**-pwd**

**-ls**

**-hadoop jar WordCount.jar WordCount.WordCount1 dir2**

**-hadoop fs -cat dir2/part-r-00000**

**Assignment No 4 Facebook**

import pandas as pd

import numpy as np

df = pd.read\_csv('dataset\_Facebook.csv',sep = ';')

df

df.head()

df.info()

df.dropna(how = 'any' , axis = 0)

df1 = df.loc[1:245, ['Category', 'Lifetime Post Total Reach','Type','Total Interactions']]

df1

df2 = df.loc[245:500, ['Post Month','Post Weekday','Post Hour','Lifetime Post Consumers']]

df2

df.transpose()

df1.transpose()

df2.transpose()

df.sort\_values(by = 'Category')

df.sort\_index()

df.sort\_index(ascending = False)

df.head()

df1.shape

df2.shape

df.melt()

df\_row = pd.concat([df1,df2])

df\_row

**Assignment No 5 : Air Quality**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.linear\_model import LogisticRegression

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import accuracy\_score

Data = pd.read\_csv('airquality.csv')

Data

Data.head()

Data.info()

Data.isnull().sum()

DataCleaned = Data.dropna()

DataCleaned

Data1 = Data.loc[1:55, ['Ozone','Solar.R','Wind']]

Data1

Data2 = Data.loc[55:111, ['Temp','Month','Day']]

Data2

Data\_main=pd.concat([Data1,Data2])

Data\_main

# Sample dataset

Data\_main = {'Month': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12],

'Value': [100, 120, 90, 110, 130, 140, 150, 160, 170, 180, 190, 200]}

# Create DataFrame

Data\_main = pd.read\_csv('AirQuality.csv')

# Map numerical values to month names

month\_mapping = {

1: 'January',

2: 'February',

3: 'March',

4: 'April',

5: 'May',

6: 'June',

7: 'July',

8: 'August',

9: 'September',

10: 'October',

11: 'November',

12: 'December'

}

# Apply mapping to the 'Month' column

Data\_main['Month'] = Data\_main['Month'].map(month\_mapping)

Data\_main

Data\_main.head()

Data\_main['Temp'].unique()

Data\_main.Temp.value\_counts()

Data\_main.count()

from sklearn.linear\_model import LinearRegression

# Load the dataset

data = pd.read\_csv('AirQuality.csv') # Replace 'AirQuality.csv' with the actual filename

# Drop rows with missing values

data.dropna(subset=['Ozone', 'Solar.R'], inplace=True)

# Select features and target variable

X = data[['Solar.R']]

y = data['Ozone']

# Create and fit the model

model = LinearRegression()

model.fit(X, y)

# Make predictions

y\_pred = model.predict(X)

# Plot the data points

plt.scatter(X, y, color='blue', label='Data points')

# Plot the regression line

plt.plot(X, y\_pred, color='red', label='Linear regression')

# Add labels and legend

plt.title('Linear Regression Model for Ozone and Solar.R')

plt.xlabel('Solar Radiation (Solar.R)')

plt.ylabel('Ozone Concentration (Ozone)')

plt.legend()

# Show plot

plt.show()

**Assignment No 5 : Heart**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import random as rd

ds=pd.read\_csv("heart.csv")

ds

ds.head()

ds.info()

#Data Cleaning:

ds.isnull().sum()

ds.dropna

#Data Integration

ds1 = ds.loc[1:150, ['age','sex','cp','trestbps','chol','fbs','restecg']]

ds1

ds2 = ds.loc[150:300, ['thalach','exang','oldpeak','slope','ca','thal','target']]

ds2

ds\_main=pd.concat([ds1,ds2])

ds\_main

#Data Transformation

import pandas as pd

# Load the dataset

heart\_data = pd.read\_csv('heart.csv')

# Choose the column to transform

column\_to\_transform = 'restecg'

# Apply transformation

heart\_data[column\_to\_transform] = heart\_data[column\_to\_transform].replace({1: True, 0: False})

# Save the transformed dataset

heart\_data.to\_csv('transformed\_heart.csv', index=False)

heart\_data

ds['oldpeak']= ds['oldpeak'].astype(int)

ds.dtypes

ds.tail()

ds.groupby('target').mean()

#Error Correcting

ds.head()

ds['ca'].unique()

ds.ca.value\_counts()

ds.count()

# Load the dataset

heart\_data = pd.read\_csv('heart.csv')

# Choose two columns for plotting

x\_column = 'age'

y\_column = 'chol'

# Plot scatter plot

plt.scatter(heart\_data[x\_column], heart\_data[y\_column], color='skyblue')

plt.title(f'Scatter plot between {x\_column} and {y\_column}')

plt.xlabel(x\_column)

plt.ylabel(y\_column)

plt.grid(True)

plt.show()

**Assignment No 7 : Air Quality**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import random as rd

ds\_airquality=pd.read\_csv("AirQuality.csv")

ds\_airquality

ds\_airquality.info()

import seaborn as sns

%matplotlib notebook

plt.scatter(x='Ozone',y='Solar.R',data=ds\_airquality,c='g')

plt.scatter(x='Ozone',y='Solar.R',data=ds\_airquality, c='r',marker='+')

plt.ylabel('Ozone')

plt.ylabel('Ozone and Solar.R')

plt.legend(title='scatter\_plot')

plt.show()

plt.figure()

sns.lmplot(x='Ozone' , y='Solar.R', data=ds\_airquality, fit\_reg=False)

plt.figure()

ds\_airquality.plot(kind='hist')

plt.figure()

sns.displot(ds\_airquality.Wind)

plt.figure()

plt.xlabel('Month')

plt.ylabel('count')

plt.legend()

ds\_airquality.Month.value\_counts().plot(kind='bar')

plt.figure()

sns.countplot(ds\_airquality.Month)

plt.figure()

sns.lineplot(x='Temp', y='Month', data=ds\_airquality)

import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))

ds\_airquality['Month'].value\_counts().plot(kind='pie', utopct='%1.1f%%', startangle=140)

plt.title('Distribution of Months')

plt.ylabel('')

plt.show()

**Assignment No 7 : Heart**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import random as rd

ds\_heart=pd.read\_csv("Heart.csv")

ds\_heart

ds\_heart.info()

import seaborn as sns

%matplotlib notebook

plt.scatter(x='age',y='restecg',data=ds\_heart,c='g')

plt.scatter(x='age',y='chol',data=ds\_heart, c='r',marker='+')

plt.ylabel('age')

plt.ylabel('chol and restecg')

plt.legend(title='scatter\_plot')

plt.show()

plt.figure()

sns.lmplot(x='age' , y='chol', data=ds\_heart, fit\_reg=False)

plt.figure()

ds\_heart.plot(kind='hist')

plt.figure()

sns.displot(ds\_heart.age)

plt.figure()

plt.xlabel('cp')

plt.ylabel('count')

plt.legend()

ds\_heart.cp.value\_counts().plot(kind='bar')

plt.figure()

sns.countplot(ds\_heart.cp)

plt.figure()

sns.lineplot(x='age', y='chol', data=ds\_heart)

import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))

ds\_heart['cp'].value\_counts().plot(kind='pie', autopct='%1.1f%%', startangle=140)

plt.title('Distribution of Chest Pain Types')

plt.ylabel('')

plt.show()