

## VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES - TECHNICAL CAMPUS

### Grade A++ Accredited Institution by NAAC

NBA Accredited for MCA Programme; Recognized under Section 2(f) by UGC; Affiliated to GGSIP University, Delhi; Recognized by Bar Council of India and AICTE An ISO 9001:2015 Certified Institution

**SCHOOL OF ENGINEERING & TECHNOLOGY**

# B.Tech Programme: CSE

Course Title: Data Warehouse & Data Mining

Course Code: CIE 425P

# Submitted To:

**Submitted By:**

**Name: Ishaan Jain**

**Enrollment No: 06117702722**

**Branch & Section: CSE B**

**Dr. Shalini Gambhir**

**Assistant Professor**



## VISION OF INSTITUTE

To be an educational institute that empowers the field of engineering to build a sustainable future by providing quality education with innovative practices that supports people, planet and profit.

## MISSION OF INSTITUTE

To groom the future engineers by providing value-based education and awakening students' curiosity, nurturing creativity and building

capabilities to enable them to make significant contributions to the world.

**EXPERIMENT - 1**

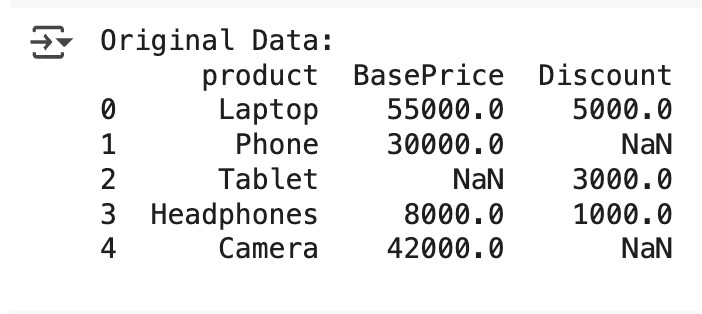
**AIM: Study of ETL process and its tools.**

**THEORY:**

**DATASET:**

**CODE:**

**# upload, read and print csv** import pandas as pd df = pd.read\_csv('product\_sales.csv') print("Original Data:") print(df)

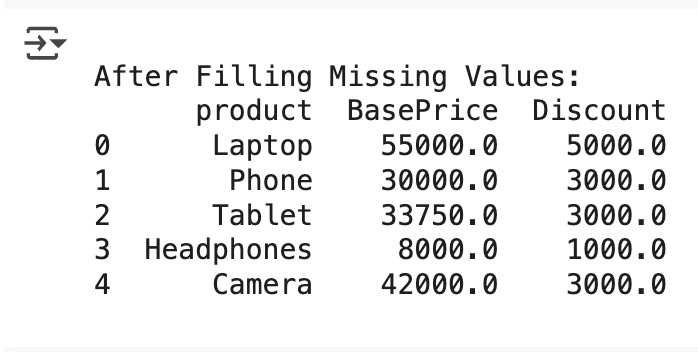


# #Missing values - mean of that column #for BasePrice

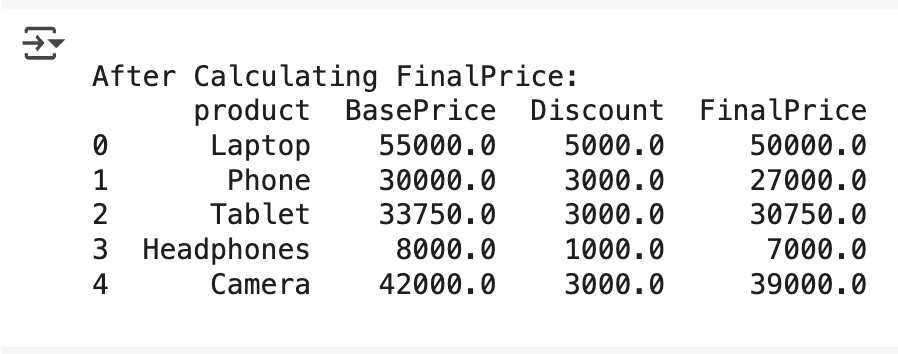
basic\_mean = df['BasePrice'].mean() df['BasePrice'] = df['BasePrice'].fillna(basic\_mean)

# # for Discount

bonus\_mean = df['Discount'].mean() df['Discount'] = df['Discount'].fillna(bonus\_mean) print("\nAfter Filling Missing Values:") print(df)



**# Create a new column 'FinalPrice'** df['FinalPrice'] = df['BasePrice'] - df['Discount'] print("\nAfter Calculating FinalPrice:") print(df)

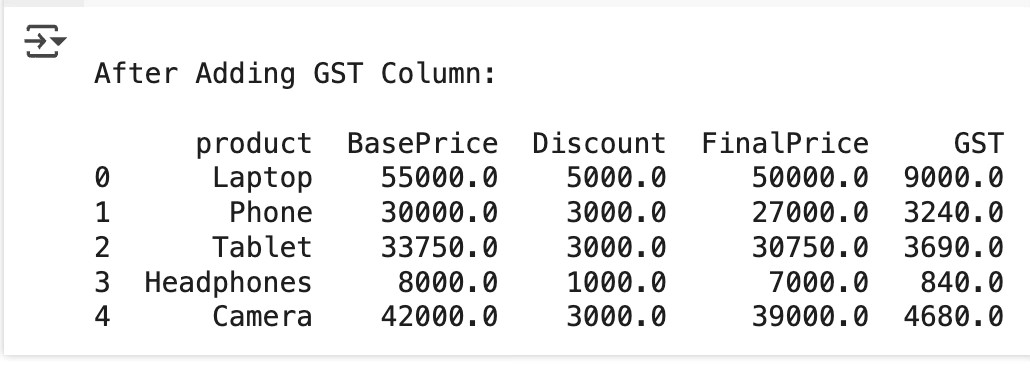


**# Define a function to calculate gst** def calculate\_gst(FinalPrice): if FinalPrice >= 40000: return 0.18 \* FinalPrice else:

return 0.12 \* FinalPrice

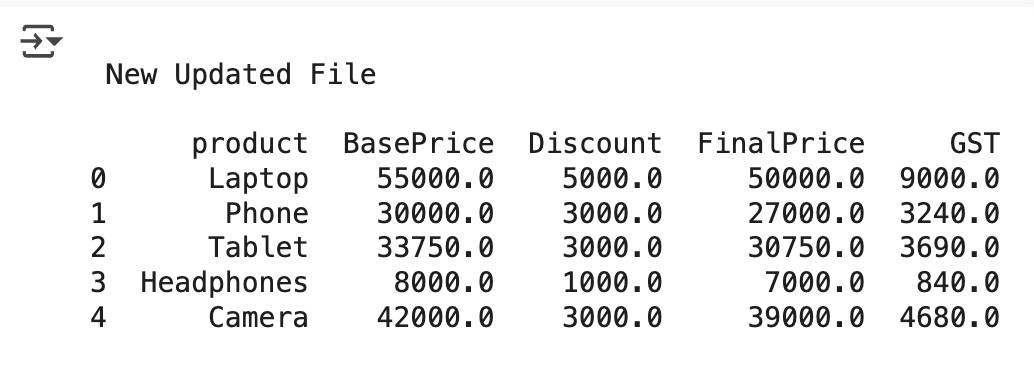
**# new gst column** df['GST'] = df['FinalPrice'].apply(calculate\_gst)

# Print after adding GST column print("\nAfter Adding GST Column:\n") print(df)

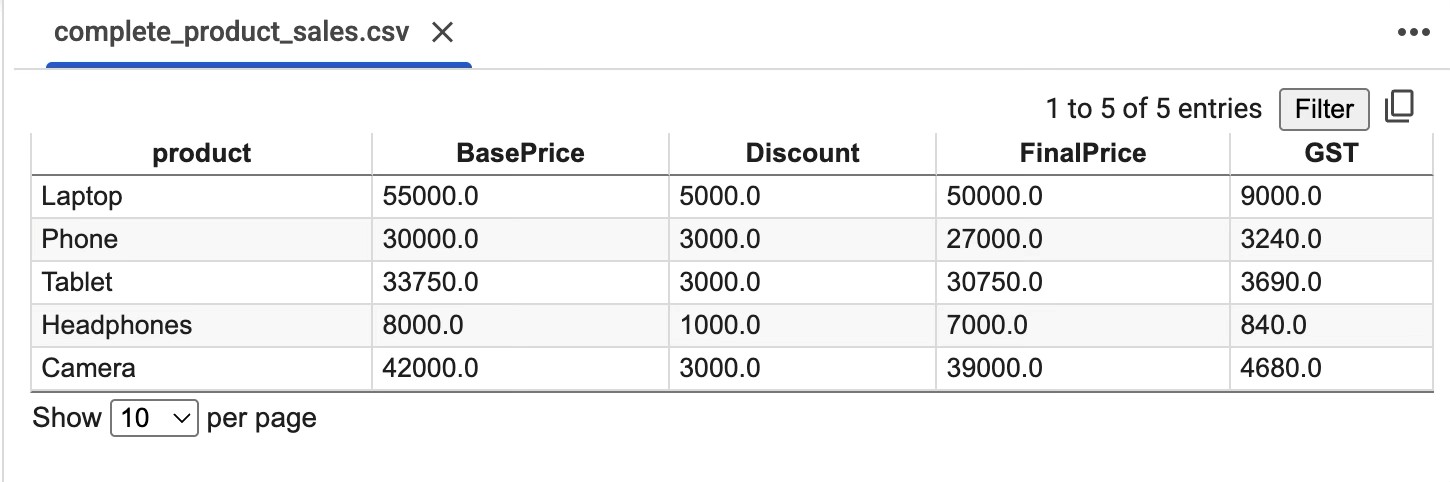


**# Save the transformed DataFrame** df.to\_csv('complete\_product\_sales.csv', index=False)

#load and print the new file transformed\_df = pd.read\_csv('complete\_product\_sales.csv') print("\n New Updated File \n") print(transformed\_df)



**OUTPUT:**



**LEARNING OUTCOME:**

# EXPERIMENT - 2

**AIM: Program of Data warehouse cleansing to input names from users (inconsistent) and format them.**

**THEORY:**

**DATASET:**

**CODE:**

import re

def clean\_name(name: str) -> str: name = name.replace("@", "a") name = name.replace("!", "i") name = name.replace("0", "o") name = re.sub(r'[^A-Za-z\s]', '', name) name = " ".join(name.split()) return name.title()

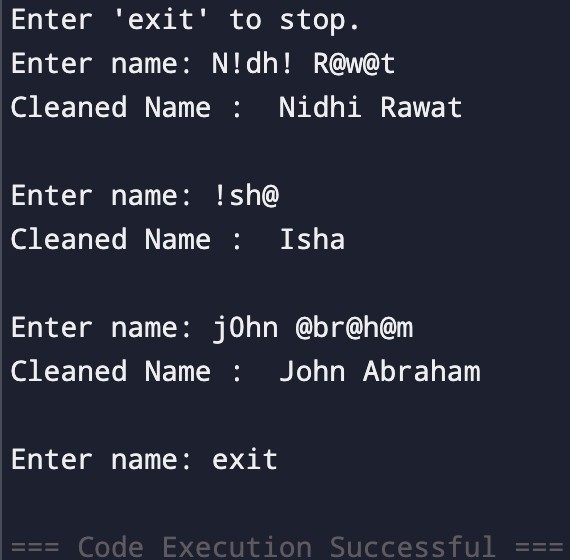
print("Enter 'exit' to stop.") while(True):

name = input("Enter name: ") if(name == "exit"):

break

print("Cleaned Name : ",clean\_name(name),"\n")

**OUTPUT:**



**LEARNING OUTCOME:**

# EXPERIMENT - 3

**AIM: Program of Data warehouse cleansing to remove redundancy in data.**

**THEORY:**

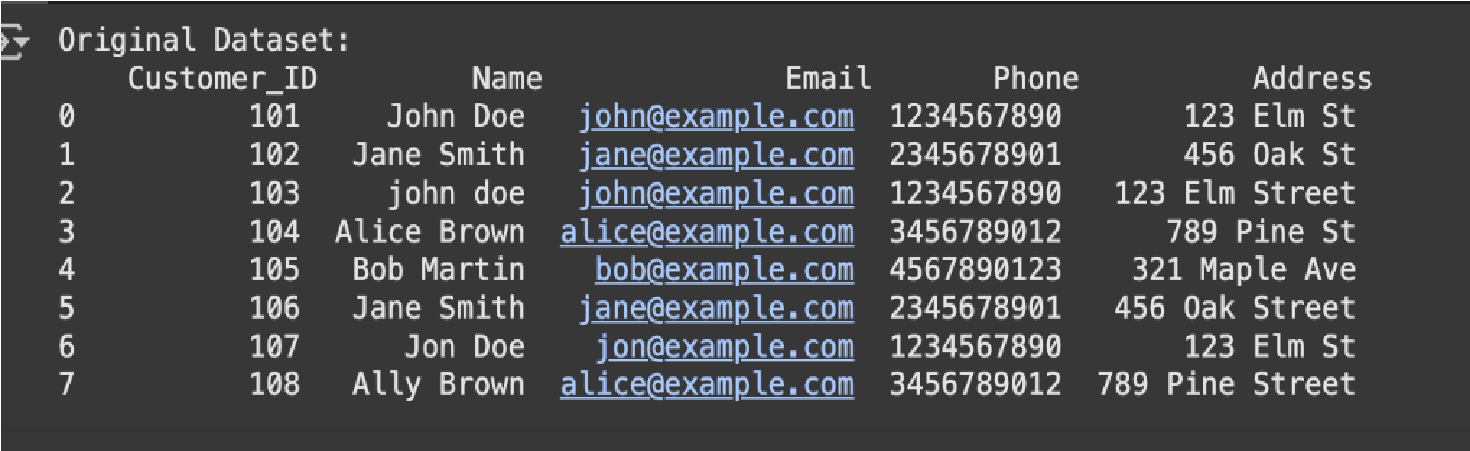
**DATASET:**

**CODE:**

from fuzzywuzzy import fuzz import pandas as pd import matplotlib.pyplot as plt

## #Load csv

df = pd.read\_csv("dataset/customer\_data.csv") print("Original Dataset:\n", df)



## #Standardize names and addresses

df['Name'] = df['Name'].str.strip().str.title() df['Address'] = df['Address'].str.strip().str.title() print("\nAfter Standardization\n", df)

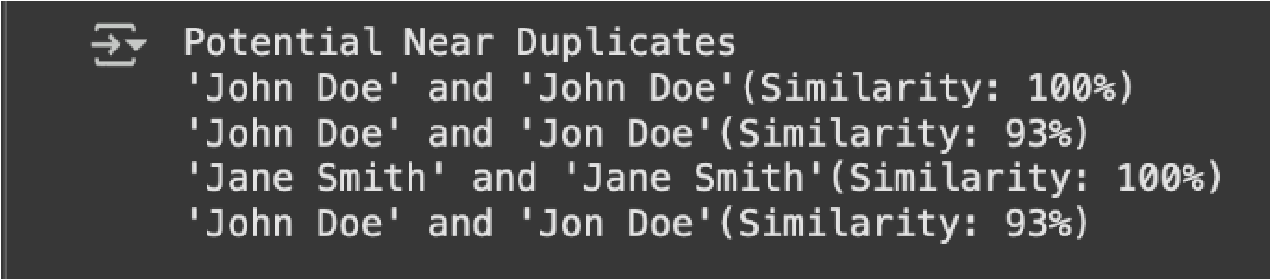


**#Remove exact duplicates** df\_no\_duplicates = df.drop\_duplicates() print("After removing duplicates\n", df\_no\_duplicates)



**#Detect near Duplicates using fuzzy matching** names = df\_no\_duplicates['Name'].tolist() print("Potential Near Duplicates"); for i in range(len(names)): for j in range(i + 1, len(names)):

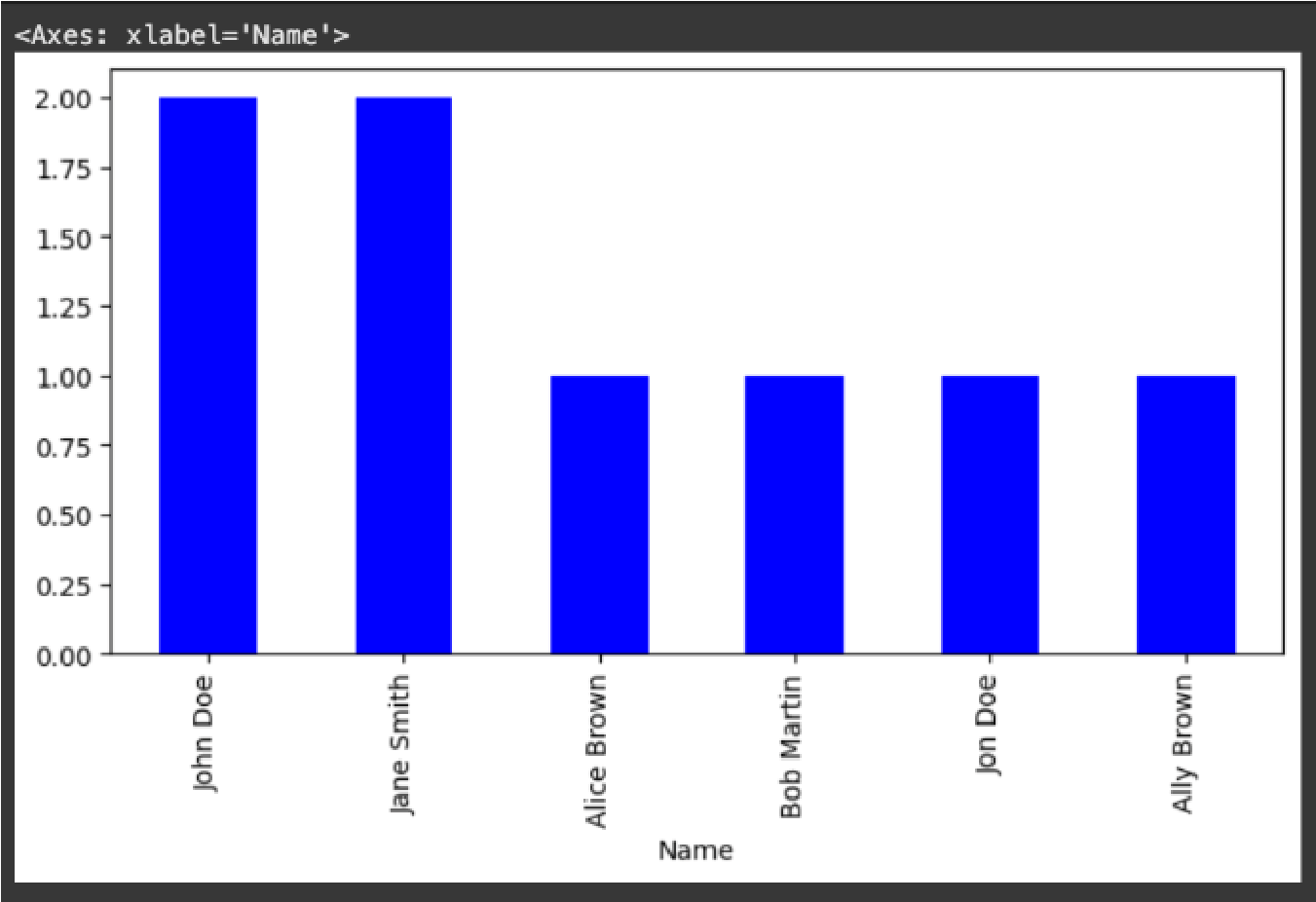
similarity = fuzz.ratio(names[i], names[j]); if similarity > 80: # Threshold for near duplicates print(f"'{names[i]}' and '{names[j]}' (Similarity: {similarity}%)")



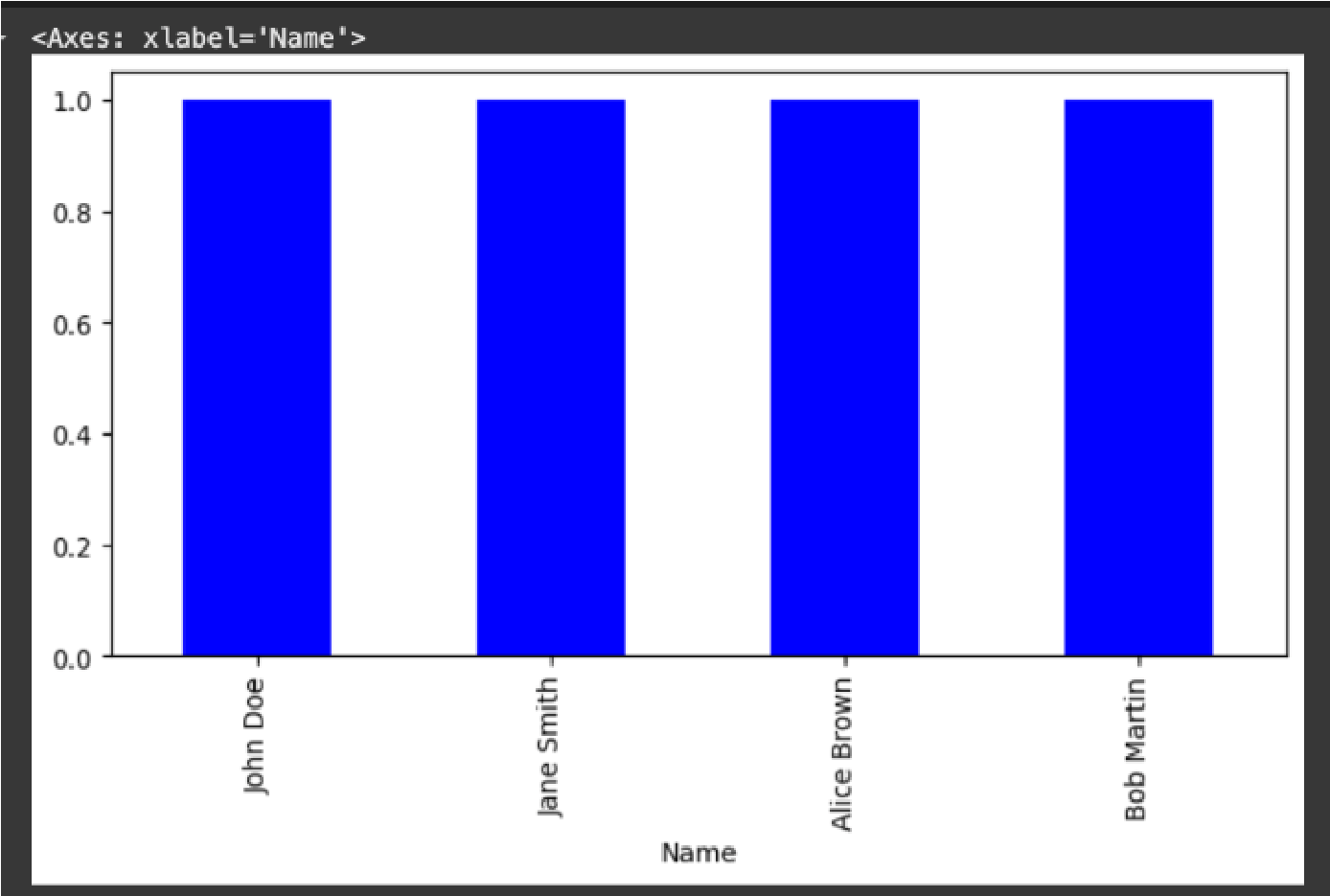
df\_no\_duplicates=df\_no\_duplicates.drop\_duplicates(subset= ['Phone'], keep='first') print("After Removing Duplicates : \n", df\_no\_duplicates)



plt.figure(figsize=(8,4)) df['Name'].value\_counts().plot(kind='bar', color='blue')



plt.figure(figsize=(8,4)) df\_no\_duplicates['Name'].value\_counts().plot(kind='bar', color='blue')



df\_no\_duplicates.to\_csv("cleaned\_data.csv", index=False) print("File Saved”)

**OUTPUT:**



**LEARNING OUTCOME:**