

Machine Learning Lab Assignment 1

Name - K Dheeraj Reddy

Reg No. - BL.EN.U4CSE21076

CSE - B ; Sem - V

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
```

```
data_frame=pd.read_excel(r"Lab Session1 Data(1).xlsx",sheet_name='Purchase data')
```

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

```
A=np.array(data_frame.iloc[:,1:-1].values)
```

```
C=np.array(data_frame.iloc[:, -1].values)
```

```
print("Matrix A:-")
```

```
print(A)
```

```
print("Matrix C:-")
```

```
print(C)
```

```
print("Rank of Matrix A:-",np.linalg.matrix_rank(A))
```

```
print("Rank of Matrix C:-",np.linalg.matrix_rank(C))
```

```
inverse=np.linalg.pinv(A)
```

```
print("Inverse Matrix of A:-",inverse)
```

```
Pseudo_inv=np.matmul(inverse,C)
```

```
print("Pseudo inverse is the actual cost of each product is : ",Pseudo_inv)
```

```

table=np.array(data_frame['Payment (Rs)'])
number=len(table)
new_row=[]
for i in range(0,number):
    if table[i]>200:
        new_row.append("RICH")
    else:
        new_row.append("POOR")

data_frame.insert(loc = 5,column = 'Label',value = new_row)
print("New Data Excel Sheet for Purchase Data is:-")
print(data_frame)

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

```

```
import pandas as pd
import numpy as np
import statistics as sp
import matplotlib.pyplot as plt

data = pd.read_excel(r"Lab Session1 Data (1).xlsx", sheet_name='IRCTC Stock
Price')

p_data = data['Price']
m_price = sp.mean(p_data)
v_price = sp.variance(p_data)

print("The mean value of the Prices is:-", m_price)
print("The Variance is:-", v_price)

data['Date'] = pd.to_datetime(data['Date'])

wednesday_data = data[data['Date'].dt.day_name() == 'Wednesday']
sample_mean_wednesday = sp.mean(wednesday_data['Price'])

print("Sample mean on Wednesdays:-", sample_mean_wednesday)
print("Population mean (overall mean):-", m_price)

april_data = data[data['Date'].dt.month == 4]

sample_mean_april = sp.mean(april_data['Price'])
print("Sample mean in April:-", sample_mean_april)

loss_probability = len(data[data['Chg%'] < 0]) / len(data)
print("Probability of making a loss over the stock:-", loss_probability)

profit_on_wednesday_probability =
len(wednesday_data[wednesday_data['Chg%'] > 0]) / len(wednesday_data)
print("Probability of making a profit on
Wednesday:--",profit_on_wednesday_probability)
```

```
conditional_profit_probability = len(wednesday_data[wednesday_data['Chg%'] >
0]) / len(wednesday_data)
print("Conditional probability of making profit on Wednesday:-",
conditional_profit_probability)
```

```
plt.scatter(data['Date'].dt.weekday, data['Chg%'])
plt.xlabel("Day of the Week")
plt.ylabel("Chg%")
plt.title("Chg% Data vs. Day of the Week")
plt.xticks(np.arange(7), ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'])
plt.show()
```