	
Leaf Fraction	0.1
Number of Surrogate Rules	2
Split Size	.
Split Search	
Exhaustive	5000
Node Sample	20000
Subtree	
Assessment Measure	Decision
Score	
Subseries	Best Assessment Value
Number of Iterations	1
Create H Statistic	No
Variable Selection	Yes
Report	
Observation Based Importance	No
Number Single Var Importance	5

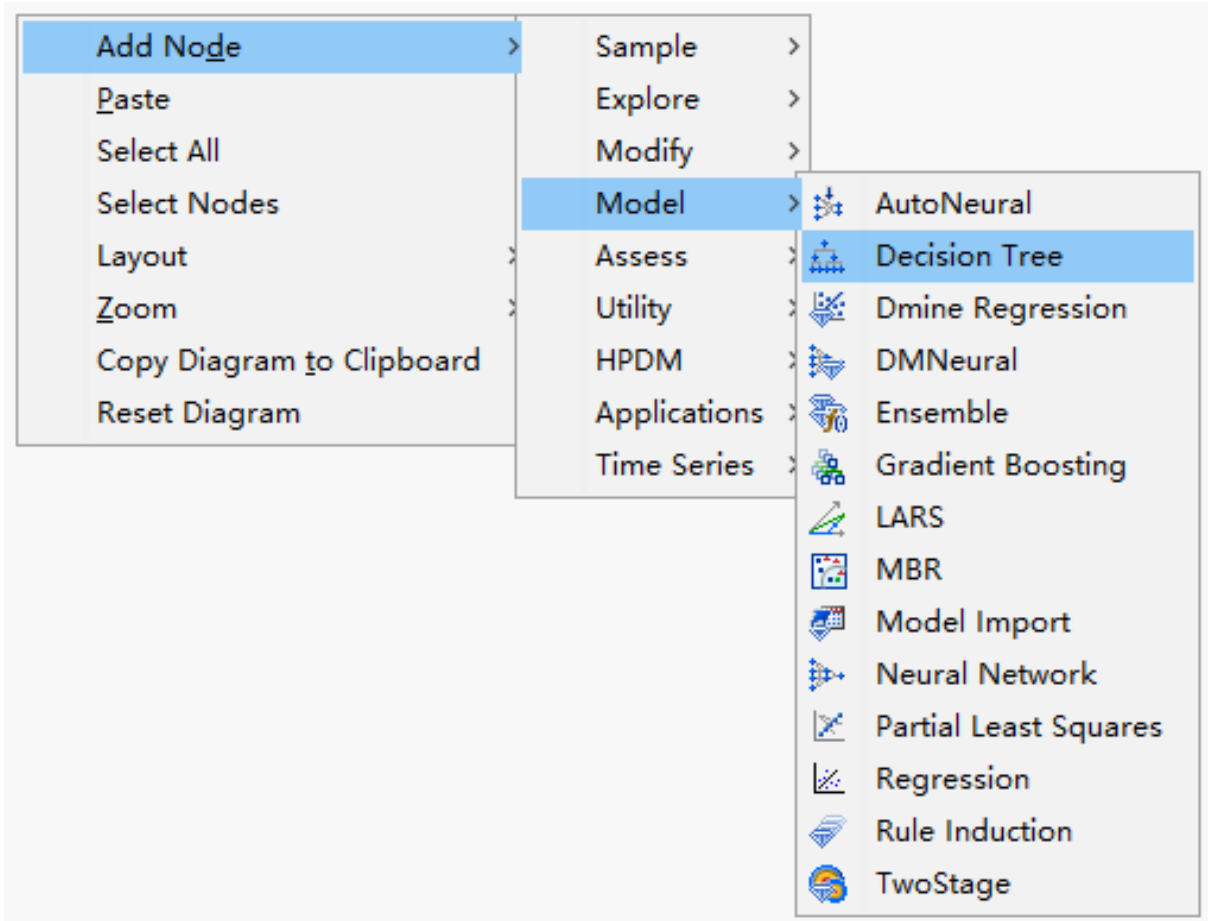
General

General Properties





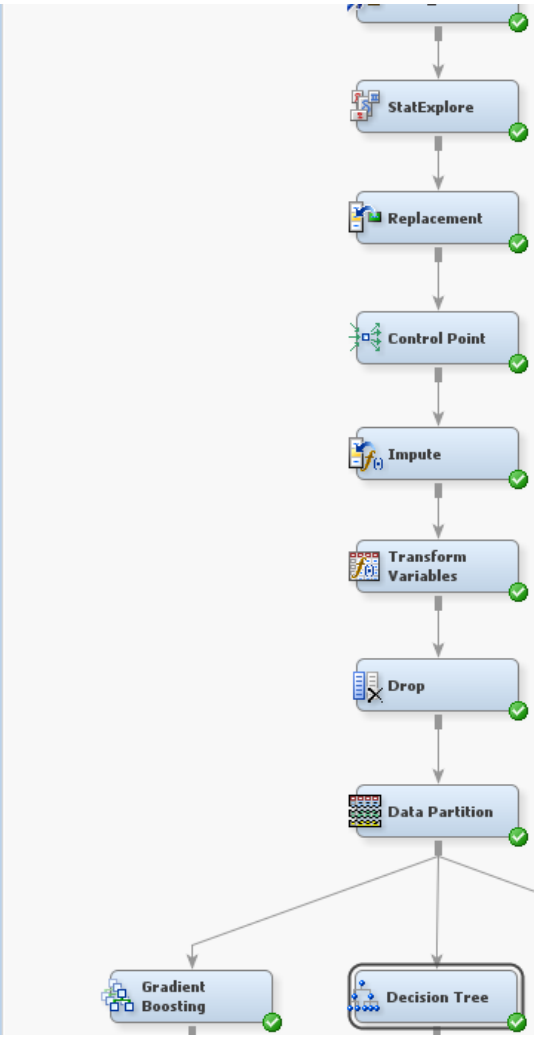
Decision Tree

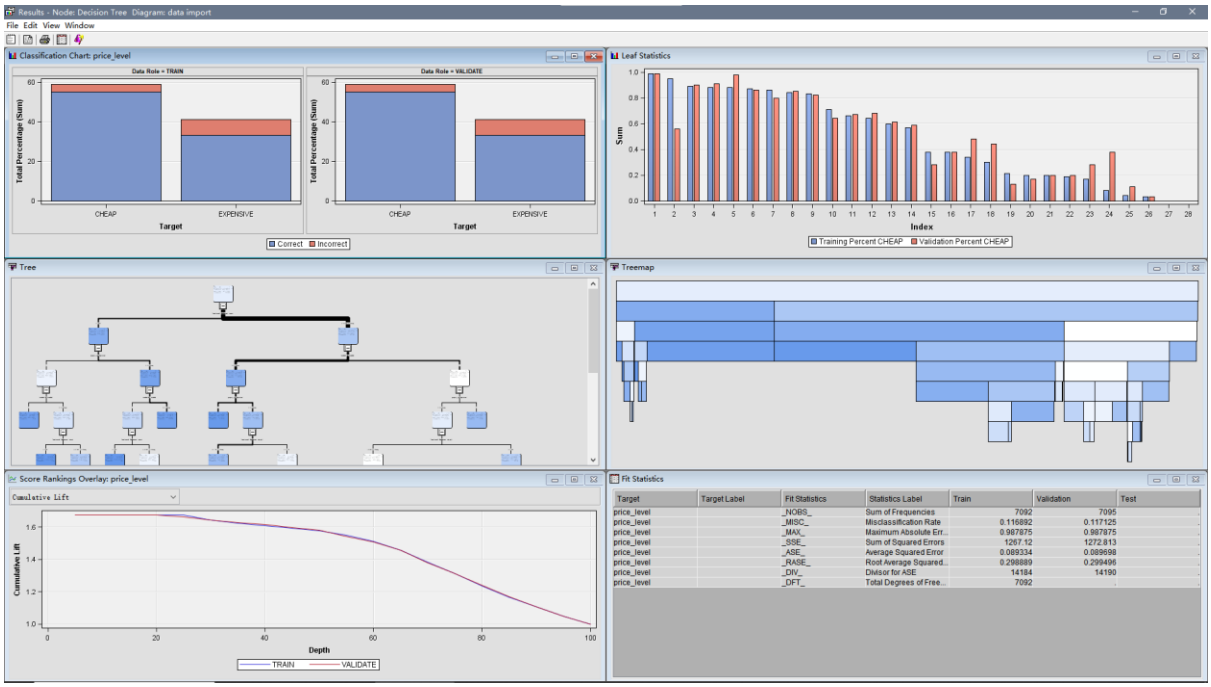


General	
Node ID	Tree3
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Interactive	
Import Tree Model	No
Tree Model Data Set	
Use Frozen Tree	No
Use Multiple Targets	No
Splitting Rule	
Interval Target Criterion	ProbF
Nominal Target Criterion	ProbChisq
Ordinal Target Criterion	Entropy
Significance 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	8
Number of Rules	5
Number of Surrogate Rules	4
Split Size	.
Split Search	
Use Decisions	No
Use Priors	No
Exhaustive	5000
Node Sample	20000
Subtree	
Method	Assessment
Number of Leaves	1
Assessment Measure	Decision
Assessment Fraction	0.25
Cross Validation	
Perform Cross Validation	No

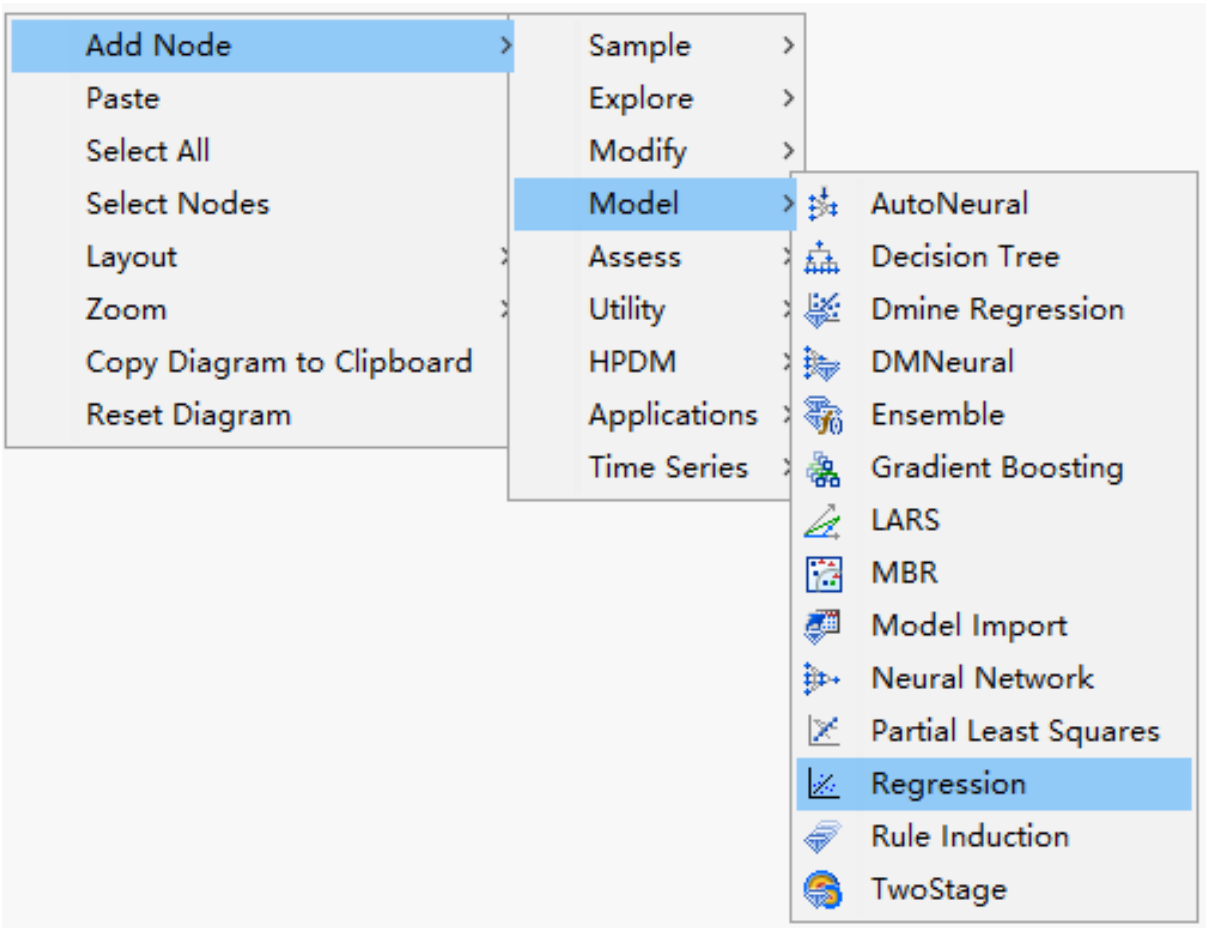
General

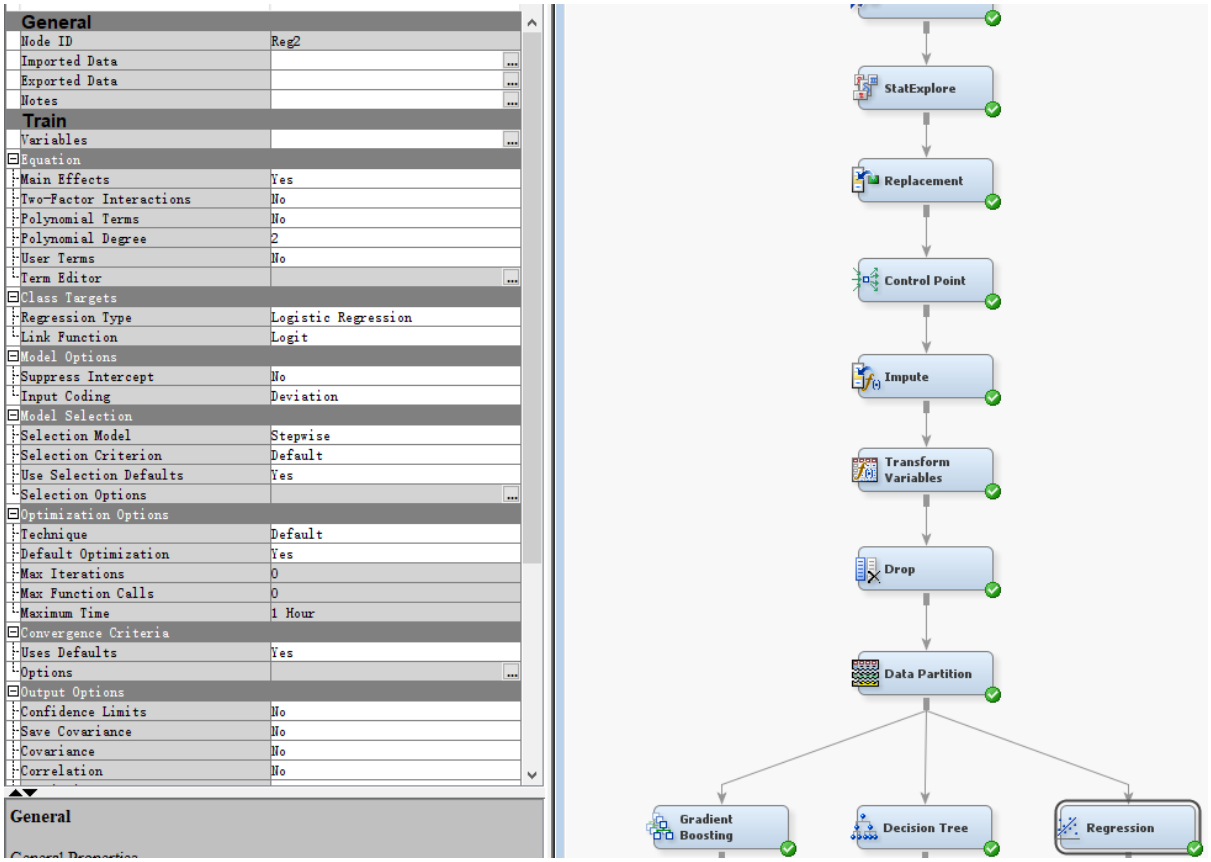
General Properties

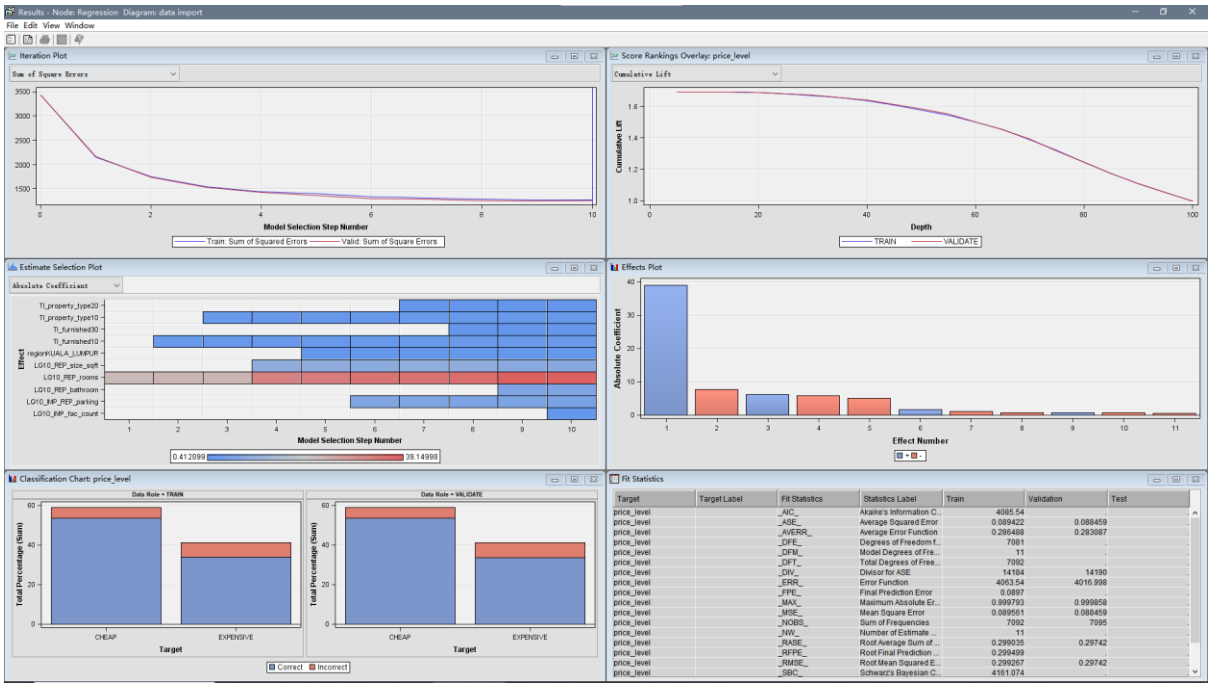




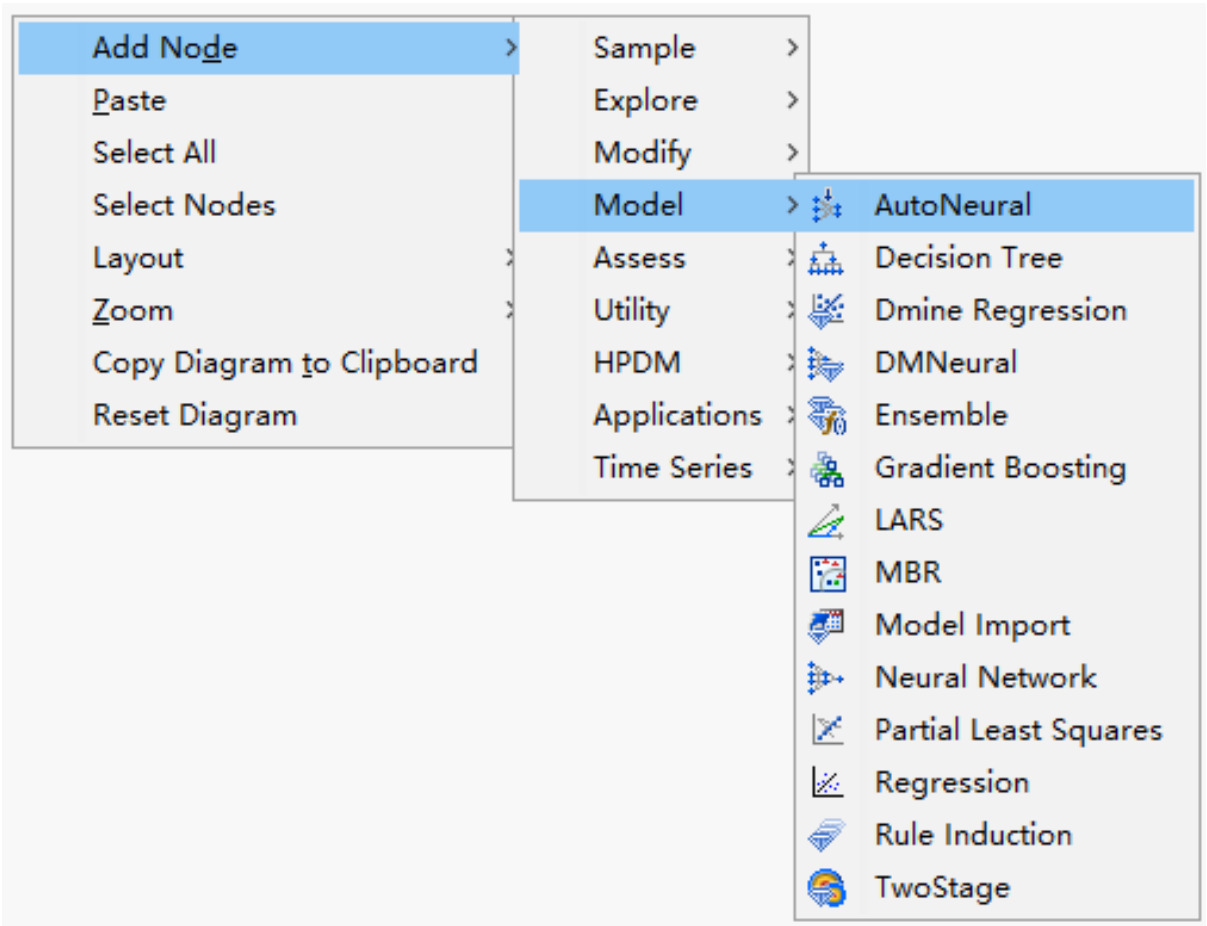
Logistic regression

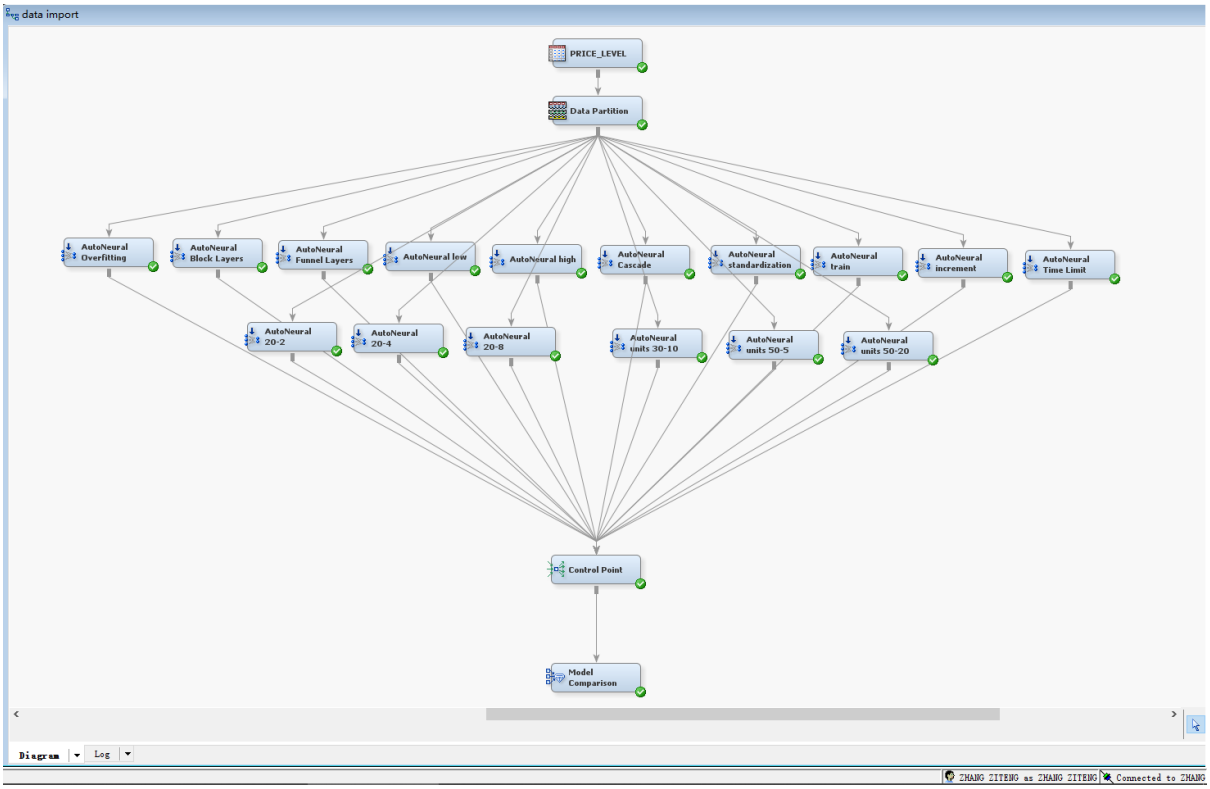


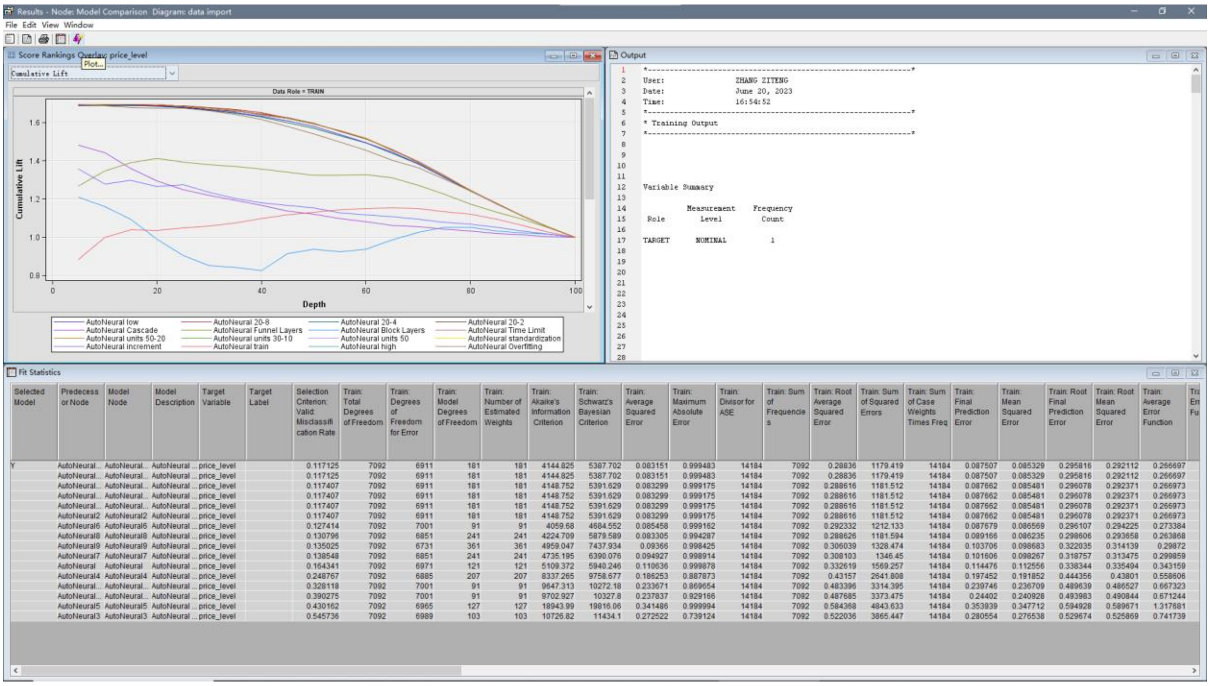

General
Node ID
Reg2
Imported Data
Exported Data
Notes
Train
Variables
Equation
Main Effects
Yes
Two-Factor Interactions
No
Polynomial Terms
No
Polynomial Degree
2
User Terms
No
Term Editor
Class Targets
Regression Type
Logistic Regression
Link Function
Logit
Model Options
Suppress Intercept
No
Input Coding
Deviation
Model Selection
Selection Model
Stepwise
Selection Criterion
Default
Use Selection Defaults
Yes
Selection Options
Optimization Options
Technique
Default
Default Optimization
Yes
Max Iterations
0
Max Function Calls
0
Maximum Time
1 Hour
Convergence Criteria
Uses Defaults
Yes
Options
Output Options
Confidence Limits
No
Save Covariance
No
Covariance
No
Correlation
No
General
StatExplore
Replacement
Control Point
Impute
Transform Variables
Drop
Data Partition
Gradient Boosting
Decision Tree
Regression

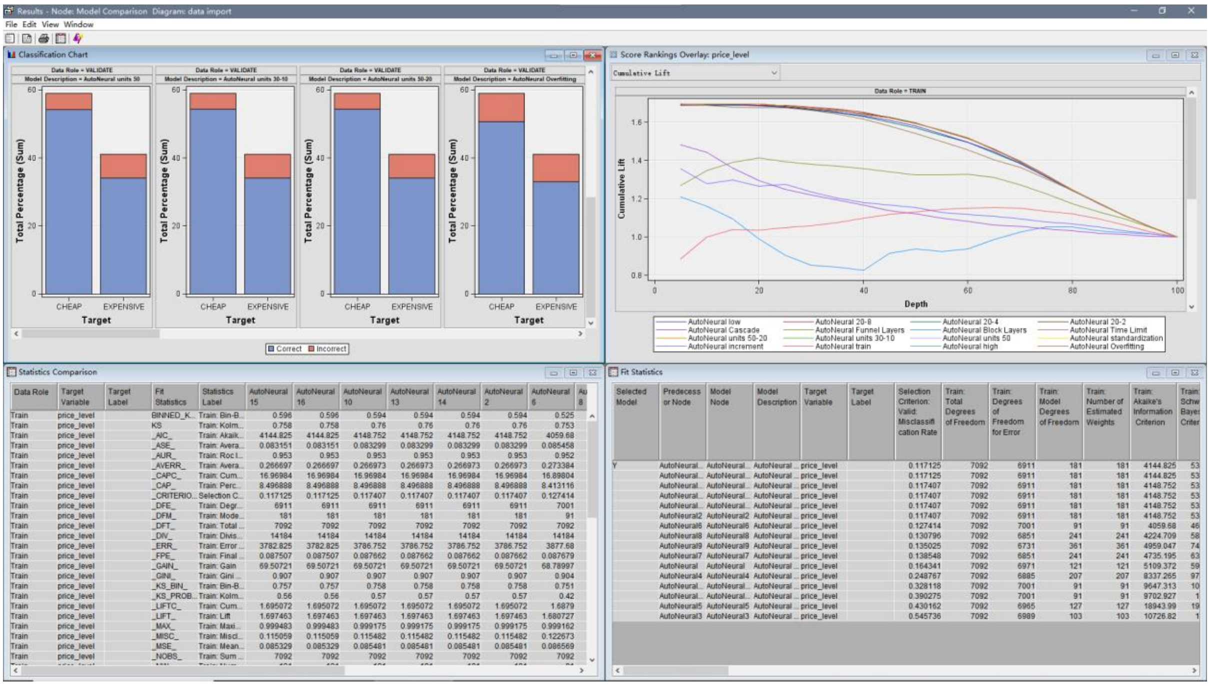

Results - Node: Regression Diagram: data import
File Edit View Window
Iteration Plot
Sum of Square Errors
Model Selection Step Number
Train: Sum of Squared Errors
Valid: Sum of Square Errors
Score Rankings Overlay: price_level
Cumulative Lift
Depth
TRAIN
VALIDATE
Estimate Selection Plot
Absolute Coefficient
Effect
Model Selection Step Number
Effects Plot
Absolute Coefficient
Effect Number
Classification Chart: price_level
Data Role = TRAIN
Data Role = VALIDATE
Total Percentage (Sum)
CHEAP
EXPENSIVE
Target
Correct
Incorrect
Fit Statistics
Target
Target Label
Fit Statistics
Statistics Label
Train
Validation
Test

Neural network comparation



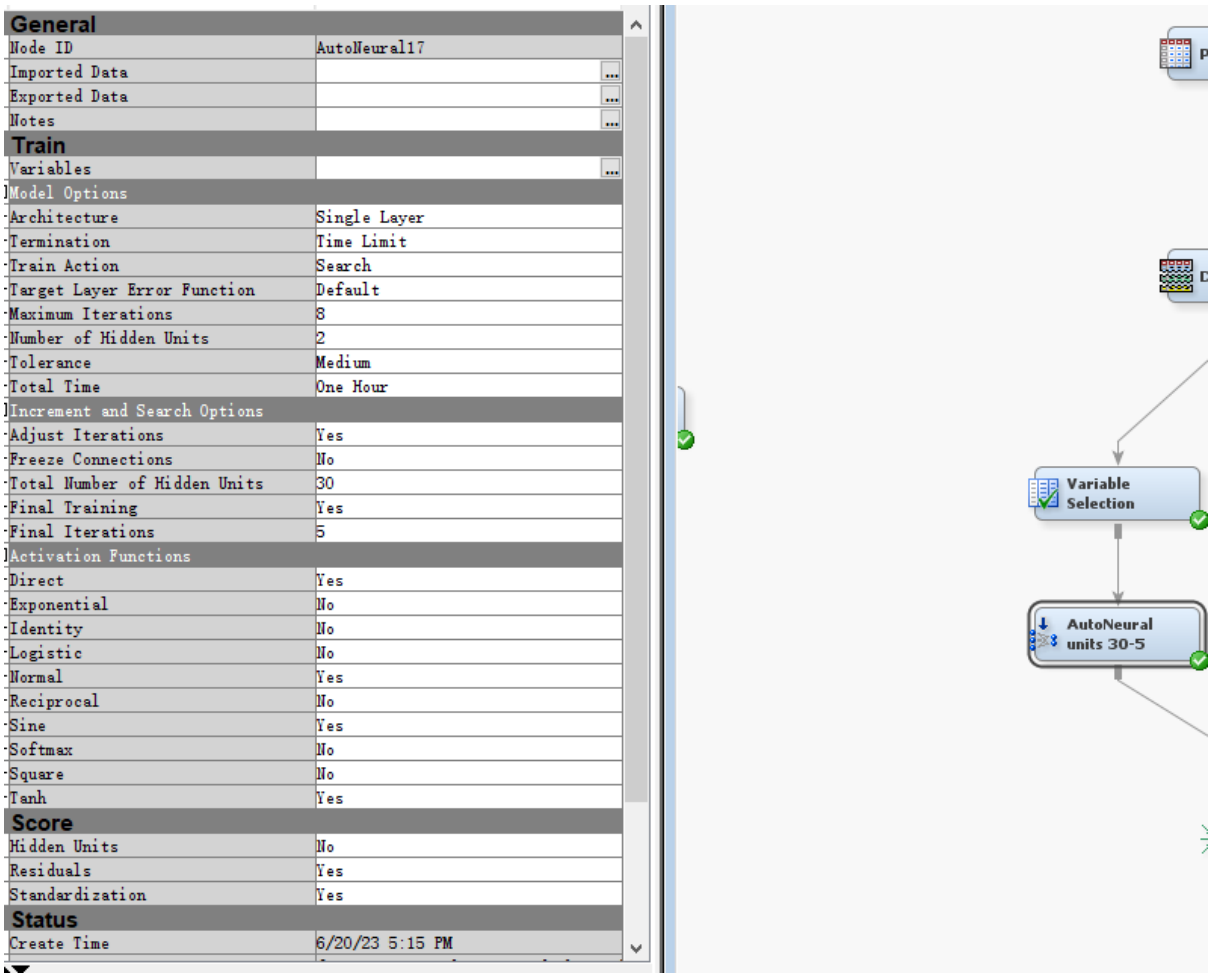


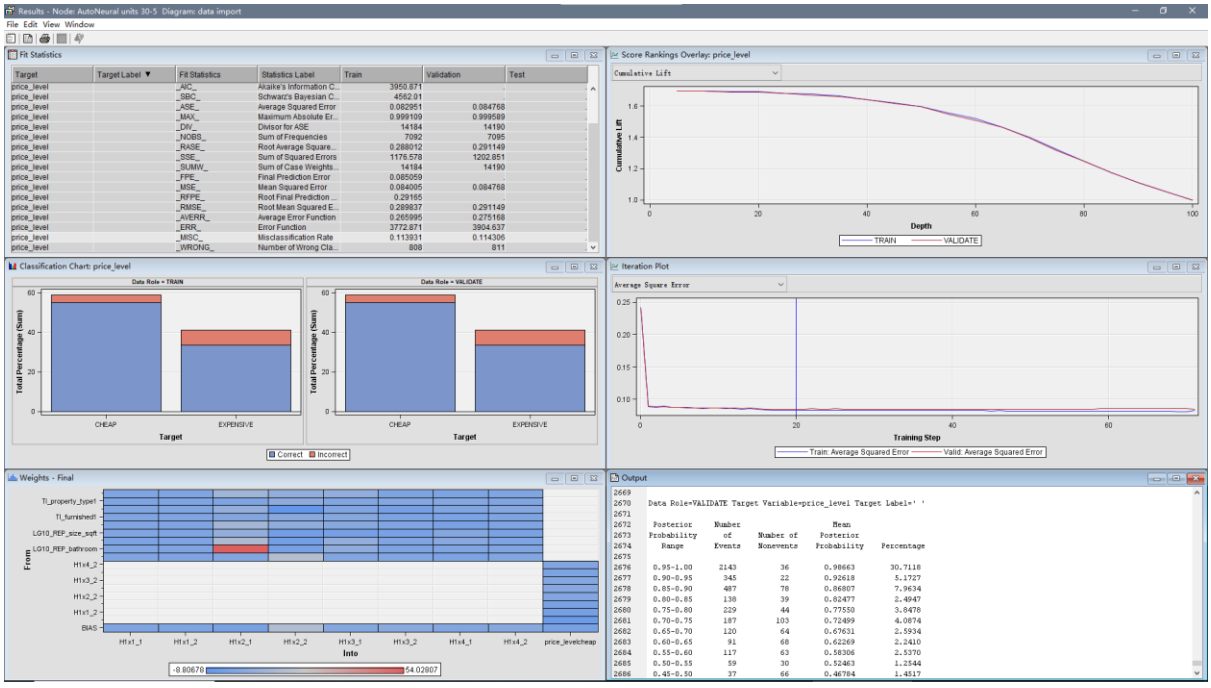

Results - Node: Model Comparison Diagram: data import
Score Rankings Overlay: price_level
Output
Fit Statistics


Results - Node: Model Comparison Diagram: data import
Classification Chart
Score Rankings Overlay: price_level
Statistics Comparison
Fit Statistics

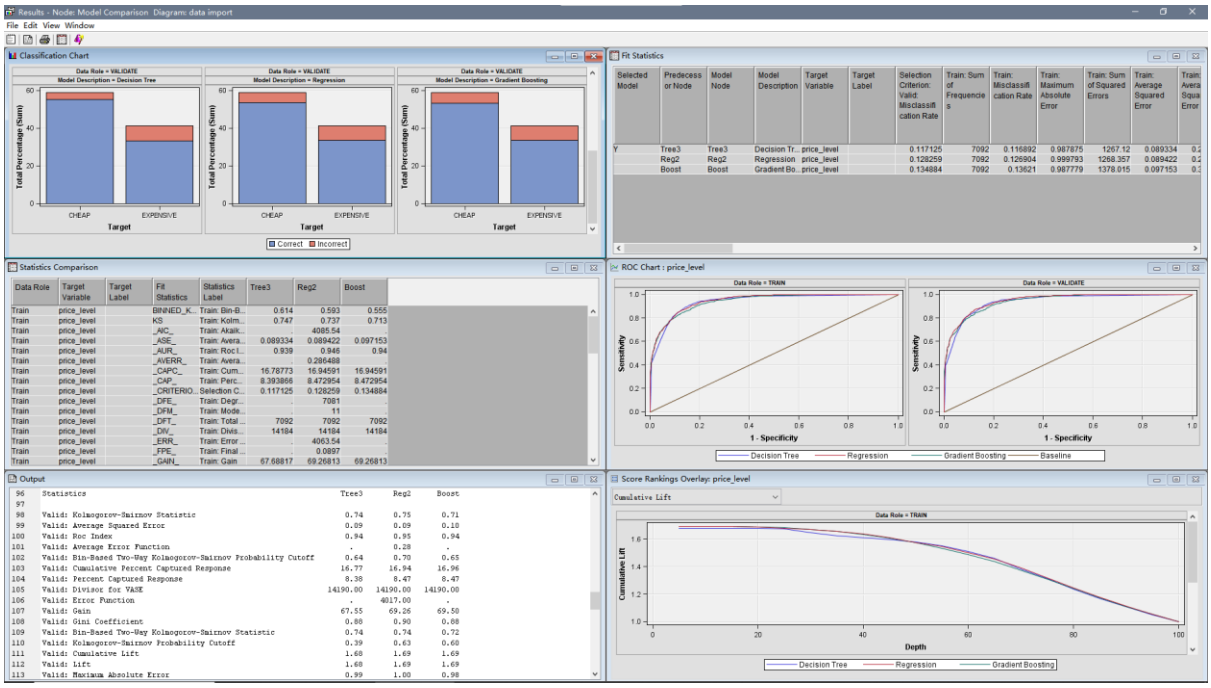
Best Neural network:

General	
Node ID	AutoNeural17
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Model Options	
Architecture	Single Layer
Termination	Time Limit
Train Action	Search
Target Layer Error Function	Default
Maximum Iterations	8
Number of Hidden Units	2
Tolerance	Medium
Total Time	One Hour
Increment and Search Options	
Adjust Iterations	Yes
Freeze Connections	No
Total Number of Hidden Units	30
Final Training	Yes
Final Iterations	5
Activation Functions	
Direct	Yes
Exponential	No
Identity	No
Logistic	No
Normal	Yes
Reciprocal	No
Sine	Yes
Softmax	No
Square	No
Tanh	Yes
Score	
Hidden Units	No
Residuals	Yes
Standardization	Yes
Status	
Create Time	6/20/23 5:15 PM

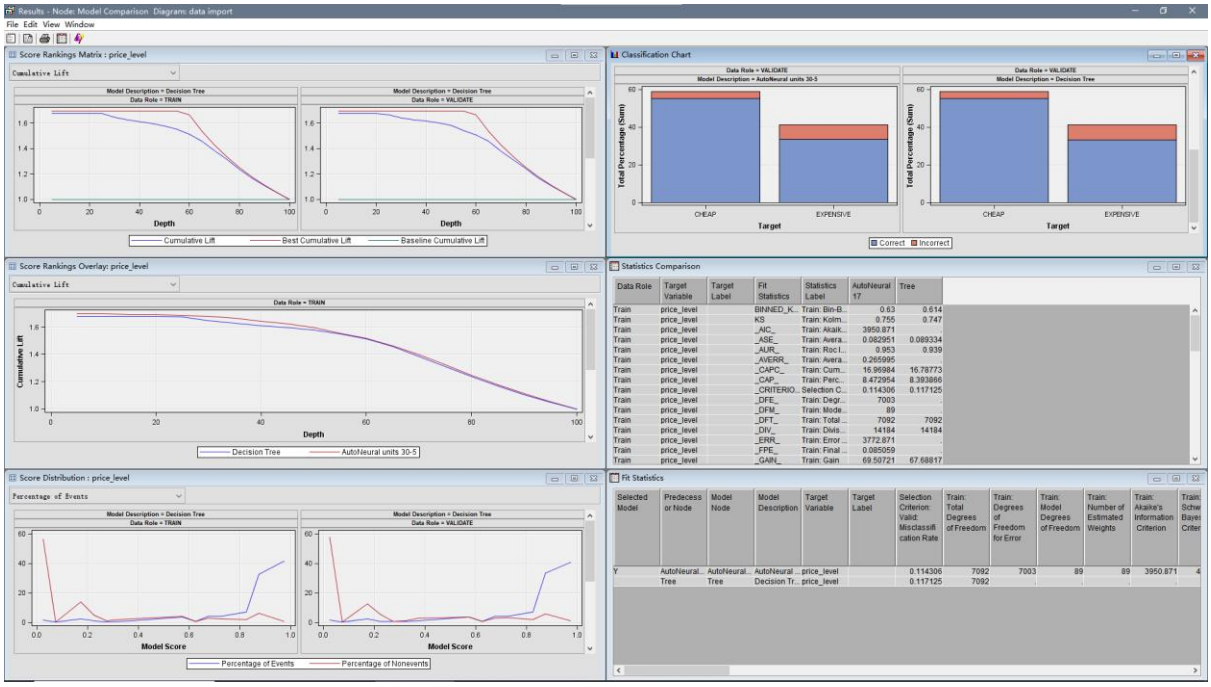


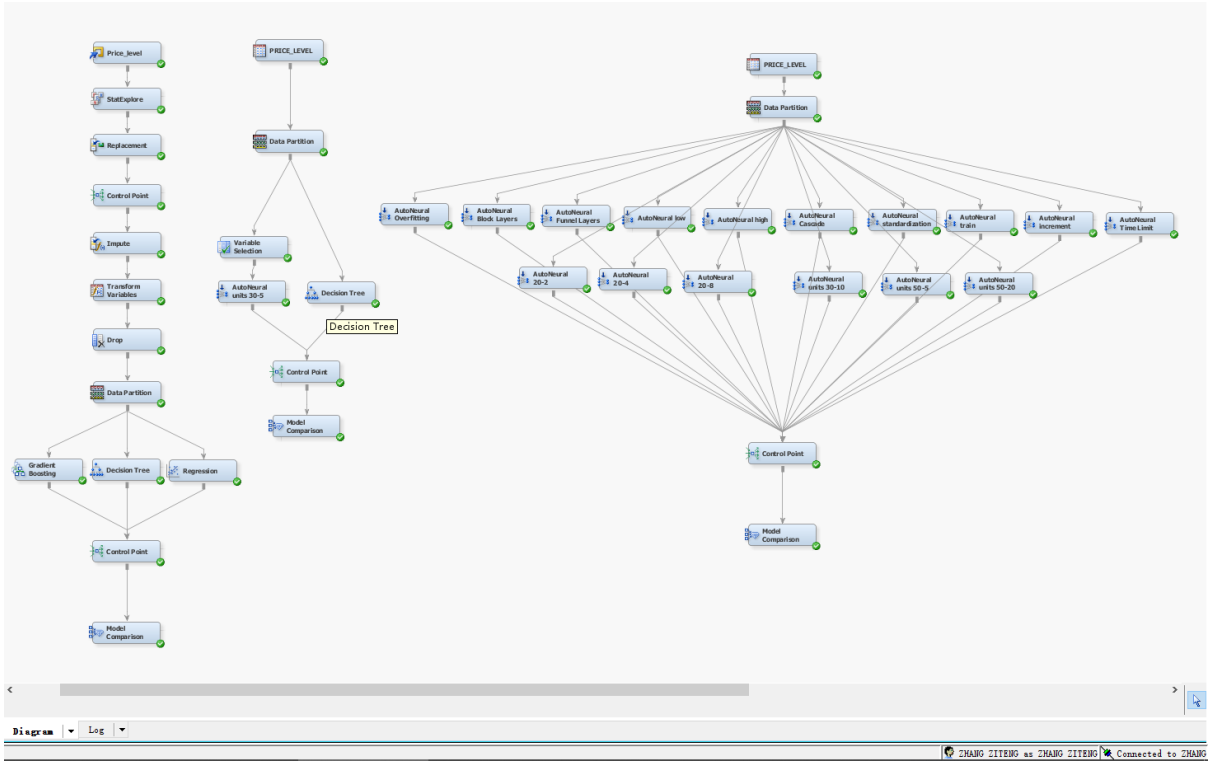


Logistic vs Decision tree vs Gradient boosting – winner is DT



Decision tree VS neural network





Regression model