**DEPARTMENT OF INFORMATION TECHNOLOGY**

**Course: Data Mining & Business Intelligence Lab (ITL601)**

**B.Tech. (Information Technology) – Semester VI**

**Academic Year: 2023-24 (Even Semester)**

**PRACTICAL 3**

**Aim:** Implementation of Data Pre-processing using Python

**Lab Objective:** Writing a python code to perform cleaning, handling and binning on a self created database.

**Theory:**

**a. Data Cleaning (Average Method):**

Data cleaning is a critical phase in preparing datasets for analysis, and the Average Method serves as a straightforward solution for handling missing values. By replacing missing entries with the mean of non-missing values in the respective columns, this method offers simplicity and efficiency. The mean calculation ensures that imputed values align with the overall statistical trends in the data, making it a pragmatic choice, especially in cases of missing completely at random scenarios.

**b. Handling Missing Data (Binning Method):**

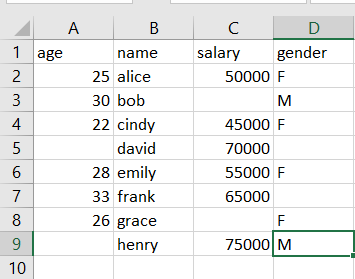
Handling missing data is an essential aspect of preparing datasets for analysis. The Binning Method is employed to categorize numerical data into discrete bins or intervals. This method is particularly useful when dealing with continuous data and aims to simplify complex numerical information. In the context of missing data, the Binning Method categorizes numerical values into predefined ranges, providing a structured approach to representing the data. Missing values can be placed into appropriate bins, enabling the preservation of the data's distribution while addressing the absence of specific values.

**c. Data Transformation (Decimal Scaling Method):**

Data transformation, crucial for meaningful analysis, includes techniques like Decimal Scaling. This method normalizes numerical attributes by shifting the decimal point, ensuring a consistent scale. The process involves determining the power of 10 required for scaling and multiplying each data point accordingly. Decimal Scaling preserves relative relationships between data points, preventing the undue influence of individual features. However, its sensitivity to outliers necessitates consideration of the dataset's specific characteristics.

Schema Designs / <Code with Output>:

* **Dataset –**



* **Data Pre-processing** –

1. **Data Cleaning (Average Method)**

**Code –**

import pandas as pd

import numpy as np

df = pd.read\_csv("expt3.csv ")

# mean

df2 = int((df["salary"].mean()))

df.fillna(df2,inplace=True)

print("-------------------------------")

print("Get the mean of the column: ", df2)

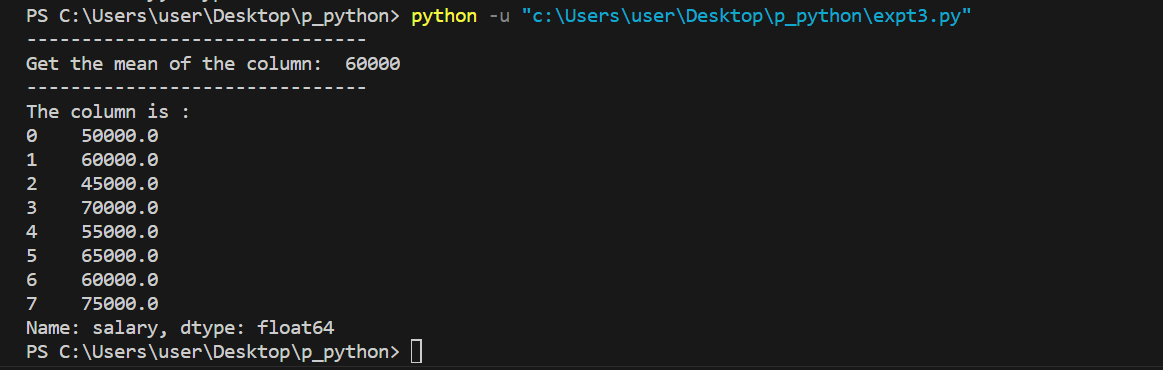
print("-------------------------------")

specific\_column = df["salary"]

print("The column is : ")

print(specific\_column)

**Output –**



1. **Handling Missing Data (Binning Method)**

**Code -**

import pandas as pd

import numpy as np

df = pd.read\_csv("expt3.csv ")

# binning

df['salary'] = pd.to\_numeric(df['salary'], errors='coerce')

mean\_salary = df['salary'].mean()

df['salary'].fillna(mean\_salary, inplace=True)

bins = [40000, 60000]

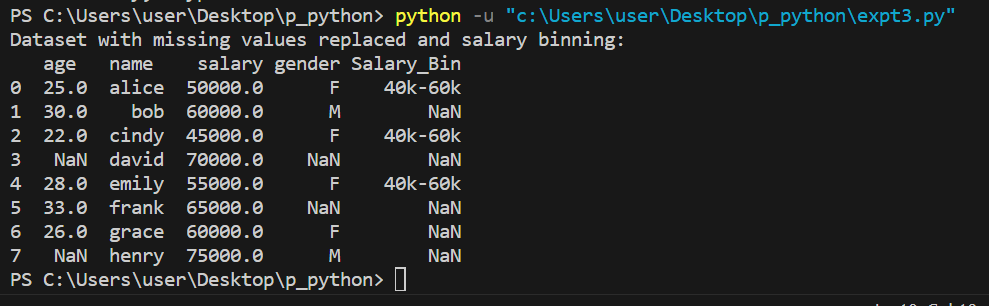
labels = ['40k-60k']

df['Salary\_Bin'] = pd.cut(df['salary'], bins=bins, labels=labels, right=False)

print("Dataset with missing values replaced and salary binning:")

print(df)

**Output -**



1. **Data Transformation (Decimal Scaling Method)**

**Code-**

import pandas as pd

import numpy as np

df = pd.read\_csv("expt3.csv ")

# binning

df['salary'] = pd.to\_numeric(df['salary'], errors='coerce')

mean\_salary = df['salary'].mean()

df['salary'].fillna(mean\_salary, inplace=True)

bins = [40000, 60000]

labels = ['40k-60k']

df['Salary\_Bin'] = pd.cut(df['salary'], bins=bins, labels=labels, right=False)

print("Dataset with missing values replaced and salary binning:")

print(df)

#Decimal Scaling

max\_salary = df['salary'].max()

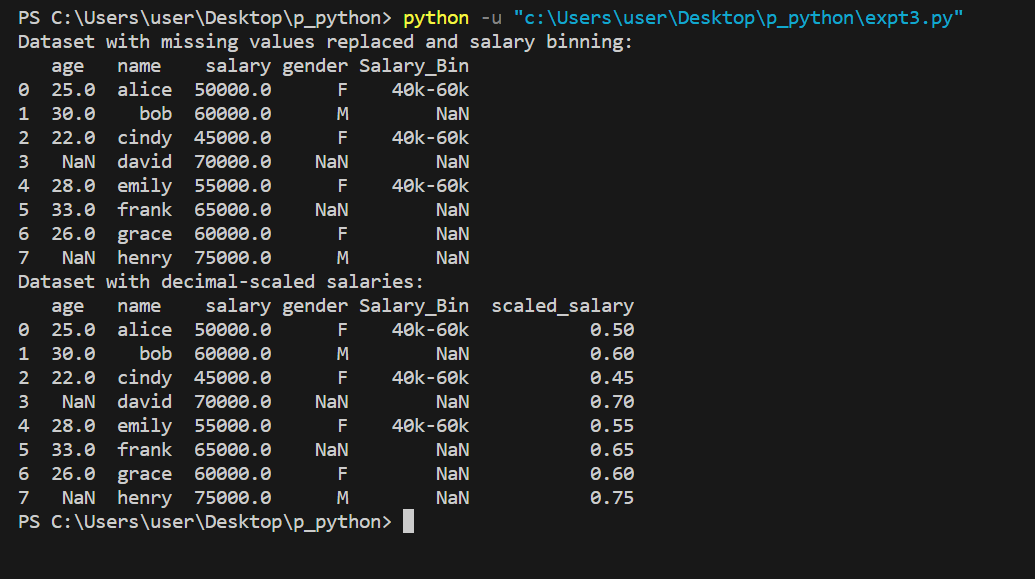
scaling\_factor = 10 \*\* len(str(int(max\_salary)))

df['scaled\_salary'] = df['salary'] / scaling\_factor

print("Dataset with decimal-scaled salaries:")

print(df)

**Output:**



**Conclusion:** In conclusion, the Python implementation of data pre-processing demonstrates key techniques essential for enhancing the quality and usability of datasets. The Average Method efficiently addresses missing values by substituting them with the mean of the non-missing values within respective columns. The Binning Method, applied when handling missing data, partitions the dataset into quantile-based bins and fills missing values with bin medians, offering a nuanced approach. Additionally, the Decimal Scaling Method provides a means of data transformation by normalizing numerical attributes to a common scale, facilitating consistent representation.

**Lab Outcome:** Performed Handling, Binning and Cleaning in Python Code.

**Submitted Details -**

**Name of Student: Parth Malviya**

**Roll No.: 22**

**Date of Performance: 23/1/24**

**Date of Submission: 28/1/24**