Experiment No: 04	TE AI&DS
Date of Performance:	Roll No: 9696
Aim: Apply data exploration and Data preprocessing techniques to organize data for data mining	
Related CO3: Apply data exploration and Data preprocessing techniques to organize and prepare data for data mining	

Rubrics for assessment of Experiment:

Sr. No	Parameters	Exceed Expectations(EE)	Meet Expectations (ME)	Below Expectations (BE)
1	Timeline (2)	Early or on time (2)	One session late (1)	More than one session late (0)
2	Preparedness (2)	Knows the basic theory related to the experiment very well. (2)	Managed to explain the theory related to the experiment. (1)	Not aware of the theory to the point. (1)
3	Effort (3)	Done expt on their own. (3)	Done expt with help from other. (2)	Just managed. (1)
4	Documentation(2)	Lab experiment is documented in proper format and maintained neatly. (2)	Documented in proper format but some formatting guidelines are missed. (1)	Experiments not written in proper format (0.5)
5	Result (1)	Specific conclusion.(1)	Partially specific conclusion. (0.5)	Not specific at all. (0)

Assessment Marks:

Timeline(2)	Preparedness(2)	Effort(3)	Documentation(2)	Result(1)	Total(10)

Theory:

Data Exploration:

Measure of Central Tendency

A measure of central tendency (also referred to as measures of centre or central location) is a summary measure that attempts to describe a whole set of data with a single value that represents the middle or centre of its distribution.

There are three main measures of central tendency: the mean, the median and the mode. Each of these measures describes a different indication of the typical or central value in the distribution.

What is the mean?

The mean is the sum of the value of each observation in a dataset divided by the number of observations. This is also known as the arithmetic average.

Looking at the retirement age distribution again:

54, 54, 54, 55, 56, 57, 57, 58, 58, 60, 60

The mean is calculated by adding together all the values (54+54+54+55+56+57+57+58+58+60+60 = 623) and dividing by the number of observations (11) which equals 56.6 years.

The population mean is indicated by the Greek symbol μ (pronounced 'mu'). When the mean is calculated on a distribution from a sample it is indicated by the symbol $\bar{x}$ (pronounced X-bar).

What is the median?

The median is the *middle value* in distribution when the values are arranged in ascending or descending order.

The median divides the distribution in half (there are 50% of observations on either side of the median value). In a distribution with an odd number of observations, the median value is the middle value.

Looking at the retirement age distribution (which has 11 observations), the median is the middle value, which is 57 years:

54, 54, 54, 55, 56, 57, 57, 58, 58, 60, 60

When the distribution has an even number of observations, the median value is the mean of

the two middle values. In the following distribution, the two middle values are 56 and 57, therefore the median equals 56.5 years:

52, 54, 54, 54, 55, 56, 57, 57, 58, 58, 60, 60

The median cannot be identified for categorical nominal data, as it cannot be logically ordered.

Standard Deviation (Dispersion)

The **standard deviation** is a statistic that measures the dispersion of a dataset relative to its mean and is calculated as the square root of the variance.

Formula

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

σ = population standard deviation

N = the size of the population

x_i = each value from the population

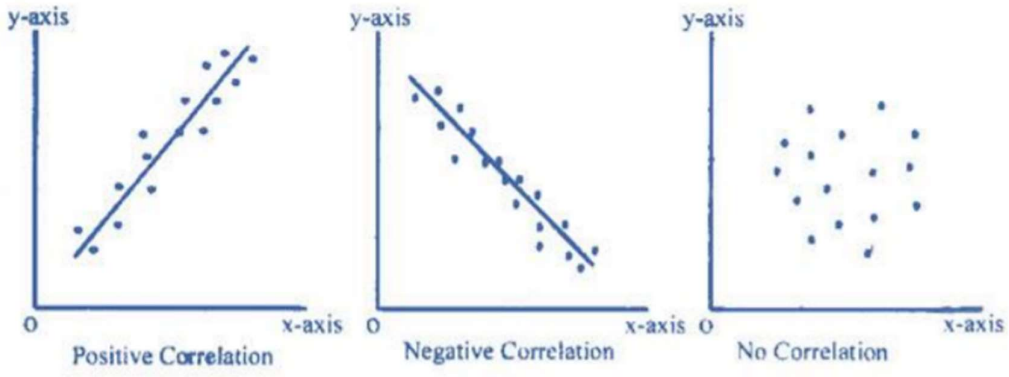
μ = the population mean

CORRELATION

Definition: The Correlation is a statistical tool used to measure the relationship between two or more variables, i.e. the degree to which the variables are associated with each other, such that the change in one is accompanied by the change in another.

Types of Correlation:

1. **Positive Correlation** A correlation in the same direction is called a positive correlation. If one variable increases the other also increases and when one variable decreases the other also decreases. For example, the length of an iron bar will increase as the temperature increases. • Price and Supply • Sales and Expenditure on Advertisement • Yield and Fertilizer Applied
2. **Negative Correlation** Correlation in the opposite direction is called a negative correlation. Here if one variable increases the other decreases and vice versa. For example, the volume of gas will decrease as the pressure increases, or the demand for a particular commodity increases as the price of such commodity decreases. Examples: • Price and Demand • Yield and Weed
3. **No Correlation or Zero Correlation** If there is no relationship between the two variables such that the value of one variable changes and the other variable remains constant, it is called no or zero correlation.



Methods of Determining Correlation:

- Pearson's Coefficient of Correlation.
- Spearman's Rank Correlation Coefficient;
- Kendall

Pearson *r* correlation

Pearson *r* correlation is the most widely used correlation statistic to measure the degree of the relationship between linearly related variables. For example, in the stock market, if we want to measure how two stocks are related to each other, Pearson *r* correlation is used to measure the degree of relationship between the two. The point-biserial correlation is conducted with the Pearson correlation formula except that one of the variables is dichotomous. The following formula is used to calculate the Pearson *r* correlation:

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

r_{xy} = Pearson r correlation coefficient between x and y

n = number of observations

x_i = value of x (for ith observation)

y_i = value of y (for ith observation)

Data preprocessing

Handling Missing Values

One of the important stages of data mining is preprocessing, where we prepare the data for mining. Real-world data tends to be incomplete, noisy, and inconsistent and an important task when preprocessing the data is to fill in missing values, smooth out noise and correct inconsistencies.

If we specifically look at dealing with missing data, there are several techniques that can be used. Choosing the right technique is a choice that depends on the problem domain — the data's domain and our goal for the data mining process.

So, there are several techniques which can be used for handling missing values.

1. Ignore the data row
This is usually done when the class label is missing (assuming your data mining goal is classification), or many attributes are missing from the row (not just one). However, you'll obviously get poor performance if the percentage of such rows is high.

2. Use a global constant to fill in for missing values
Decide on a new global constant value, like "unknown", "N/A" or minus infinity, that will be used to fill all the missing values.
This technique is used because sometimes it just doesn't make sense to try and predict the missing value.

3. Use attribute mean
Replace missing values of an attribute with the mean (or median if its discrete) value for that attribute in the database.

4. Use a data mining algorithm to predict the most probable value
The value can be determined using regression, inference based tools using Bayesian formalism, decision trees, clustering algorithms (K-Mean\Median etc.).

Feature Scaling- Normalization and Standardization

Normalization
Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling.
Here's the formula for normalization:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Here, Xmax and Xmin are the maximum and the minimum values of the feature respectively.

- When the value of X is the minimum value in the column, the numerator will be 0, and hence X' is 0
- On the other hand, when the value of X is the maximum value in the column, the numerator is equal to the denominator and thus the value of X' is 1
- If the value of X is between the minimum and the maximum value, then the value of X' is between 0 and 1

Standardization (z-score normalization_

Standardization is another scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.

Here's the formula for standardization:

$$X' = \frac{X - \mu}{\sigma}$$

μ is the mean of the feature values and σ is the standard deviation of the feature values. Note that in this case, the values are not restricted to a particular range.

Normalization is good to use when you know that the distribution of your data does not follow a Gaussian distribution. This can be useful in algorithms that do not assume any distribution of the data like K-Nearest Neighbors and Neural Networks.

Standardization, on the other hand, can be helpful in cases where the data follows a Gaussian distribution. However, this does not have to be necessarily true. Also, unlike normalization, standardization does not have a bounding range. So, even if you have outliers in your data, they will not be affected by standardization.

Binning

Data binning, bucketing is a data pre-processing method used to minimize the effects of small observation errors. The original data values are divided into small intervals known as bins and then they are replaced by a general value calculated for that bin. This has a smoothing effect on the input data and may also reduce the chances of overfitting in the case of small datasets.

There are 2 methods of dividing data into bins:

Equal Frequency Binning: bins have an equal frequency.

```
Input:[5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215]
Output:
[5, 10, 11, 13]
[15, 35, 50, 55]
[72, 92, 204, 215]
```

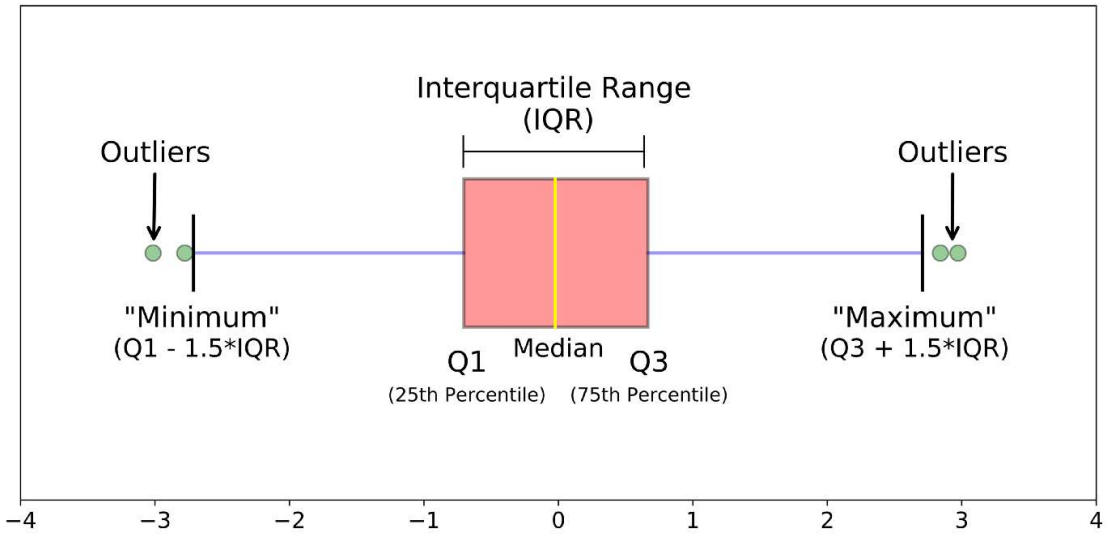
Equal Width Binning : bins have equal width with a range of each bin are defined as [min + w], [min + 2w] …. [min + nw] where w = (max – min) / (no of bins).

```
Input: [5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215]
[5, 10, 11, 13, 15, 35, 50, 55,72]

[92]

[204, 215]
```

Data visualization

BOX PLOT:
A boxplot is a standardized way of displaying the distribution of data based on a five number summary ("minimum", first quartile (Q1), median, third quartile (Q3), and "maximum"). It

can tell you about your outliers and what their values are. It can also tell you if your data is symmetrical, how tightly your data is grouped, and if and how your data is skewed.

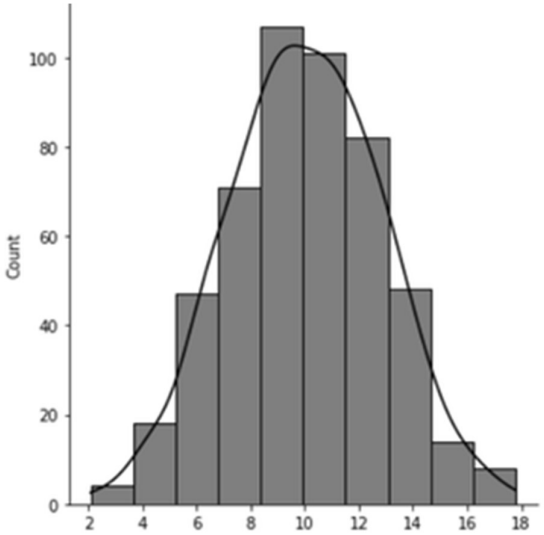


Histogram

A Histogram is a variation of a bar chart in which data values are grouped together and put into different classes. This grouping enables you to see how frequently data in each class occur in the dataset.

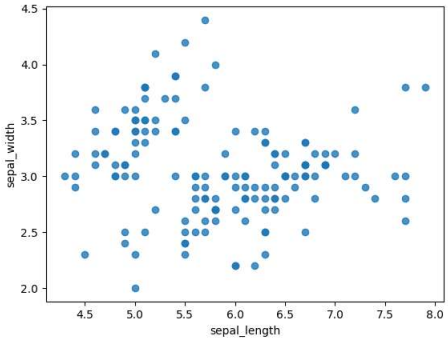
The histogram graphically shows the following:

- Frequency of different data points in the dataset.
- Location of the centre of data.
- The spread of dataset.



Scatter plot:

Scatter plots are **a commonly used data visualisation tool within data science**. They allow us to plot two numerical variables, as points, on a two-dimensional graph. From these plots, we can understand if there is a relationship between the two variables, and what the strength of that relationship is.



Implementation:

1) Load the libraries
2) Download the data set from classroom (realEstate_trans_full.csv)
3) Read the file –select appropriate file read function according to data type of file.
4) Display attributes in the data set-10 samples.
5) Describe the attributes name, count no of values, and find min, max, mean,data type, range, quartile, percentile.
6) Display correlation between any two attributes.
7) Download the dataset from classroom (realEstate_trans_less_dirty.csv)
8) Identify missing values fill them with mean.
9) Implement data normalization and standardization on real estate data.
10) Implement binning techniques on real estate data.

Code:

 # data normalization

```
import pandas as pd

csv_read = pd.read_csv('D:/DWM/Exp 4/realEstate_trans_imputed.csv')

def normalize(col):
    return (col - col.min()) / (col.max() - col.min())

def standardize(col):
    return (col - col.mean()) / col.std()

cols = ['price_mean','sq__ft']

for col in cols:
    csv_read['n_' + col] = normalize(csv_read[col])
    csv_read['s_' + col] = standardize(csv_read[col])
```

```
with open('D:/DWM/Exp 4/realEstate_trans_imputed2.csv', 'w') as write_csv:
    write_csv.write(csv_read.to_csv(sep=',', index=False))
```

 # data smoothing

```
import numpy as np
import pandas as pd

# read the data
csv_read = pd.read_csv('D:/DWM/Exp 4/realEstate_trans_imputed2.csv')

# create bins for the price that are based on the
# linearly spaced range of the price values

bins = np.linspace(
    csv_read['price_mean'].min(),
    csv_read['price_mean'].max(),
    6
)

# and apply the bins to the data
csv_read['b_price'] = np.digitize(
csv_read['price_mean'],
bins
)

# print out the counts for the bins
counts_b = csv_read['b_price'].value_counts()
print(counts_b.sort_index())

# and write to a file

with open('D:/DWM/Exp 4/realEstate_trans_binning.csv', 'w') as write_csv:
    write_csv.write(csv_read.to_csv(sep=',', index=False))

# OutPut :-
# 1    350
# 2    480
# 3    118
# 4     26
# 5      4
# 6      3
# Name: b_price, dtype: int64
```

 # handling missing value:

```
import pandas as pd
# read the data
```

```
csv_read =
pd.read_csv('/home/universe/Desktop/9689/realEstate_trans_less_dirty -
realEstate_trans_less_dirty.csv')

# impute mean in place of NaNs

csv_read['price_mean'] =
csv_read['price'].fillna(csv_read.groupby('zip')['price'].transform('mean'))

# impute median in place of NaNs
csv_read['price_median'] =
csv_read['price'].fillna(csv_read.groupby('zip')['price'].transform('median')
)

# and write to a file
with open('/home/universe/Desktop/9689/realEstate_trans_imputed.csv', 'w') as
write_csv:
    write_csv.write(csv_read.to_csv(sep=',' ,index=False))
```

 # mean, median and correlation

```
import pandas as pd
import numpy as np

dataframe =
pd.read_csv('/home/universe/Desktop/9689/realEstate_trans_less_dirty -
realEstate_trans_less_dirty.csv')
# print(dataframe.columns)
dtype = (dataframe.dtypes)
# print(dtype)

# print(dataframe.head(10))
# print(dataframe[['beds','baths', 'sq__ft']].describe)

#Mean
print(dataframe[['beds','baths', 'sq__ft']].mean())
#Mode
print(dataframe[['beds','baths', 'sq__ft']].mode())
#Median
print(dataframe[['beds','baths', 'sq__ft']].median())
#Standrad Deviation
print(dataframe[['beds','baths', 'sq__ft']].std())


# Coorelation
print(dataframe[['beds','baths', 'sq__ft']].corr(method = "pearson"))
```

```
print("\nKendall")
print(dataframe[['beds','baths', 'sq__ft']].corr(method = "kendall"))


print("\nSpearman")
print(dataframe[['beds','baths', 'sq__ft']].corr(method = "spearman"))


dataframe['price_mean'] =
dataframe['price'].fillna(dataframe.groupby('zip')['price'].transform('mean'))
dataframe['price_median'] =
dataframe['price'].fillna(dataframe.groupby('zip')['price'].transform('median'
))
```

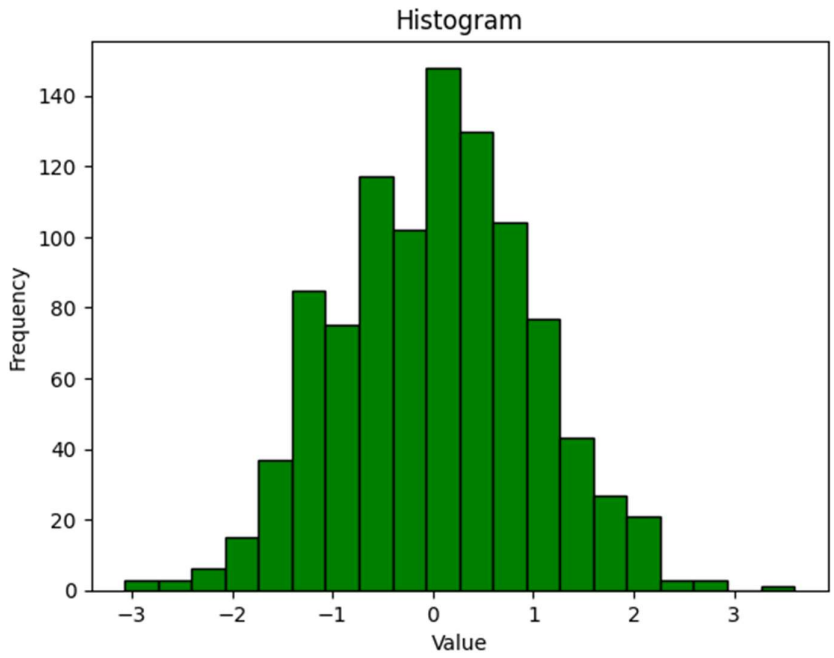
Post lab:

1. Apply data visualization for statistical description of data – in the form of histogram, scatter plot and box plot.

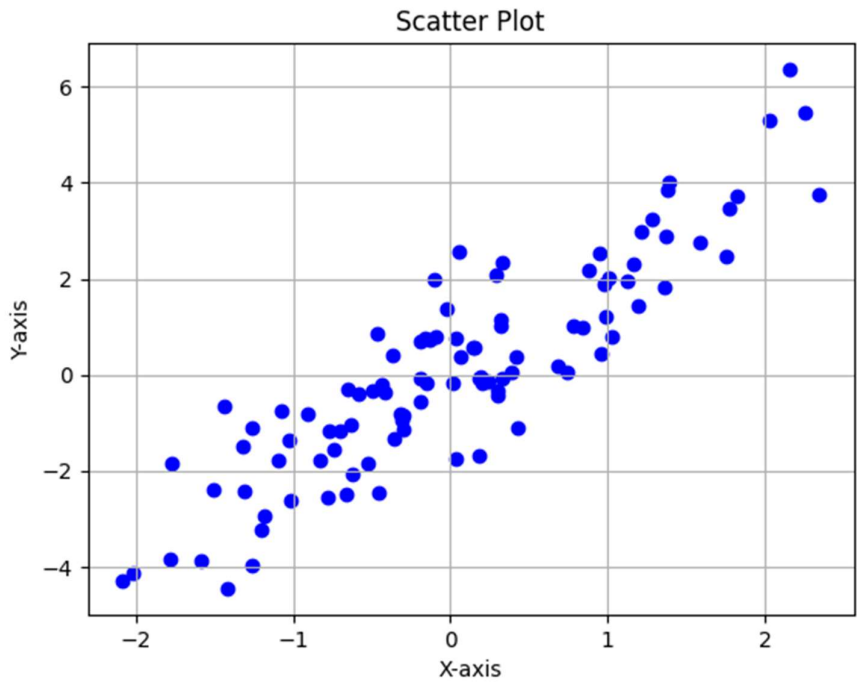
Code:

```
import matplotlib.pyplot as plt
import numpy as np
```

```
# Generate some example data
data = np.random.randn(1000)  # Replace with your own dataset

# Create a histogram
plt.hist(data, bins=20, color='green', edgecolor='black')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histogram')
plt.show()
```




```
# Scatter Plot
# Generate some example data
x = np.random.randn(100)
y = 2 * x + np.random.randn(100)  # Replace with your own dataset

# Create a scatter plot
plt.scatter(x, y, c='blue', marker='o')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Scatter Plot')
plt.grid(True)
plt.show()
```



Conclusion:
I am able to understand the different data preprocessing technique and able to implement the code.