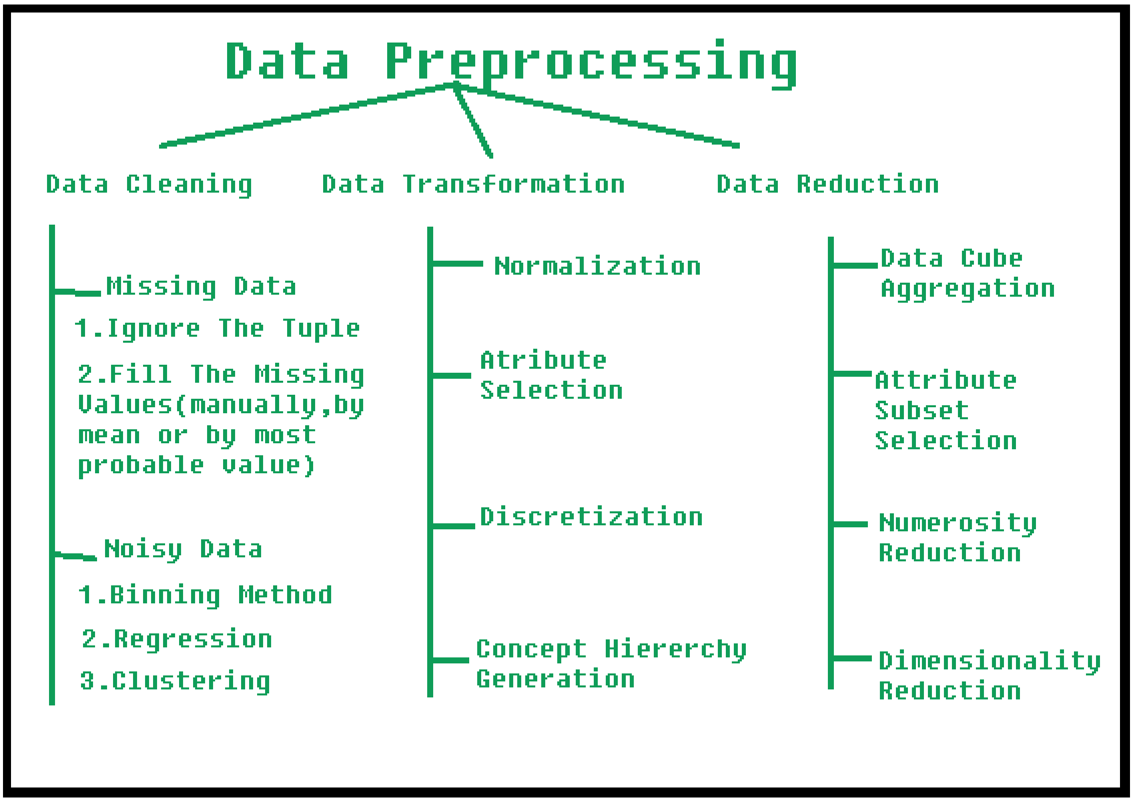
Experiment No. 1

Aim: Select a dataset and perform exploratory data analysis using Python

Theory:

Data Preprocessing:



Data preprocessing is a crucial step in the data mining process. It involves cleaning, transforming, and preparing raw data to make it suitable for analysis. The steps involved include:

- Data cleaning: This involves identifying and correcting errors or inconsistencies in the data, such as missing values, outliers, and duplicates. Various techniques can be used for data cleaning, such as imputation, removal, and transformation.

- Data Integration: This involves combining data from multiple sources to create a unified dataset. Data integration can be challenging as it requires handling data with different formats, structures, and semantics. Techniques such as record linkage and data fusion can be used for data integration.

- Data Transformation: This involves converting the data into a suitable format for analysis. Common techniques used in data transformation include normalization, standardization, and discretization. Normalization is used to scale the data to a common range, while standardization is used to transform the data to have zero mean and unit variance. Discretization is used to convert continuous data into discrete categories.

- Data Reduction: This involves reducing the size of the dataset while preserving the important information. Data reduction can be achieved through techniques such as feature selection and feature extraction. Feature selection involves selecting a subset of relevant features from the dataset, while feature extraction involves transforming the data into a lower-dimensional space while preserving the important information.

Data Transformation:

Data transformation in data mining refers to the process of converting raw data into a format that is suitable for analysis and modelling. The goal of data transformation is to prepare the data for data mining so that it can be used to extract useful insights and knowledge. Data transformation typically involves several steps, including:

- Smoothing: It is a process that is used to remove noise from the dataset using some algorithms It allows for highlighting important features present in the dataset. It helps in predicting the patterns. When collecting data, it can be manipulated to eliminate or reduce any variance or any other noise form. The concept behind data smoothing is that it will be able to identify simple changes to help predict different trends and patterns. This serves as a help to analysts or traders who need to look at a lot of data which can often be difficult to digest for finding patterns that they wouldn’t see otherwise.

- Aggregation: Data collection or aggregation is the method of storing and presenting data in a summary format. The data may be obtained from multiple data sources to integrate these data sources into a data analysis description. This is a crucial step since the accuracy of data analysis insights is highly dependent on the quantity and quality of the data used. Gathering accurate data of high quality and a large enough quantity is necessary to produce relevant results. The collection of data is useful for everything from decisions concerning financing or business strategy of the product, to pricing, operations, and marketing strategies. For example, Sales, data may be aggregated to compute monthly& annual total amounts.

- Normalization: Data normalization involves converting all data variables into a given range.

- Generalization: It converts low-level data attributes to high-level data attributes using concept hierarchy. For Example, Age initially in Numerical form (22, 25) is converted into categorical value (young, old). For example, Categorical attributes, such as house addresses, may be generalized to higher-level definitions, such as town or country.

- Attribute Construction: Where new attributes are created & applied to assist the mining process from the given set of attributes. This simplifies the original data & makes the mining more efficient.

Data Discretization:

Data discretization refers to a method of converting a huge number of data values into smaller ones so that the evaluation and management of data become easy. In other words, data discretization is a method of converting attribute values of continuous data into a finite set of intervals with minimum data loss.

- Histogram analysis: Histogram refers to a plot used to represent the underlying frequency distribution of a continuous data set. Histogram assists the data inspection for data distribution. For example, Outliers, skewness representation, normal distribution representation, etc.

- Binning: Binning refers to a data smoothing technique that helps to group a huge number of continuous values into smaller values. For data discretization and the development of idea hierarchy, this technique can also be used.

Data Visualization:

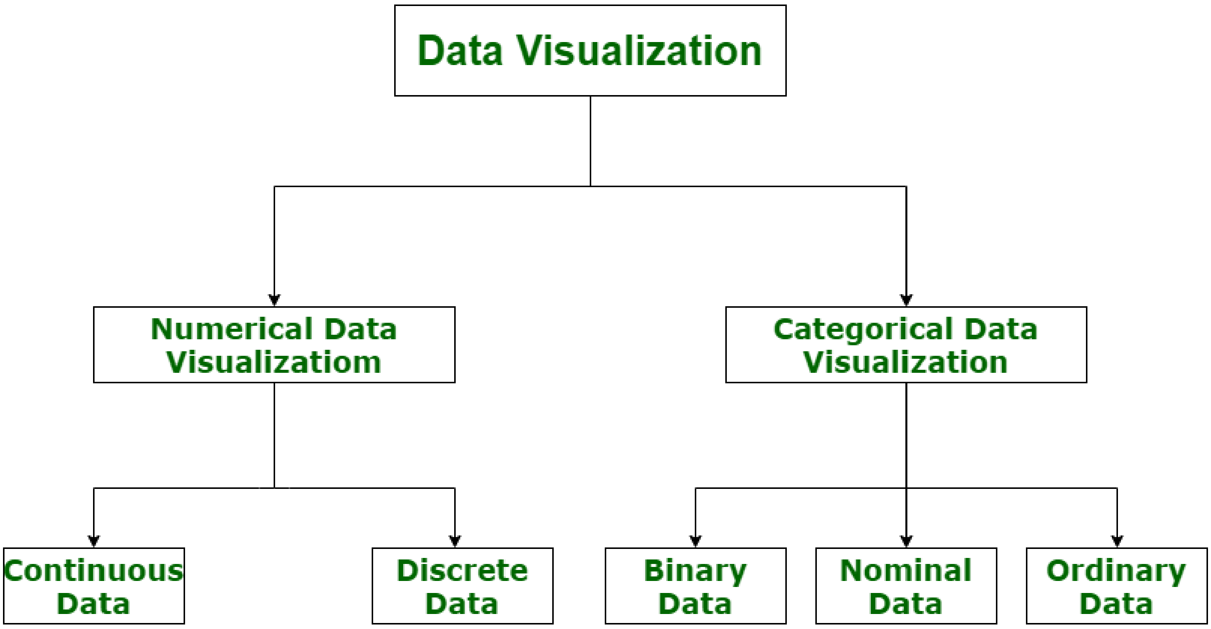
Data visualization is actually a set of data points and information that are represented graphically to make it easy and quick for user to understand. Data visualization is good if it has a clear meaning, purpose, and is very easy to interpret, without requiring context. Tools of data visualization provide an accessible way to see and understand trends, outliers, and patterns in data by using visual effects or elements such as a chart, graphs, and maps.

Categories of Data Visualization :

- Numerical Data : Numerical data is also known as Quantitative data. Numerical data is any data where data generally represents amount such as height, weight, age of a person, etc. Numerical data visualization is easiest way to visualize data. It is generally used for helping others to digest large data sets and raw numbers in a way that makes it easier to interpret into action. Numerical data is categorized into two categories :

1. Continuous Data- It can be narrowed or categorized (Example: Height measurements).
2. Discrete Data- This type of data is not “continuous” (Example: Number of cars or children’s a household has).

The type of visualization techniques that are used to represent numerical data visualization is Charts and Numerical Values. Examples are Pie Charts, Bar Charts, Averages, Scorecards, etc.



- Categorical Data : Categorical data is also known as Qualitative data. Categorical data is any data where data generally represents groups. It simply consists of categorical variables that are used to represent characteristics such as a person’s ranking, a person’s gender, etc. Categorical data visualization is all about depicting key themes, establishing connections, and lending context. Categorical data is classified into three categories :

1. Binary Data- In this, classification is based on positioning (Example: Agrees or Disagrees).
2. Nominal Data- In this, classification is based on attributes (Example: Male or Female).
3. Ordinal Data- In this, classification is based on ordering of information (Example: Timeline or processes).

The type of visualization techniques that are used to represent categorical data is Graphics, Diagrams, and Flowcharts. Examples are Word clouds, Sentiment Mapping, Venn Diagram, etc.

Code:

```
import csv
import random
from faker import Faker
from datetime import datetime, timedelta

# Set the number of entries for each dimension
num_products = 1000
num_warehouses = 100
num_suppliers = 500
num_contacts = 1000
num_entries = 1000

fake = Faker()

# Function to generate a unique ID
def generate_unique_id(existing_ids):
    new_id = fake.random_int(min=1000, max=9999)
    while new_id in existing_ids:
        new_id = fake.random_int(min=1000, max=9999)
    existing_ids.add(new_id)
    return new_id

# Function to generate random date within a range
def generate_random_date(start_date, end_date):
    time_delta = end_date - start_date
    random_days = random.randint(0, time_delta.days)
    return start_date + timedelta(days=random_days)

# Generate data and write to a single CSV file
with open('inventory_data.csv', mode='w', newline=") as file:
    writer = csv.writer(file)
    writer.writerow(["transaction_id", "product_id", "product_name", "product_category",
"product_brand", "warehouse_id", "warehouse_name", "warehouse_location", "city_id",
"supplier_id", "supplier_name", "contact_id", "phone", "email", "arrival_date",
"dispatch_date", "number_of_boxes"])

    product_ids = set()
    warehouse_ids = set()
    supplier_ids = set()
    contact_ids = set()

    for i in range(num_entries):
        transaction_id = i + 1

        # Generate product data
        product_id = generate_unique_id(product_ids)
        product_name = fake.word()
        product_category = fake.word()
        product_brand = fake.company()

        # Generate warehouse data
        warehouse_id = generate_unique_id(warehouse_ids)
        warehouse_name = fake.company()
        warehouse_location = fake.address()
        city_id = generate_unique_id(contact_ids)

        # Generate supplier data
        supplier_id = generate_unique_id(supplier_ids)
        supplier_name = fake.company()

        # Generate contact data
        contact_id = generate_unique_id(contact_ids)
        phone = fake.phone_number()
        email = fake.email()

        # Generate transaction date data
        arrival_date = generate_random_date(datetime(2020, 1, 1), datetime(2023, 8, 1))
        dispatch_date = generate_random_date(arrival_date, datetime(2023, 8, 1))
        number_of_boxes = random.randint(1, 100)

        writer.writerow([transaction_id, f"P{product_id}", product_name, product_category,
product_brand, f"W{warehouse_id}", warehouse_name, warehouse_location, f"C{city_id}",
f"S{supplier_id}", supplier_name, f"C{contact_id}", phone, email,
arrival_date.strftime('%Y-%m-%d'), dispatch_date.strftime('%Y-%m-%d'),
number_of_boxes])
```