MULTIMEDIA UNIVERSITY

Faculty Of Computing Informatics

Data Mining (TDP3301)

Trimester 2, 2022/2023

Data-Driven Analysis of Marital Satisfaction for Couples

Group Name: AIA

No.	Name	Id Number
1.	Ayat Abdulaziz Gaber Al-Khulaqi	1191202335
2.	Amin Ahmed Mohammedelhassan Alawad	1191302190
3.	Iven Low Zi Yin	1191202539

Contribution Table

Name	**Id**	**Section 1**	**Section 2**	**Section 3**
Ayat Abdulaziz Gaber Al-Khulaqi	1191202335		X	X
Amin Ahmed Mohammedelhassan Alawad	1191302190	X		
Iven Low Zi Yin	1191202539		X	

Abstract

This research project aims to analyze marital satisfaction among couples from 45 countries using data mining techniques. Marital satisfaction plays a crucial role in couples' well-being and overall mental health. By exploring and identifying predictors of marital satisfaction through classification, clustering, and association rule mining, we seek to develop a data-driven model that accurately assesses relationship quality. The study utilizes a survey dataset from married individuals, including various attributes related to marital satisfaction and demographic information. Descriptive analysis of the dataset reveals insights into the respondents' characteristics and the distribution of satisfaction levels. The findings will contribute to understanding the factors influencing marital satisfaction in 45 countries and help in developing interventions to promote healthier relationships.

Keywords: Data Mining, Classification, Clustering, Association Rule Mining (ARM), Descriptive Analysis.

Table of Contents

1. Introduction

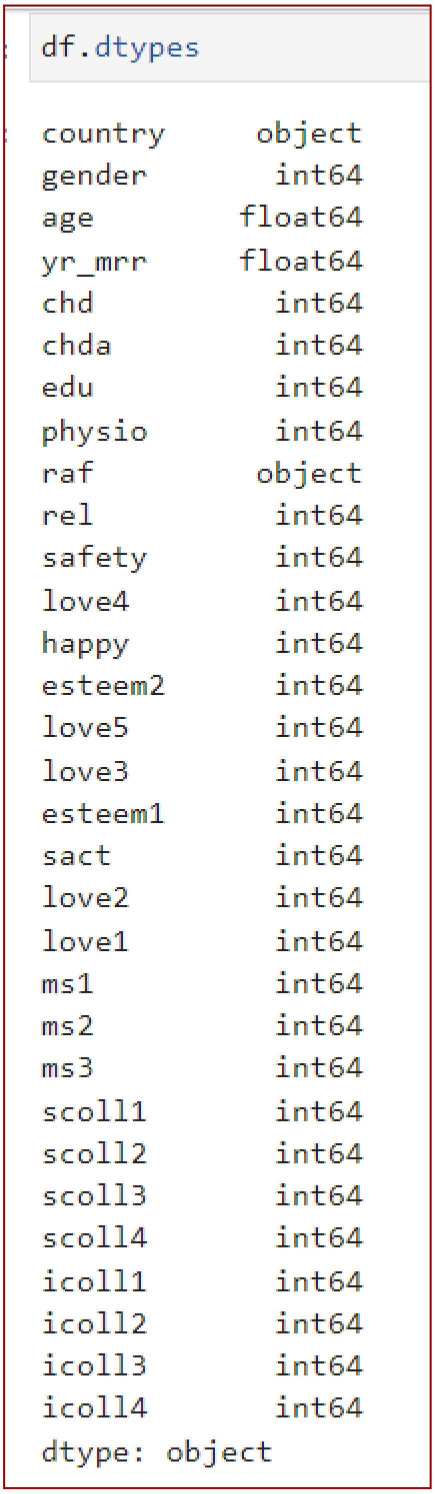
The satisfaction and well-being of couples in relationships have a significant impact on their happiness and overall mental health. Marital satisfaction, which serves as an essential measure of a couple's relationship quality, plays a crucial role in their overall well-being, happiness, and life satisfaction. Understanding the factors that contribute to marital satisfaction can cultivate healthy relationships and improve the overall quality of marital life.

However, the world continues to face challenges regarding divorce rates and family breakdown, leading to negative consequences such as poverty and social inequalities. Despite this, there is currently no suitable or highly accurate model available for the couples to assess and intervene in their relationship satisfaction. This study considers a wide range of variables, including their personal information, individual characteristics, cultural values, marital satisfaction and so on which will be discussed in the study later on.

In this study, a large-scale global dataset was provided for us to gain valuable insights and the factors of influencing marital satisfaction. This project aims to employ data mining techniques to investigate, explore data, and predictors of marital satisfaction, validate patterns, and establish causal relationships by using classification, clustering, and association rule mining techniques. By leveraging data-driven approaches rather than relying solely on traditional statistical methods, we aim to empirically develop a marital satisfaction model that can accurately assess the relationship quality of the couples.

2. About the Dataset

This survey study includes responses from married individuals in 45 countries around the world. The questionnaire consists of various questions aimed at understanding marital satisfaction and related factors. We have read through the survey questions in the questionnaire and reviewed all of the attributes in the dataset. The figure below it displays all data features included in the dataset:



```
df.dtypes

country      object
gender        int64
age         float64
yr_mrr      float64
chd           int64
chda          int64
edu           int64
physio        int64
raf          object
rel           int64
safety        int64
love4         int64
happy         int64
esteem2       int64
love5         int64
love3         int64
esteem1       int64
sact          int64
love2         int64
love1         int64
ms1           int64
ms2           int64
ms3           int64
scoll1        int64
scoll2        int64
scoll3        int64
scoll4        int64
icoll1        int64
icoll2        int64
icoll3        int64
icoll4        int64
dtype: object
```

1. **country**: Indicates the respondent's country.
2. **gender**: Indicates the respondent's gender, with options being Male (1) or Female (2).
3. **age**: Represents the respondent's age in years.
4. **yr_mrr**: Indicates the duration of the respondent's marriage in years.
5. **chd**: Represents the total number of biological children the respondent has.
6. **chda**: Indicates the number of children the respondent is currently raising in their family, which may include non-biological or adult children.
7. **edu**: Represents the respondent's level of education, ranging from no formal education (1) to a bachelor's or master's degree (5).
8. **physio**: Reflects the respondent's assessment of their material situation compared to the average in their country, with options ranging from much better (1) to much worse (5).
9. **raf**: Indicates the respondent's current religious affiliation, with multiple options such as Protestant (1), Catholic (2), Jewish (3), Muslim (4), Buddhist (5), None (6), Jehovah (7), Evangelic (8), Spiritualism (9), Other - very specific (10), Orthodox (11), and Hinduism (12).
10. **rel**: Represents the respondent's level of religiosity, ranging from Not at all (1) to Extremely Religious (7).

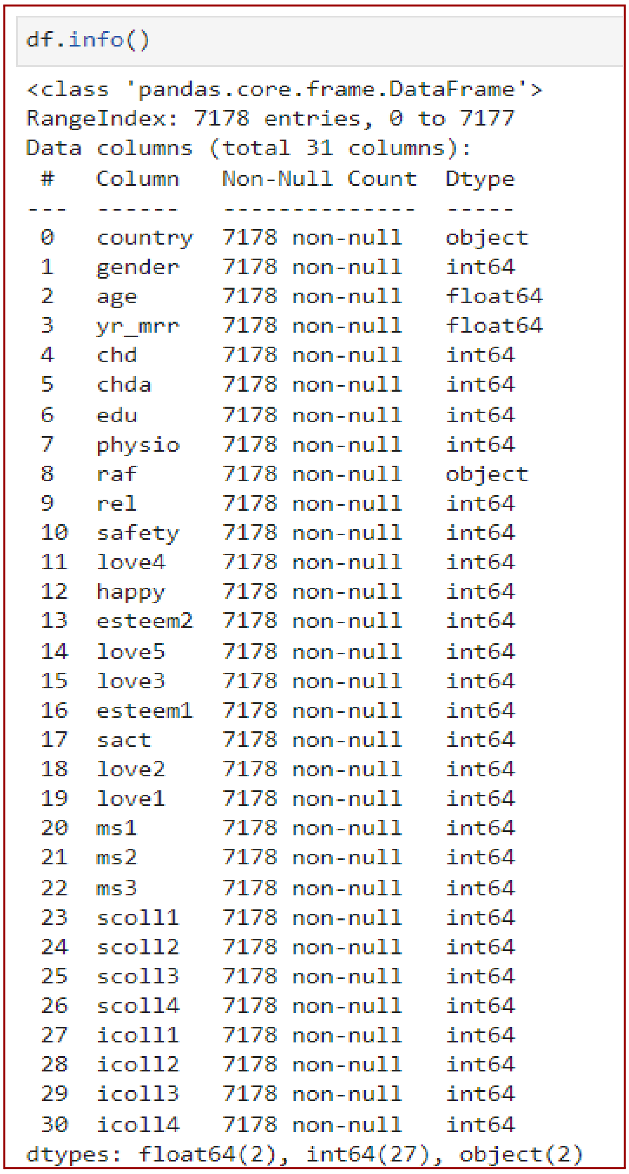
11. **safety**: Reflects the respondent's belief regarding their ability to rely on pension and social benefits when they get old, with options ranging from Agree strongly (+3) to Disagree strongly (-3).
12. **love4**: Indicates the respondent's level of enjoyment in their spouse's company, ranging from Yes (+2) to No (-2).
13. **happy**: Represents the respondent's level of happiness, ranging from Yes (+2) to No (-2).
14. **esteem2**: Indicates the respondent's perception of their spouse's attractiveness, ranging from Yes (+2) to No (-2).
15. **love5**: Represents the respondent's level of enjoyment in doing things together with their spouse, ranging from Yes (+2) to No (-2).
16. **love3**: Indicates the respondent's level of enjoyment in cuddling with their spouse, ranging from Yes (+2) to No (-2).
17. **esteem1**: Reflects the respondent's level of respect for their spouse, ranging from Yes (+2) to No (-2).
18. **sact**: Represents the respondent's sense of pride in their spouse, ranging from Yes (+2) to No (-2).
19. **love2**: Indicates whether the respondent's relationship has a romantic aspect, ranging from Yes (+2) to No (-2).
20. **love1**: Represents the respondent's level of love for their spouse, ranging from Yes (+2) to No (-2).
21. **ms1**: Three questions scales ranging from 1 to 7, where 1 represents very dissatisfied and 7 represents very satisfied, indicating the respondent's satisfaction with their marriage.
22. **ms2**: A scale ranging from 1 to 7, where 1 represents very dissatisfied and 7 represents very satisfied, indicating the respondent's satisfaction with their spouse as a spouse.
23. **ms3**: A scale ranging from 1 to 7, where 1 represents very dissatisfied and 7 represents very satisfied, indicating the respondent's satisfaction with their relationship with their spouse.
24. **scoll1**: Children's pride in parents' accomplishments on a 7-point Likert scale.
25. **scoll2**: Parents' pride in children's accomplishments on a 7-point Likert scale.
26. **scoll3**: Living arrangements of ageing parents on a 7-point Likert scale.
27. **scoll4**: Living arrangements of children until marriage on a 7-point Likert scale.

28. **icoll1**: Personal belief - children's pride in parents' accomplishments on a 7-point Likert scale.
29. **icoll2**: Personal belief - parents' pride in children's accomplishments on a 7-point Likert scale.
30. **icoll3**: Personal belief - living arrangements of aging parents on a 7-point Likert scale.
31. **icoll4**: Personal belief - living arrangements of children until marriage on a 7-point Likert scale.

3. Exploratory analysis

3.1 Overview of the data

The output of the code **df.info()** as seen in the *Figure 3.1.1*, presents information regarding a data frame named df. It reveals that the data frame comprises 7178 rows and 31 columns. These columns encompass diverse data types, namely object (string), int64 (integer), and float64 (a floating-point number). Each column is accompanied by its respective non-null count, representing the number of non-missing values. Additionally, the memory usage of the data frame is provided. Consequently, the output offers a succinct overview of the structure of the data frame and the data types associated with its columns.



```
df.info()
```

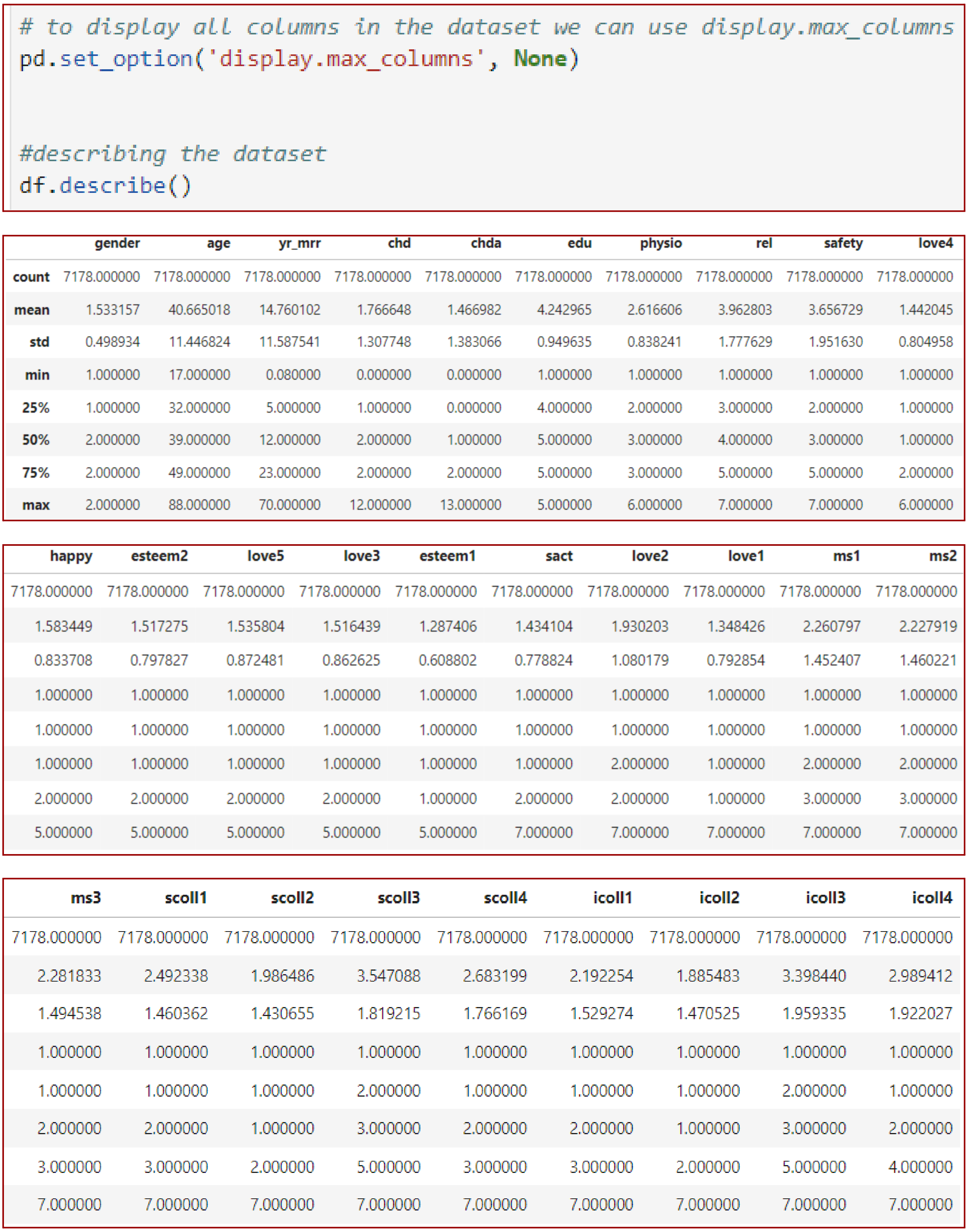
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7178 entries, 0 to 7177
Data columns (total 31 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   country  7178 non-null   object 
 1   gender   7178 non-null   int64  
 2   age      7178 non-null   float64
 3   yr_mrr   7178 non-null   float64
 4   chd      7178 non-null   int64  
 5   chda     7178 non-null   int64  
 6   edu      7178 non-null   int64  
 7   physio   7178 non-null   int64  
 8   raf      7178 non-null   object 
 9   rel      7178 non-null   int64  
 10  safety   7178 non-null   int64  
 11  love4    7178 non-null   int64  
 12  happy    7178 non-null   int64  
 13  esteem2  7178 non-null   int64  
 14  love5    7178 non-null   int64  
 15  love3    7178 non-null   int64  
 16  esteem1  7178 non-null   int64  
 17  sact     7178 non-null   int64  
 18  love2    7178 non-null   int64  
 19  love1    7178 non-null   int64  
 20  ms1      7178 non-null   int64  
 21  ms2      7178 non-null   int64  
 22  ms3      7178 non-null   int64  
 23  scoll1   7178 non-null   int64  
 24  scoll2   7178 non-null   int64  
 25  scoll3   7178 non-null   int64  
 26  scoll4   7178 non-null   int64  
 27  icoll1   7178 non-null   int64  
 28  icoll2   7178 non-null   int64  
 29  icoll3   7178 non-null   int64  
 30  icoll4   7178 non-null   int64  
dtypes: float64(2), int64(27), object(2)
```

Figure 3.1.1: Data Overview

Figure 3.1 provides the descriptive statistics of the dataset, we performed using describe() function. It returns the description of the data in the DataFrame and provides the summary statistics such as count, mean, standard deviation, and five-number summary. Looking at the result, it can be seen that:

- **age**: The age of the respondents is around 40.67 years, with a standard deviation of approximately 11.45. The minimum age is 17, and the maximum age is 88.
- **yr_mrr**: The average duration of marriage is approximately 14.76 years, with a standard deviation of around 11.59. The minimum duration is 0.08 years, and the maximum is 70 years.
- **edu**: education level of the respondents ranges from no formal education (coded as 1) to a bachelor's or master's degree (coded as 5). The mean value is approximately 4.24, indicating a relatively high level of education among the respondents.
- **physio**: reflects the respondents' assessment of their material situation compared to the average in their country. The mean value is approximately 2.62, suggesting that, on average, respondents perceive their material situation as somewhat better than the country's average.
- **rel**: The level of religiosity of the respondents ranges from "Not at all" (coded as 1) to "Extremely Religious" (coded as 7). The mean value is around 3.96, indicating a moderate level of religiosity among the respondents.
- **safety**: the respondents' belief regarding their ability to rely on pension and social benefits when they get old. The mean value is approximately 3.66, suggesting that, on average, respondents have a moderate belief in the reliability of these benefits.
- **love4**, **happy**, **esteem2**, **love5**, **love3**, **esteem1**, and **sact**: represent different aspects of the respondents' feelings and attitudes towards their spouse. The mean values range from 1.29 to 2.68, indicating varying degrees of enjoyment, happiness, attractiveness, and satisfaction in different aspects of the relationship.
- **love2** and **love1**: represent the presence of a romantic aspect and the level of love in the respondents' relationship, respectively. The mean values are 2.26 and 1.99, indicating a moderate level of romantic aspect and love in the relationship.
- **ms1**, **ms2**, and **ms3**: represent the respondents' satisfaction with their marriage, spouse as a spouse, and relationship with their spouse, respectively. The mean values range from 1.29 to 2.49, indicating moderate levels of satisfaction in these aspects.
- **coll1**, **scoll2**, **scoll3**, **scoll4**, **icoll1**, **icoll2**, **icoll3**, and **icoll4**: represent various aspects related to children's pride in parents' accomplishments, parents' pride in children's

accomplishments, and living arrangements of aging parents and children until marriage. They are measured on a 7-point Likert scale, and the mean values range from 2.19 to 3.55, indicating moderate levels of agreement or satisfaction in these aspects.



```
# to display all columns in the dataset we can use display.max_columns
pd.set_option('display.max_columns', None)


#describing the dataset
df.describe()
```

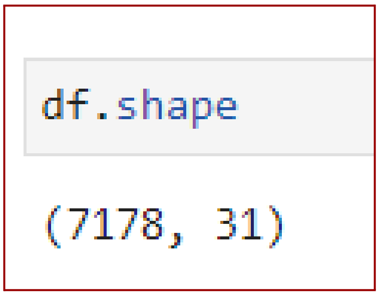
	gender	age	yr_mrr	chd	chda	edu	physio	rel	safety	love4
count	7178.000000	7178.000000	7178.000000	7178.000000	7178.000000	7178.000000	7178.000000	7178.000000	7178.000000	7178.000000
mean	1.533157	40.665018	14.760102	1.766648	1.466982	4.242965	2.616606	3.962803	3.656729	1.442045
std	0.498934	11.446824	11.587541	1.307748	1.383066	0.949635	0.838241	1.777629	1.951630	0.804958
min	1.000000	17.000000	0.080000	0.000000	0.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	1.000000	32.000000	5.000000	1.000000	0.000000	4.000000	2.000000	3.000000	2.000000	1.000000
50%	2.000000	39.000000	12.000000	2.000000	1.000000	5.000000	3.000000	4.000000	3.000000	1.000000
75%	2.000000	49.000000	23.000000	2.000000	2.000000	5.000000	3.000000	5.000000	5.000000	2.000000
max	2.000000	88.000000	70.000000	12.000000	13.000000	5.000000	6.000000	7.000000	7.000000	6.000000

	happy	esteem2	love5	love3	esteem1	sact	love2	love1	ms1	ms2
count	7178.000000	7178.000000	7178.000000	7178.000000	7178.000000	7178.000000	7178.000000	7178.000000	7178.000000	7178.000000
mean	1.583449	1.517275	1.535804	1.516439	1.287406	1.434104	1.930203	1.348426	2.260797	2.227919
std	0.833708	0.797827	0.872481	0.862625	0.608802	0.778824	1.080179	0.792854	1.452407	1.460221
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
50%	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	2.000000	1.000000	2.000000	2.000000
75%	2.000000	2.000000	2.000000	2.000000	1.000000	2.000000	2.000000	1.000000	3.000000	3.000000
max	5.000000	5.000000	5.000000	5.000000	5.000000	7.000000	7.000000	7.000000	7.000000	7.000000

	ms3	scoll1	scoll2	scoll3	scoll4	icoll1	icoll2	icoll3	icoll4
count	7178.000000	7178.000000	7178.000000	7178.000000	7178.000000	7178.000000	7178.000000	7178.000000	7178.000000
mean	2.281833	2.492338	1.986486	3.547088	2.683199	2.192254	1.885483	3.398440	2.989412
std	1.494538	1.460362	1.430655	1.819215	1.766169	1.529274	1.470525	1.959335	1.922027
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	1.000000	1.000000	1.000000	2.000000	1.000000	1.000000	1.000000	2.000000	1.000000
50%	2.000000	2.000000	1.000000	3.000000	2.000000	2.000000	1.000000	3.000000	2.000000
75%	3.000000	3.000000	2.000000	5.000000	3.000000	3.000000	2.000000	5.000000	4.000000
max	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000

Figure 3.1: Descriptive statistics of Data

Figure 3.2 shows the number of data contained in the dataset. By using shape it enables us to obtain the number of rows and columns of the data. It shows that we have 7178 rows of data and 31 columns of attributes.

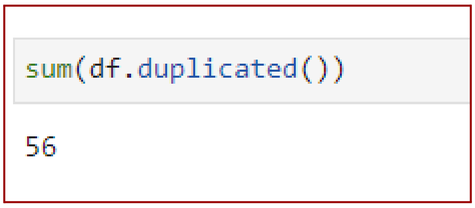


```
df.shape
```

```
(7178, 31)
```

Figure 3.2: Shape of the Data

Figure 3.3 shows that there are 56 duplicated rows in the dataset. However, it's important to note that the data was collected from a survey where each row represents an individual state, even if there are duplicates. Therefore, dropping these duplicated rows is not necessary since they represent distinct states.

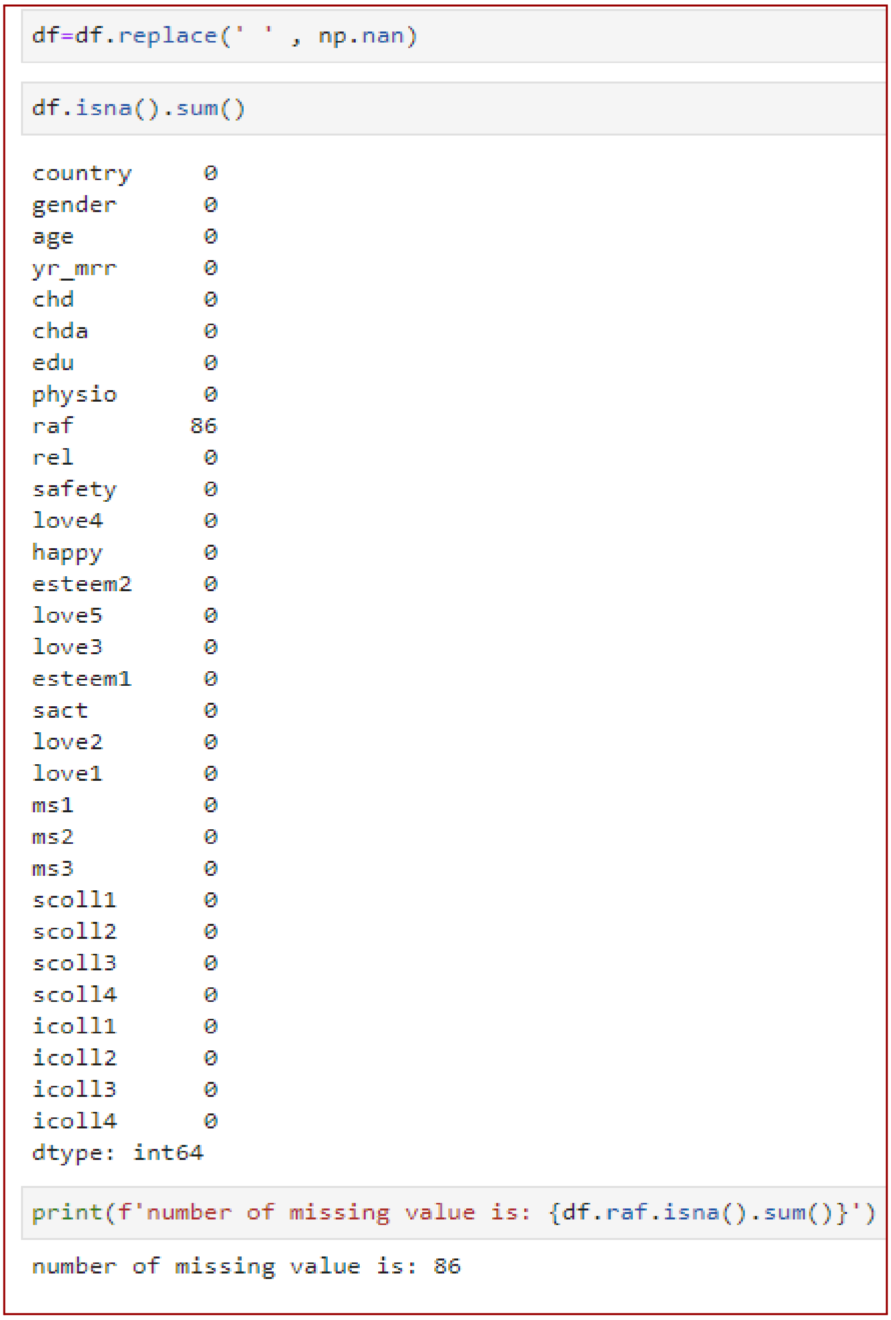


```
sum(df.duplicated())
```

```
56
```

Figure 3.3: Checking for duplicates

As seen in *Figure 3.4*, it illustrates the information of the data attributes which includes the data type and missing values in the dataset. The value zero indicates that there are no missing values in that particular column, but there are 86 missing values in the "RAF" column which represents the current religious affiliation. The numbers in the column represent the different religions of each person, we assume that the empty values do not represent any religion. As a result, we filled in the missing values with the number 6, which corresponds to none in the original survey as shown in *Figure 3.5*.



```
df=df.replace(' ' , np.nan)
```

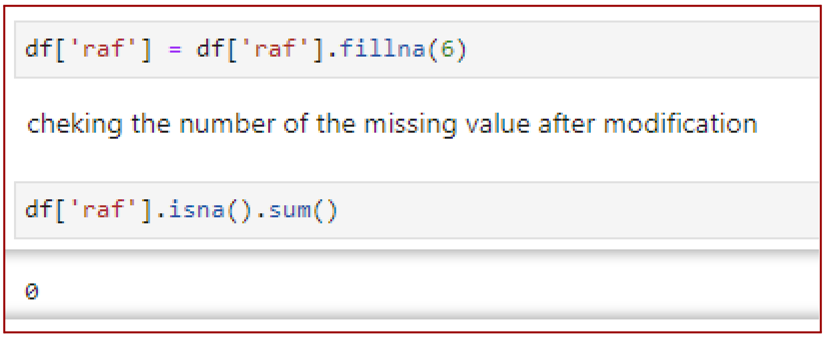
```
df.isna().sum()
```

```
country     0
gender      0
age         0
yr_mrr      0
chd         0
chda        0
edu         0
physio      0
raf        86
rel         0
safety      0
love4       0
happy       0
esteem2     0
love5       0
love3       0
esteem1     0
sact        0
love2       0
love1       0
ms1         0
ms2         0
ms3         0
scoll1      0
scoll2      0
scoll3      0
scoll4      0
icoll1      0
icoll2      0
icoll3      0
icoll4      0
dtype: int64
```

```
print(f'number of missing value is: {df.raf.isna().sum()}')
```

```
number of missing value is: 86
```

Figure 3.4: Missing Values



```
df['raf'] = df['raf'].fillna(6)
```

cheking the number of the missing value after modification

```
df['raf'].isna().sum()
```

```
0
```

Figure 3.5: Fill empty values

The skewness can be indicated based on the skewness value. If it is 0, then it indicates a perfectly symmetric distribution where the data is evenly distributed around the mean. A positive skewness value indicates a right-skewed or positively skewed distribution, where the tail of the distribution extends more towards higher values, and the majority of the data is concentrated towards the lower values. Conversely, a negative skewness value indicates a left-skewed or negatively skewed distribution, where the tail of the distribution extends more towards lower values, and the majority of the data is concentrated towards the higher values.

As seen in *Figure 3.6*, features such as chd, chda, love4, happy, esteem2, love5, love3, esteem1, sact, love2, love1, ms1, ms2, ms3, scoll1, scoll2, scoll4, and icoll1 all have positive values, indicating a right-skewed or positively skewed distribution. Age, yr_mrr, and icoll4 are close to 0, indicating an almost symmetric distribution. However, for edu, it has a negative skewness value, indicating a left-skewed or negatively skewed distribution.

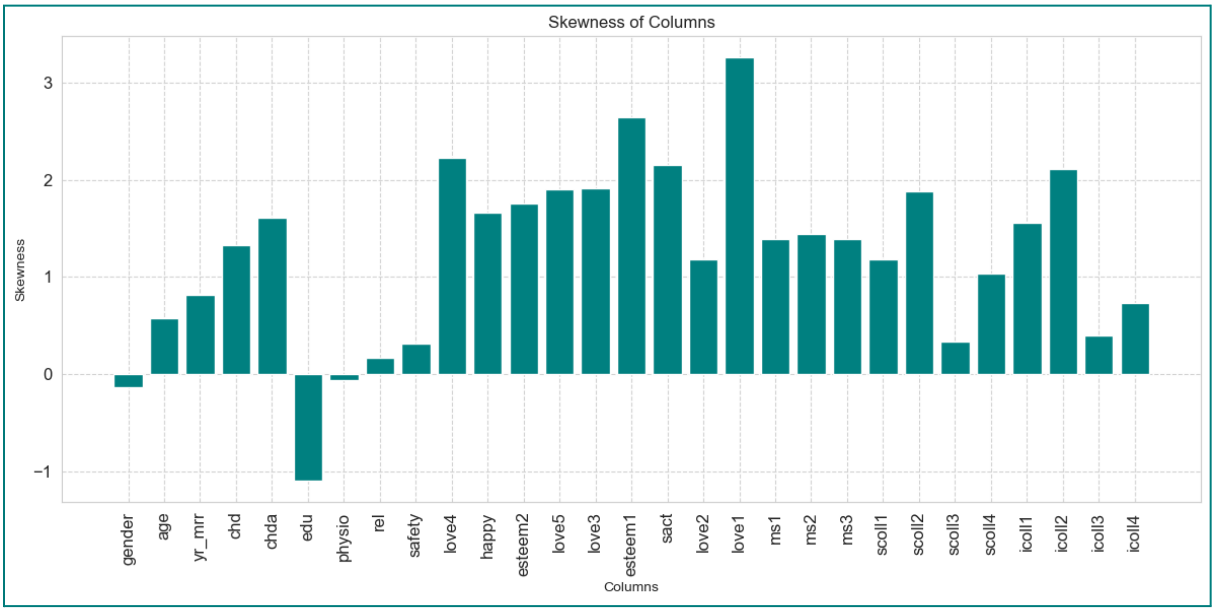


Figure 3.6: Skewness Analysis

Figure 3.7 represents the gender count in the dataset by percentage, it can be seen that there are 53.3% of females in the data, more than male which consisted of 46.7%.

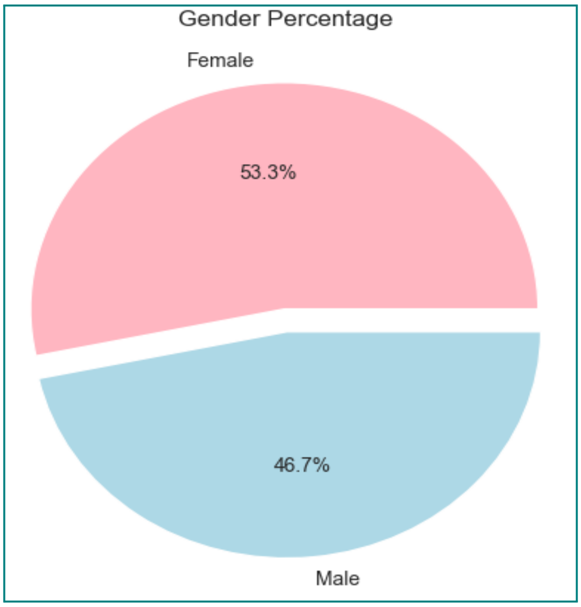


Figure 3.7: Gender Percentage in Pie Chart

The graph below illustrates the distribution of ages within the dataset. The graph reveals that the age of 30.0 exhibits the highest frequency, closely followed by ages 32.0 and 35.0. As the age increases, there is a gradual decrease in frequency, with a noticeable decline commencing from age 50.0. Certain age groups demonstrate relatively lower frequencies, notably those above 54.0 and below 20.0. These age categories are depicted by shorter bars on the graph, indicating a smaller population size.

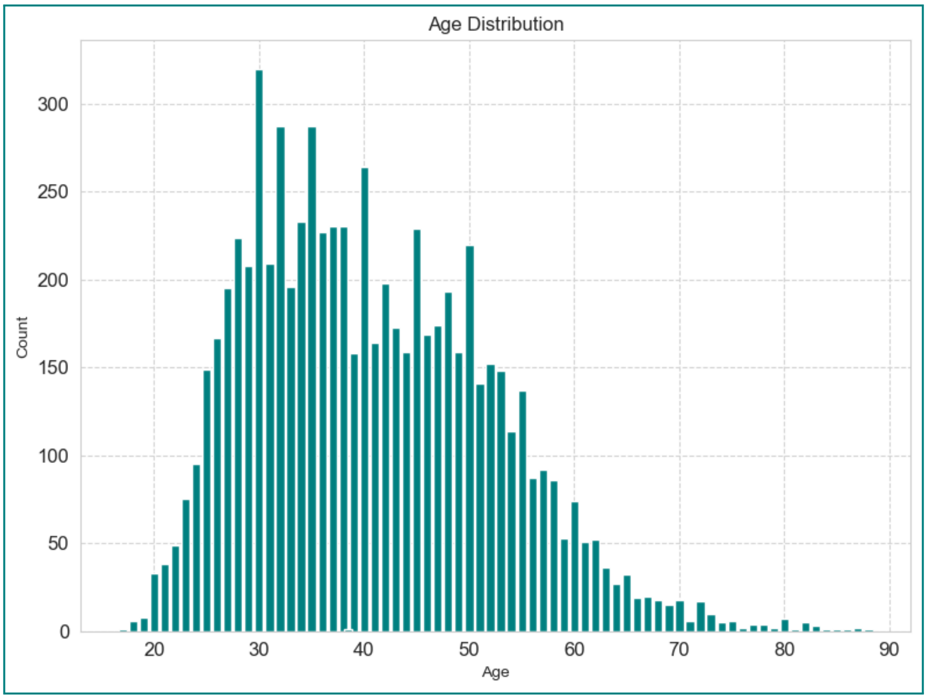


Figure 3.8: Age Distribution

In *Figure 3.9*, each data point represents an age group on the x-axis and the corresponding years of marriage on the y-axis. The plot shows a positive association between age and years of marriage, indicating that as a person gets older, they tend to have been married for a longer duration.

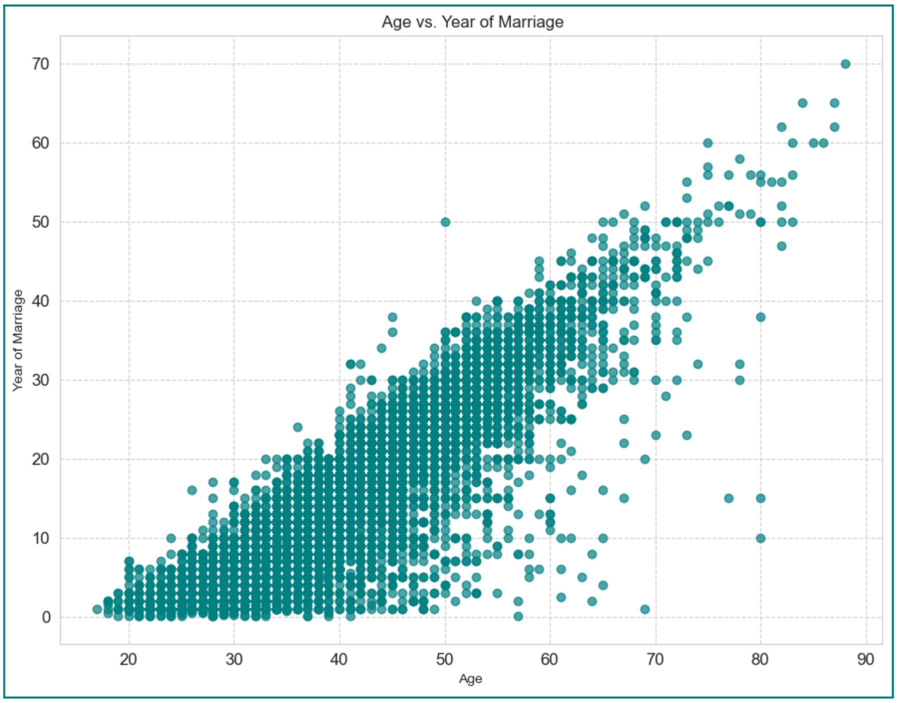


Figure 3.9: Age compared to Years of marriage

In *Figure 3.10*, the bar chart represents the total number of respondents from each country. Croatia, Iran, and Nigeria have the highest number of respondents, while Indonesia, Canada, and Romania have the lowest number of respondents.

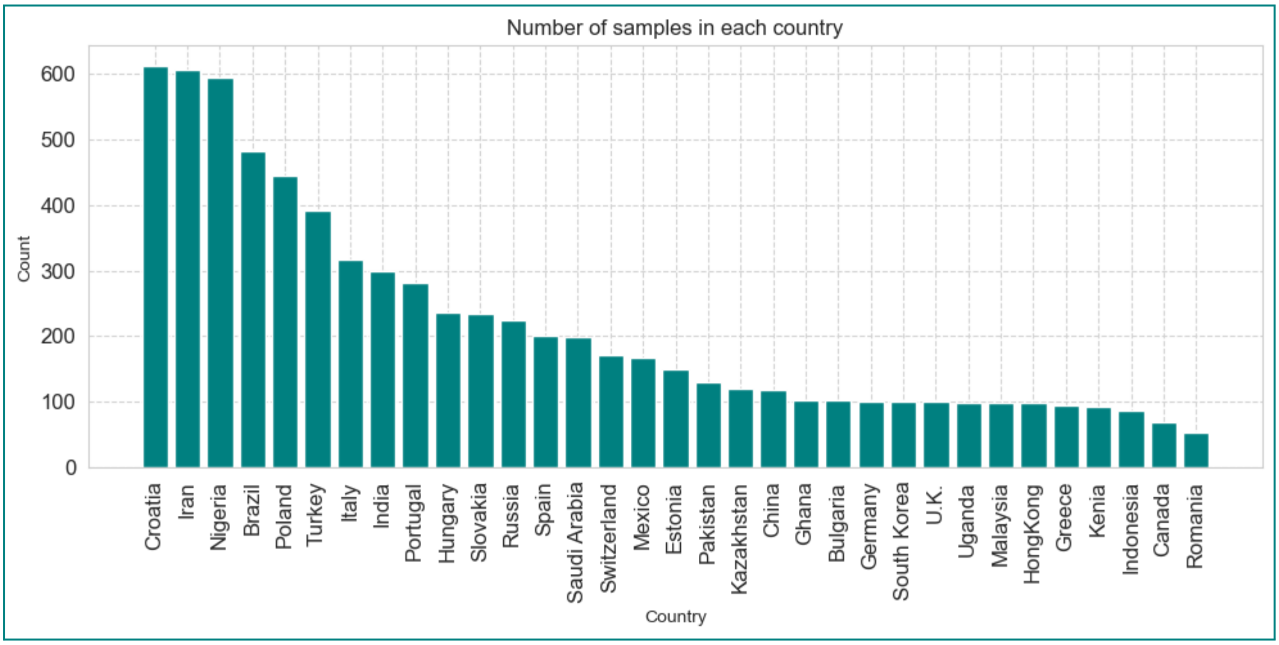


Figure 3.10: Count Number of Samples in each Country

The interactive map presented in *Figure 3.11* showcases markers that represent each country. These markers offer details about the country's name and the count of respondents from that particular country. This visualization provides a visual overview of the respondent distribution across various countries, effectively highlighting those with the highest and lowest number of respondents.

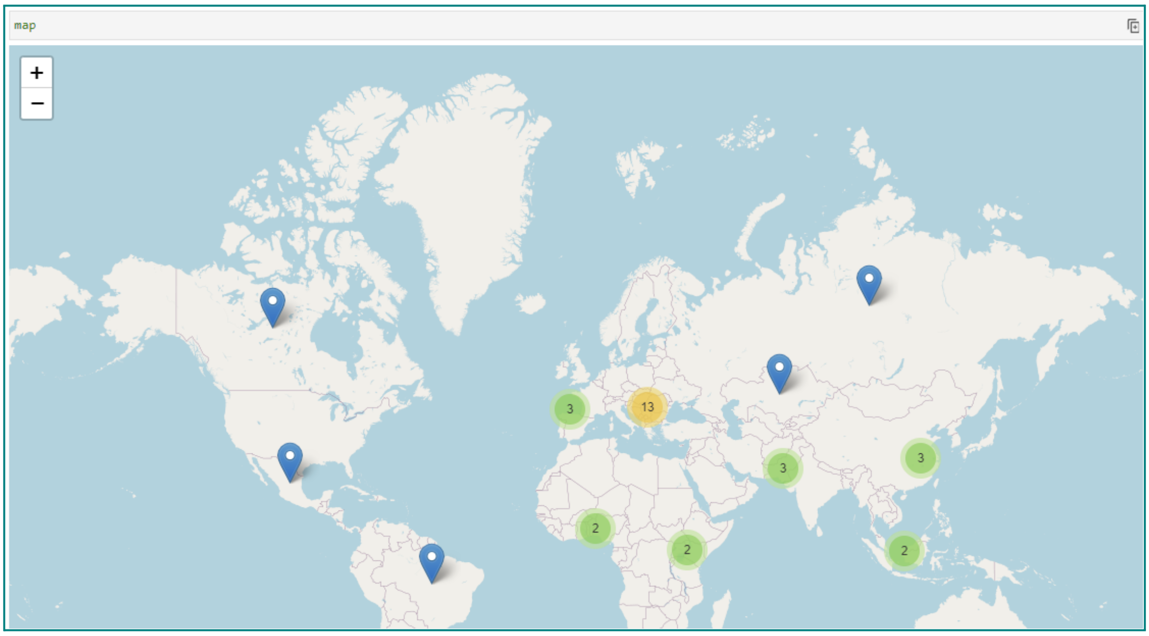


Figure 3.11: Plotting the countries on the Map

The correlation matrix HeatMap, shown in *Figure 3.12*, illustrates the strongest and weakest associations between variables. Among the highly correlated pairs are 'ms1' and 'ms2' (correlation coefficient: 0.89), 'ms2' and 'ms3' (correlation coefficient: 0.88), and 'age' and 'yr_mrr' (correlation coefficient: 0.88). Conversely, the pairs with the weakest correlations include 'edu' and 'yr_mrr' (correlation coefficient: -0.24), 'physio' and 'edu' (correlation coefficient: -0.21), and 'edu' and 'chd' (correlation coefficient: -0.20).

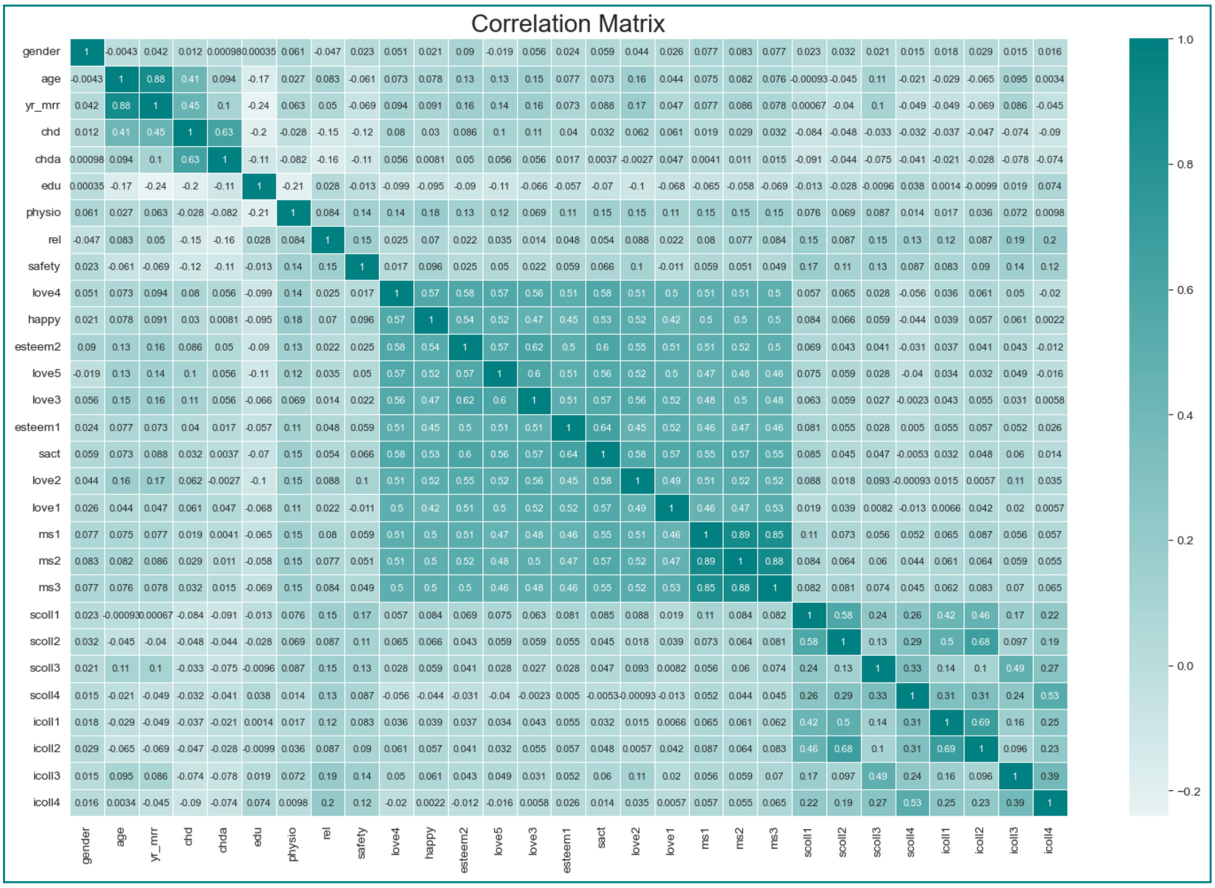


Figure 3.12: Correlation Matrix (Heat Map)

As seen in *Figure 3.13* majority of respondents for ms1 rated their satisfaction between 1 (very dissatisfied) and 5. A smaller number of respondents rated their satisfaction as 6 or 7 (very satisfied). For ms2, the responses were similar to ms1, with most respondents falling in the range of 1 to 5. There were fewer respondents who rated their satisfaction as 6 or 7. Lastly, for ms3, the majority of respondents expressed satisfaction levels between 1 and 5, with a smaller proportion rating their satisfaction as 6 or 7.

Based on the results, we can conclude that the levels of satisfaction reported by the respondents vary across different aspects of their marriage and relationship. The majority of respondents rated their satisfaction levels between 1 and 5, indicating a range of satisfaction levels from very dissatisfied to moderately satisfied. However, there is also a notable proportion of respondents who reported higher levels of satisfaction, rating their experiences as 6 or 7 (very satisfied).

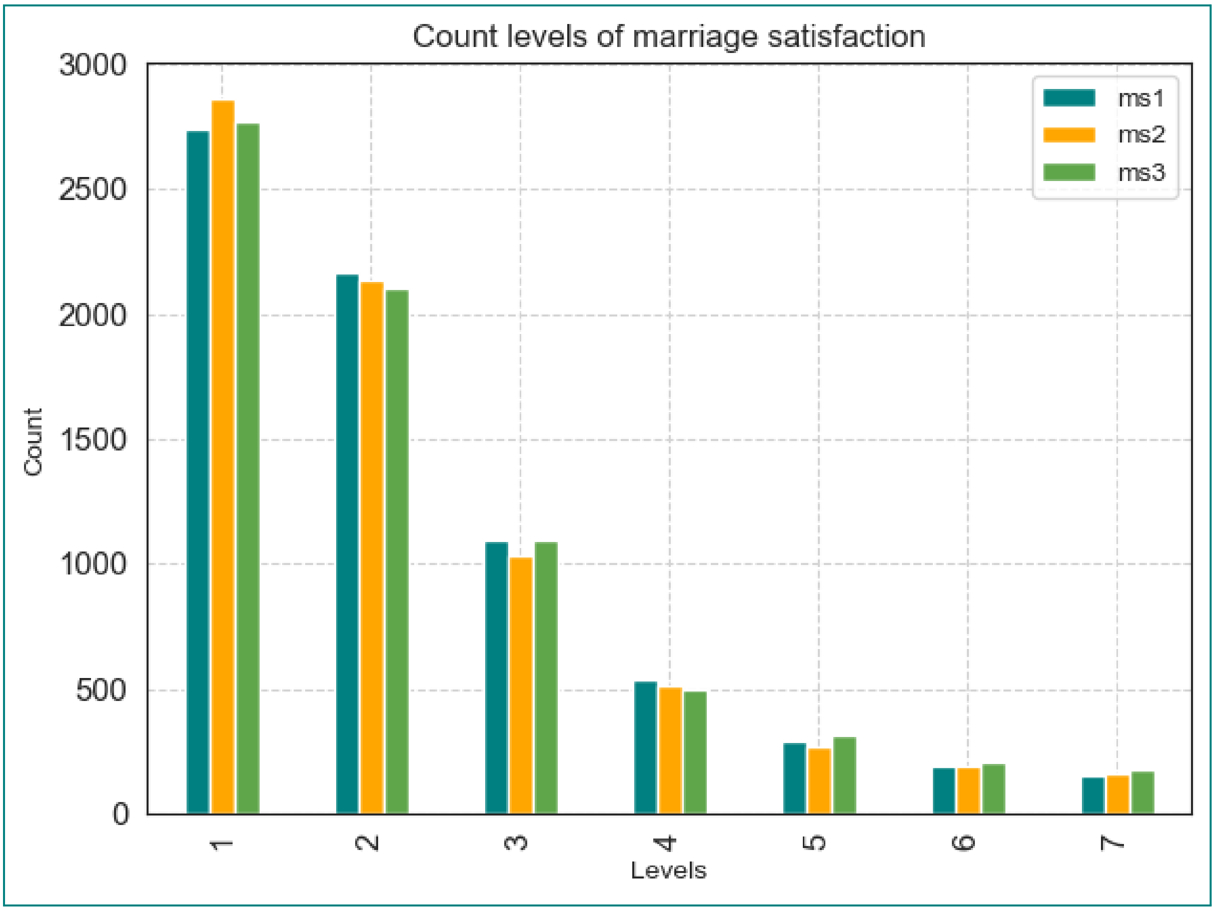


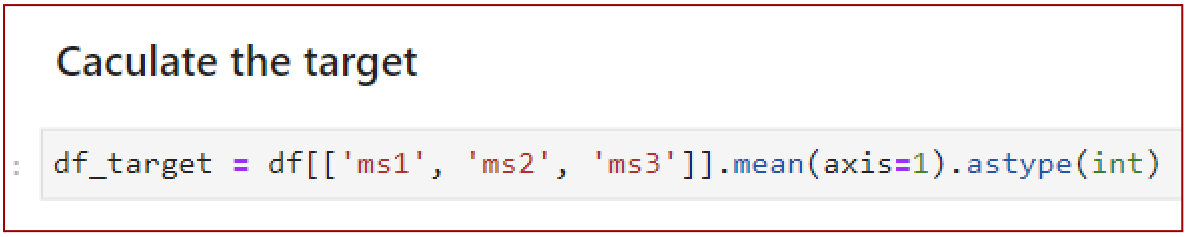
Figure 3.13: Comparing the columns that represent marriage satisfaction

4. Feature selection

In this section, we will discuss our features selected for each machine learning technique. Feature selection is the process of selecting a subset of relevant features for use in model construction. Feature selection is able to reduce computational time, cost of the modelling, and improve the performance of the model. Based on our explore and finding on the datasets, we have found that there are a few conditions that needs to be fulfilled when choosing the feature selection method:

- The dataset has 30 features, eliminating low correlated features is necessary to reduce computational expenses during training.
- Flexibility: The method can be applied with different machine learning techniques.
- Automated searching process without manual feature engineering is desired.
- The method should enhance interpretability by reducing the dataset's dimensionality.

With the conditions above, we have identified that **Recursive Feature Elimination** (RFE) satisfies these conditions. RFE eliminates low correlated features, it is able to work with various algorithms, it offers an automated feature selection process, and improves interpretability by reducing dimensionality. Therefore, RFE is the suitable method for feature selection. The goal of RFE is to select features by recursively considering smaller and smaller sets of features. For the target, we will be setting up ms1, ms2, and ms3. Taking up the mean of these values as our target variable as shown in figure 4.1.

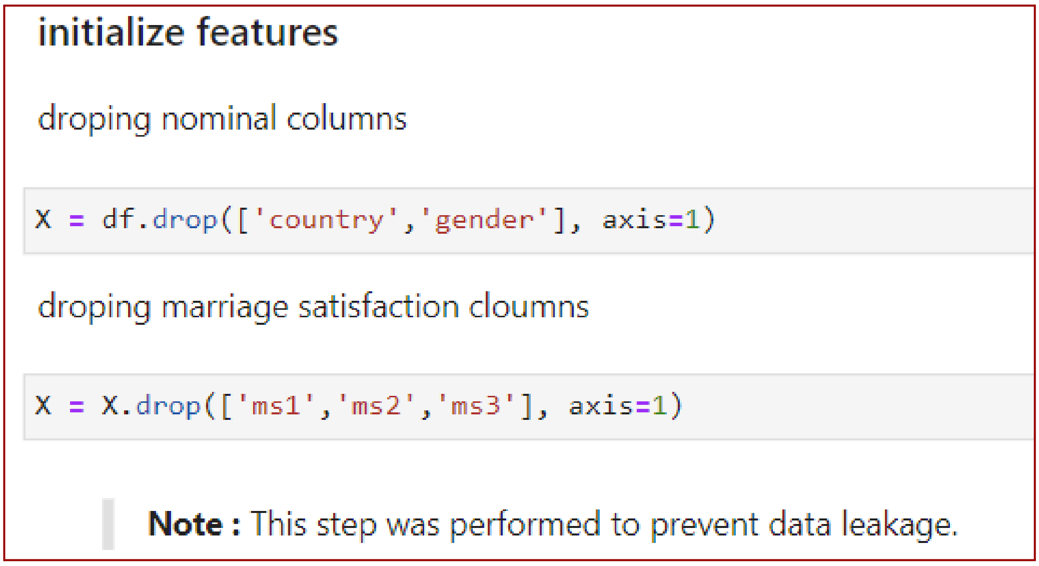


Caculate the target

```
df_target = df[['ms1', 'ms2', 'ms3']].mean(axis=1).astype(int)
```

Figure 4.1: Target Variable

As seen in *Figure 4.2* we dropped country and gender from the original data frame because they are not relevant or informative for predicting marital satisfaction. Also, form X we dropped ms1, ms2 and ms3, which represent the marital satisfaction scores, were also dropped from the feature matrix X. This is done to prevent target leakage, which means using information in the features that would not be available in a real-world scenario when making predictions.



initialize features

droping nominal columns

```
X = df.drop(['country','gender'], axis=1)
```

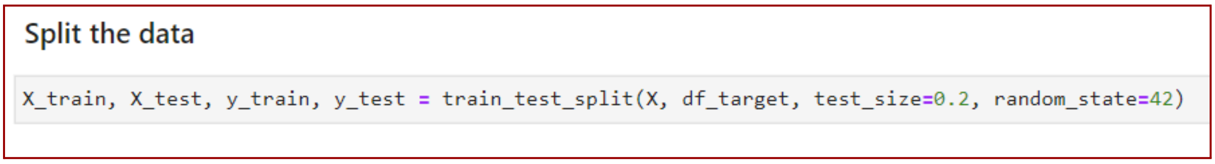
droping marriage satisfaction cloumns

```
X = X.drop(['ms1','ms2','ms3'], axis=1)
```

> **Note :** This step was performed to prevent data leakage.

Figure 4.2: Initialize Features

After preparing the data, the code divides the dataset into two separate sets: a training set and a testing set as shown in *Figure 4.3*. This is done to evaluate the performance of the model on unseen data. The train_test_split function is used for this purpose. The testing set is 20% of the total dataset, while the training set is the remaining 80%. The random state parameter is set to 42, ensuring that the same random split is reproducible.

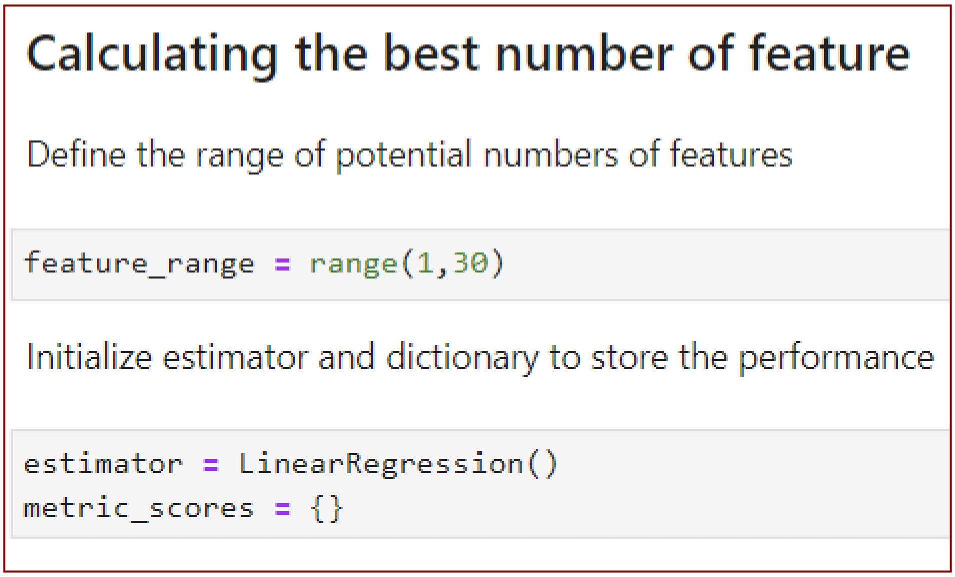


Split the data

```
X_train, X_test, y_train, y_test = train_test_split(X, df_target, test_size=0.2, random_state=42)
```

Figure 4.3: Split the Data

The code illustrated in *Figure 4.4* considers different numbers of features from 1 to 29. For each number of features, it uses a Linear Regression model and a technique called Recursive Feature Elimination (RFE). RFE helps identify the most important features by repeatedly removing the least important ones. This process is performed for each number of features in the range.



Calculating the best number of feature

Define the range of potential numbers of features

```
feature_range = range(1,30)
```

Initialize estimator and dictionary to store the performance

```
estimator = LinearRegression()
metric_scores = {}
```

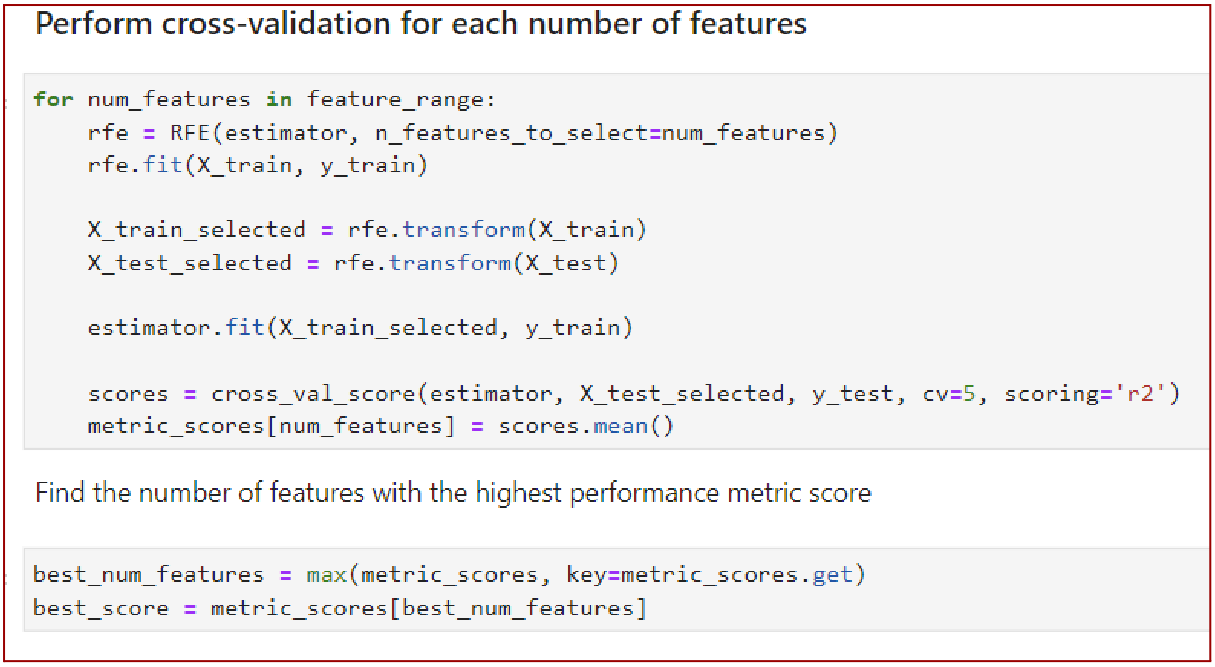
Figure 4.4: Calculating the Best Number of Feature

After selecting the most important features, they are transformed for both the training and testing datasets using the RFE transform as seen in *Figure 4.5*. Then, the estimator (Linear Regression) is trained using the selected features and the training data.

To evaluate the model's performance, cross-validation is performed on the transformed testing data using a technique called 5-fold cross-validation. This means that the testing data is divided into five equal parts, and the model is trained and tested five times, each time using a different part as the testing set. Performance scores, specifically R-squared scores, are calculated to measure how well the model fits the data.

The average performance score across the five cross-validation folds is calculated and stored in the metric_scores dictionary for each number of features. This dictionary keeps track of the performance scores for different numbers of features.

To find the best number of features, the code selects the key (number of features) with the highest value (performance score) from the metric_scores dictionary using the max function and specifying the key=metric_scores.get parameter. This identifies the number of features that resulted in the highest performance score.



Perform cross-validation for each number of features

```
for num_features in feature_range:
    rfe = RFE(estimator, n_features_to_select=num_features)
    rfe.fit(X_train, y_train)

    X_train_selected = rfe.transform(X_train)
    X_test_selected = rfe.transform(X_test)

    estimator.fit(X_train_selected, y_train)

    scores = cross_val_score(estimator, X_test_selected, y_test, cv=5, scoring='r2')
    metric_scores[num_features] = scores.mean()
```

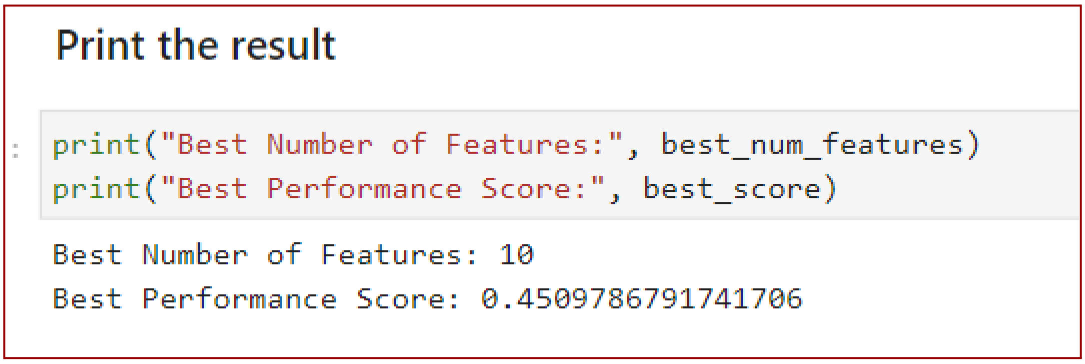
Find the number of features with the highest performance metric score

```
best_num_features = max(metric_scores, key=metric_scores.get)
best_score = metric_scores[best_num_features]
```

Figure 4.5: Perform Cross-Validation for each Number of Features

The best number of features is found to be 10 as shown in *Figure 4.6*, indicating that out of all the available features, including 10 of them provides the optimal result for predicting marital satisfaction.

The best performance score achieved is 0.451 (45.1%), the low performance score could be due to the selected features not capturing all the relevant information, the linear regression model's limitations, potential dataset limitations or noise, and the exclusion of influential factors. Improvements are needed to accurately predict marital satisfaction.



Print the result

```
print("Best Number of Features:", best_num_features)
print("Best Performance Score:", best_score)
```

```
Best Number of Features: 10
Best Performance Score: 0.4509786791741706
```

Figure 4.6: Displaying the Results

5. Pattern discovery

Based on our exploration and findings of the dataset, we have researched, tried performing other data mining techniques and we came up with the idea of performing three classification techniques, one clustering, and one association rule mining techniques. We have selected these methods, namely Decision Trees, Naive Bayes, and Support Vector Machine (SVM), K-Means clustering, and Association Rule Mining (ARM). Here are the reasons for selecting these methods:

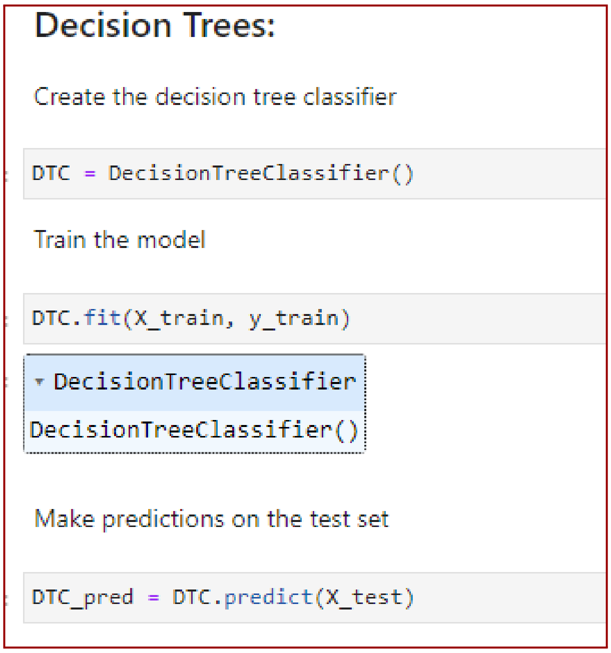
5.1 Classification

The selection of these three methods, namely Decision Trees, Naive Bayes, and Support Vector Machines (SVM), is based on their versatility, popularity, and effectiveness in various data mining tasks. Here are the reasons for selecting these methods:

5.1.1 Decision Tree

- Interpretability: Decision trees provide clear and interpretable rules for decision-making. The generated tree structure can be easily understood and visualized, making it useful for explaining patterns and insights to stakeholders.
- Non-Parametric Nature: Decision trees do not require strong assumptions about the underlying data distribution, making them suitable for different types of data.

The code shown in *Figure 5.1.1.1* implements a decision tree classifier for making predictions. It trains the classifier using the training data and then uses it to predict the output labels for the test data. The decision tree classifier is created, trained with the training data, and the predictions are stored in the variable DTC_pred.



Decision Trees:

Create the decision tree classifier

```
DTC = DecisionTreeClassifier()
```

Train the model

```
DTC.fit(X_train, y_train)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

Make predictions on the test set

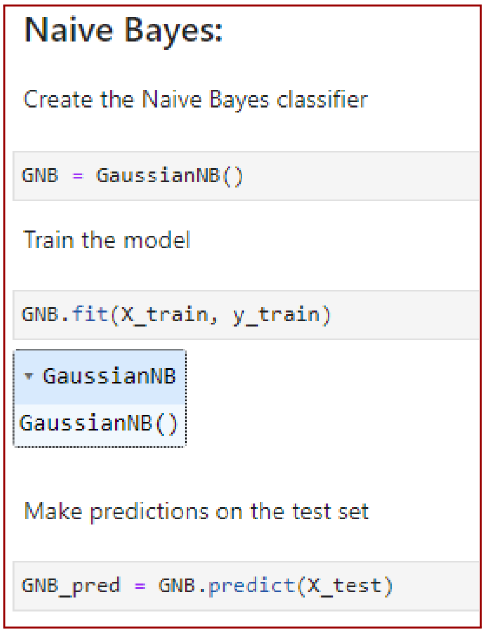
```
DTC_pred = DTC.predict(X_test)
```

Figure 5.1.1.1: Decision Tree

5.1.2 Naive Bayes

- Simplicity and Efficiency: Naive Bayes is computationally efficient and simple to implement. It performs well even with limited training data, making it suitable for large datasets.
- Probabilistic Framework: Naive Bayes leverages probability theory to calculate the likelihood of an instance belonging to a certain class. It can handle missing values even though there are none in our data and works well in high - dimensional data.
- Independence Assumption: Although the assumption of independence between features is often violated in real-world scenarios, Naive Bayes can still provide reasonable results and performs particularly well when the independence assumption holds or is approximately true.

As shown in *Figure 5.1.2.1* the code implements the Naive Bayes classifier. It creates the classifier, trains it with the training data, and uses it to make predictions on the test set. The predictions are stored in the variable GNB_pred.



Naive Bayes:

Create the Naive Bayes classifier

```
GNB = GaussianNB()
```

Train the model

```
GNB.fit(X_train, y_train)
```

```
▾ GaussianNB
GaussianNB()
```

Make predictions on the test set

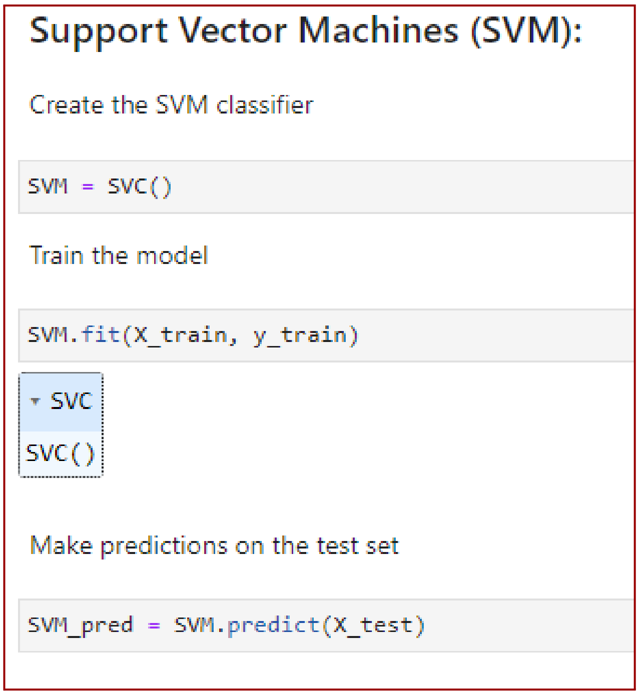
```
GNB_pred = GNB.predict(X_test)
```

Figure 5.1.2.1: Naive Bayes

5.1.3 Support Vector Machines (SVM)

- Effective in High-Dimensional Spaces: SVM performs well in high-dimensional feature spaces, making it suitable for our data.
- Interpretable Results: SVM is less sensitive to outliers compared to other algorithms. The use of the kernel trick allows SVM to capture complex relationships between data points.
- Non-Hierarchical Clustering: SVM can be used for both classification and regression tasks. It offers different kernel functions, such as linear, polynomial, and radial basis function (RBF), providing flexibility in capturing various data patterns.

The code applies Support Vector Machines (SVM) as seen in *Figure 5.1.3.1*, a machine learning technique, to create a classifier. The SVM classifier is trained using the training data and then used to predict labels for the test set. The predicted labels are stored in the variable SVM_pred. In summary, this code demonstrates the implementation of SVM for classification tasks, training the model, and generating predictions.



Support Vector Machines (SVM):

Create the SVM classifier

```
SVM = SVC()
```

Train the model

```
SVM.fit(X_train, y_train)
```

```
▾ SVC
SVC()
```

Make predictions on the test set

```
SVM_pred = SVM.predict(X_test)
```

Figure 5.1.3.1: Support Vector Machines (SVM)

5.2 Clustering

To find the optimal number of clusters, we have performed using the Elbow Method. It iterates through different numbers of clusters (ranging from 1 to 10) and performs the K-means clustering algorithm for each number. The within-cluster sum of squares (WCSS) is calculated and stored in a list.

Next, a plot is generated as shown in *Figure 5.2.1* to visualize the WCSS values for different numbers of clusters. The plot helps identify the "elbow" point, which indicates a significant drop in WCSS. In this case, the plot suggests that 5 clusters may be a reasonable choice.

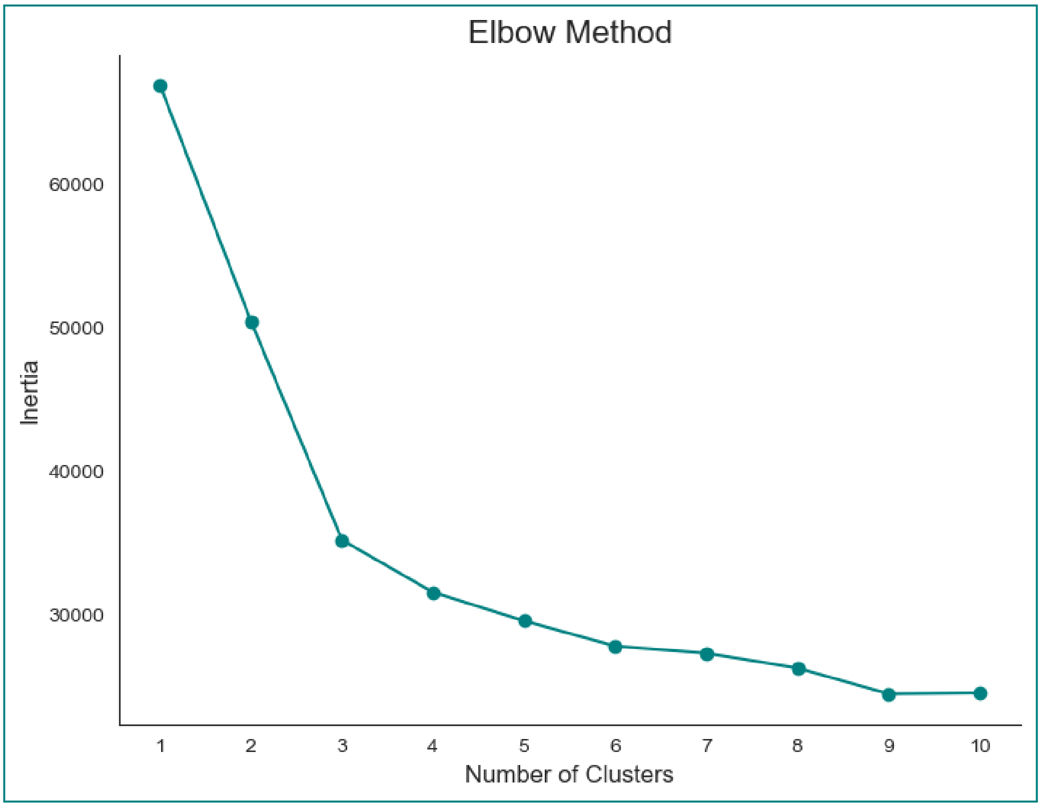
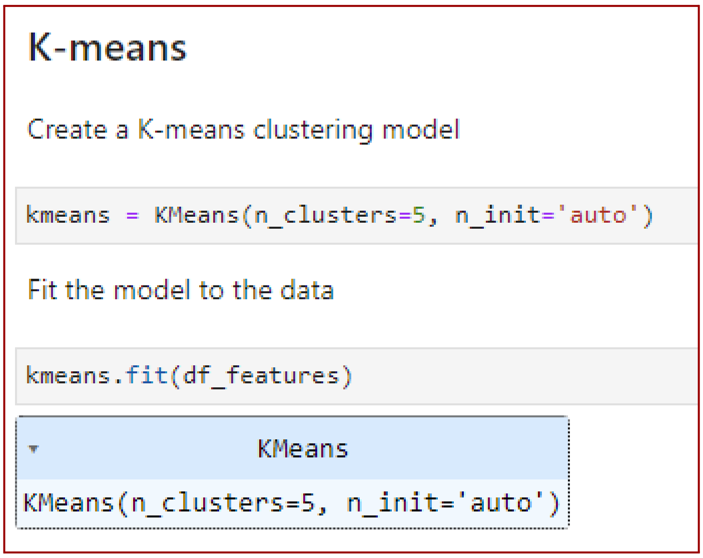


Figure 5.2.1: Elbow Method

5.2.1 K-means Clustering

- Simplicity and Efficiency: K-means is a simple and computationally efficient clustering algorithm. It is easy to understand and implement, making it suitable for a wide range of datasets.
- Interpretable Results: K-means produces clusters that are represented by their centroids. These centroids can be easily interpreted and analyzed to gain insights into the characteristics of the different clusters. This interpretability is valuable for understanding the underlying structure of the data.
- Non-Hierarchical Clustering: Unlike hierarchical clustering algorithms that create a tree-like structure, K-means generates flat clusters. This characteristic is beneficial when the aim is to assign each data point to a single cluster rather than capturing nested or overlapping clusters.

Subsequently, a K-means clustering model is created with 5 clusters, and the model is fitted to the dataset. The code as seen in *Figure 5.2.1.1* allows us to analyze the dataset's clustering structure and determine an appropriate number of clusters based on the Elbow Method.



K-means

Create a K-means clustering model

```
kmeans = KMeans(n_clusters=5, n_init='auto')
```

Fit the model to the data

```
kmeans.fit(df_features)
```

```
KMeans
KMeans(n_clusters=5, n_init='auto')
```

Figure 5.2.1.1: K-Means

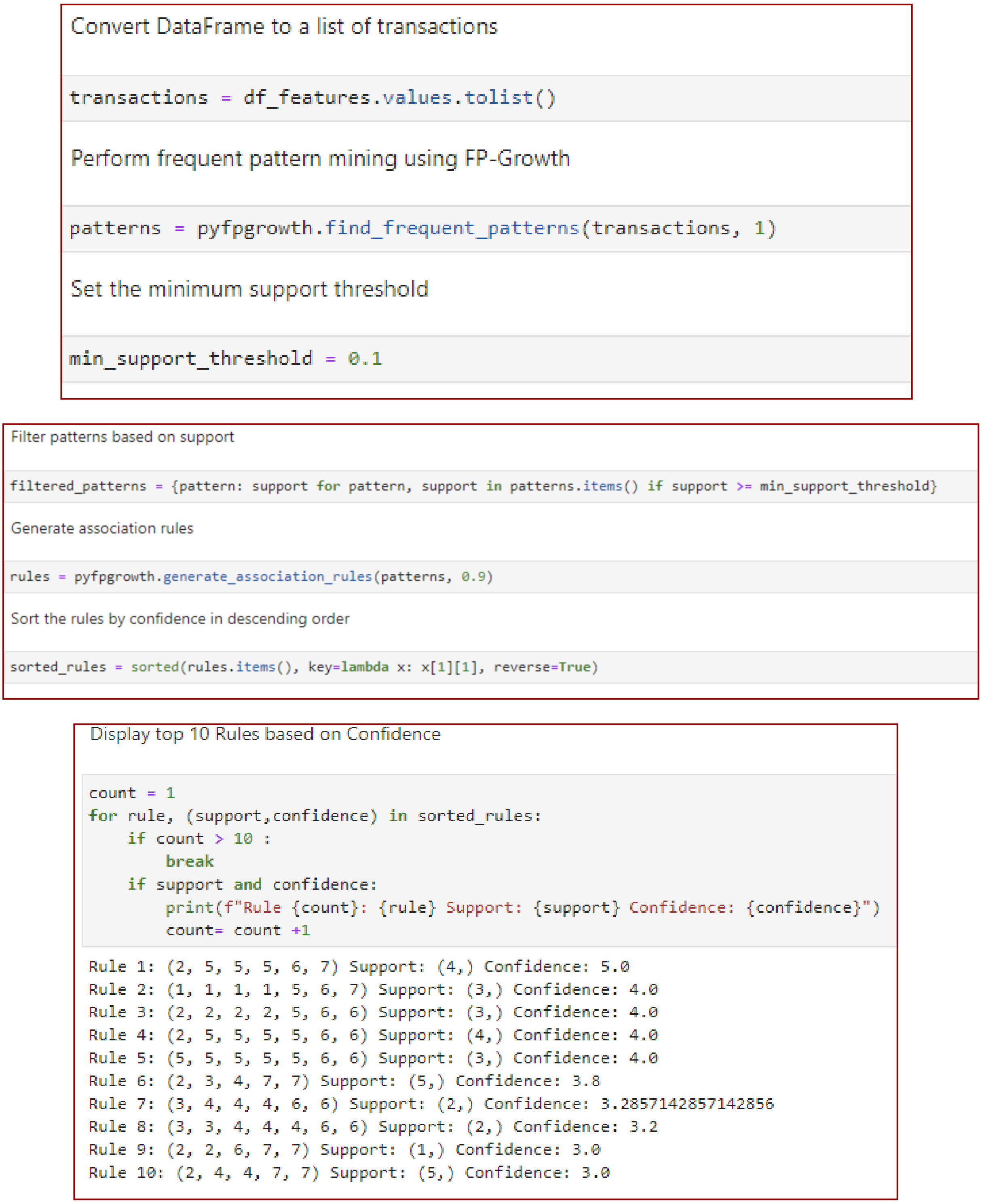
5.3 Association Rule Mining (ARM)

5.3.1 FP-Growth algorithm

- Efficiency: FP-Growth is fast, especially when dealing with large datasets. It can be achieved by building a compact data structure called an FP-tree, which allows for efficient pattern mining without the need for repeated database scans.
- Handling Non-Binary Data: FP-Growth can handle non-binary (integer or continuous) data directly without requiring explicit binary encoding. This allows us to work with your original values between 1 and 7 without any transformation.
- Flexibility: FP-Growth allows us to specify different measures for evaluating association rules, such as support, confidence, or lift. This flexibility enables us to customise the mining process based on the specific requirements.

We used the FP-Growth algorithm for Association Rule Mining (ARM). It starts by converting the dataset into a list of transactions. Then, it performs frequent pattern mining using FP-Growth and sets a minimum support threshold. The patterns that meet the support threshold are filtered and stored. Next, association rules are generated based on the filtered patterns. The rules are sorted by confidence in descending order.

The output as seen in *Figure 5.3.1.1*, shows the top 10 rules along with their support and confidence values. Each rule represents a combination of items or features in the dataset that have a certain level of support and confidence. These rules provide insights into the relationships and associations among the features.



Convert DataFrame to a list of transactions

```
transactions = df_features.values.tolist()
```

Perform frequent pattern mining using FP-Growth

```
patterns = pyfpgrowth.find_frequent_patterns(transactions, 1)
```

Set the minimum support threshold

```
min_support_threshold = 0.1
```

Filter patterns based on support

```
filtered_patterns = {pattern: support for pattern, support in patterns.items() if support >= min_support_threshold}
```

Generate association rules

```
rules = pyfpgrowth.generate_association_rules(patterns, 0.9)
```

Sort the rules by confidence in descending order

```
sorted_rules = sorted(rules.items(), key=lambda x: x[1][1], reverse=True)
```

Display top 10 Rules based on Confidence

```
count = 1
for rule, (support,confidence) in sorted_rules:
    if count > 10 :
        break
    if support and confidence:
        print(f"Rule {count}: {rule} Support: {support} Confidence: {confidence}")
        count= count +1
```

```
Rule 1: (2, 5, 5, 5, 6, 7) Support: (4,) Confidence: 5.0
Rule 2: (1, 1, 1, 1, 5, 6, 7) Support: (3,) Confidence: 4.0
Rule 3: (2, 2, 2, 2, 5, 6, 6) Support: (3,) Confidence: 4.0
Rule 4: (2, 5, 5, 5, 5, 6, 6) Support: (4,) Confidence: 4.0
Rule 5: (5, 5, 5, 5, 5, 6, 6) Support: (3,) Confidence: 4.0
Rule 6: (2, 3, 4, 7, 7) Support: (5,) Confidence: 3.8
Rule 7: (3, 4, 4, 4, 6, 6) Support: (2,) Confidence: 3.2857142857142856
Rule 8: (3, 3, 4, 4, 4, 6, 6) Support: (2,) Confidence: 3.2
Rule 9: (2, 2, 6, 7, 7) Support: (1,) Confidence: 3.0
Rule 10: (2, 4, 4, 7, 7) Support: (5,) Confidence: 3.0
```

Figure 5.3.1.1 FP-Growth algorithm

6. Results and Discussion

In this section, we will discuss the results of the model which have been used for predicting. *Figure 6.1* provides the comparison of the evaluation metrics with our 3 models selected which are Decision Tree, Gaussian Naive Bayes, and SVM.

From the *Figure 6.1* it can be seen that SVM tops the graph among the other 2 models, having highest accuracy (0.53), precision (0.49), recall (0.53), and f1 score (0.49) compared to the other 2 models. Gaussian Naive Bayes are able to achieve 0.52 of accuracy, 0.33 of precision and recall, 0.32 for F1 score. Whereas Decision Trees are able to achieve 0.50 of accuracy, 0.35 of precision, 0.31 of recall and 0.33 for f1 score.

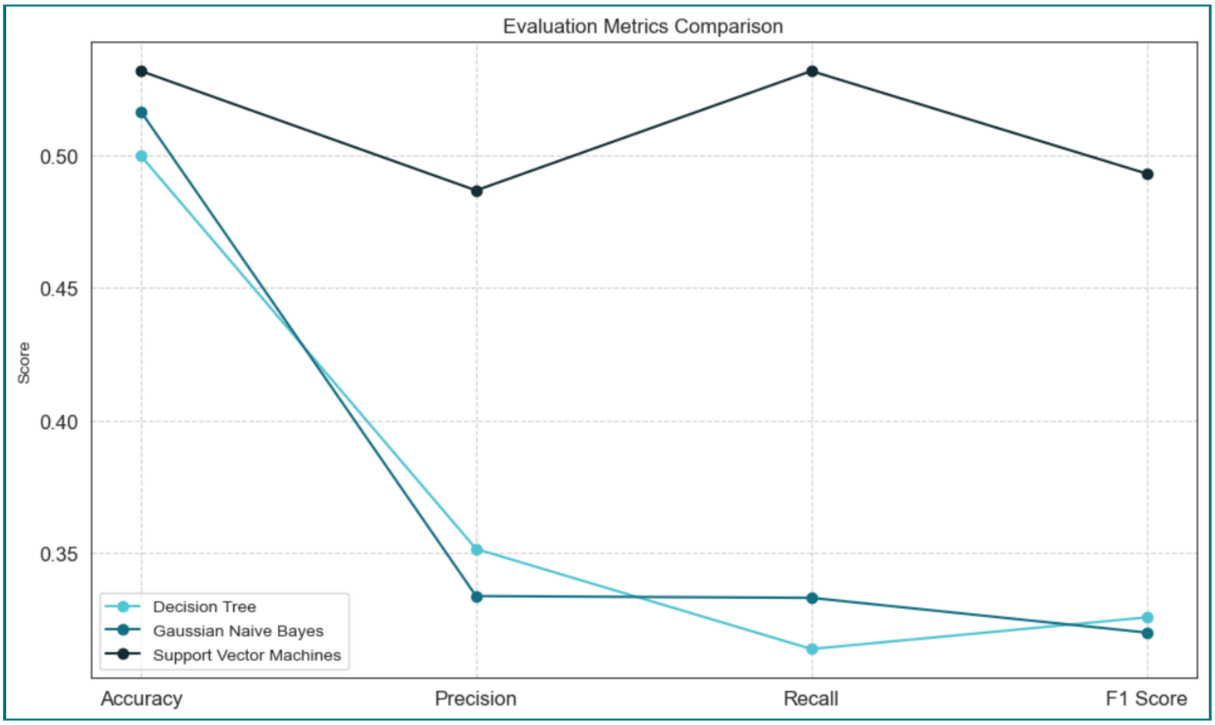
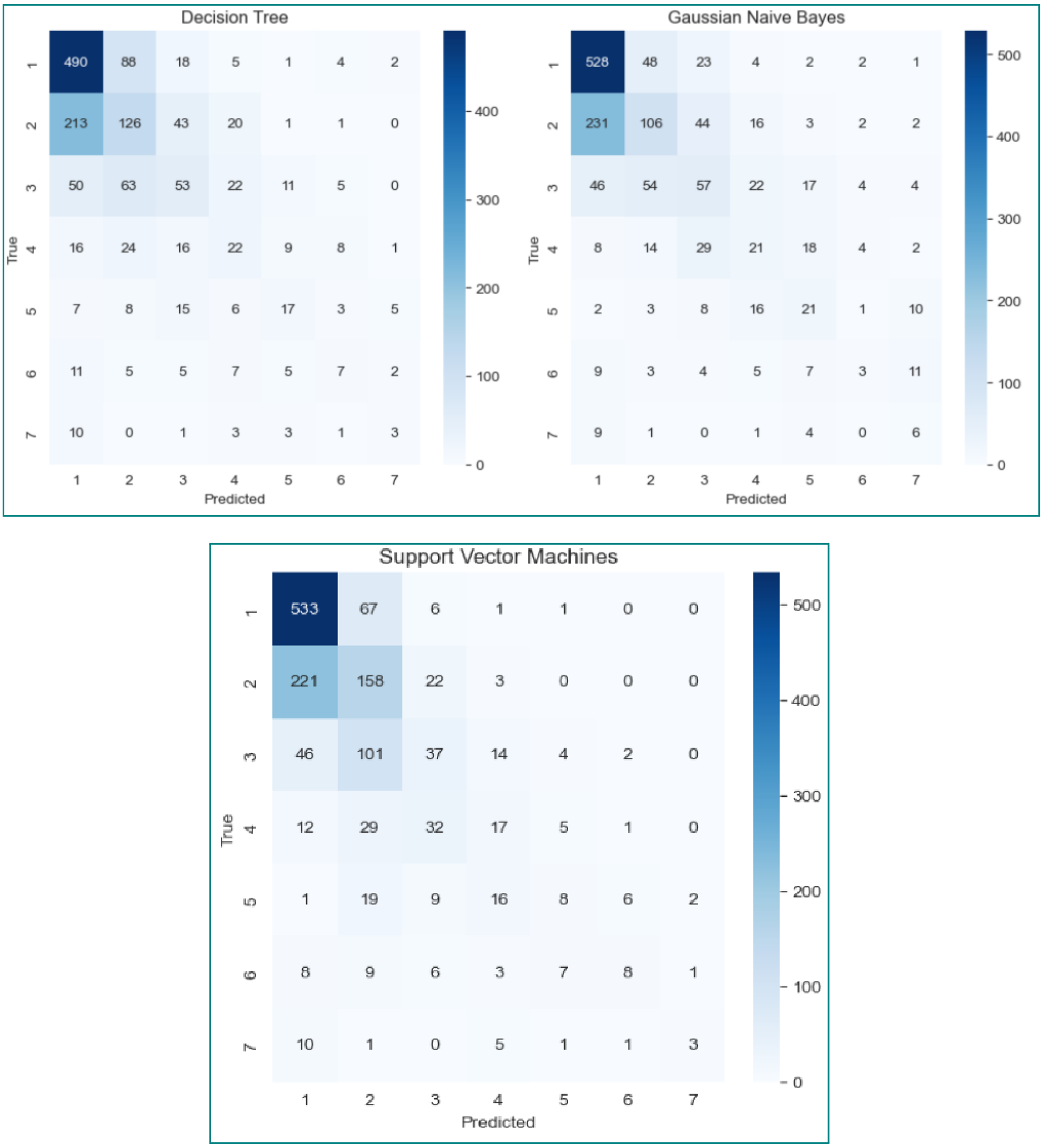


Figure 6.1: Evaluation Metrics Comparison

Figure 6.2 shows the confusion matrix between the 3 models, it can be seen that SVM and Naive Bayes exhibit better performance than Decision Tree.



Decision Tree

	1	2	3	4	5	6	7
1	490	88	18	5	1	4	2
2	213	126	43	20	1	1	0
3	50	63	53	22	11	5	0
4	16	24	16	22	9	8	1
5	7	8	15	6	17	3	5
6	11	5	5	7	5	7	2
7	10	0	1	3	3	1	3

Gaussian Naive Bayes

	1	2	3	4	5	6	7
1	528	48	23	4	2	2	1
2	231	106	44	16	3	2	2
3	46	54	57	22	17	4	4
4	8	14	29	21	18	4	2
5	2	3	8	16	21	1	10
6	9	3	4	5	7	3	11
7	9	1	0	1	4	0	6

Support Vector Machines

	1	2	3	4	5	6	7
1	533	67	6	1	1	0	0
2	221	158	22	3	0	0	0
3	46	101	37	14	4	2	0
4	12	29	32	17	5	1	0
5	1	19	9	16	8	6	2
6	8	9	6	3	7	8	1
7	10	1	0	5	1	1	3

Figure 6.2: Confusion Matrix

Figure 6.3 shows the Silhouette Coefficients Score which measures the quality of clustering for each individual data point and provides an average score for the entire dataset and separate each page cluster with different color.

a mean Silhouette Coefficients Score of 0.20 for the five clusters indicates a moderate level of separation with some overlap or ambiguity between data points from different clusters.

While the clustering results are better than random assignment, there is still room for improvement in terms of strengthening the cohesion within each cluster.

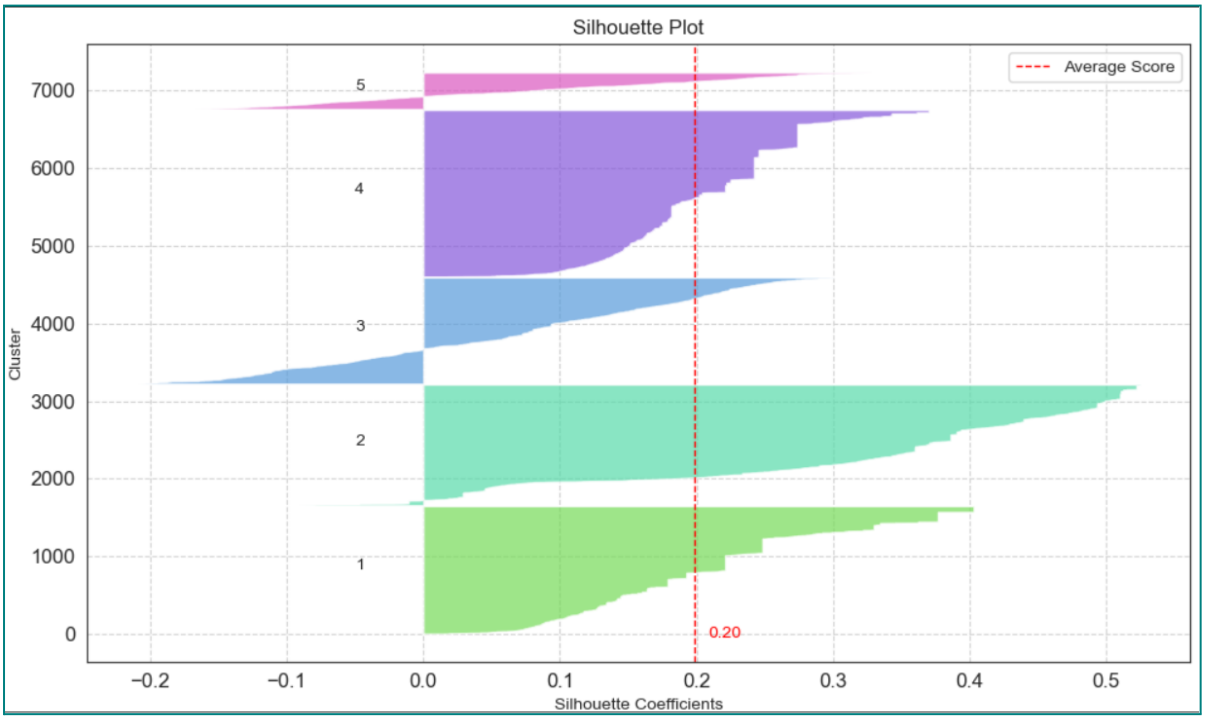


Figure 6.3: K-Means Silhouette Coefficients Score

Figure 6.4 is a comparison btween Support and Confidence of the top Confidence rules, Support: Support indicates how often a particular itemset or rule appears in the dataset as a proportion of the total transactions or instances, will Confidence measures the reliability or strength of an association rule.

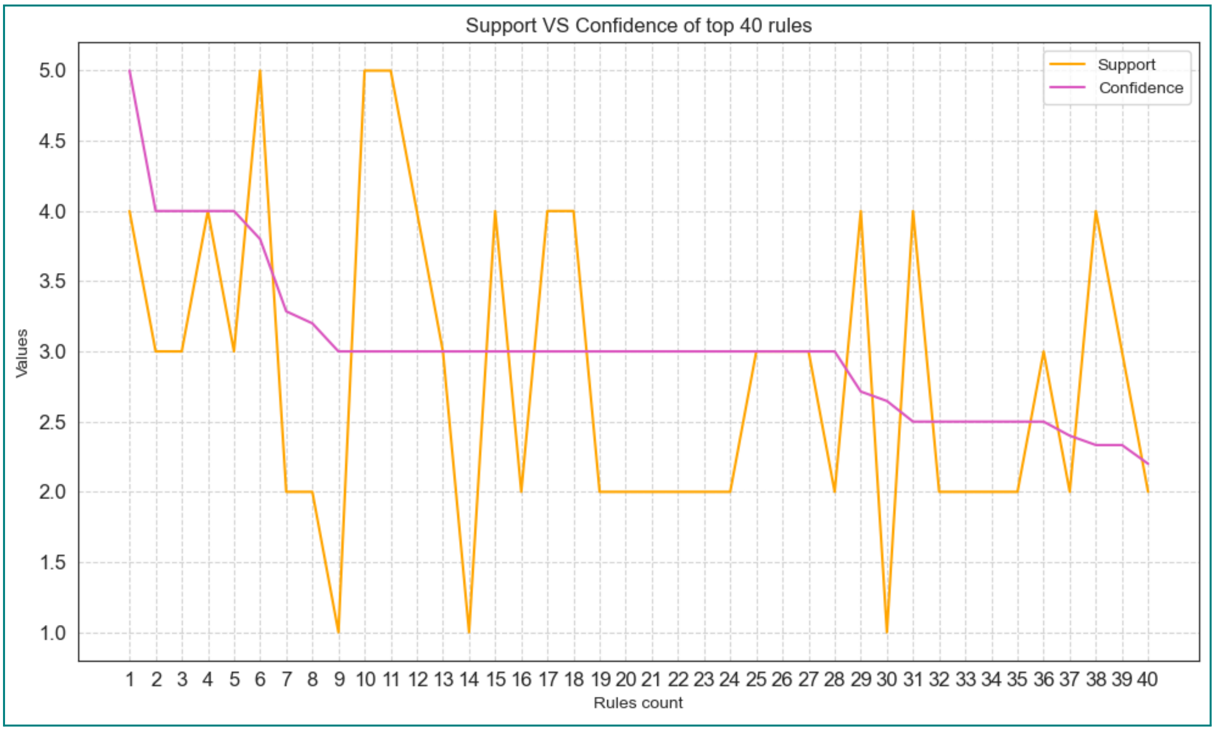


Figure 6.4 Association Rule Mining (Support VS Confidence of top 40 rules)

Figure 6.5 illustrates the frequency distribution of the rules, showcasing the number of repetitions each rule has.

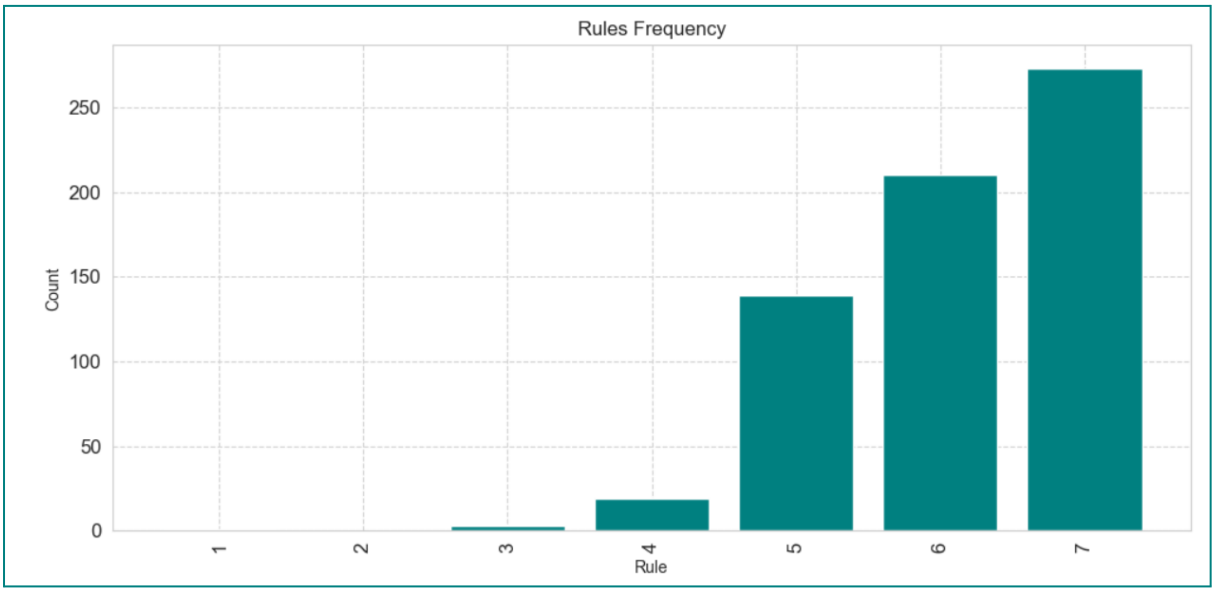


Figure 6.5 Association Rule Mining (Rules Frequency)

Overall, it can be seen that our accuracy results in our model are not that high, since we took the mean of ms1, ms2, and ms3 as our target variables might be the reason that affects our accuracies. Therefore, splitting the target variables can be our future enhancements of this project. Not only that, we can also explore more advanced modelling techniques such as ensemble models or deep learning architectures to improve the accuracy and predictive power of the models.

Throughout completing the project, we have found some limitations of this study, the first limitation is that the dataset may not capture all the relevant factors that contribute to relationship satisfaction. It focuses on variables such as demographics, personal beliefs, and satisfaction ratings. Other important aspects like communication, conflict resolution, and emotional dynamics may not be fully represented. We can overcome them by expanding the dataset with additional variables that capture a broader range of factors influencing relationship satisfaction. This can include measures of communication patterns and style, conflict resolution strategies, emotional intimacy, and shared activities. It will provide a more nuanced understanding of relationship dynamics.

Besides, other limitations include self-reporting bias. As the data relies on self-reported information from the respondents, it may cause biases and inaccuracies. Respondents may provide socially desirable responses or may not accurately recall certain aspects of their relationship, leading to potential measurement errors as a result. To overcome these, we can combine the data with qualitative methods such as interviews or focus groups. It can offer deeper insights into participants' experiences, perceptions, and subjective interpretations. It can provide a more comprehensive understanding of relationship satisfaction.

7. Conclusion

In this project, we aimed to employ data mining techniques to develop a marital satisfaction model for the couples. By analyzing a survey dataset from married individuals in 45 countries, we explored various factors related to marital satisfaction and identified predictors using Decision Tree, Naive Bayes, and SVM for classification, K-Means clustering, and FP-Growth algorithm for association rule mining techniques.

Our accuracy of the model was beyond average, SVM are able to achieve the highest accuracy among the models. The reason our accuracy model is only able to achieve beyond average was because we took the mean values from ms1, ms2, and ms3. Future enhancements of this project includes splitting the dataset into individual variables to improve the accuracy and perform using advanced modelling techniques to achieve better accuracies.

Overall, this project contributes to the development of a data-driven marital satisfaction model for the couples in the world. The insights gained from this analysis can assist in understanding the factors that influence relationship quality and inform interventions to promote healthier and more successful marriages. 2 limitations were addressed such as limited scope and self-reporting bias. These 2 limitations can be overcome by diversifying the dataset and performing a mixed-method approach. Further research can be conducted to validate and refine the model, enabling its application in real-world scenarios to support couples in assessing and improving their relationship satisfaction.