Faculty of Computer Science & Information Technology

WIE3007 Data Mining and Warehousing

Semester 1, 2023/2024

Group Project - Project Report

Lecturer

Prof. Dr. Teh Ying Wah

Prepared By

Name	Matric Number
Chai Nam Chi	U2005421/1
Cheong Hui Ting	U2005292/1
Hong Jia Herng	U2005313/1
Lee Hui Xin	U2005353/1
Liewn Wan Chyi	U2005418/1

Table of Contents

1.0 Introduction to the Selected Dataset

A dataset called Adult Data Set (Census Income Dataset) is downloaded from Kaggle (https://www.kaggle.com/datasets/kritidoneria/adultdatasetxai). The dataset is US Census Data extracted from the 1994 census data donated to UC Irvine's Machine Learning Repository.

2.0 Understanding the Dataset

2.1 Loading the Dataset

A dataset metadata file is provided by the website which helped us to build fundamentals understanding on the dataset. The dataset consists of 32,561 instances across 15 variables. The dataset is used for predicting income levels of US citizens based on other independent variables such as age, education levels and workclass.

The raw file is imported into the workspace using the File Import node. The Results panel shows the data summary that has been imported

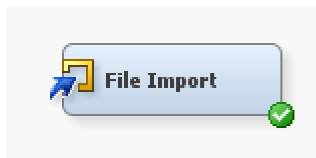


Figure 2.1: File Import node to import data from local storage.

2.2 Dataset Summary

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat	Label
12	_0	Num	8			0
5	_13	Num	8			13
1	_39	Num	8	BEST12.	BEST32.	
13	_40	Num	8			40
11	_2174	Num	8			2174
3	_77516	Num	8			77516
7	_Adm_clerical	Char	17			Adm-clerical
4	_Bachelors	Char	12			Bachelors
10	_Male	Char	6			Male
6	_Never_married	Char	21			Never-married
8	_Not_in_family	Char	14			Not-in-family
2	_State_gov	Char	16			State-gov
14	_United_States	Char	18			United-States
9	_White	Char	18			White
15	___50K	Char	5			<=50K

Figure 2.2: File Import summary.

From the results of File Import, we can see that the dataset does not have a header, therefore the data summary shows the first row as header instead. Therefore, we add header to the columns according to the metadata description from the website using Talend Data Prep. After that, we export the updated data file and update the file path for the File Import node and Run the tasks again.

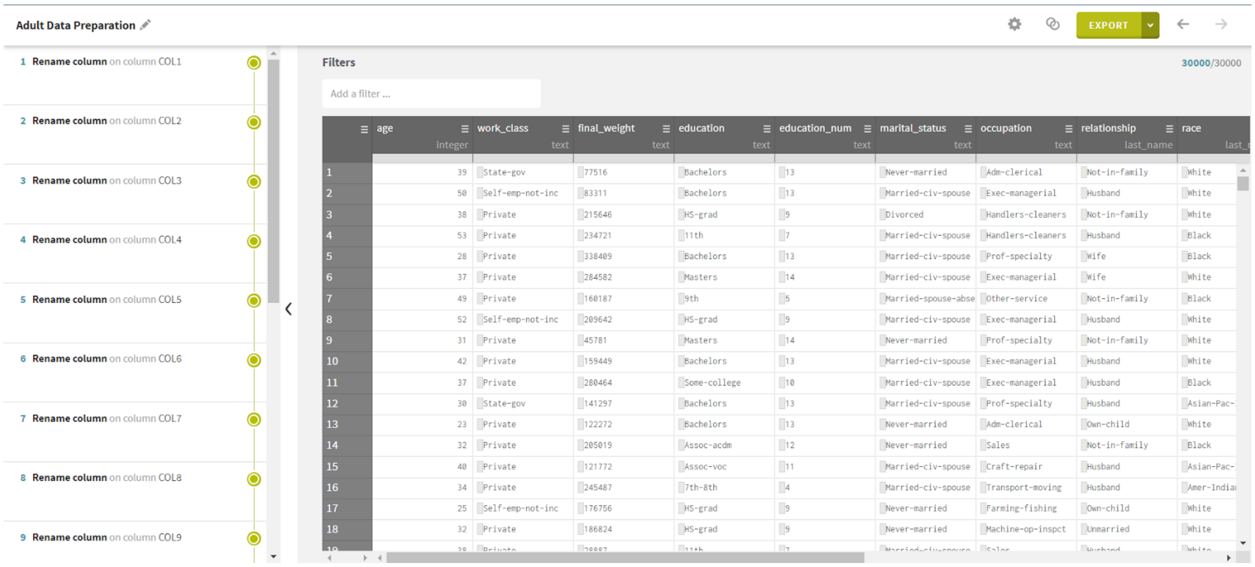


Figure 2.3: Replacing variables with meaningful names using Talend Data Prep.

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat
1	age	Num	8	BEST12.	BEST32.
11	capital_gain	Num	8	BEST12.	BEST32.
12	capital_loss	Num	8	BEST12.	BEST32.
4	education	Char	12	$12.	$12.
5	education_num	Num	8	BEST12.	BEST32.
3	final_weight	Num	8	BEST12.	BEST32.
15	gross_income	Char	5	$5.	$5.
13	hours_per_week	Num	8	BEST12.	BEST32.
6	marital_status	Char	21	$21.	$21.
14	native_country	Char	18	$18.	$18.
7	occupation	Char	17	$17.	$17.
9	race	Char	18	$18.	$18.
8	relationship	Char	14	$14.	$14.
10	sex	Char	6	$6.	$6.
2	work_class	Char	16	$16.	$16.

Figure 2.4: Updated variable summary.

Time series and row ID are synthesized for the purpose of sequence analysis, association rule analysis and time series clustering.

	id	timestamp	age	work_class	final_weight	education	education_num	marital_status	occupation	relationship	race	sex	capital_gain	capital_loss	hours_per_weel
21251	77	2022-08-30 11:00:00	32	Private	272376	Assoc-acdm	12	Never-married	Adm-clerical	Not-in-family	White	Female	0	0	4
27460	17	2023-05-16 04:00:00	28	Private	163772	HS-grad	9	Married-civ-spouse	Other-service	Husband	Other	Male	0	0	4
26623	44	2023-04-11 07:00:00	26	Private	39092	Some-college	10	Married-civ-spouse	Exec-managerial	Husband	White	Male	4064	0	5
4410	73	2020-09-27 18:00:00	60	Private	181953	HS-grad	9	Married-civ-spouse	Transport-moving	Husband	White	Male	0	0	2
9286	3	2021-04-18 22:00:00	27	Private	401723	HS-grad	9	Never-married	Adm-clerical	Not-in-family	Black	Female	0	0	4

Figure 2.5 Synthesized timestamp and id columns.

2.3 Column Metadata

The table below shows the clear description of columns.

Columns	Description	Datatype
age	The age of adult	Numerical (interval)
capital_gain	The income of the adult from investment sources other than working salary	Numerical (interval)
capital_loss	The loss of adult on the investment	Numerical (interval)
education	The highest education level of the adult	Categorical (ordinal)
education_num	The numerical representation of the “education” variable	Numerical (interval)
final_weight	The number of units in the target population that the responding unit represents	Numerical (interval)
gross_income	The income group of the adult, either more tham $50,000 or less than or equal to $50,000	Categorical (ordinal)
hours_per_week	The working hours of the adult per week	Numerical (interval)
marital_status	The marital status of the adult	Categorical (ordinal)
native_country	The country where the adult born in	Categorical (ordinal)
occupation	The job title of the adult	Categorical (ordinal)
race	The race of the adult	Categorical (ordinal)
relationship	The relationship status of the adult	Categorical (ordinal)
sex	The gender of the adult	Categorical (ordinal)
work_class	The company category that the adult worked at	Categorical (ordinal)

3.0 Application of SAS SEMMA Methodology

3.1 Sample

We had understood that this dataset is used for predicting income levels of US citizens. Before starting the sampling, we changed the role of gross_income as target variable by right clicking on the File Import node and 'Edit Variables'.

Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
age	Input	Interval	No		No	.	.
capital_gain	Input	Interval	No		No	.	.
capital_loss	Input	Interval	No		No	.	.
education	Input	Nominal	No		No	.	.
education_num	Input	Interval	No		No	.	.
final_weight	Input	Interval	No		No	.	.
gross_income	Target	Nominal	No		No	.	.
hours_per_we	Input	Interval	No		No	.	.
marital_status	Input	Nominal	No		No	.	.
native_country	Input	Nominal	No		No	.	.
occupation	Input	Nominal	No		No	.	.
race	Input	Nominal	No		No	.	.
relationship	Input	Nominal	No		No	.	.
sex	Input	Nominal	No		No	.	.
work_class	Input	Nominal	No		No	.	.

Figure 3.1.1: gross_income is changed to Target role.

3.1.1 Stratified Sampling to Alleviate Target Variable Class Imbalance

By clicking the Explore button at the bottom of the Variables window, we can see the distribution of the target variable. We plotted a pie chart for easier understanding. The chart shows that the target groups are not balanced. Therefore, we chose the **stratified sampling method** to make the target group balanced. The **stratify strategy** is set to **Equal**. **Size Percentage** is set to **100** to make sure the sample covers the whole population and is representative enough of the dataset.

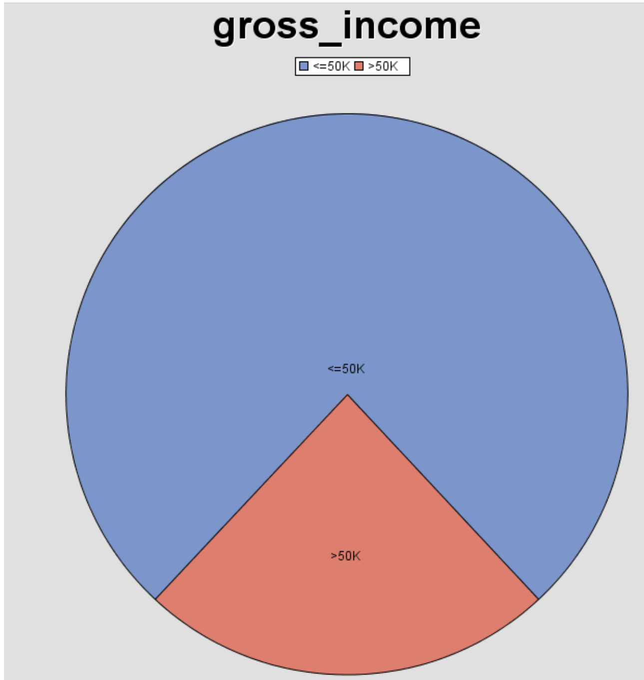


Figure 3.1.2: Initial distribution of gross_income.

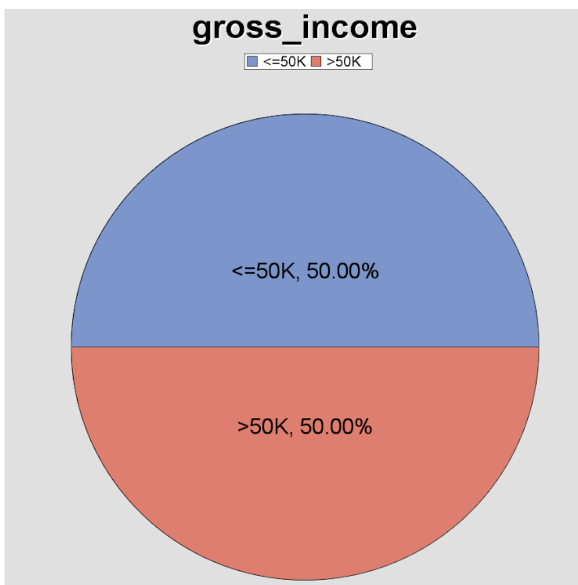


Figure 3.1.3: Distribution of gross_income categories after sampling.

3.1.2 Data Partitioning

In this section, we divided the dataset into two sets, 80% as the training set and 20% as the validation set. The training set is being used to fit the model for obtaining the best set of model parameters and the validation set is being used to test the generalization of the trained model to new unseen data. The partitioning method we use is known as stratified partitioning, in which the goal is to maintain the same distribution of target classes in both the training and validation sets as in the original dataset. A default random seed of 12345 is set to ensure the reproducibility of the experiment.

Property	Value
General	
Node ID	Part
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Output Type	Data
Partitioning Method	Stratified
Random Seed	12345
Data Set Allocations	
Training	80.0
Validation	20.0
Test	0.0
Report	
Interval Targets	Yes
Class Targets	Yes
Status	
Create Time	1/21/24 1:47 PM
Run ID	4783045b-583b-bd4c-b476-d
Last Error	
Last Status	Complete
Last Run Time	1/21/24 2:07 PM
Run Duration	0 Hr. 0 Min. 1.92 Sec.
Grid Host	
User-Added Node	No

Figure 3.1.4: Properties of Data Partition node.

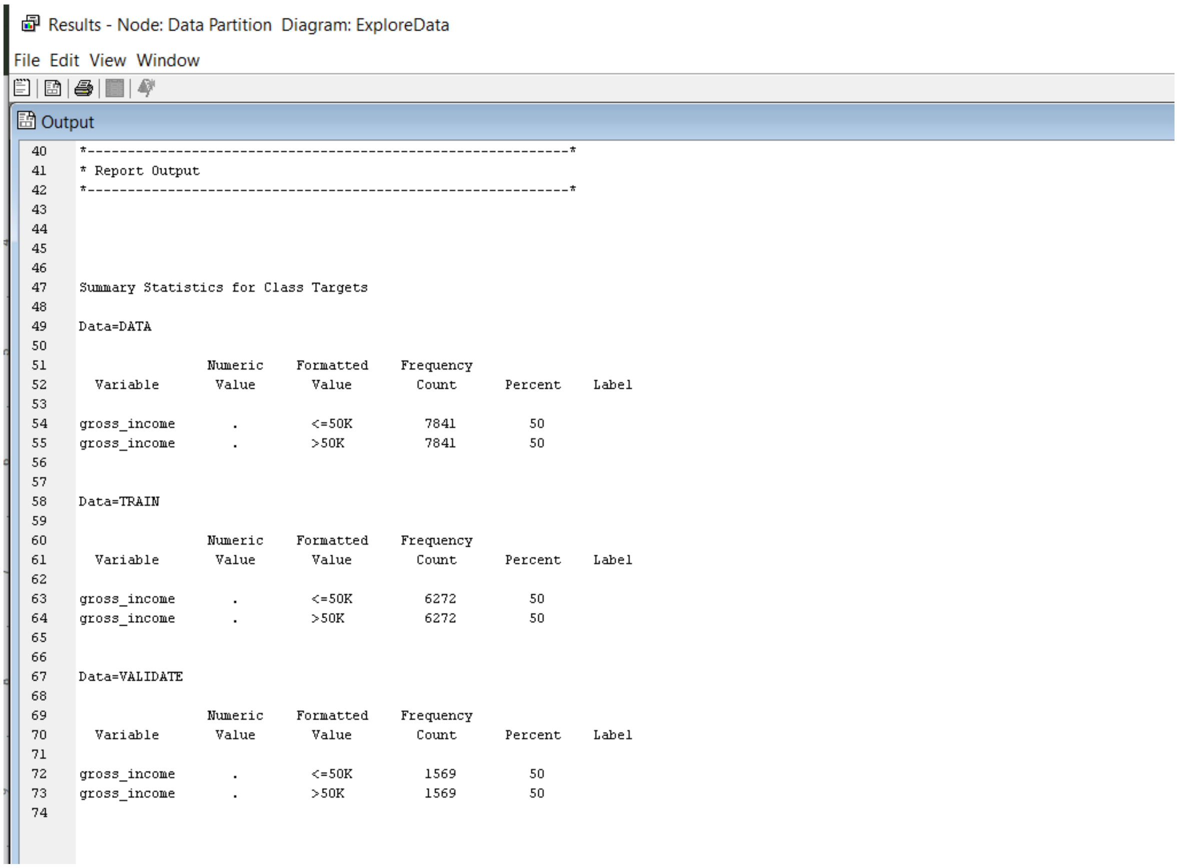


Figure 3.1.5: Data partition result.

3.2 Explore

3.2.1 Chi Square & Feature Importance

Chi-Square values are used to determine the relationship between the target variable (i.e. gross_income) with the independent variables, i.e the categorical variables. Cramer's V is used to determine how strong are the relationships between the target and independent variables. Based on the matrix below, relationship, marital_status, occupation and education have the strongest relationship with gross_income.

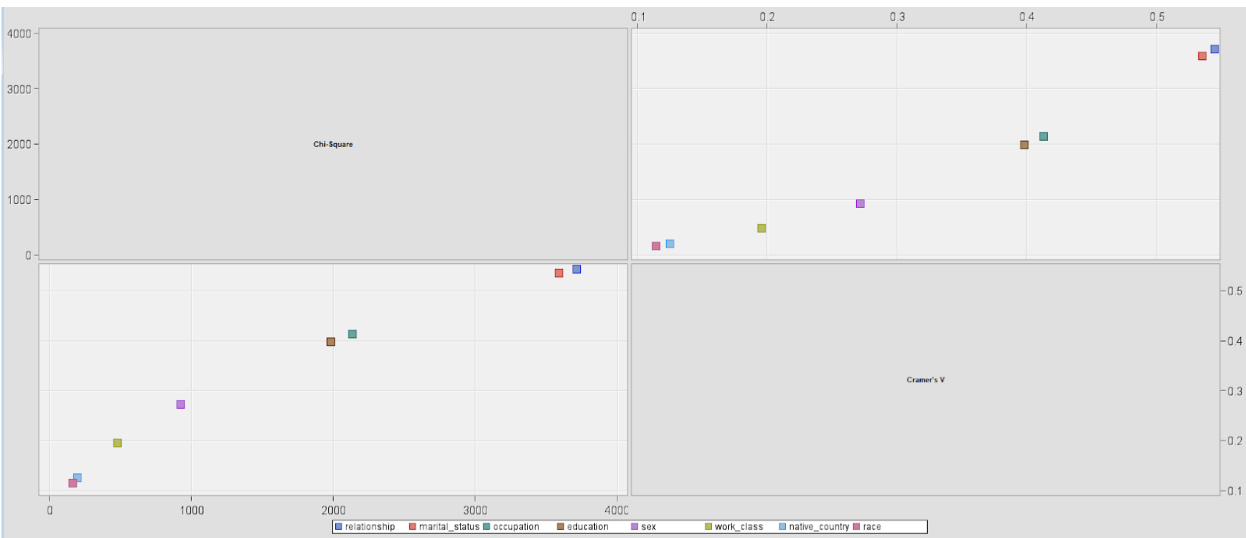


Figure 3.2.1: Chi-Square and Cramer's V matrix plot for all categorical variables.

Variable worth analysis looks into the worth of all variables including nominal and continuous variables to predict the target variable. According to the bar chart below, relationship, marital_status, occupation, age, education_num, education and hours_per_week have the worth value higher than median worth value.

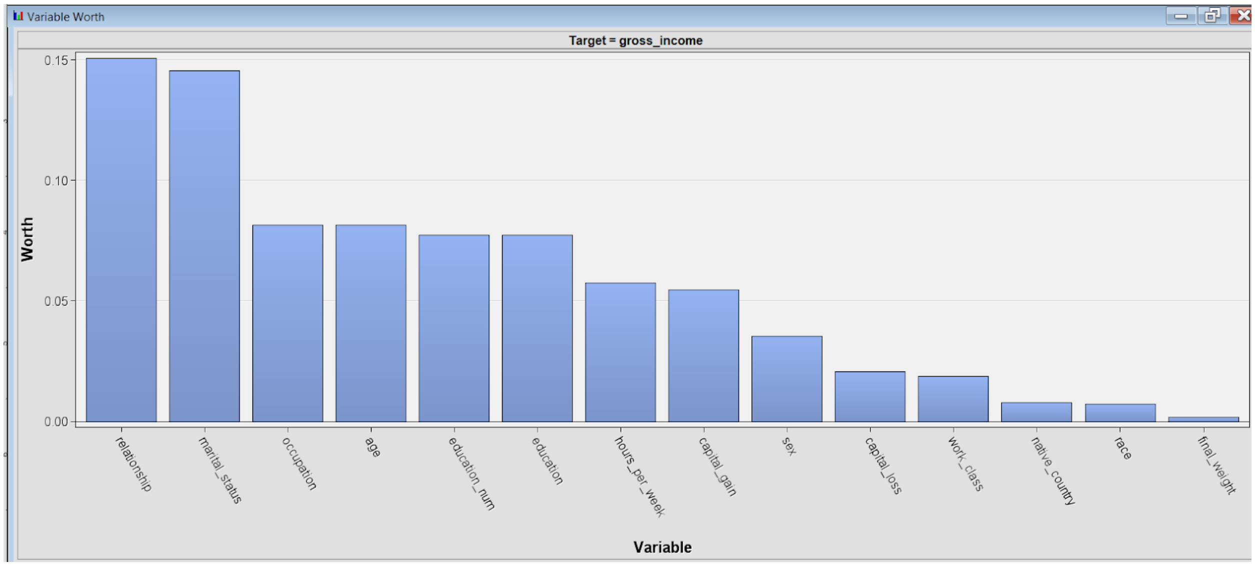


Figure 3.2.2: Bar chart showing worth value for all variables.

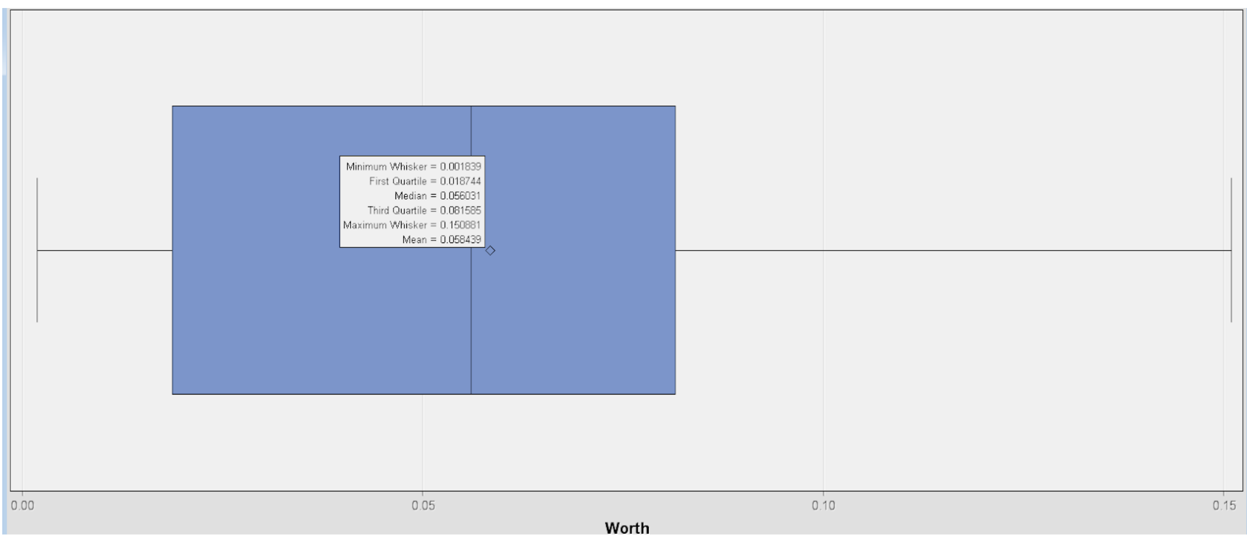


Figure 3.2.3: Box plot showing the distribution of variable worth values.

3.2.2 Descriptive Analysis

3.2.2.1 Univariate and Multivariate Analysis

All columns are explored with the sampling setting of Random and Max to load all data from the sample for the exploration.

- **Class Variables**

a. relationship

The values in this column are clean. Since it has a strong relationship with the target variable, it will be accepted as an input variable.

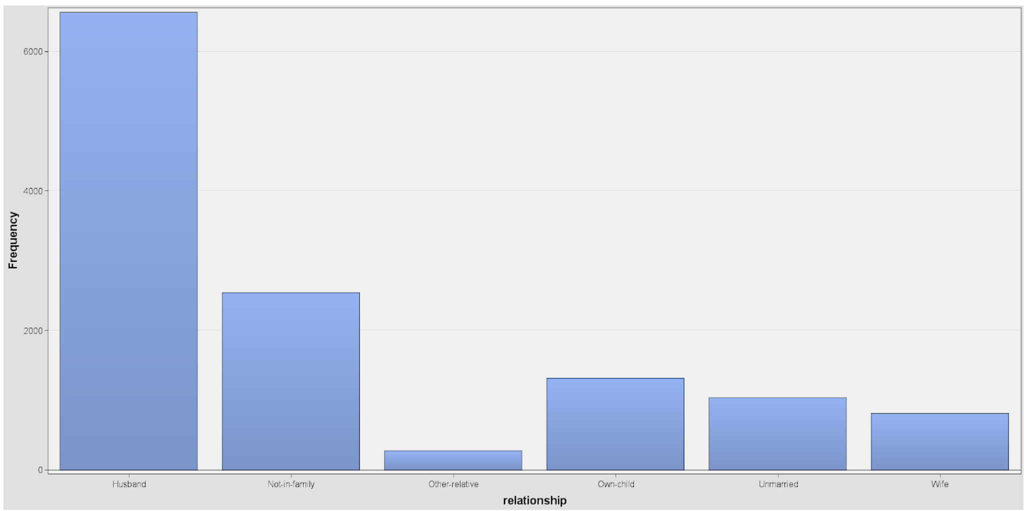


Figure 3.2.4: Bar chart showing value distribution in relationship column.

b. marital_status

The values in this column are clean. Since it has strong relationship with the target variable, it will be accepted as input variable

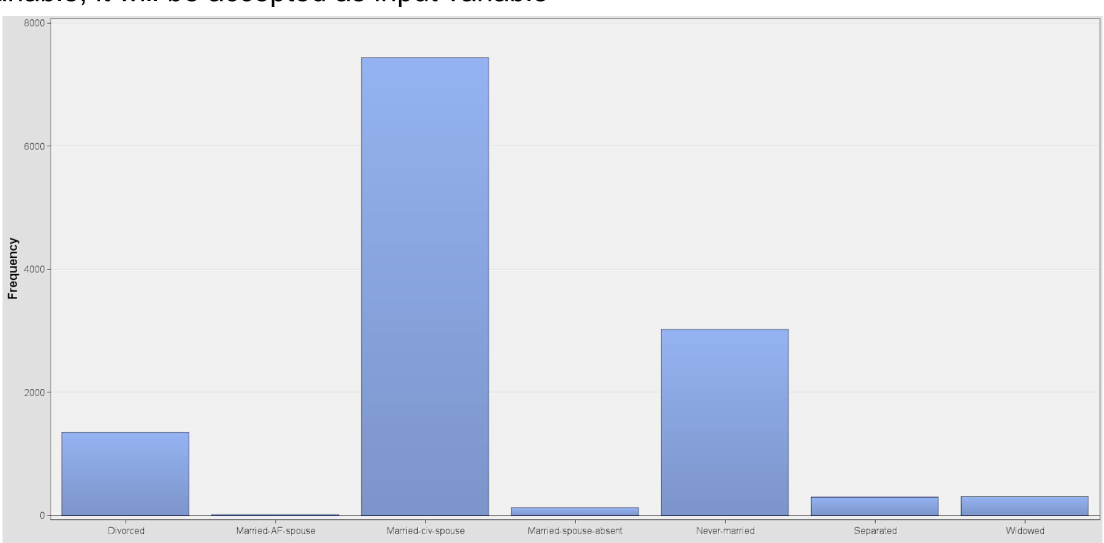


Figure 3.2.5: Bar chart showing value distribution in marital_status column.

c. occupation

Unknown values is spotted in thie variable and it is marked as '?'. These values will be converted into null and the inference model will be used to inference the missing values. This variable has strong relationship with the target variable, therefore it will be accepted as target variable.

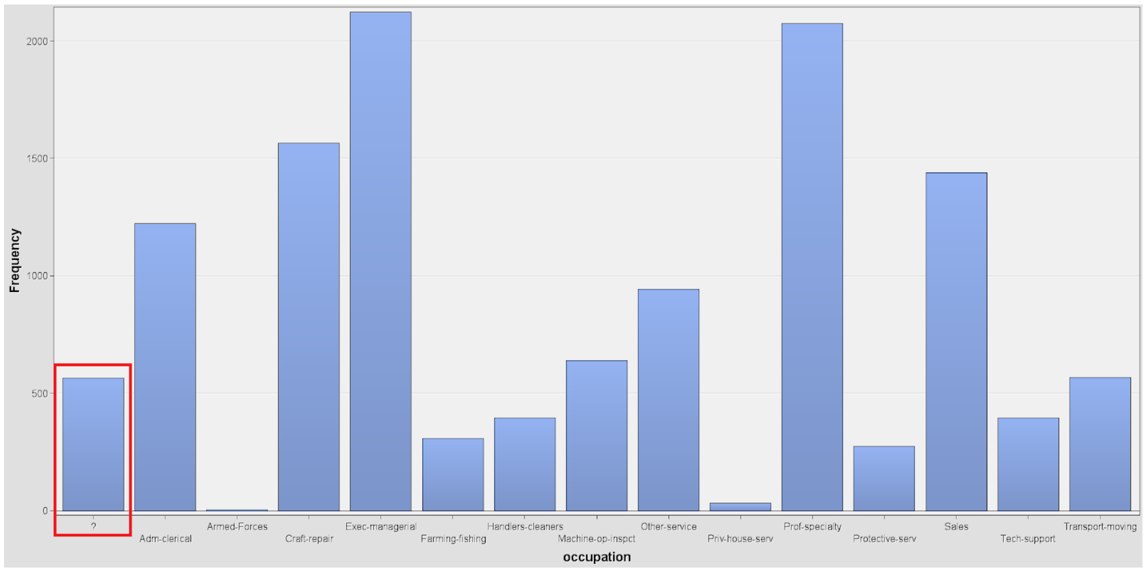


Figure 3.2.6: Bar chart showing value distribution in occupation column.

d. education

The value in this column is considered clean. However, education_num provides the same information as this variable, therefore it will be rejected as an input variable.

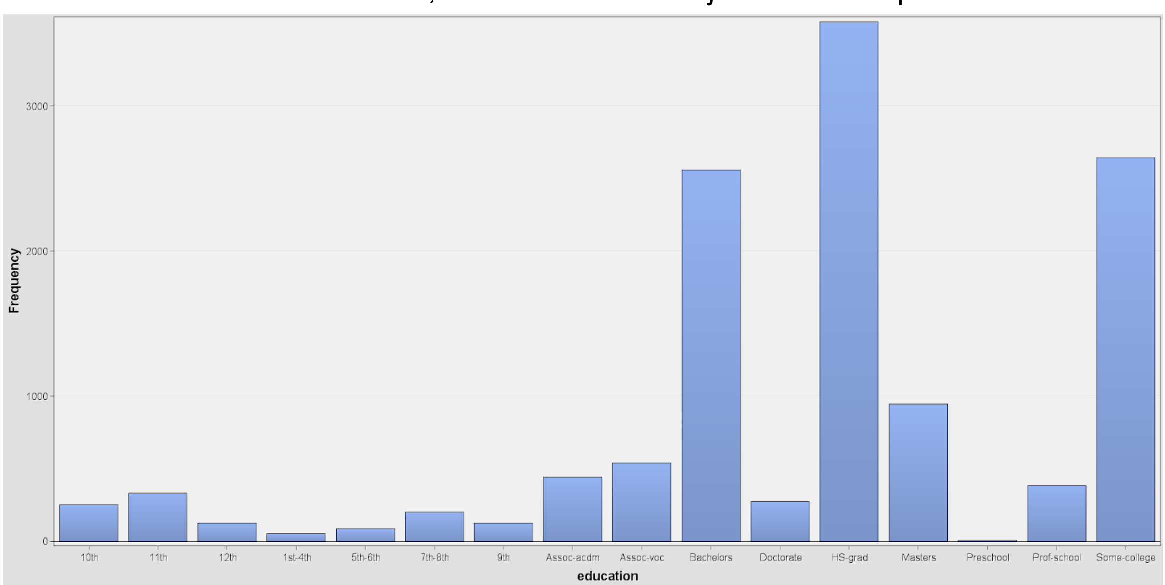


Figure 3.2.7: Bar chart showing value distribution in education column.

e. sex

The values of this variable are clean. It can be accepted as an input variable to explore for more insights.

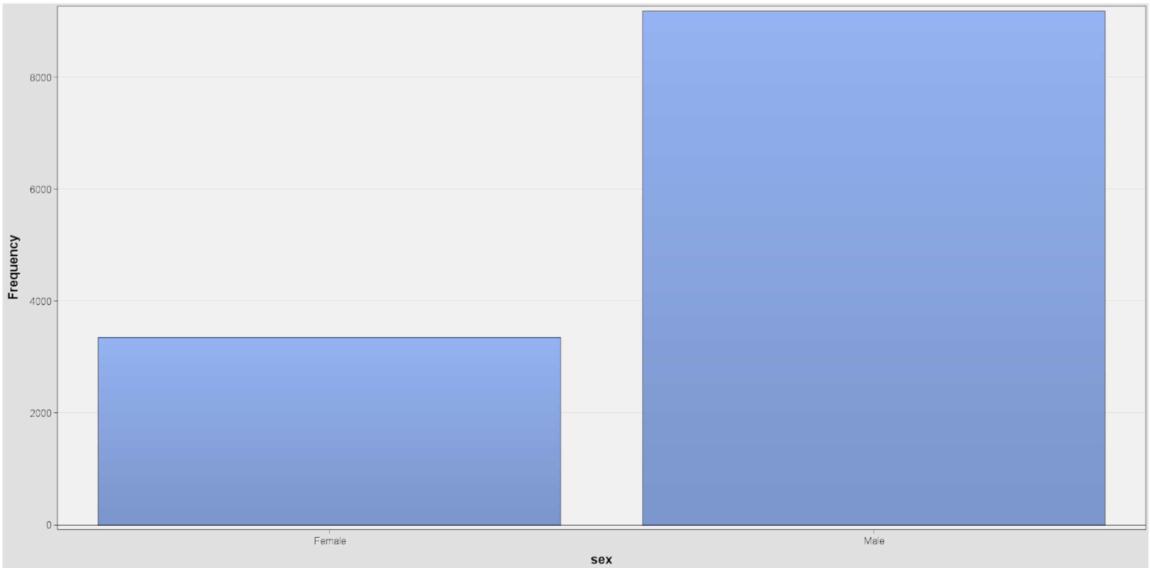


Figure 3.2.8: Bar chart showing value distribution in sex column.

f. work_class

Unknown value is detected in this variable and marked as '?'. These values will be converted to null and an inference model will be used to inference the missing values. It is commonly known that the working class will affect income level, therefore this variable will be accepted as input variable.

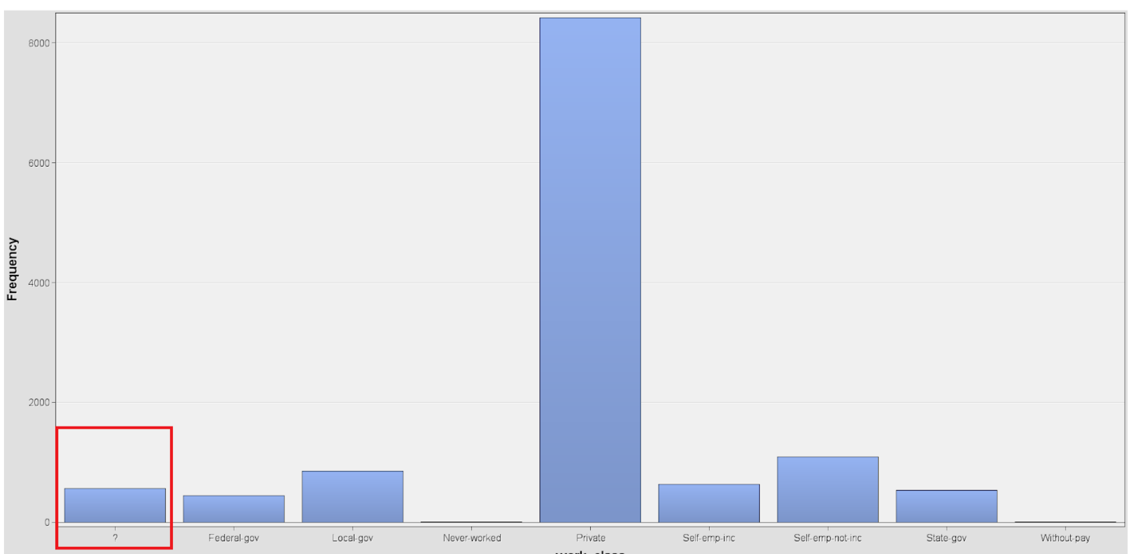


Figure 3.2.9: Bar chart showing value distribution in work_class.

g. native_country

Most of the values in this variable are United States and the other values only consists of a very small portion. Therefore, the values other than Unted States will be grouped as ‘Other’. Poor countries, developing countries, and developed countries have different income level, therefore this column will be accepted as input variable to provide more insight.

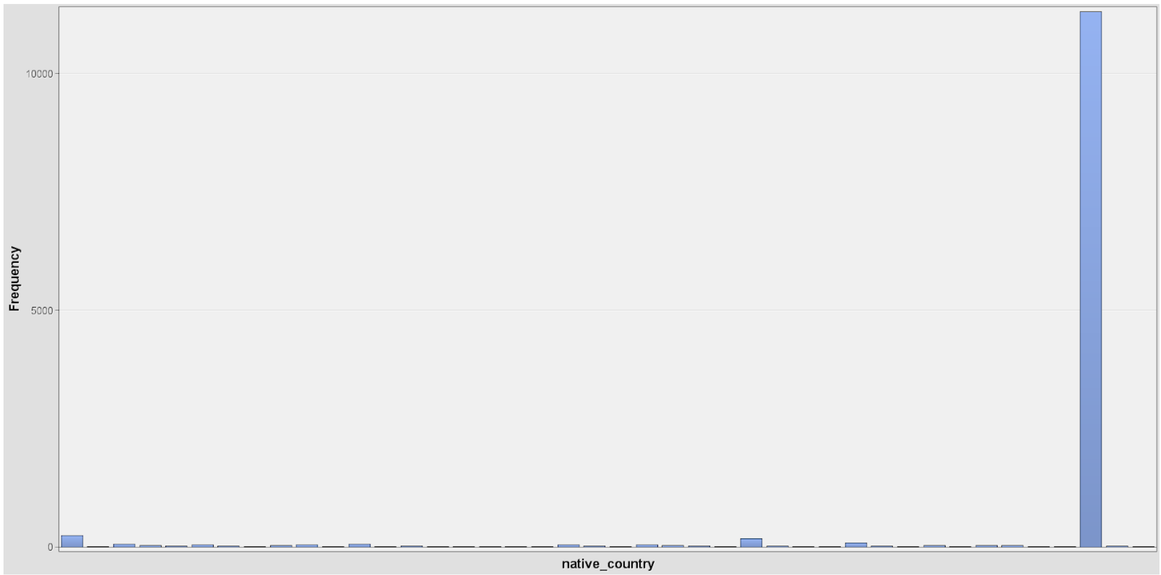


Figure 3.2.10: Bar chart showing distribution of values in native_country column.

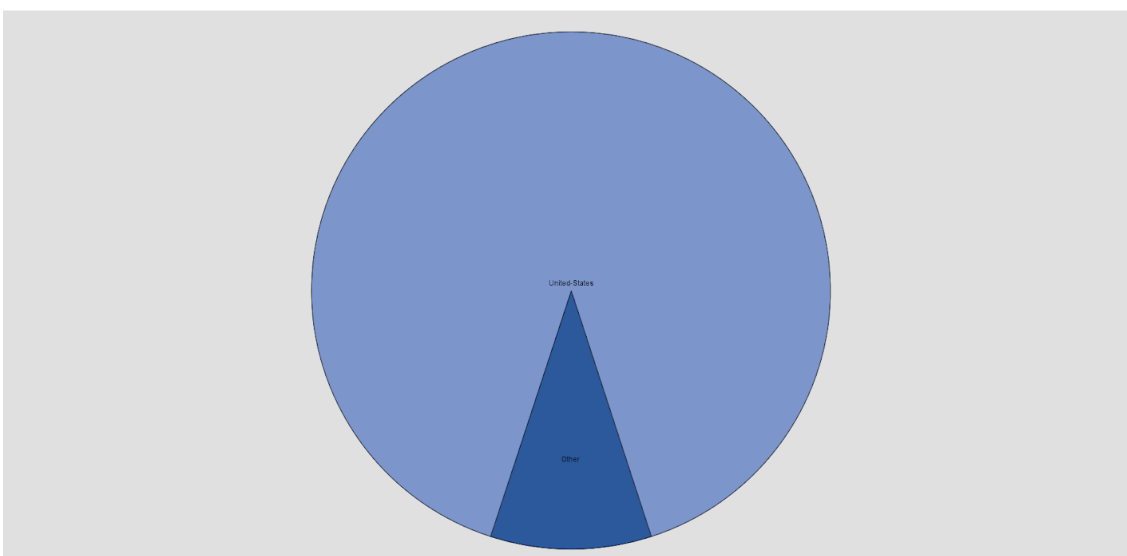


Figure 3.2.11: Pie plot suggested that a big portion of values are the United States and the other values can be grouped as ‘Other’.

h. race

The values in this column are clean. This column can be used with other columns to provide more insights for example working class. Therefore, it will be accepted as an input variable.

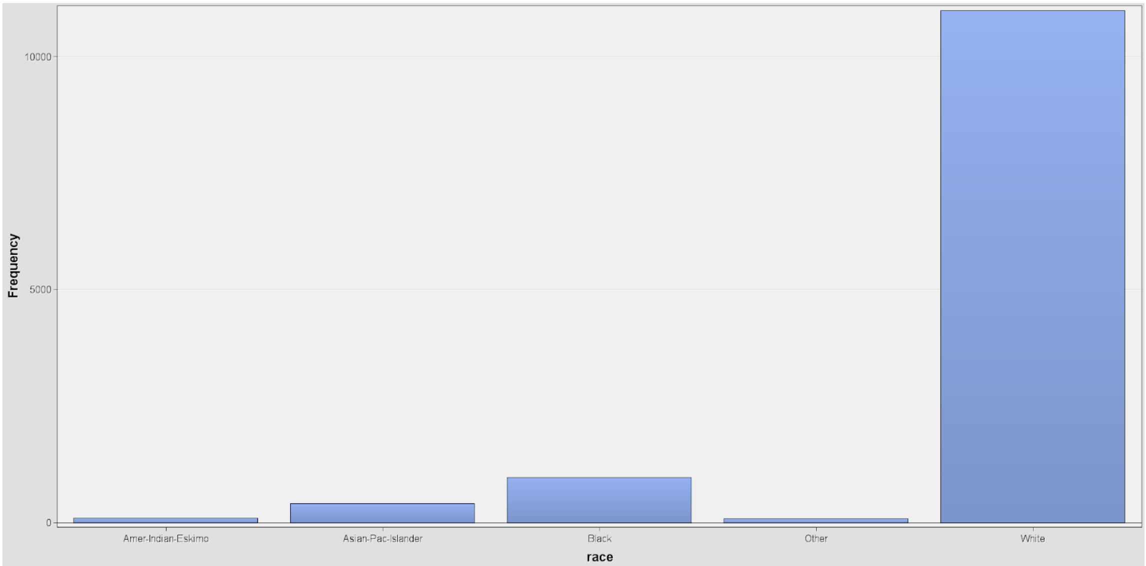


Figure 3.2.12: Bar chart showing value distribution in race column.

Figure 3.2.13: Histogram plot of work_class, race and gross_income.

- **Interval Variables**

a. age

The values in the age columns are reasonable and clean. Due to its strong relationship and worth with the target variable, it will be accepted as input variable.

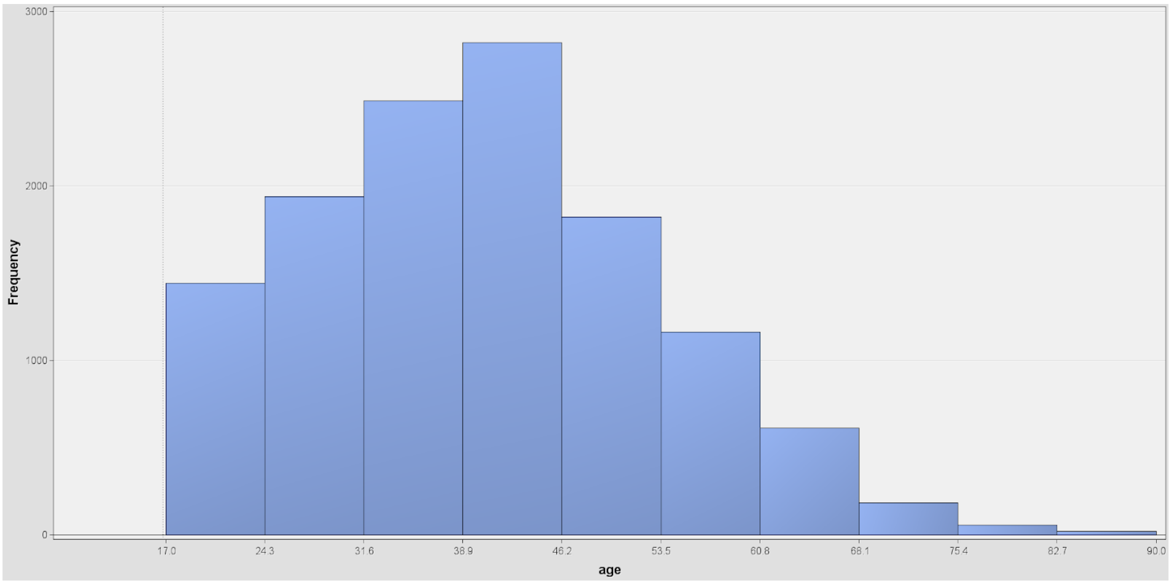


Figure 3.2.14: Histogram showing distribution of age values.

b. education_num

This column is the number representation of the education column, which means the minimum year of education. The values in this column are clean and no missing values detected. Due to its strong relationship and high worth to the target variable, it will be accepted as target variable.

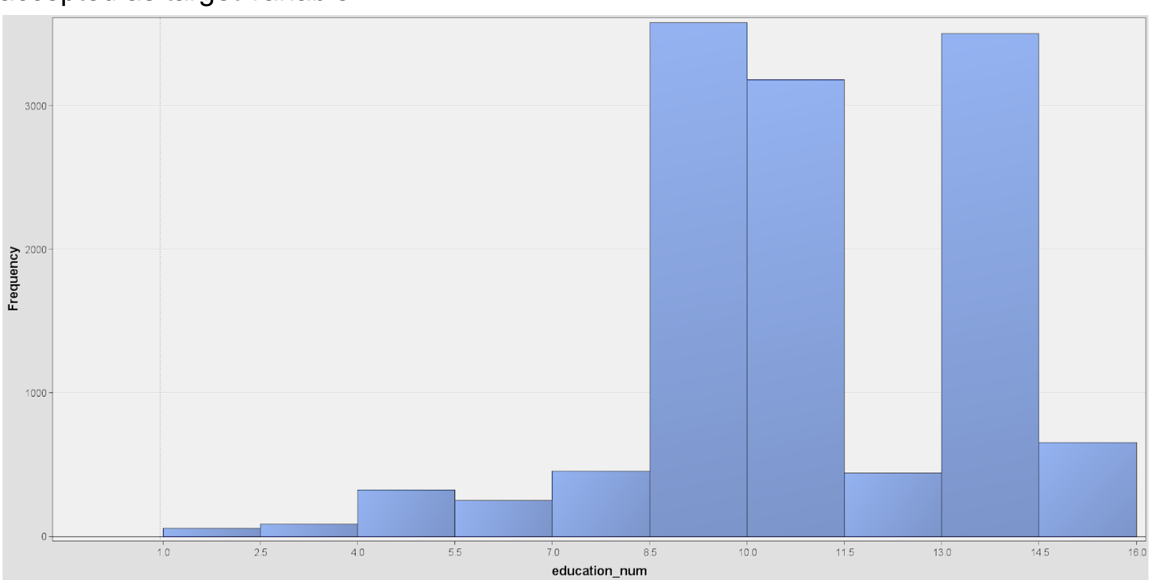


Figure 3.2.15: Histogram showing value distribution in education_num column.

c. **hours_per_week**

The values in this column are clean and no missing values detected. Income levels for some working classes are highly related to working hours, therefore this column will be accepted as input variable.

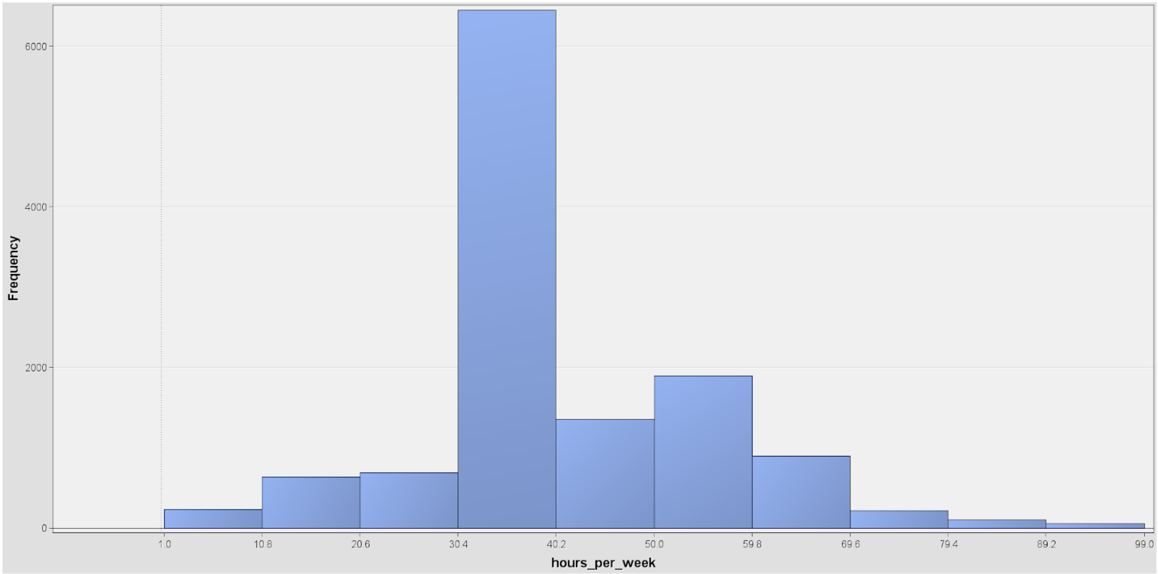


Figure 3.2.16: Histogram showing value distribution in hours_per_week column.

d. **capital_gain**

Most of the values are cluttered at 0, therefore it cannot provide more insights for the target variable. It will be rejected for input.

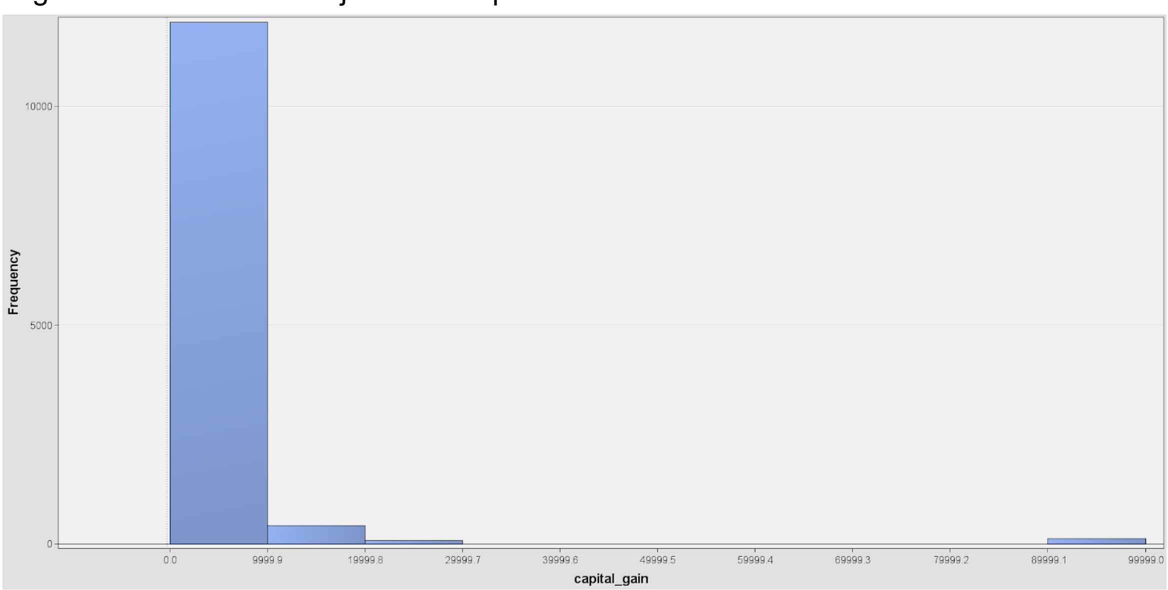


Figure 3.2.17: Histogram plot showing value distribution of capital_gain column.

e. **capital_loss**

Most of the values are cluttered at 0, therefore it cannot provide more insights for the target variable. It will be rejected for input.

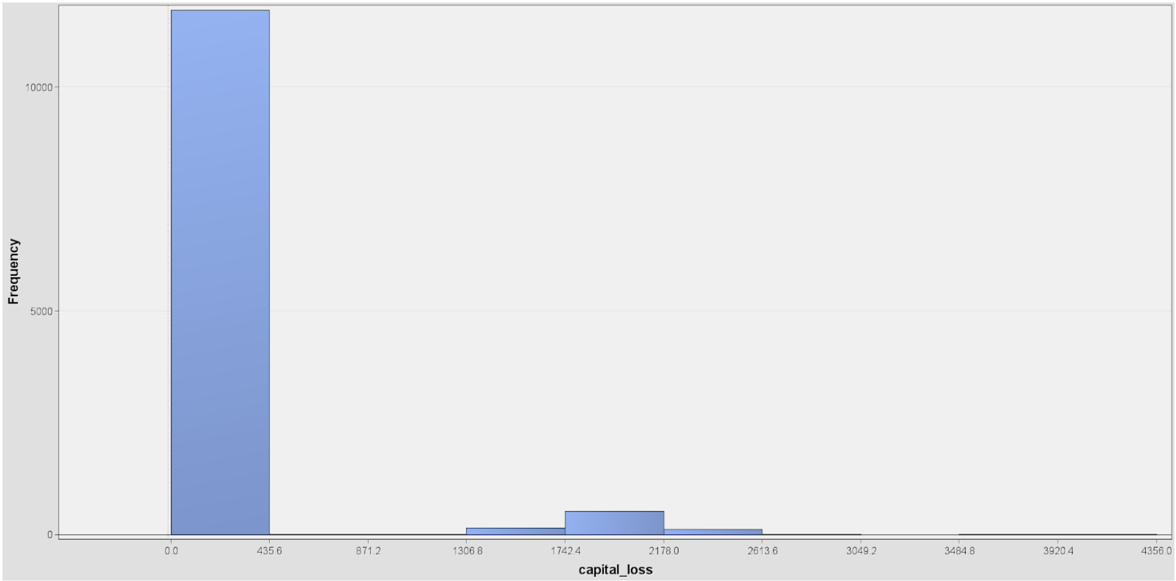


Figure 3.2.18: Histogram plot showing value distribution of capital_loss column.

f. final_weight

This column is the weight added for identifying different demography. It does not provide more insights for the target variable, therefore it will be rejected for input.

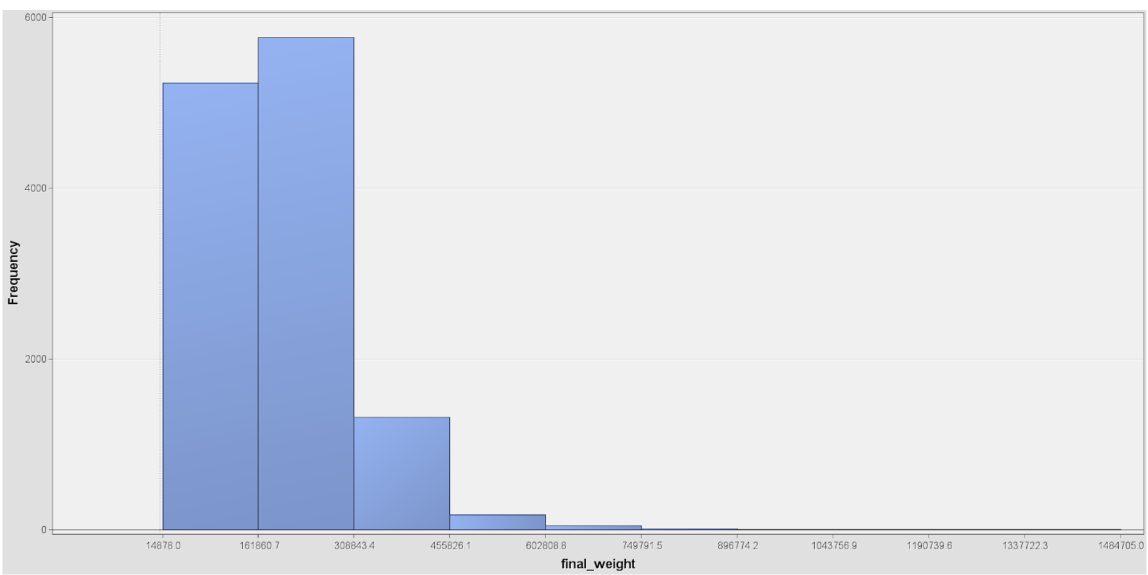


Figure 3.2.19: Histogram plot showing distribution of values in final_weight.

3.2.3 Association Rule Analysis

Association rule analysis has been done to explore the types of occupations that would be done by the same person. Table below shows that an armed-forces would probably also be a private house servant and transport moving workers probably an armed-forces as well. There are 100 rules generated on the occupation column.

Rule Description

Map	Rule
RULE1	Armed-Forces ==> Priv-house-serv
RULE2	Transport-moving & Armed-Forces ==> Priv-house-serv
RULE3	Tech-support & Armed-Forces ==> Priv-house-serv
RULE4	Sales & Armed-Forces ==> Priv-house-serv
RULE5	Protective-serv & Armed-Forces ==> Priv-house-serv
RULE6	Prof-specialty & Armed-Forces ==> Priv-house-serv
RULE7	Other-service & Armed-Forces ==> Priv-house-serv
RULE8	Machine-op-inspct & Armed-Forces ==> Priv-house-serv
RULE9	Handlers-cleaners & Armed-Forces ==> Priv-house-serv
RULE10	Farming-fishing & Armed-Forces ==> Priv-house-serv
RULE11	Exec-managerial & Armed-Forces ==> Priv-house-serv
RULE12	Craft-repair & Armed-Forces ==> Priv-house-serv
RULE13	Armed-Forces & Adm-clerical ==> Priv-house-serv
RULE14	Armed-Forces & ? ==> Priv-house-serv
RULE15	Transport-moving & Tech-support & Armed-Forces ==> Priv-house-serv
RULE16	Transport-moving & Sales & Armed-Forces ==> Priv-house-serv
RULE17	Transport-moving & Protective-serv & Armed-Forces ==> Priv-house-serv
RULE18	Transport-moving & Prof-specialty & Armed-Forces ==> Priv-house-serv
RULE19	Transport-moving & Other-service & Armed-Forces ==> Priv-house-serv
RULE20	Transport-moving & Machine-op-inspct & Armed-Forces ==> Priv-house-serv
RULE21	Transport-moving & Handlers-cleaners & Armed-Forces ==> Priv-house-serv
RULE22	Transport-moving & Farming-fishing & Armed-Forces ==> Priv-house-serv
RULE23	Transport-moving & Exec-managerial & Armed-Forces ==> Priv-house-serv
RULE24	Transport-moving & Craft-repair & Armed-Forces ==> Priv-house-serv
RULE25	Transport-moving & Armed-Forces & Adm-clerical ==> Priv-house-serv
RULE26	Transport-moving & Armed-Forces & ? ==> Priv-house-serv
RULE27	Tech-support & Sales & Armed-Forces ==> Priv-house-serv
RULE28	Tech-support & Protective-serv & Armed-Forces ==> Priv-house-serv
RULE29	Tech-support & Prof-specialty & Armed-Forces ==> Priv-house-serv
RULE30	Tech-support & Other-service & Armed-Forces ==> Priv-house-serv
RULE31	Tech-support & Machine-op-inspct & Armed-Forces ==> Priv-house-serv
RULE32	Tech-support & Handlers-cleaners & Armed-Forces ==> Priv-house-serv
RULE33	Tech-support & Farming-fishing & Armed-Forces ==> Priv-house-serv
RULE34	Tech-support & Exec-managerial & Armed-Forces ==> Priv-house-serv
RULE35	Tech-support & Craft-repair & Armed-Forces ==> Priv-house-serv
RULE36	Tech-support & Armed-Forces & Adm-clerical ==> Priv-house-serv
RULE37	Tech-support & Armed-Forces & ? ==> Priv-house-serv
RULE38	Sales & Protective-serv & Armed-Forces ==> Priv-house-serv
RULE39	Sales & Prof-specialty & Armed-Forces ==> Priv-house-serv
RULE40	Sales & Other-service & Armed-Forces ==> Priv-house-serv

Figure 3.2.20: Association rules table.

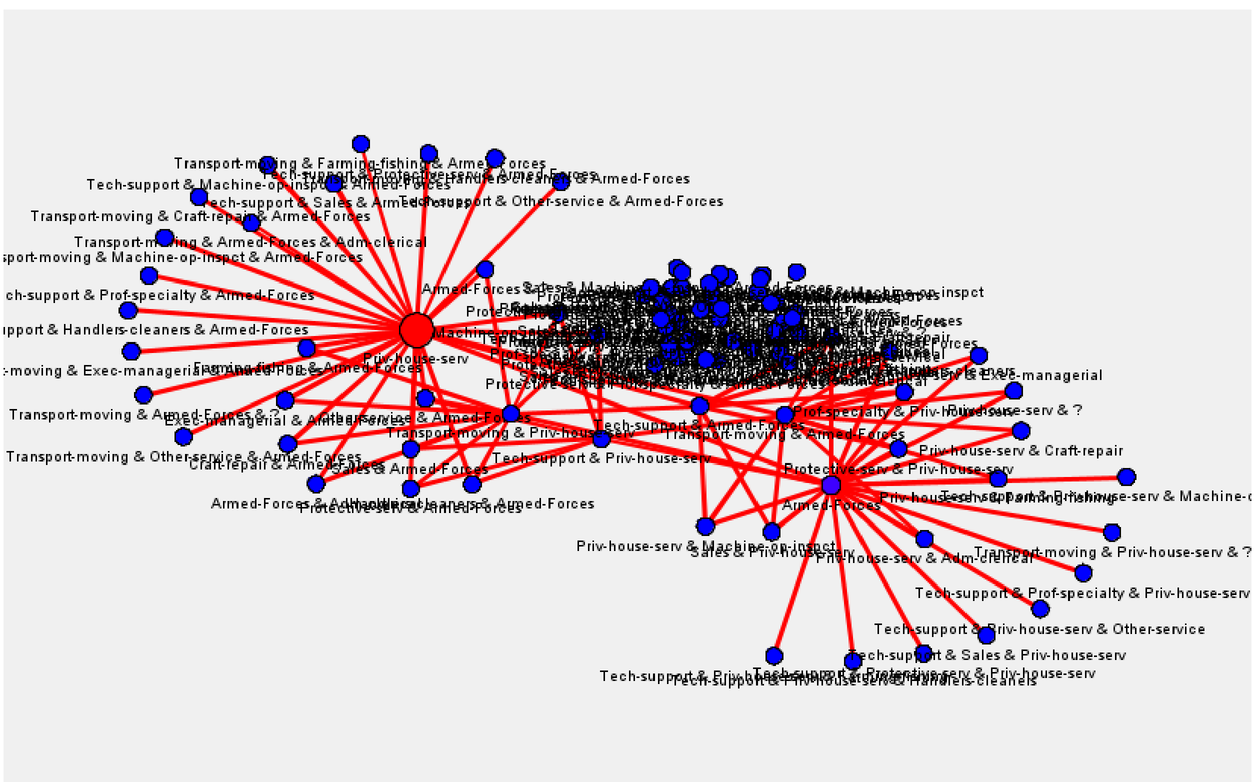


Figure 3.2.21: Link graph of occupation association rules.

3.2.4 Sequence Analysis

Sequence analysis has been done on the occupation column to find out the sequence of a person taking different jobs. The top rules show that most of the people will have unknown occupation values after the first job, while from RULE 14 onwards, people will choose a administrative clerical job and craft repair job after the first job.

Map	Rule
RULE1	? ==> ?
RULE2	Adm-clerical ==> ?
RULE3	Craft-repair ==> ?
RULE4	Exec-managerial ==> ?
RULE5	Farming-fishing ==> ?
RULE6	Handlers-cleaners ==> ?
RULE7	Machine-op-inspct ==> ?
RULE8	Other-service ==> ?
RULE9	Prof-specialty ==> ?
RULE10	Protective-serv ==> ?
RULE11	Sales ==> ?
RULE12	Transport-moving ==> ?
RULE13	? ==> Adm-clerical
RULE14	Adm-clerical ==> Adm-clerical
RULE15	Craft-repair ==> Adm-clerical
RULE16	Exec-managerial ==> Adm-clerical
RULE17	Farming-fishing ==> Adm-clerical
RULE18	Handlers-cleaners ==> Adm-clerical
RULE19	Machine-op-inspct ==> Adm-clerical
RULE20	Other-service ==> Adm-clerical
RULE21	Prof-specialty ==> Adm-clerical
RULE22	Protective-serv ==> Adm-clerical
RULE23	Sales ==> Adm-clerical
RULE24	Tech-support ==> Adm-clerical
RULE25	Transport-moving ==> Adm-clerical
RULE26	? ==> Craft-repair
RULE27	Adm-clerical ==> Craft-repair
RULE28	Craft-repair ==> Craft-repair
RULE29	Exec-managerial ==> Craft-repair
RULE30	Farming-fishing ==> Craft-repair
RULE31	Handlers-cleaners ==> Craft-repair
RULE32	Machine-op-inspct ==> Craft-repair
RULE33	Other-service ==> Craft-repair
RULE34	Prof-specialty ==> Craft-repair
RULE35	Protective-serv ==> Craft-repair
RULE36	Sales ==> Craft-repair
RULE37	Transport-moving ==> Craft-repair
RULE38	? ==> Exec-managerial
RULE39	Adm-clerical ==> Exec-managerial
RULE40	Craft-repair ==> Exec-managerial

Figure 3.2.22: Sequence analysis rules table.

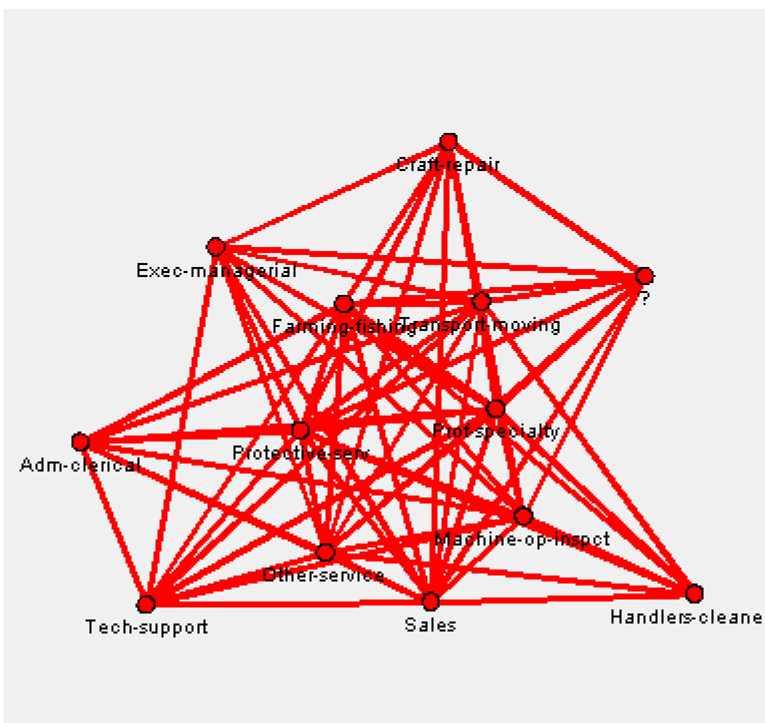


Figure 3.2.23: Link graph of sequence analysis rules.

3.2.5 Time Series Clustering

Time series clustering has been done on the marital_status, occupation and relationship for capital_gain to find out similar time series of capital gain. The results show that TS-297 is the most similar with TS-2 and TS-13 is the most similar with TS-8.

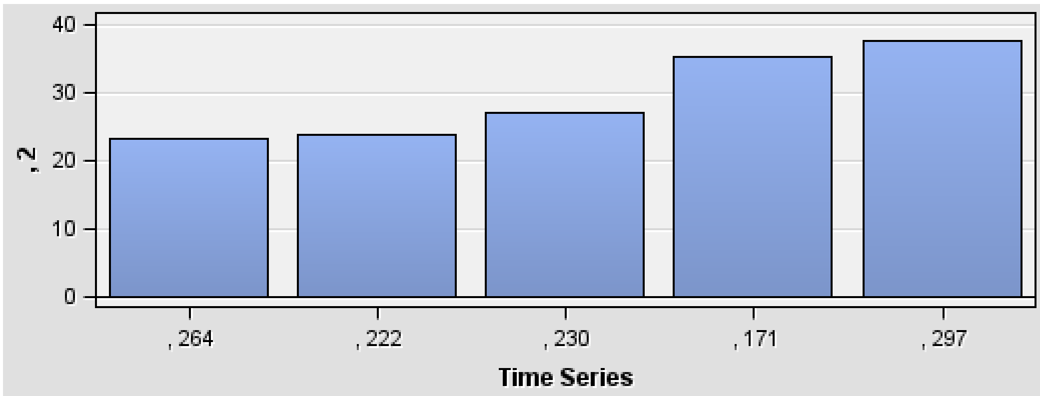


Figure 3.2.24: Bar graph of time series similar measure for TS-2.

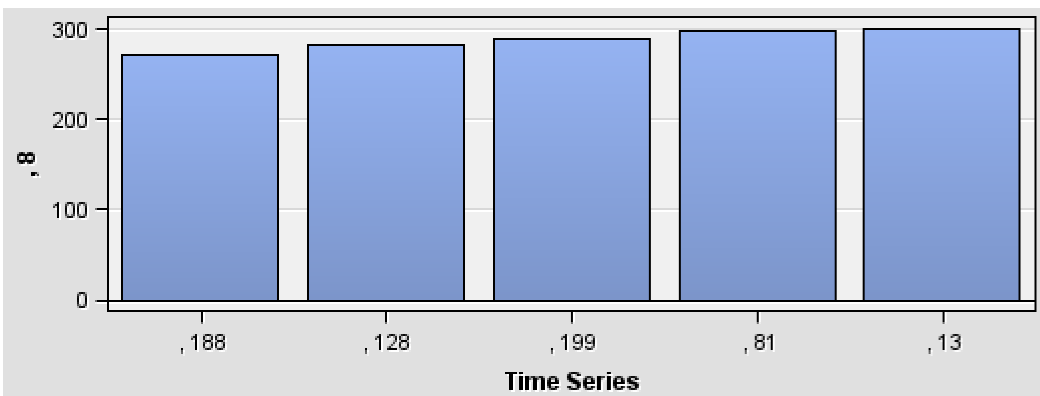


Figure 3.2.25: Bar graph of time series similar measure for TS-8.

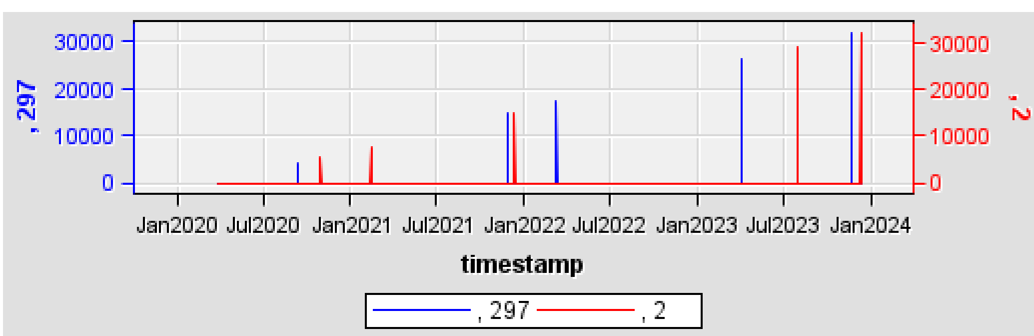


Figure 3.2.26: Chart showing patterns of TS-2 versus TS-297.

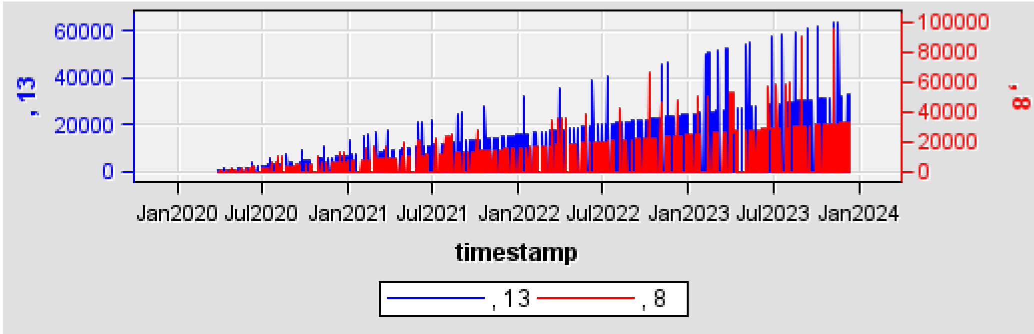


Figure 3.2.27: Chart showing patterns of TS-8 versus TS-13.

3.2.6 Summary

Column	Findings	Actions
relationship	- Strong relationship with target_variable. - Clean data.	- Accepted as input.
marital_status	- Strong relationship with target_variable. - Clean data.	- Accepted as input.
occupation	- Strong relationship with target_variable. - Unknown values detected.	- Accepted as input. - Assume unknown values as missing values. - Impute missing value.
education	- Strong relationship with target_variable. - Text representation of education_num	- Rejected.
sex	- Can be combined with the marital_status column. - Clean data.	- Accepted as input.
work_class	- Unknown values detected. - Can be combined with other variables to provide more insights.	- Accepted as input. - Assume unknown values as missing values. - Inference missing value.

native_country	- Majority of values are the United-States. - No missing values detected.	- Group the minority groups as ‘Others’. - Accepted as input.
race	- Clean data. - Can be combined with other variables to provide more insights.	- Accepted as input.
age	- Strong relationship with target_variable. - Clean data.	- Accepted as input.
education_num	- Strong relationship with target_variable. - Clean data.	- Accepted as input.
hours_per_week	- Strong relationship with target_variable. - Clean data.	- Accepted as input.
capital_gain	- Values cluttered at 0.	- Rejected.
capital_loss	- Values cluttered at 0.	- Rejected.
final_weight	- Does not provide more insights.	- Rejected.

3.3 Modify

Based on the previous section, these are the variables that need further processing before going into the modeling step.

Variable	Action
age	- Accepted as input by default, no outliers found.
capital_gain	- Need to be dropped, most of the values are ‘0’ that are not useful.
capital_loss	- Need to be dropped, most of the values are ‘0’ that are not useful.
education	- Need to be dropped, since “education_year” brings the same insight as this.
education_year	- Accepted as input by default, no outliers found.
final_weight	- Need to be dropped, irrelevant for predicting income.
gross_income	- Set as target to perform income prediction.
hours_per_week	- Accepted as input by default, no outliers found.
marital_status	- Accepted as input.
native_country	- Accepted as input. - Currently 90.15% is United-States, all country values are then replaced with mode value.
occupation	- Accepted as input. - Impute rows with ‘?’ symbol with inference algorithm.
race	- Accepted as input by default, no outliers found.
relationship	- Need to be dropped, “marital-status” and “gender” are more representative and contribute the similar meaning as this variable.
sex	- Accepted as input by default, no outliers found.
work_class	- Accepted as input. - Impute rows with ‘?’ symbol with inference algorithm.

Class Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
TRAIN	education	INPUT	16	0	HS-grad	28.55	Some-college	21.06
TRAIN	marital_status	INPUT	7	0	Married-civ-spouse	59.27	Never-married	24.07
TRAIN	native_country	INPUT	42	0	United-States	90.18	?	1.86
TRAIN	occupation	INPUT	15	0	Exec-managerial	16.93	Prof-specialty	16.53
TRAIN	race	INPUT	5	0	White	87.63	Black	7.70
TRAIN	relationship	INPUT	6	0	Husband	52.36	Not-in-family	20.26
TRAIN	sex	INPUT	2	0	Male	73.27	Female	26.73
TRAIN	work_class	INPUT	9	0	Private	67.16	Self-emp-not-inc	8.71
TRAIN	gross_income	TARGET	2	0	<=50K	50.00	>50K	50.00

Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
age	INPUT	40.41151	12.88093	12544	0	17	40	90	0.391217	-0.16807
capital_gain	INPUT	2035.503	10206.72	12544	0	0	0	99999	8.514124	77.7758
capital_loss	INPUT	125.1272	482.4649	12544	0	0	0	4356	3.734889	12.68897
education_num	INPUT	10.61264	2.59945	12544	0	1	10	16	-0.30945	0.368729
final_weight	INPUT	189554.4	106105.9	12544	0	14878	177675	1484705	1.634636	7.998727
hours_per_week	INPUT	42.17841	12.39242	12544	0	1	40	99	0.277618	2.959748

Figure 3.3.1: Dataset preview before modifying.

The above figure shows the dataset statistics after partitioning them into training and validation sets.

3.3.1 Replacement

We use the ‘Replacement’ node to perform cleaning on some values as shown in Figure 3.3.2. This step helps to prepare for the imputation step later.

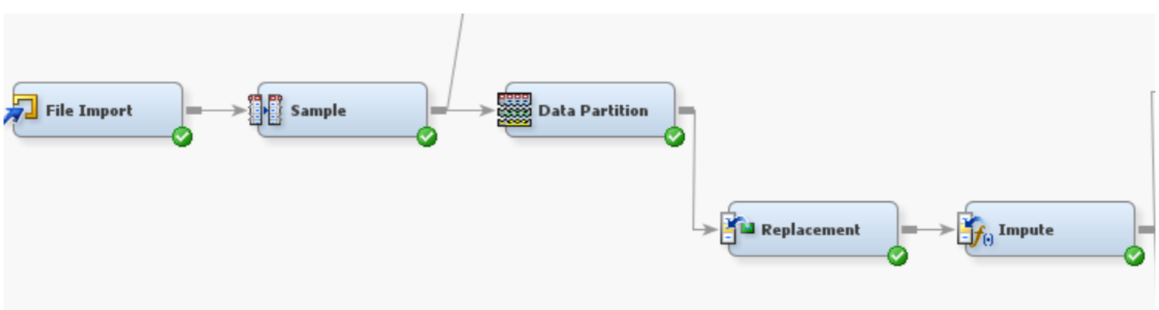


Figure 3.3.2: Snapshot of the “cleaning” steps in SAS Enterprise Miner.

The intentionally filled value in the “native_country” variable which is “?”, will be replaced with the “_MISSING_” keyword in SAS Enterprise Miner for imputation later, and other countries will be replaced with “Others”.

At the same time, we also replace the intentional data for “occupation” and “work_class” variables with “_MISSING_” as shown in figure below to be filled in using an inference algorithm in the next process.

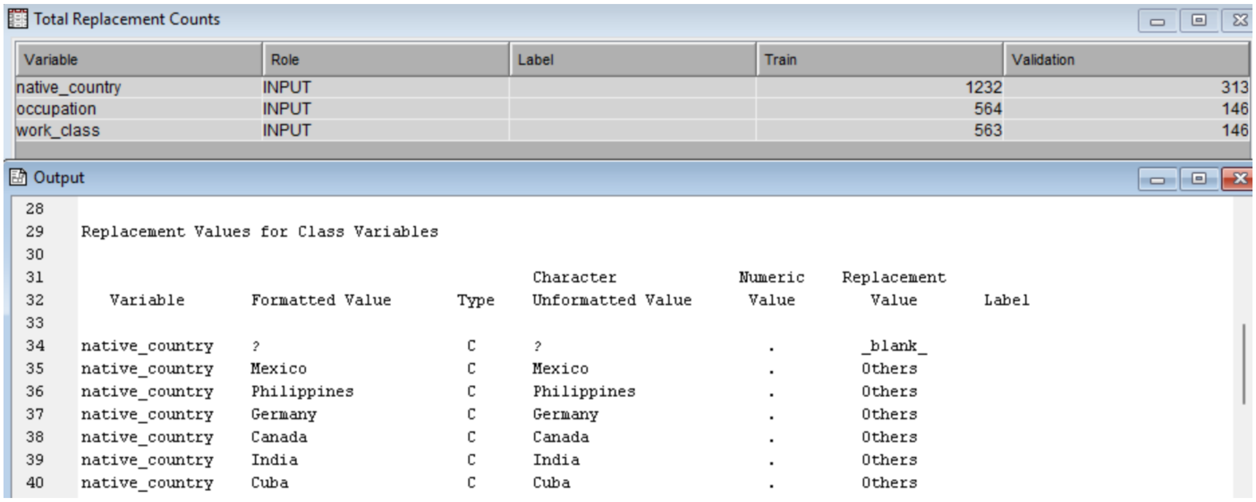
Variable	Formatted Value	Replacement Value	Frequency Count	Type	Character Unformatted Value	Numeric Value
marital_status	Widowed		304	C	Widowed	.
marital_status	Separated		301	C	Separated	.
marital_status	Married-spouse-absent		124	C	Married-spouse-absent	.
marital_status	Married-AF-spouse		12	C	Married-AF-spouse	.
marital_status	_UNKNOWN_	_DEFAULT_	.	C		.
native_country	United-States		11312	C	United-States	.
native_country	?	_MISSING_	233	C	?	.
native_country	Mexico	Others	172	C	Mexico	.
native_country	Philippines	Others	79	C	Philippines	.
native_country	Germany	Others	60	C	Germany	.
native_country	Canada	Others	57	C	Canada	.
native_country	India	Others	47	C	India	.
native_country	Cuba	Others	42	C	Cuba	.
native_country	England	Others	39	C	England	.
native_country	Italy	Others	39	C	Italy	.
native_country	El-Salvador	Others	36	C	El-Salvador	.
native_country	Puerto-Rico	Others	34	C	Puerto-Rico	.
native_country	South	Others	34	C	South	.
native_country	China	Others	30	C	China	.
native_country	Jamaica	Others	27	C	Jamaica	.
native_country	Taiwan	Others	26	C	Taiwan	.
occupation	?	_MISSING_	564	C	?	.
occupation	Handlers-cleaners		396	C	Handlers-cleaners	.

Figure 3.3.3: Replacement editor of the replacement node.

Summary of replacement values :

Variable	Value	Replacement Value
native_country	?	_MISSING_
	Mexico, Philippines… (countries other than US)	Others
occupation	?	_MISSING_
work_class	?	_MISSING_

After running the replacement node, the result in Figure 3.3.4 shows that the missing values in the dataset, which are 1232 observations for native_country, 564 observations for occupation and 563 observations for work_class are replaced. This step also includes setting countries that are not the United States as others.



Total Replacement Counts

Variable	Role	Label	Train	Validation
native_country	INPUT		1232	313
occupation	INPUT		564	146
work_class	INPUT		563	146

Output

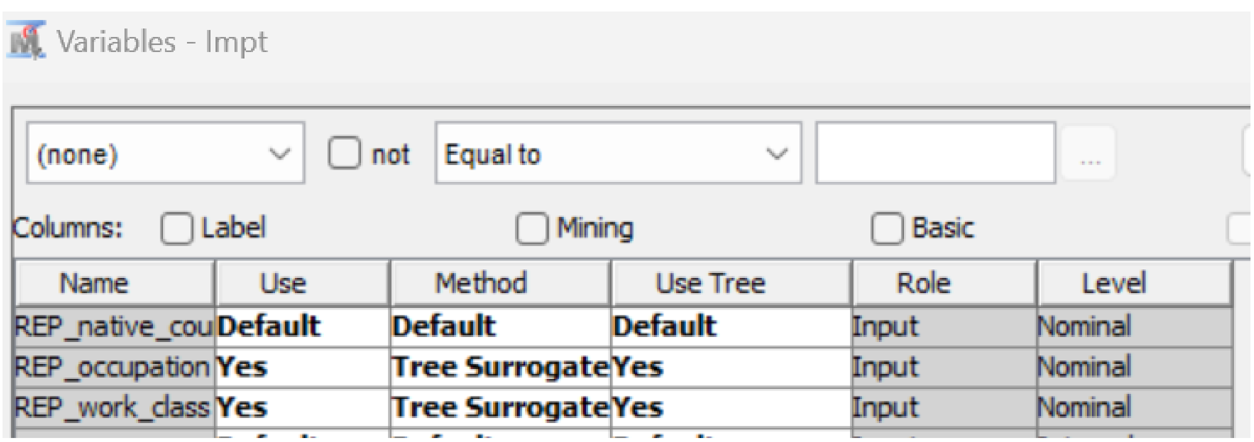
```
28
29   Replacement Values for Class Variables
30
31                                                  Character          Numeric   Replacement
32     Variable       Formatted Value      Type     Unformatted Value   Value       Value        Label
33
34   native_country   ?                      C      ?                     .        _blank_
35   native_country   Mexico                 C      Mexico                .        Others
36   native_country   Philippines            C      Philippines           .        Others
37   native_country   Germany                C      Germany               .        Others
38   native_country   Canada                 C      Canada                .        Others
39   native_country   India                  C      India                 .        Others
40   native_country   Cuba                   C      Cuba                  .        Others
```

Figure 3.3.4: Execution results up until the replacement node.

3.3.2 Imputation

Next, we use the “Impute” node to replace all missing values in REP_native_country with mode. Other than that, we also fill in the missing values in “workclass” and “occupation” variables by using an inference algorithm.

To perform tree surrogate, select edit variables at “Impute” node, and modify the value for “Use” and "Use Tree” to Yes for the “REP_workclass” and “REP_occupation” variables, the result from the replacement node. Besides, the “Method” is changed to “Tree Surrogate”. Surrogate splits are especially relevant in situations where there are missing values for the primary split variable and decides which branch to follow.



Variables - Impt

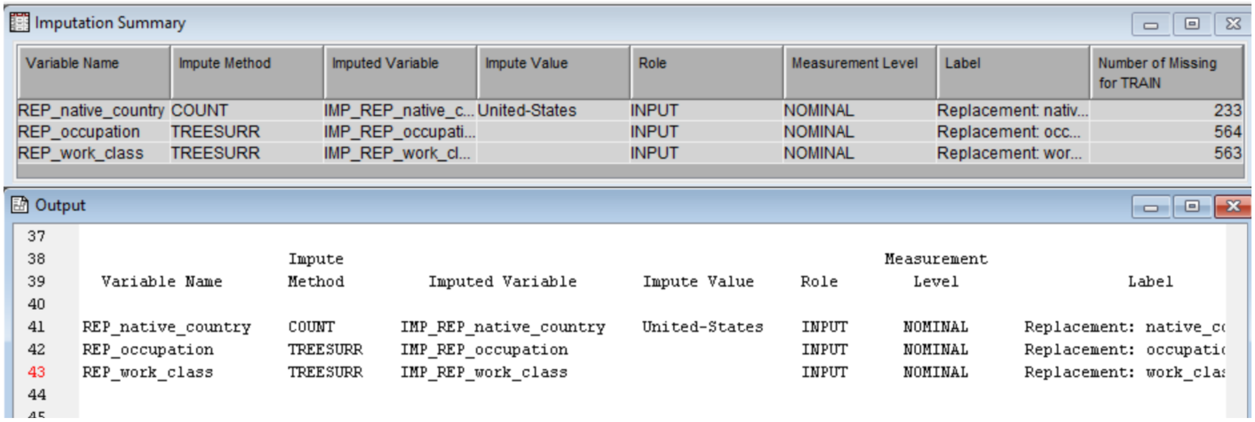
(none) | not | Equal to

Columns: Label | Mining | Basic

Name	Use	Method	Use Tree	Role	Level
REP_native_cou	Default	Default	Default	Input	Nominal
REP_occupation	Yes	Tree Surrogate	Yes	Input	Nominal
REP_work_class	Yes	Tree Surrogate	Yes	Input	Nominal

Figure 3.3.5: Variable editor of the imputation node.

Refer to Figure 3.3.6, 233 observations for REP_native_country are filled in with mode, United States while 564 observations for REP_occupation and 563 observations are filled in after running imputation.



Imputation Summary

Variable Name	Impute Method	Imputed Variable	Impute Value	Role	Measurement Level	Label	Number of Missing for TRAIN
REP_native_country	COUNT	IMP_REP_native_c...	United-States	INPUT	NOMINAL	Replacement: nativ...	233
REP_occupation	TREESURR	IMP_REP_occupati...		INPUT	NOMINAL	Replacement: occ...	564
REP_work_class	TREESURR	IMP_REP_work_cl...		INPUT	NOMINAL	Replacement: wor...	563

Output

```
                      Impute                                                    Measurement
   Variable Name      Method        Imputed Variable      Impute Value    Role       Level             Label

 REP_native_country   COUNT      IMP_REP_native_country   United-States   INPUT     NOMINAL     Replacement: native_co
 REP_occupation       TREESURR   IMP_REP_occupation                       INPUT     NOMINAL     Replacement: occupatio
 REP_work_class       TREESURR   IMP_REP_work_class                       INPUT     NOMINAL     Replacement: work_clas
```

Figure 3.3.6: Execution results up until the imputation node.

Figure 3.3.7 shows that all "?" values are now replaced after running the 'Impute' node.

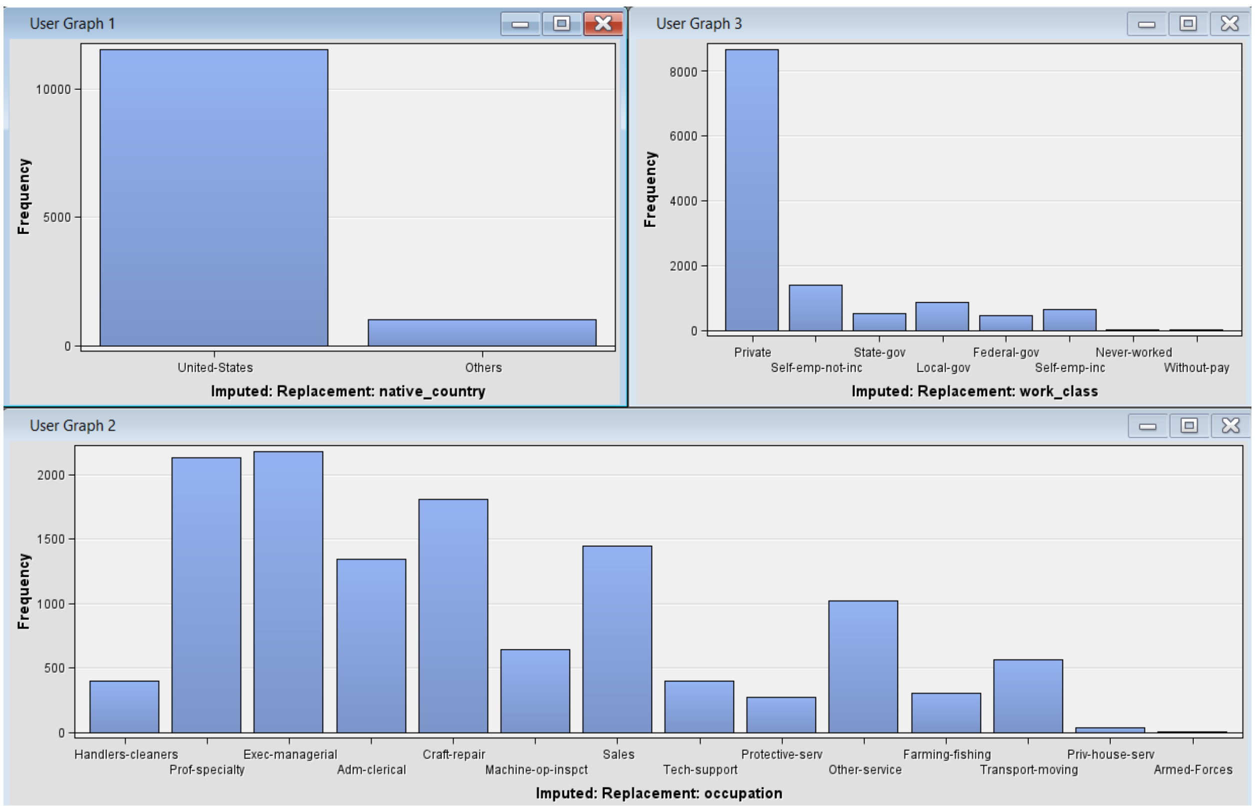


Figure 3.3.7: Bar plots of imputed variables.

3.3.4 Deletion of Variables

Lastly, some variables are identified to be dropped using the "Drop" node as they are not bringing meaning in the prediction process. Those variables are:

Variables	Reason
capital_loss capital_gain	Most of the values are ‘0’ that are not useful.
relationship	‘marital-status’ and ‘gender’ made up the ‘relationship’ variable. Keeping ‘relationship’ variable will be redundant.
final_weight	Irrelevant for predicting income.
education	Is another representation of “education_year”.

Before dropping the variables:

Name	Drop	Role	Level
IMP_REP_native	**Default**	Input	Nominal
IMP_REP_occup	**Default**	Input	Nominal
IMP_REP_work_	**Default**	Input	Nominal
WARN	**Default**	Assessment	Nominal
dataobs	**Yes**	ID	Interval
age	**Default**	Input	Interval
capital_gain	**Yes**	Input	Interval
capital_loss	**Yes**	Input	Interval
education	**Yes**	Input	Nominal
education_num	**Default**	Input	Interval
final_weight	**Yes**	Input	Interval
gross_income	**Default**	Target	Binary
hours_per_week	**Default**	Input	Interval
marital_status	**Default**	Input	Nominal
native_country	**Yes**	Rejected	Nominal
occupation	**Yes**	Rejected	Nominal
race	**Default**	Input	Nominal
relationship	**Yes**	Input	Nominal
sex	**Default**	Input	Nominal
work_class	**Yes**	Rejected	Nominal

Figure 3.3.8: List of variables before “Drop” node.

After running “Drop” node:

Name	Use	Report	Role	Level
IMP_REP_native	Default	No	Input	Nominal
IMP_REP_occup	Default	No	Input	Nominal
IMP_REP_work_	Default	No	Input	Nominal
age	Default	No	Input	Interval
education_num	Default	No	Input	Interval
gross_income	Default	No	Target	Binary
hours_per_week	Default	No	Input	Interval
marital_status	Default	No	Input	Nominal
race	Default	No	Input	Nominal
sex	Default	No	Input	Nominal

Figure 3.3.9: List of variables after "Drop" node.

3.3.2 Summary

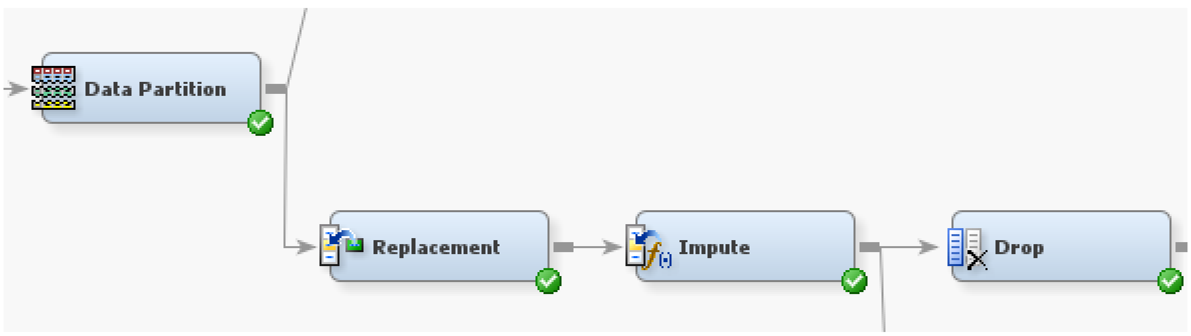


Figure 3.3.10: Entire pipeline of "Modify" phase based on SAS Enterprise Miner.

The "Modify" step includes "Replacement", "Impute", and "Drop". After running through the "Modify" pipeline, the following is the resulting variable for modeling:

Variable	Description
IMP_REP_native_country	Replace "?" with "_MISSING_", impute missing values with "United-States".
IMP_REP_occupation	Replace "?" with "_MISSING_", impute missing with inference algorithm.
IMP_REP_work_occupation	Replace "?" with "_MISSING_", impute missing with inference algorithm.
age	No change.
education_num	No change.
hours_per_week	No change.

marital_status	No change.
race	No change.
sex	No change.
gross_income (target)	No change.

3.4 Model

3.4.1 Decision Tree

The project incorporates the use of decision trees as one of its classification models. A decision tree is a non-parametric algorithm employed in supervised learning, suitable for both regression and classification tasks. Its structure comprises a hierarchical tree with a root node, branches, internal nodes, and leaf nodes (IBM, 2023).

In this project, we have used a decision tree node and set its “Maximum Depth” to 10 because we have 9 variables as input like what has been shown in Figure 3.4.1 below. Thus, using a depth of 10 will be most likely the best option for our case. Since, this is a classification problem, the parameters in the Subtree section such as Method that specify how to construct the sub-tree in terms of selection methods was set as Largest which selects the full tree, and the Assessment Measure was set to Misclassification. Cross validation parameter is also enabled to perform cross validation for each subtree in the sequence. The complete parameters configured for both trees are displayed in Figure 3.4.1.

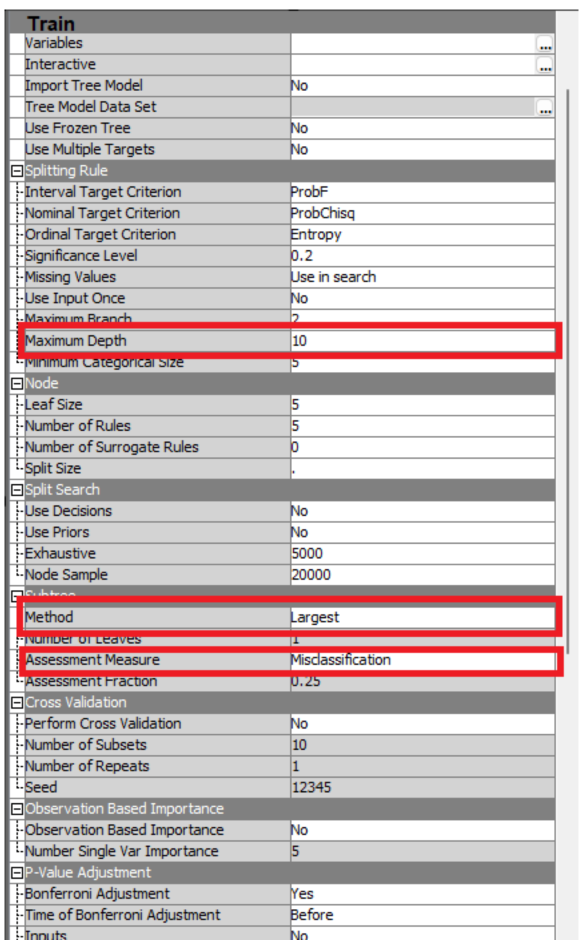


Figure 3.4.1: Parameters of decision tree

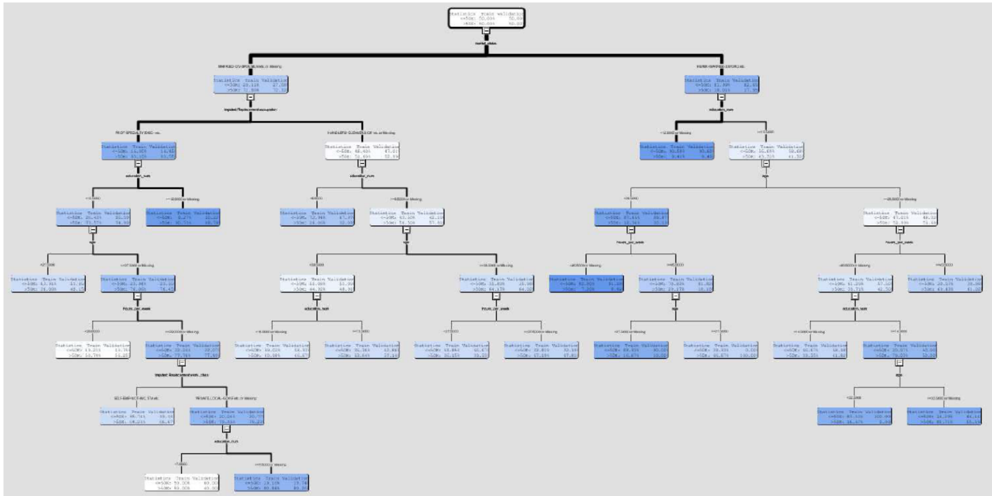


Figure 3.4.2: Architecture of decision tree

3.4.2 Support Vector Machine

In this project, SVM is also included as one of our classification models. Support Vector Machine (SVM) is a robust classification and regression technique that maximises the predictive accuracy of a model without overfitting the training data. SVM is particularly suited to analysing data with very large numbers (for example, thousands) of predictor fields (IBM, 2021).

The SVM model is created with the HP SVM node in SAS and below is the parameter values.

General	
Node ID	HPSVM
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Maximum Iterations	25
Use Missing as Level	No
Tolerance	1.0E-6
Penalty	1.0
Optimization Method	
Optimization Method	Interior Point
Interior Point Options	...
Active Set Options	...
Status	
Create Time	1/20/24 10:25 AM
Run ID	2010b0f7-6d34-4b45-9d3f-1b2987b7545f
Last Error	
Last Status	Complete
Last Run Time	1/20/24 10:29 AM
Run Duration	0 Hr. 0 Min. 8.53 Sec.
Grid Host	
User-Added Node	No

Figure 3.4.3: The parameter values of SVM

3.4.3 Random Forest

Random forest is another frequently used machine learning algorithm whereby it combines the output of multiple decision trees to reach a single result. Similar to decision tree algorithm, it is also capable of handling both classification and regression problems. Hence, we have decided to use random forest algorithm as one of the models to solve our classification problem.

We have use the HP Forest node as our random forest model. Almost all of the parameters are using the default value provided by SAS, which can be seen in Figure 3.4.4. The maximum number of trees parameter is altered and set to 40 instead of the default of 100. The reason behind that is to prevent overfitting. Random forests are a type of ensemble model, which means they are composed of multiple individual decision trees. As the number of trees in the forest increases, the model becomes more complex and may start to fit the noise in the training data, rather than the underlying pattern. This can result in poor generalization to new data. By limiting the number of trees in the forest, we can help to ensure that the model is able to generalize well to new data. Additionally, it's also important to note that as more trees are added, training time will increase and make the model more computationally expensive.

Train	
Variables	...
Tree Options	
Maximum Number of Trees	40
Seed	12345
Type of Sample	Proportion
Proportion of Obs in Each Sample	0.6
Number of Obs in Each Sample	.
Splitting Rule Options	
Maximum Depth	50
Missing Values	Use In Search
Minimum Use In Search	1
Number of Variables to Consider in Split Sea	.
Significance Level	0.05
Max Categories in Split Search	30
Minimum Category Size	5
Exhaustive	5000
Node Options	
Method for Leaf Size	Default
Smallest Percentage of Obs in Node	1.0E-5
Smallest Number of Obs in Node	1
Split Size	.
Use as Modeling Node	Yes
Score	
Variable Selection	Yes
Variable Importance Method	Loss Reduction
Number of Variables to Consider	25
Cutoff Fraction	0.01
Status	
Create Time	1/20/24 10:25 AM
Run ID	5180ac74-33da-d544-9849-d8cc8e9e80ce
Last Error	
Last Status	Complete
Last Run Time	1/21/24 12:54 PM
Run Duration	0 Hr. 0 Min. 6.49 Sec.
Grid Host	
User-Added Node	No

Figure 3.4.4: The parameter values of Random Forest

3.4.4 Gradient Boosting

Gradient boosting is a machine learning ensemble technique that combines the predictions of multiple weak learners, typically decision trees, sequentially. It aims to improve overall predictive performance by optimizing the model’s weights based on the errors of previous iterations, gradually reducing prediction errors and enhancing the model’s accuracy (Saini, 2024).

We opted for this algorithm due to its compatibility with the classification nature of our project's problem. Additionally, it is readily accessible in SAS Enterprise Miner. The project incorporates Gradient Boosting through the utilization of the Gradient Boosting node. Since this is a classification problem, the parameter in the Subtree section, “Assessment Measure” was set to Misclassification.

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	2
Minimum Categorical Size	5
Reuse Variable	1
Categorical Bins	30
Interval Bins	100
Missing Values	Use in search
Performance	Disk
Node	
Leaf Fraction	0.001
Number of Surrogate Rules	0
Split Size	.
Split Search	
Exhaustive	5000
Node Sample	20000
Subtree	
Assessment Measure	Misclassification
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
Status	
Create Time	1/20/24 10:25 AM
Run ID	23c1a799-8eaf-2f4c-987f-e476100c81fc
Last Error	
Last Status	Complete
Last Run Time	1/21/24 4:37 PM
Run Duration	0 Hr. 0 Min. 14.27 Sec.
Grid Host	
User-Added Node	No

Figure 3.4.5: The parameter values of Grafient Boosting

3.4.5 Neural Network (AutoNeural)

Neural networks, also known as artificial neural networks (ANNs) is a subset of machine learning and are at the heart of deep learning algorithms. It mimics the way that human biological neurons process and send signals to one another (Nicholson, 2020). In this project, we used neural networks as one of our classification models. We chose this algorithm because it is suitable for our project problem type (classification) and also it is available in SAS Enterprise Miner. Neural networks were implemented in the project by using the AutoNeural node.

We have use AutoNeural node as our neural network model and since the AutoNeural node offers an automatic way to explore alternative network architectures and hidden unit counts, there are less parameters to tune compared to Neural Network nodes. The train action is set to 'Search' so that the model will sequentially increase the network complexity. The number of hidden units is 2 and maximum iterations is 8. This means that for every two hidden unit added, 8 iterations will be run to find the optimum weights. The training process will be terminated if overfitting occurs. The tolerance is configured as 'Medium' to prevent preliminary training from occurring. The model parameters shown in Figure 3.4.6.

Train	
Variables	...
Model Options	
Architecture	Single Layer
Termination	Overfitting
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	No
Status	
Create Time	1/20/24 10:25 AM
Run ID	2e16d03a-343c-b245-9aee-b5b33cb6f244
Last Error	
Last Status	Complete
Last Run Time	1/20/24 10:28 AM
Run Duration	0 Hr. 2 Min. 45.16 Sec.
Grid Host	
User-Added Node	No

Figure 3.4.6: The parameter values of AutoNeural

3.5 Assess

3.5.1 Misclassification Rate

Misclassification rate is the percentage of times a machine learning model makes an incorrect prediction. Thus, a lower misclassification rate shows a better performing model. The misclassification rate of all the machine learning models can be obtained through the results of the Model Comparison node. The results are shown in Figure 3.5.1.

Selected Model	Model Node	Model Description	Valid: Misclassification Rate
Y	Boost	Gradient Boosting	0.19216
	HPDMForest	HP Forest	0.19503
	AutoNeural	AutoNeural	0.19981
	Tree	Decision Tree	0.20714
	HPSVM	HP SVM	0.21256

Figure 3.5.1: Misclassification rate of models

From the results, we can see that the best performing model is Gradient Boosting with a misclassification rate of 0.192, followed by random forest with misclassification rate of 0.195, AutoNeural with misclassification rate of 0.200, decision tree and finally support vector machine with misclassification rate of 0.207 and 0.213 respectively. The models look like they are performing well. However, misclassification rate also comes with its own limitations.

3.5.2 Precision

Precision is the ratio of correctly classified positive samples to a total number of classified positive samples. The formula of precision is as follows:

$$precision = \frac{TP}{TP + FP}$$

$$where:$$

$$TP = True\ positive$$

$$FP = False\ positive$$

Precision measures the reliability of the machine learning models when classifying results as positive and is helpful especially in situations where the dataset is severely imbalanced. To calculate precision, the classification matrix for all the models is shown in Figure 3.5.2.

Model Node	Model Description	Data Role	Target	Target Label	False Negative	True Negative	False Positive	True Positive
Tree	Decision Tree	TRAIN	gross_income		1177	5082	1190	5095
Tree	Decision Tree	VALIDATE	gross_income		328	1247	322	1241
HPSVM	HP SVM	TRAIN	gross_income		1054	4644	1628	5218
HPSVM	HP SVM	VALIDATE	gross_income		267	1169	400	1302
HPDMForest	HP Forest	TRAIN	gross_income		942	4928	1344	5330
HPDMForest	HP Forest	VALIDATE	gross_income		257	1214	355	1312
Boost	Gradient Boosting	TRAIN	gross_income		912	4821	1451	5360
Boost	Gradient Boosting	VALIDATE	gross_income		230	1196	373	1339
AutoNeural	AutoNeural	TRAIN	gross_income		811	4524	1748	5461
AutoNeural	AutoNeural	VALIDATE	gross_income		202	1144	425	1367

Figure 3.5.2: Classification matric for all models

From the results, the precision of each model can be calculated. The calculation results are shown in Table 2. Only the validation set is used to calculate precision.

Model	Precision
Decision Tree	0.794
HP SVM	0.765
HP Forest	0.787
Gradient Boosting	0.782
AutoNeural	0.763

Table 3.5.1: Precision of models

From the results, decision tree had the highest precision of 0.794, meaning that it has the highest reliability when predicting gross income.

3.5.3 Recall

Recall is the ratio between the number of positive samples correctly classified as positive to the total number of positive samples. The formula for recall is as follows:

$$recall = \frac{TP}{TP + FN}$$

$$where:$$

$$TP = True\ Positive$$

$$FN = False\ negative$$

Recall helps to measure the model's overall ability to detect positive samples. The higher the recall, the more positive samples were detected by the model. Recall is suitable for use when the dataset is imbalanced. From Figure 3.5.2, recall can be calculated and only the validation set was used to calculate recall. The calculation results are shown in Table 3.5.2.

Model	Recall
Decision Tree	0.791
HP SVM	0.830
HP Forest	0.836
Gradient Boosting	0.853
AutoNeural	0.871

Table 3.5.2: Recall of models

From the results, AutoNeural had the highest recall of 0.871, meaning that it has the highest overall ability to predict gross income.

3.5.4 F1-Score

F1-score is the harmonic mean of the precision and the recall. The goal of the F1-score is used to combine precision and recall into a single number. The formula for F1-score is as follows:

$$F1 = 2\,x\,\frac{Precision\ x\ Recall}{Precision + Recall}$$

From Table 3.5.1 and Table 3.5.2, we can calculate the F1-score for each machine learning model. The calculation results are shown in Table 3.5.3.

Model	F1-score
Decision Tree	0.792
HP SVM	0.796
HP Forest	0.811
Gradient Boosting	0.816
AutoNeural	0.814

Table 3.5.3: F1-score of models

From the results, the model with the best F1-score is Gradient Boosting, with an F1-score of 0.816. Thus, it can be concluded that overall, Gradient Boosting is the best performing model.

4.0 Conclusion

In conclusion, the project consisted of obtaining the raw data and preprocessing by adding column names and filling in missing values. The cleaned dataset was then exported and used as the data source for modeling work. Besides that, the 5 machine learning algorithms were trained on 1 set of data each. Next, the models were evaluated using misclassification rate, precision, recall and F1-score. The best performing model was Gradient boosting, with an F1-score of 0.816.

In terms of future works, the dataset could include more balanced data to correctly classify more categories of more than 50k gross income. Meanwhile, more machine learning models such as Logistic Regression can be fit using the Regression node, to compare more models to find the models with the best score. Lastly, most of the models that were used had default parameters. In the future, hyperparameter tuning could be done to further improve the machine learning model's predictive quality.

References

IBM. (2021, August 17). About SVM. Retrieved January 22, 2024, from https://www.ibm.com/docs/en/spss-modeler/saas?topic=models-about-svm

IBM. (2024, January 22). *What is a decision tree | IBM.* Retrieved January 22, 2024, from https://www.ibm.com/my-en/topics/decision-trees

Nicholson, C. (2020). A Beginner's Guide to Neural Networks and Deep Learning. Pathmind. Retrieved January 22, 2024, from https://wiki.pathmind.com/neural-network

Saini, A. (2024, January 10). *Gradient Boosting Algorithm: A complete guide for beginners*. Analytics Vidhya. Retrieved January 22, 2024 from https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/