Experiment No 8

* 1. **Aim/Purpose of the Experiment**

To familiarize the students with data visualization using two feature variables.

* 1. **Learning Outcomes**

Knowledge of the Data cleaning, Data preparation and data visualization using bivariate analysis in python.

* 1. **Prerequisites**

Basic knowledge of programming, python syntax, matplotlib, seaborn, different libraries.

* 1. **Materials/Equipment/Apparatus / Devices/Software required**

Jupyter Notebook.

* 1. **Introduction and Theory**

Bivariate analysis is a statistical method used to examine the relationship between two variables. In Python, you can perform bivariate analysis using libraries such as NumPy, Pandas, and Matplotlib/Seaborn for data manipulation, analysis, and visualization. Here's a brief outline of the process:

**Bivariate Analysis**

import warnings

warnings.filterwarnings("ignore")

import pandas as pd, numpy as np

import matplotlib.pyplot as plt, seaborn as sns

%matplotlib inline

**Session- 2, Data Cleaning**

inp0= pd.read\_csv("bank\_marketing\_updated\_v1.csv")

inp0.head()

**Segment- 3, Fixing the Rows and Columns**

inp0=pd.read\_csv("bank\_marketing\_updated\_v1.csv", skiprows= 2)

inp0.head()

inp0.drop("customerid", axis=1, inplace=True)

inp0.head()

inp0['job']=inp0.jobedu.apply(lambda x: x.split(",")[0])

inp0.head()

inp0['education']=inp0.jobedu.apply(lambda x: x.split(",")[1])

inp0.head()

inp0.drop('jobedu',axis= 1, inplace= True)

inp0.head()

inp0[inp0.month.apply(lambda x: isinstance(x,float))== True]

inp0.isnull().sum()

**Segment- 4, Impute/Remove missing values**

inp0.age.isnull().sum()

inp0.shape

float(100.0\*20/45211)

inp1=inp0[-inp0.age.isnull()].copy()

inp1.shape

inp1.month.isnull().sum()

float(100.0\*50/45191)

month\_mode=inp1.month.mode()[0]

month\_mode

inp1.month.fillna(month\_mode, inplace= True)

inp1.month.value\_counts(normalize= True)

inp1.month.isnull().sum()

0

inp1.response.isnull().sum()

30

float(100.0\*30/45191)

0.06638489964816004

inp1= inp1[~inp1.response.isnull()]

inp1.isnull().sum()

inp1.pdays.describe()

inp1.loc[inp1.pdays<0,"pdays"]=np.NaN

inp1.pdays.describe()

**Session- 4, Bivariate and Multivariate Analysis**

plt.scatter(inp1.salary, inp1.balance)

plt.show()

inp1.plot.scatter(x='age', y='balance')

plt.show()

sns.pairplot(data=inp1, vars=["salary","balance", "age"])

plt.show()

sns.heatmap( inp1[["salary","balance", "age"]].corr(), annot= True,

cmap= "Reds")

plt.show()

**Segment- 4, Numerical categorical variable**

**Salary vs response**

inp1.groupby("response")["salary"].mean()

inp1.groupby("response")["salary"].median()

sns.boxplot(data=inp1,x="response", y="salary")

plt.show()

sns.boxplot(data=inp1,x="response", y="balance")

plt.show()

inp1.groupby("response")["balance"].mean()

inp1.groupby("response")["balance"].median()

def p75(x):

return np.quantile(x, 0.75)

response

inp1.groupby("response")["balance"].aggregate(["mean","median",p75])

inp1.groupby("response")

["balance"].aggregate(["mean","median"]).plot.bar()

plt.show()

**Education vs salary**

inp1.groupby("education")["salary"].mean()

inp1.groupby("education")["salary"].median()

**Job vs salary**

inp1.groupby('job')['salary'].mean()

inp1.groupby('job')['salary'].median()

**Segment- 5, Categorical categorical variable**

inp1["response\_flag"]=np.where(inp1.response=="yes", 1, 0)

inp1.response.value\_counts()

inp1.response.value\_counts(normalize= True)

inp1.response\_flag.mean()

**Education vs response rate**

inp1.groupby("education")["response\_flag"].mean()

**Marital vs response rate**

inp1.groupby(["marital"])["response\_flag"].mean()

inp1.groupby(["marital"])["response\_flag"].mean().plot.barh()

plt.show()

**Loans vs response rate**

inp1.groupby(["loan"])["response\_flag"].mean().plot.bar()

plt.show()

**Housing loans vs response rate**

inp1.groupby(["housing"])["response\_flag"].mean().plot.bar()

plt.show()

**Age vs response**

plt.show()

inp1.groupby(['job'])['response\_flag'].mean().plot.barh()

plt.show()