Data Science Slips Solutions :---

Slip 1:--

Q.2 A) Write a Python program to create a Pie plot to get the frequency of the three species of the Iris data (Use iris.csv)

```
import pandas as pd
import matplotlib.pyplot as plt
iris = pd.read_csv("iris.csv")
ax=plt.subplots(1,1,figsize=(10,8))
iris['Species'].value_counts().plot.pie(explode=[0.1,0.1,0.1],autopct=
'%1.1f%%',shadow=True,figsize=(10,8))
plt.title("Iris Species %")
plt.show()
```

B) Write a Python program to view basic statistical details of the data.(Use wineequality-red.csv

```
import pandas as pd
data = pd.read_csv("iris.csv")
print(data.describe())
```

Slip 2:--

Q.2 A) Write a Python program for Handling Missing Value. Replace missing value of salary, age column with mean of that column.(Use Data.csv file).

```
import pandas as pd
import numpy as np

df = pd.read_csv("/Users/hitesh/Downloads/Placement_Data_Full_Class.csv")

df.head()

df.fillna(df.mean())
df.fillna(df.median())


df['salary'] = df['salary'].fillna(df['salary'].mode()[0])
```

Q.2 B) Write a Python program to generate a line plot of name Vs salary

```
# importing the required libraries
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# define data values
Data = read_csv('\File Path')
x = Data['Name'] # X-axis points
y = Data['Salaty']# Y-axis points

plt.plot(x, y) # Plot the chart
plt.show() # display
```

Slip nos 3:--

Q.2 A)Write a Python program to create box plots to see how each feature i.e. Sepal Length, Sepal Width, Petal Length, Petal Width are distributed across the three species. (Use iris.csv dataset)

```
import pandas as pd
import matplotlib.pyplot as plt
iris = pd.read_csv("iris.csv")
# Drop id column
new_data = iris.drop('Id',axis=1)
new_data.hist(edgecolor='black', linewidth=1.2)
fig=plt.gcf()
fig.set_size_inches(12,12)
plt.show()
```

Q.2 B) Write a Python program to view basic statistical details of the data (Use Heights and Weights Dataset)

```
import pandas as pd
data = pd.read_csv("iris.csv")
print(data.describe())
```

Slips nos 4:--

Q.2 A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate color, labels and styling options.

```
import math
import random
import matplotlib.pyplot as plt
# create random data
no_of_balls = 25
x = [random.triangular() for i in range(no_of_balls)]
y = [random.gauss(0.5, 0.25) for i in range(no_of_balls)]
colors = [random.randint(1, 4) for i in range(no_of_balls)]
areas = [math.pi * random.randint(5, 15)**2 for i in
range(no_of_balls)]
# draw the plot
plt.figure()
plt.scatter(x, y, s=areas, c=colors, alpha=0.85)
plt.hist(x , y ,edgecolor = ‘ivory’,  colors – ‘red’)
plt.plot(x ,y)
plt.boxplot(x ,y)
plt.axis([0.0, 1.0, 0.0, 1.0])
plt.xlabel("X")
plt.ylabel("Y")
plt.show()
```

Q.2 B) Write a Python program to print the shape, number of rows-columns, data types, feature names and the description of the data(Use User_Data.csv)

```
import pandas as pd
iris_data = pd.read_csv("iris.csv")
print("\nKeys of Iris dataset:")
print(iris_data.keys())
print("\nNumber of rows and columns of Iris dataset:")
print(iris_data.shape)
```

Slips nos 5:--

Q.2 A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate color, labels and styling options.

```
import math
import random
import matplotlib.pyplot as plt
# create random data
no_of_balls = 25
x = [random.triangular() for i in range(no_of_balls)]
y = [random.gauss(0.5, 0.25) for i in range(no_of_balls)]
colors = [random.randint(1, 4) for i in range(no_of_balls)]
areas = [math.pi * random.randint(5, 15)**2 for i in
range(no_of_balls)]
# draw the plot
plt.figure()
plt.scatter(x, y, s=areas, c=colors, alpha=0.85)
plt.hist(x , y ,edgecolor = ‘ivory’,  colors – ‘red’)
plt.plot(x ,y)
plt.boxplot(x ,y)
plt.axis([0.0, 1.0, 0.0, 1.0])
plt.xlabel("X")
plt.ylabel("Y")
plt.show()
```

Q.2 B) Write a Python program to print the shape, number of rows-columns, data types, feature names and the description of the data(Use User_Data.csv)

```
import pandas as pd
iris_data = pd.read_csv("iris.csv")
print("\nKeys of Iris dataset:")
print(iris_data.keys())
print("\nNumber of rows and columns of Iris dataset:")
print(iris_data.shape)
```

Slips nos 6:--

Q.2 A) Write a Python program for Handling Missing Value. Replace missing value of salary, age column with mean of that column.(Use Data.csv file).

```
import pandas as pd
import numpy as np

df = pd.read_csv("/Users/hitesh/Downloads/Placement_Data_Full_Class.csv")

df.head()

df.fillna(df.mean())
df.fillna(df.median())

df['salary'] = df['salary'].fillna(df['salary'].mode()[0])
```

Q.2 B) Write a Python program to generate a line plot of name Vs salary

```
# importing the required libraries
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# define data values
Data = read_csv(‘\File Path’)
x = Data[‘Name’] # X-axis points
y = Data[‘Salaty’]# Y-axis points

plt.plot(x, y) # Plot the chart
plt.show() # display
```

Slips nos 7:-→

Q.2) Write a Python program to perform the following tasks : a. Apply OneHot coding on Country column. b. Apply Label encoding on purchased column (Data.csv have two categorical column the country column, and the purchased column).

```
#importing libraries
import pandas as pd
import numpy as np
from sklearn.preprocessing import OneHotEncoder

#Retrieving data
data = pd.read_csv('Employee_data.csv')

# Converting type of columns to category
data[‘Country’]=data['Country'].astype('category')
data['Purchased']=data['Purchased'].astype('category')


#Assigning numerical values and storing it in another columns
data['Gen_new']=data['Country'].cat.codes
data['Rem_new']=data['Purchased'].cat.codes


#Create an instance of One-hot-encoder
enc=OneHotEncoder()

#Passing encoded columns
'''
NOTE: we have converted the enc.fit_transform() method to array because the
fit_transform method
of OneHotEncoder returns SpiPy sparse matrix this enables us to save space when we
have huge number of categorical variables
'''
enc_data=pd.DataFrame(enc.fit_transform(data[['Gen_new','Rem_new']]).toarray())

#Merge with main
New_df=data.join(enc_data)

print(New_df)
# Import libraries
```

```
import numpy as np
import pandas as pd

# Import dataset
df = pd.read_csv('../../data/Iris.csv')

df['species'].unique()

# Import label encoder
from sklearn import preprocessing

# label_encoder object knows how to understand word labels.
label_encoder = preprocessing.LabelEncoder()

# Encode labels in column 'species'.
df['Country']= label_encoder.fit_transform(df['Country'])

df['Country'].unique()
df['Purchased']= label_encoder.fit_transform(df['Purchased'])

df['Purchased'].unique()
```

Slips nos 8:--

Q.2) Write a program in python to perform following task : [15] Standardizing Data (transform them into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1) (Use winequality-red.csv)

Standardize the Data :--

```
# Standardize time series data

from pandas import read_csv

from sklearn.preprocessing import StandardScaler

from math import sqrt

# load the dataset and print the first 5 rows

series = read_csv('daily-minimum-temperatures-in-me.csv', header=0, index_col=0)

print(series.head())

# prepare data for standardization

values = series.values

values = values.reshape((len(values), 1))

# train the standardization

scaler = StandardScaler()

scaler = scaler.fit(values)

print('Mean: %f, StandardDeviation: %f' % (scaler.mean_, sqrt(scaler.var_)))

# standardization the dataset and print the first 5 rows

normalized = scaler.transform(values)

for i in range(5):

        print(normalized[i])

# inverse transform and print the first 5 rows

inversed = scaler.inverse_transform(normalized)

for i in range(5):

 print(inversed[i])
```

Slips nos9:--

Q.2 A) Generate a random array of 50 integers and display them using a line chart, scatter plot. Apply appropriate color, labels and styling options.

```
import math
import random
import matplotlib.pyplot as plt
# create random data
no_of_balls = 25
x = [random.triangular() for i in range(no_of_balls)]
y = [random.gauss(0.5, 0.25) for i in range(no_of_balls)]
colors = [random.randint(1, 4) for i in range(no_of_balls)]
areas = [math.pi * random.randint(5, 15)**2 for i in
range(no_of_balls)]
# draw the plot
plt.figure()
plt.scatter(x, y, s=areas, c=colors, alpha=0.85)
plt.hist(x , y ,edgecolor = ‘ivory’,  colors – ‘red’)
plt.plot(x ,y)
plt.boxplot(x ,y)
plt.axis([0.0, 1.0, 0.0, 1.0])
plt.xlabel("X")
plt.ylabel("Y")
plt.show()
```

Q.2 B) Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in a pie chart.

```
# Import libraries

from matplotlib import pyplot as plt

import numpy as np




# Creating dataset
```

```
cars = ['AUDI', 'BMW', 'FORD',

                    'TESLA', 'JAGUAR', 'MERCEDES']


data = [23, 17, 35, 29, 12, 41]


# Creating plot

fig = plt.figure(figsize =(10, 7))

plt.pie(data, labels = cars)


# show plot

plt.show()
```

Q.2 C) Write a program in python to perform following task (Use winequality-red.csv) [5] Import Dataset and do the followings: a) Describing the dataset b) Shape of the dataset c) Display first 3 rows from dataset.

```
import pandas as pd
data = pd.read_csv("iris.csv")
print("Shape of the data:")
print(data.shape)
print("\nData Type:")
print(type(data))
print("\nFirst 3 rows:")
print(data.head(3))
```

Slip nos 10:--

Q.2 A) Write a python program to Display column-wise mean, and median for SOCRHeightWeight dataset.

```
import pandas as pd
import numpy as np
raw_data = {
        'Height': [175, 199, 158, 181],

        'Weight': [88, 92, 95, 70]}
df = pd.DataFrame(raw_data)
df
```

```
df['Height'].mean()
```

```
df['weight'].median()
df.describe()
```

Q.2 B) Write a python program to compute sum of Manhattan distance between all pairs of points.

```
# Manhattan distance between all
# the pairs of given points

# Return the sum of distance
# between all the pair of points.
def distancesum (x, y, n):
    sum = 0

    # for each point, finding distance
    # to rest of the point
    for i in range(n):
        for j in range(i+1,n):
            sum += (abs(x[i] - x[j]) +
                            abs(y[i] - y[j]))

    return sum

# Driven Code
x = [ -1, 1, 3, 2 ]
y = [ 5, 6, 5, 3 ]
n = len(x)
print(distancesum(x, y, n) )

# This code is contributed by "Sharad_Bhardwaj".
```

Slips nos 11:--

Q.2 A) Write a Python program to create a Pie plot to get the frequency of the three species of the Iris data (Use iris.csv)

```
import pandas as pd
import matplotlib.pyplot as plt
iris = pd.read_csv("iris.csv")
ax=plt.subplots(1,1,figsize=(10,8))
iris['Species'].value_counts().plot.pie(explode=[0.1,0.1,0.1],autopct=
'%1.1f%%',shadow=True,figsize=(10,8))
plt.title("Iris Species %")
plt.show()
```

B) Write a Python program to view basic statistical details of the data.(Use wineequality-red.csv

```
import pandas as pd
data = pd.read_csv("iris.csv")
print(data.describe())
```

Slips nos 12:--

Q.2 A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate color, labels and styling options.

```
import math
import random
import matplotlib.pyplot as plt
# create random data
no_of_balls = 25
x = [random.triangular() for i in range(no_of_balls)]
y = [random.gauss(0.5, 0.25) for i in range(no_of_balls)]
colors = [random.randint(1, 4) for i in range(no_of_balls)]
areas = [math.pi * random.randint(5, 15)**2 for i in
range(no_of_balls)]
# draw the plot
plt.figure()
plt.scatter(x, y, s=areas, c=colors, alpha=0.85)
plt.hist(x , y ,edgecolor = ‘ivory’,  colors – ‘red’)
plt.plot(x ,y)
plt.boxplot(x ,y)
plt.axis([0.0, 1.0, 0.0, 1.0])
plt.xlabel("X")
plt.ylabel("Y")
plt.show()
```

Q.2 B) Write a Python program to create data frame containing column name, salary, department add 10 rows with some missing and duplicate values to the data frame. Also drop all null and empty values. Print the modified data frame.

```
Import pandas as pd
import numpy as np

exam data = {
           ‘salary’:[100000 , 1255000  , 45680335 ,
45444,4444,787156,4568844,645694,646845,7984546]
‘Department’:[‘DA , ‘DS’ , ‘Execl Analyst’ , ‘Developer’,
‘JAVA Developer’ , ‘C# Developer’ ,’Python Developer’ , ‘SDE’
, ‘ ‘ , ‘’ , nan ,’ ‘ ]

}
df = pd.DataFrame(exam_data , index=labels)
print(df)

df.isnull()

df.notnull()
```

Slips nos 13:-

Q.2 A) Write a Python program to create a graph to find relationship between the petal length and petal width.(Use iris.csv dataset)

```
import pandas as pd
import matplotlib.pyplot as plt
iris = pd.read_csv("iris.csv")
fig = iris[iris.Species=='Iris-setosa'].plot.scatter(x='PetalLengthCm',y='PetalWidthCm',color='orange', label='Setosa')
iris[iris.Species=='Iris-versicolor'].plot.scatter(x='PetalLengthCm',y='PetalWidthCm',color='blue', label='versicolor',ax=fig)
iris[iris.Species=='Iris-virginica'].plot.scatter(x='PetalLengthCm',y='PetalWidthCm',color='green', label='virginica', ax=fig)
fig.set_xlabel("Petal Length")
fig.set_ylabel("Petal Width")
fig.set_title(" Petal Length VS Width")
fig=plt.gcf()
fig.set_size_inches(12,8)
plt.show()
```

Q.2 B) Write a Python program to find the maximum and minimum value of a given flattened array.

```
import numpy as np
a = np.arange(4).reshape((2,2))
print("Original flattened array:")
print(a)
print("Maximum value of the above flattened array:")
print(np.amax(a))
print("Minimum value of the above flattened array:")
print(np.amin(a))
```

Slips nos 14 :--

Q. 2 A) Write a Python NumPy program to compute the weighted average along the specified axis of a given flattened array.

```
import numpy as np
a = np.arange(4).reshape((2,2))
print("Original flattened array:")
print(a)
print("Maximum value of the above flattened array:")
print(np.amax(a))
print("Minimum value of the above flattened array:")
print(np.amin(a))
```

Q. 2 B) Write a Python program to view basic statistical details of the data (Use advertising.csv)

```
import pandas as pd
data = pd.read_csv("advertising.csv")
print(data.describe())
```

Slips nos 15:--

Q.2 A) Generate a random array of 50 integers and display them using a line chart, scatter plot, histogram and box plot. Apply appropriate color, labels and styling options.

```
import math
import random
import matplotlib.pyplot as plt
# create random data
no_of_balls = 25
x = [random.triangular() for i in range(no_of_balls)]
y = [random.gauss(0.5, 0.25) for i in range(no_of_balls)]
colors = [random.randint(1, 4) for i in range(no_of_balls)]
areas = [math.pi * random.randint(5, 15)**2 for i in
range(no_of_balls)]
# draw the plot
plt.figure()
plt.scatter(x, y, s=areas, c=colors, alpha=0.85)
plt.hist(x , y ,edgecolor = ‘ivory’,  colors – ‘red’)
plt.plot(x ,y)
plt.boxplot(x ,y)
plt.axis([0.0, 1.0, 0.0, 1.0])
plt.xlabel("X")
plt.ylabel("Y")
plt.show()
```

Q.2 B) Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in a pie chart.

```
# Import libraries

from matplotlib import pyplot as plt

import numpy as np
```

```
# Creating dataset
cars = ['DS', 'DSA', 'C++',
           'DA', 'C', 'JAVA']

data = [23, 17, 35, 29, 12, 41]

# Creating plot
fig = plt.figure(figsize =(10, 7))
plt.pie(data, labels = cars)

# show plot
plt.show()
```

```
Slip nos 16:--
```

Q.2 A) Write a python program to create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in a pie chart and bar chart

```
# Import libraries

from matplotlib import pyplot as plt

import numpy as np



# Creating dataset

cars = ['DS', 'DSA', 'C++',

           'DA', 'C', 'JAVA']


data = [23, 17, 35, 29, 12, 41]


# Creating plot

fig = plt.figure(figsize =(10, 7))

plt.pie(data, labels = cars)


# show plot

plt.show()
```

Q.2 B) Write a python program to create a data frame for students' information such as name, graduation percentage and age. Display average age of students, average of graduation percentage.

```
import pandas as pd
import numpy as np
```

```
extract data = {

‘Name’ : [‘Hitesh’ ,’Imran Qureshi’ , ‘Faizan’]

‘Grad_Per’ : [50 , 100 , 50  ]

‘Age’ : [19 ,30 , 31]


}

df = pd.DataFrame(exam_data , index=labels)
print(df)

df.percentile()

df.mean()

df[‘Grad_Per’].mean()
```