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)
```