Practical 1:-

dftemp = df.dropna(subset=['temperature'])

print("temp null")

print(dftemp)

print("==================")

mean\_temp = df['temperature'].mean()

df['temperature'] = df['temperature'].fillna(mean\_temp)

print(df)

practical 2:-

new\_df = df

col = ['Maths\_Score']

new\_df.boxplot(col)

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

ax.scatter(df['Placement\_Score'],df['Placement offer count'])

plt.show()

outliers = np.where((df[col]<lwr\_bound)|(df[col]>uppr\_bound))

practical 3:-

dfSum = df.isnull().sum()

all\_mean = df.mean(numeric\_only = True)

dfAge = df['Age'].mean()

numeric\_df = df.select\_dtypes(include=[int,float])

iris\_set1 = (df['Species']=="Iris-setosa")

print(df[iris\_set1].describe())

practical 8:-

import seaborn as sns

sns.distplot(dataset['Fare'])

sns.jointplot(x='Age', y='Fare', data=dataset)

sns.pairplot(data=dataset)

sns.rugplot(dataset['Fare'])

sns.barplot(dataset,x='Age',y='Fare')

practical9:-

df1=df.dropna()

df2=df1.drop(['Name', 'SibSp','Ticket','Pclass','PassengerId','Parch','Cabin','Embarked'], axis=1)

sns.set\_theme(style="ticks", color\_codes=True)

sns.countplot(x ='Sex', data = df2)

plt.show()

sns.countplot(x ='Survived', data = df2)

plt.show()

sns.set\_style("whitegrid")

sns.boxplot(x = 'Sex', y = 'Age', data = df2)

sns.boxplot(x = df2['Sex'],

y = df2['Age'],

hue = df2['Survived'],

palette = 'Set2')

Practical 10:-

fig, axes = plt.subplots(2, 2, figsize=(16, 8))

axes[0,0].set\_title("Distribution of First Column")

axes[0,0].hist(df["SepalLengthCm"]);

axes[0,1].set\_title("Distribution of Second Column")

axes[0,1].hist(df["SepalWidthCm"]);

axes[1,0].set\_title("Distribution of Third Column")

axes[1,0].hist(df["PetalLengthCm"]);

axes[1,1].set\_title("Distribution of Fourth Column")

axes[1,1].hist(df["PetalWidthCm"]);

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

dataset = pd.read\_csv('IrisDataset.csv')

print(dataset.head())

gk = dataset.groupby("species")

print(gk.first())

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

x= dataset.iloc[:,:4].values

y= dataset['species'].values

x\_train ,x\_test, y\_train, y\_test = train\_test\_split(x,y,test\_size=0.2)

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

x\_train = sc.fit\_transform(x\_train)

x\_test = sc.fit\_transform(x\_test)

from sklearn.naive\_bayes import GaussianNB

classifier = GaussianNB()

classifier.fit(x\_train,y\_train)

y\_pred = classifier.predict(x\_test)

print(y\_pred)

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test,y\_pred)

from sklearn.metrics import accuracy\_score

print("Accuracy: ", accuracy\_score(y\_test, y\_pred))

print(cm)

df=pd.DataFrame({'Real Values':y\_test,'Prediected values':y\_pred})

print(df)