A1.

Code:

import pandas as pd

import numpy as np

data=pd.read\_excel(r"Lab Session1 Data.xlsx",sheet\_name="Purchase data")

d1=data.iloc[0:10,1:4]

A=d1.to\_numpy()

d3=data.iloc[0:10,4:5]

C=d3.to\_numpy()

com=np.linalg.pinv(A)

X=np.matmul(com,C)

cost = np.linalg.pinv(A) @ C

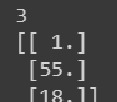
re = np.hstack((A, C))

dimensional=np.linalg.matrix\_rank(re)

print(dimensional)

print(cost)

Output :



A2.

Code:

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report

d1=pd.read\_excel('Lab Session1 Data.xlsx',sheet\_name='Purchase data')

d1.drop(d1.iloc[:,5:22],inplace=True,axis=1)

A=d1.iloc[:,1:-1].values

C=d1.iloc[:,-1].values

A=np.array(A)

C=np.array(C)

print("matrix of A:")

print(A)

print("matrix of C:")

print(C)

rank=np.linalg.matrix\_rank(A)

print("rank of A:", rank)

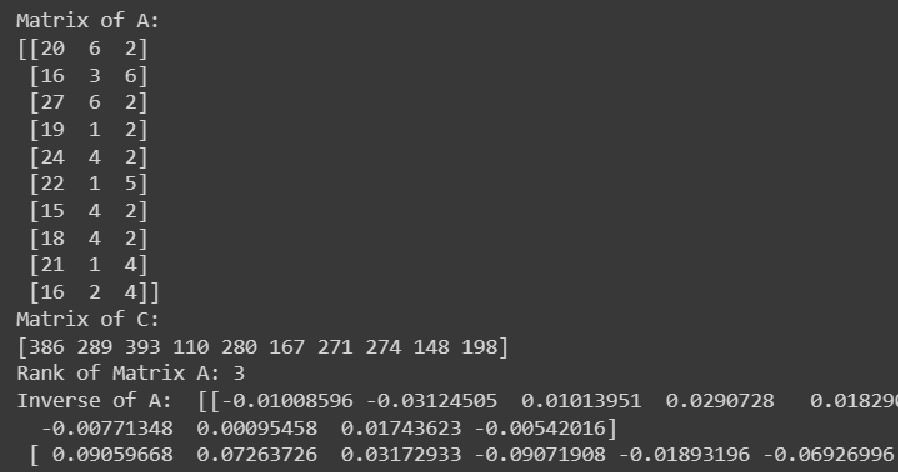
inverse=np.linalg.pinv(A)

print("inverse of A: ",inverse)

Pseudo\_inv=np.matmul(inverse,C)

print("Pseudo inverse is  : ",Pseudo\_inv)

Output:



A3.

Code:

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report

data\_frame = pd.read\_excel(r"Lab Session1 Data.xlsx", sheet\_name="Purchase data")

df=data\_frame.iloc[0:10,0:5]

df['Label'] = df['Payment (Rs)'].apply(lambda x: 'RICH' if x > 200 else 'POOR')

print(df)

X = df.drop(['Customer', 'Payment (Rs)', 'Label'], axis=1)

y = df['Label']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

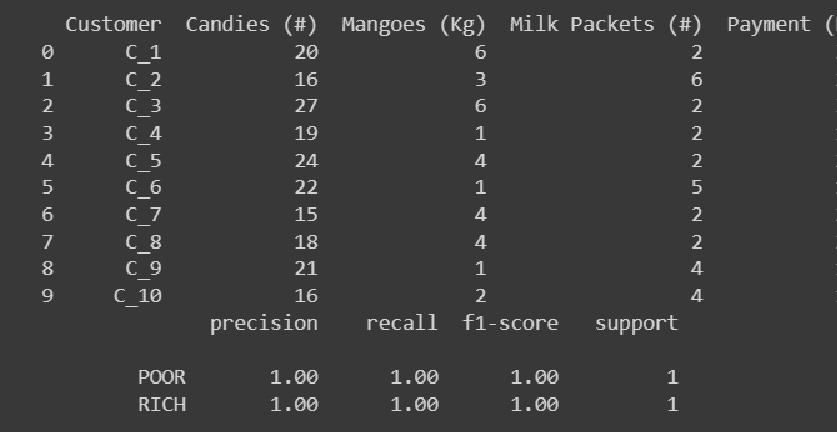
model = RandomForestClassifier(random\_state=42)

model.fit(X\_train\_scaled, y\_train)

y\_pred = model.predict(X\_test\_scaled)

print(classification\_report(y\_test, y\_pred))

Output:



A4.

Code:

import numpy as np

import pandas as pd

from matplotlib import pyplot as plt

df = pd.read\_excel(r"Lab Session1 Data.xlsx",sheet\_name="IRCTC Stock Price")

price = df['Price']

mean = np.mean(price)

variance = np.var(price)

print('Mean for price :', mean)

print('Variance for Price : ', variance)

wednesday\_data = price[df['Day'] == 'Wed']

sp\_mean = np.mean(wednesday\_data)

print('Sample mean:', sp\_mean)

apr\_data = price[df['Month'] == 'Apr']

apr\_mean = np.mean(apr\_data)

print('April mean:', apr\_mean)

chg\_data = df['Chg%']

is\_loss = np.where(chg\_data > 0, False, True)

prob\_of\_loss = np.mean(is\_loss)

print('Probability of making a loss:', prob\_of\_loss)

wednesday\_data = df[df['Day'] == 'Wed']

profit\_on\_wed = np.mean(wednesday\_data['Chg%'] > 0)

print('Probability of making a profit on Wednesday:', profit\_on\_wed)

probability\_of\_wed = np.mean(df['Day'] == 'Wed')

#cp = conditional probability

cp = profit\_on\_wed / probability\_of\_wed

print('Conditional probability of making a profit ,', cp)

plt.scatter(df['Day'], df['Chg%'])

plt.xlabel('Day')

plt.ylabel('Chg%')

plt.title('Chg% vs Day')

plt.show()

Output:

