\*\*Assignment No 1\*\*

import numpy as np

import pandas as pd

df=pd.read\_csv("/content/Iris (1).csv")

df

df.size

#total no of rows and columns

df.shape

#display type of each column

df.dtypes

#sorting data in ascending order by considering column sepal length

df.sort\_values(by="SepalLengthCm", ascending=False)

# Slice the data: 11 to 20 rows, and only two columns, sepal.length and Species

df.loc[11:20,["SepalLengthCm","type"]]

#rename the column Species to Type

df.rename(columns={"Species":"type"})

\*\*Assignment no 2\*\*

Consider the given dataset StudentsPerformanceTest1

1. Check that is there any missing values in dataframe as a whole

2. is there any missing values across each column

3. count of missing values across each column

4. count row wise missing value

5. count of missing values of a gender column.

6. groupby count of missing values of a column , consider column gender and score

7. replace the missing value of score column with average value of the column

df2=pd.read\_csv("/content/Student Performance Missing values.csv")

df2

#1.Check that is there any missing values in dataframe as a whole

df2.isnull()

#2.is there any missing values across each column

df2.isnull().any()

#3.count of missing values across each column

df2.isnull().sum()

#4.count row wise missing value

df2.isnull().sum(axis=1)

#5.count of missing values of a gender column.

df2.Gender.isnull().sum()

#6.groupby count of missing values of a column , consider column gender and score

df2.groupby(["Gender"])["Math Score"].apply(lambda x: x.isnull().sum())

#7.replace the missing value of score column with average value of the column

ab=df2["Math Score"].mean()

df2["Math Score"].fillna(value=ab,inplace=True)

df2

\*\*Assignment no 3\*\*

#Perform the label encoding , by considering Species as target variable.

from sklearn.preprocessing import LabelEncoder

abc=LabelEncoder()

df["New\_Species"]=abc.fit\_transform(df["type"])

df

\*\*Assignment no 4\*\*

#Perform the one hot Encoding , by considering Species as target variable.

dummies=pd.get\_dummies(df.Species)

dummies

\*\*Assignment no 5\*\*

#Perform the Dummy Variable Encoding , by considering Species as target variable.

dummies1=pd.get\_dummies(df.Species,drop\_first=True)

dummies1

\*\*Assignment No 6\*\*

#. Creation of Dataset " StudentsPerformance" using Microsoft Excel.

sp=pd.read\_csv("/content/Student Performance.csv")

sp

sp.info()

sp.head()

sp.tail()

sp.describe()

sp.size

sp.mean()

sp.mode()

sp.median()

\*\*Assignment no 7\*\*

#1. Load the Academic Performance dataset in data frame object.

df2

#2. Check null values in the dataset.

df2.isnull()

#3. Check missing values in dataset and replace the null values with standard null value NaN

df2.replace("na",np.NaN)

df2

#4. Replace the missing value of Math Score with Mean Value

xyz=df2["Math Score"].mean

df2["Math Score"].fillna(value=xyz,inplace=True)

df2

#5. Replace the missing value of Reading Score with standard deviation

xyz1=df2["Reading Score"].std()

df2["Reading Score"].fillna(value=xyz1,inplace=True)

df2

#6. Replace the missing value of place with common value "Nashik"

df2["Region"].fillna(value="nashik",inplace=True)

df2

\*\*Assignment NO 8\*\*

#1. Load the Academic Performance dataset in data frame object.

df2

#2. Check null values in the dataset.

df2.isnull()

#3.Count the number of null values in complete data set

#Dropping rows with at least 1 null value

df2.dropna()

#5.To Drop rows if all values in that row are missing

df2.dropna(how = 'all')

#6.To Drop columns with at least 1 null value.

df2.dropna(axis = 1)

#7.To drop rows with at least 1 null value in CSV file. making new data frame with dropped NA values

new\_data = df2.dropna(axis = 0, how ='any')

new\_data

\*\*Assignment no 9\*\*

import matplotlib.pyplot as plt

#1.. Detect the outlier using BoxPlot.

col = ['Math Score', 'Reading Score' , 'Writing Score','Placement Score']

df2.boxplot(col)

#2.Handle the outlier using Quantile based flooring and capping

sp=pd.read\_csv("/content/Student Performance.csv")

sp\_stud=sp

ninetieth\_percentile = np.percentile(sp\_stud['math score'], 90)

b = np.where(sp\_stud['math score']>ninetieth\_percentile,

ninetieth\_percentile, sp\_stud['math score'])

print("New array:",b)

sp\_stud.insert(1,"m score",b,True)

sp\_stud

\*\*Assignment no 10\*\*

#1.scatter plot outliers

fig, ax = plt.subplots(figsize = (18,10))

ax.scatter(sp['placement score'], sp['placement offer count'])

#2.Handle the outlier using Quantile based flooring and capping

sp=pd.read\_csv("/content/Student Performance.csv")

sp\_stud=sp

ninetieth\_percentile = np.percentile(sp\_stud['math score'], 90)

b = np.where(sp\_stud['math score']>ninetieth\_percentile,

ninetieth\_percentile, sp\_stud['math score'])

print("New array:",b)

sp\_stud.insert(1,"m score",b,True)

sp\_stud

\*\*Assignment no 11\*\*

from scipy import stats

#1.z score outliers

z = np.abs(stats.zscore(sp['math score']))

z

threshold=0.18

sample\_outliers=np.where(z<threshold)

sample\_outliers

#2. replace the outliers with the median value.

#Replace the upper bound outliers using median value

median=np.median(sorted\_rscore)

refined\_sp=sp

refined\_sp['reading score'] = np.where(refined\_sp['reading score'] >upr\_bound, median,refined\_sp['reading score'])

refined\_sp

#Replace the lower bound outliers using median value

refined\_sp['reading score'] = np.where(refined\_sp['reading score'] <lwr\_bound, median,refined\_sp['reading score'])

\*\*Assignment no 12\*\*

sorted\_rscore= sorted(sp['reading score'])

q1 = np.percentile(sorted\_rscore, 25)

q3 = np.percentile(sorted\_rscore, 75)

print(q1,q3)

IQR=q3-q1

lwr\_bound = q1-(1.5\*IQR)

upr\_bound = q3+(1.5\*IQR)

print(lwr\_bound, upr\_bound)

r\_outliers = []

for i in sorted\_rscore:

if (i<lwr\_bound or i>upr\_bound):

r\_outliers.append(i)

print(r\_outliers)

#remove outliers

updated\_sp=sp

for i in sample\_outliers:

updated\_sp.drop(i,inplace=True)

updated\_sp