Q.1 write a R program to calculate the multiplication table using a function.

```
num = as.integer(readline(prompt = "Enter a number: "))
for(i in 1:10)
{
print(paste(num,'x', i, '=', num*i))
}
```

Q.2 Write a python program the Categorical values in numeric format for a given dataset.

```
import numpy as np

        import pandas as pd

dataset = pd.read_csv("play_tennis.csv")

         dataset
         from sklearn import preprocessing
         le = preprocessing.LabelEncoder()
         dataset['outlook'] = le.fit_transform(dataset.outlook)
         dataset['temp'] = le.fit_transform(dataset.temp)
         dataset['humidity'] = le.fit_transform(dataset.humidity)
         dataset['play'] = le.fit_transform(dataset.play)
         x = dataset.iloc[:,:-1].values
         y = dataset.iloc[:,5].values
         from sklearn.preprocessing import StandardScaler
         st_x = StandardScaler()
         x1 = st_x.fit_transform(x)
         print(x1)
```

Q.1 Consider the student data set It can be downloaded from: https://drive.google.com/open?id=1oakZCv7g3mlmCSdv9J8kdSaqO5_6dIOw Write a programme in python to apply simple linear regression and find out mean absolute error, mean squared error and root mean squared error.

```
import numpy as nm

import pandas as pd

data_set= pd.read_csv('student_scores.csv')

print(data_set)

y = data_set['Scores'].values.reshape(-1, 1)

X = data_set['Hours'].values.reshape(-1, 1)

print(X)
print(y)

print(X.shape)

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)

print(X_train)

print(X_test)

from sklearn.linear_model import LinearRegression

regressor = LinearRegression()

regressor.fit(X_train, y_train)

print(regressor.intercept_)

print(regressor.coef_)

score = regressor.predict([[9.5]])

print(score)

y_pred = regressor.predict(X_test)

print(y_pred)

from sklearn.metrics import mean_absolute_error, mean_squared_error

mae = mean_absolute_error(y_test, y_pred)

mse = mean_squared_error(y_test, y_pred)

rmse = nm.sqrt(mse)

print(mae)

print(mse)

print(rmse)

print('Actual',y_test)

print('Predicted',y_pred)
```

Q.2 Write a R program to reverse a number and also calculate the sum of digits of that number.

```
n = as.integer(readline(prompt = "Enter a number :"))
```

```
sum = 0
while (n > 0) {
  r = n %% 10
  sum = sum + r
  n = n %/% 10
}
print(paste("Sum of digit is :", sum))
```

Q.1 Write a python program the Categorical values in numeric format for a given dataset.

```
import numpy as np
import pandas as pd
dataset = pd.read_csv("play_tennis.csv")
dataset
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
dataset['outlook'] = le.fit_transform(dataset.outlook)
dataset['temp'] = le.fit_transform(dataset.temp)
dataset['humidity'] = le.fit_transform(dataset.humidity)
dataset['play'] = le.fit_transform(dataset.play)
x = dataset.iloc[:,:-1].values
y = dataset.iloc[:,5].values
from sklearn.preprocessing import StandardScaler
st_x = StandardScaler()
x1 = st_x.fit_transform(x)
print(x1)
```

Q.2 Write a R program to create a data frame using two given vectors and display the duplicate elements

```
a = c(10,20,10,10,40,50,20,30)

b = c(10,30,10,20,0,50,30,30)

print("Original data frame:")

ab = data.frame(a,b)

print(ab)

print("Duplicate elements of the said data frame:")

print(duplicated(ab))
print("Unique rows of the said data frame:")

print(unique(ab))
```

Q.1 Write a R program to calculate the multiplication table using a function.

```
num = as.integer(readline(prompt = "Enter a number: "))
for(i in 1:10)
{
print(paste(num,'x', i, '=', num*i))
}
```

Q.2 Consider following dataset

```
    weather=['Sunny','Sunny','Overcast','Rainy','Rainy','Rainy','Overcast','S
      unny','Sunny','Rainy','Sunny','Overcast','Overcast','Rainy']
      temp=['Hot','Hot','Hot','Mild','Cool','Cool','Cool','Mild','Cool','Mild','Mi
      ld','Mild','Hot','Mild']
     play=['No','No','Yes','Yes','Yes','No','Yes','No','Yes','Yes','Yes','Yes','Y
     es','No'].
```

QQ. **Use Naïve Bayes algorithm to predict[0:Overcast, 2:Mild] tuple belongs to which class whether to play the sports or not**.

```
    weather=['Sunny','Sunny','Overcast','Rainy','Rainy','Rainy','Overcast','S
      unny','Sunny','Rainy','Sunny','Overcast','Overcast','Rainy']
      temp=['Hot','Hot','Hot','Mild','Cool','Cool','Cool','Mild','Cool','Mild','Mi
      ld','Mild','Hot','Mild']
     play=['No','No','Yes','Yes','Yes','No','Yes','No','Yes','Yes','Yes','Yes','Y
     es','No'].
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
wheather_encoded = le.fit_transform(weather)
print(wheather_encoded)
temp_encoded = le.fit_transform(temp)
label = le.fit_transform(play)
print("Temp:",temp_encoded)
print("Play:",label)
features = list(zip(wheather_encoded,temp_encoded))
print(features)
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(features,label)
predicted = model.predict([[0,2]])
```

```
print("Predicted Value:",predicted)
```

Q.1 Write a python program to find all null values in a given data set and remove them.
(Download dataset from github.com)

Q.2 Consider the student data set It can be downloaded from:
https://drive.google.com/open?id=1oakZCv7g3mlmCSdv9J8kdSaqO5_6dIOw
Write a programme in python to apply simple linear regression and find out mean absolute error, mean squared error and root mean squared error.

```
import numpy as nm

import pandas as pd

data_set= pd.read_csv('student_scores.csv')

print(data_set)

y = data_set['Scores'].values.reshape(-1, 1)

X = data_set['Hours'].values.reshape(-1, 1)

print(x)
```

Q.1 Write a python program to splitting the dataset into training and testing set.

```
import numpy as np

import pandas as pd

dataset = pd.read_csv("play_tennis.csv")

dataset

from sklearn import preprocessing

le = preprocessing.LabelEncoder()

dataset['outlook'] = le.fit_transform(dataset.outlook)

dataset['temp'] = le.fit_transform(dataset.temp)

dataset['humidity'] = le.fit_transform(dataset.humidity)

dataset['wind'] = le.fit_transform(dataset.wind)

dataset['play'] = le.fit_transform(dataset.play)
```

```
x=dataset.iloc[:,:-1].values

print(x)

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

print(y)

from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)

print(x_train)

print(x_test)
```

Q.2 Write a script in R to create a list of employees and perform the following:

a. Display names of employees in the list.
b. Add an employee at the end of the list.
c. Remove the third element of the list.

Q.1 Write a R program to create a data frame using two given vectors and display the duplicate elements.

```
a = c(10,20,10,10,40,50,20,30)

b = c(10,30,10,20,0,50,30,30)

print("Original data frame:")

ab = data.frame(a,b)

print(ab)

print("Duplicate elements of the said data frame:")

print(duplicated(ab))

print("Unique rows of the said data frame:")

print(unique(ab))
```

Q.2 Write a Python program build Decision Tree Classifier using Scikit-learn package for diabetes data set (download database from https://www.kaggle.com/uciml/pima-indians-diabetes-database)

Q.1 Write a python program to splitting the dataset into training and testing set.

```
import numpy as np

import pandas as pd

dataset = pd.read_csv("play_tennis.csv")

dataset

from sklearn import preprocessing

le = preprocessing.LabelEncoder()

dataset['outlook'] = le.fit_transform(dataset.outlook)

dataset['temp'] = le.fit_transform(dataset.temp)

dataset['humidity'] = le.fit_transform(dataset.humidity)

dataset['wind'] = le.fit_transform(dataset.wind)

dataset['play'] = le.fit_transform(dataset.play)

x=dataset.iloc[:,:-1].values

print(x)

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

print(y)

from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)

print(x_train)

print(x_test)
```

Q.2 Consider following dataset
weather=['Sunny','Sunny','Overcast','Rainy','Rainy','Rainy','Overcast','S
unny','Sunny','Rainy','Sunny','Overcast','Overcast','Rainy']
temp=['Hot','Hot','Hot','Mild','Cool','Cool','Cool','Mild','Cool','Mild','Mi

ld','Mild','Hot','Mild']
play=['No','No','Yes','Yes','Yes','No','Yes','No','Yes','Yes','Yes','Yes','Yes','No']. Use Naïve Bayes algorithm to predict[0:Overcast, 2:Mild] tuple belongs to which class whether to play the sports or not.

Q.1 Write a R program to reverse a number and also calculate the sum of digits of that number.

```
n = as.integer(readline(prompt = "Enter a number :"))
sum = 0
while (n > 0) {
  r = n %% 10
  sum = sum + r
  n = n %/% 10
}

print(paste("Sum of digit is :", sum))
```

Q.2 Write a Python Programme to read the dataset ("Iris.csv"). dataset download from (https://archive.ics.uci.edu/ml/datasets/iris) and apply Apriori algorithm.

Q.1 Consider following observations/data. And apply simple linear regression and find out estimated coefficients b0 and b1.(use numpy package)
x= [0, 1, 2, 3, 4, 5, 6, 7, 8, 9,11,13]
y = ([1, 3, 2, 5, 7, 8, 8, 9, 10, 12,16, 18]

```
import numpy as np

import pandas as pd

dataset = pd.read_csv("play_tennis.csv")

dataset

from sklearn import preprocessing

le = preprocessing.LabelEncoder()
```

```
dataset['outlook'] = le.fit_transform(dataset.outlook)
dataset['temp'] = le.fit_transform(dataset.temp)
dataset['humidity'] = le.fit_transform(dataset.humidity)
dataset['wind'] = le.fit_transform(dataset.wind)
dataset['play'] = le.fit_transform(dataset.play)
x=dataset.iloc[:,:-1].values
print(x)
y=dataset.iloc[:,4].values
print(y)
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
print(x_train)
print(x_test)
```

Q.2 Write a R program to create a data frame using two given vectors and display the duplicate elements [15]

```
a = c(10,20,10,10,40,50,20,30)
b = c(10,30,10,20,0,50,30,30)
print("Original data frame:")
ab = data.frame(a,b)
print(ab)
print("Duplicate elements of the said data frame:")
print(duplicated(ab))
print("Unique rows of the said data frame:")
print(unique(ab))
```

Q.1 Write a R program to reverse a number and also calculate the sum of digits of that number.

```
n = as.integer(readline(prompt = "Enter a number :"))
sum = 0
while (n > 0) {
  r = n %% 10
  sum = sum + r
  n = n %/% 10
}
print(paste("Sum of digit is :", sum))
```

Q.2 Consider following observations/data. And apply simple linear regression and find out estimated coefficients b1 and b1 Also analyse the performance of the model (Use sklearn package)

x = np.array([1,2,3,4,5,6,7,8])
y = np.array([7,14,15,18,19,21,26,23])

```
 import numpy as np

from sklearn.linear_model import LinearRegression

x= np.array([1,2,3,4,5,6,7,8]).reshape((-1, 1))

print(x)

y = np.array([7,14,15,18,19,21,26,23])

print(y)

model = LinearRegression()

model.fit(x, y)

x_new = np.array(9).reshape((-1, 1))

y_new_pred = model.predict(x_new)

print(y_new_pred)

print('Slope:- ', model.coef_)
```

Q.1 Write a python program to implement multiple Linear Regression model for a car dataset. Dataset can be downloaded from: https://www.w3schools.com/python/python_ml_multiple_regression.asp

```
import pandas

from sklearn import linear_model

df = pandas.read_csv("data.csv")

print(df)

X = df[['Weight', 'Volume']]

print(X)

y = df['CO2']

print(y)
```

Q.2 Write a R program to calculate the sum of two matrices of given size.

```
m1 = matrix(c(1, 2, 3, 4, 5, 6), nrow = 2)

print("Matrix-1:")

print(m1)

m2 = matrix(c(0, 1, 2, 3, 0, 2), nrow = 2)

print("Matrix-2:")

print(m2)

result = m1 + m2

print("Result of addition")

print(result)
```

Q.1 Write a python programme to implement multiple linear regression model for stock market data frame as follows:

Stock_Market

```
import pandas as pd
```

```
from sklearn import linear_model
data = {'year':
[2017,2017,2017,2017,2017,2017,2017,2017,2017,2017,2017,2017,2016,2016,2016,2016,2016,
2016,2016,2016,2016,2016,2016,2016],
'month': [12,11,10,9,8,7,6,5,4,3,2,1,12,11,10,9,8,7,6,5,4,3,2,1],
'interest_rate':
[2.75,2.5,2.5,2.5,2.5,2.5,2.5,2.25,2.25,2.25,2,2,2,1.75,1.75,1.75,1.75,1.75,1.75,1.75,1.75,
1.75,1.75,1.75],
'unemployment_rate':
[5.3,5.3,5.3,5.3,5.4,5.6,5.5,5.5,5.5,5.6,5.7,5.9,6,5.9,5.8,6.1,6.2,6.1,6.1,6.1,5.9,6.2,6.2,6.1],
'index_price':
[1464,1394,1357,1293,1256,1254,1234,1195,1159,1167,1130,1075,1047,965,943,958,97
1,949,884,866,876,822,704,719]
}
df = pd.DataFrame(data)
print(df)
x = df[['interest_rate','unemployment_rate']]
print(x)
y = df['index_price']
print(y)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)
print(X_train)
print(X_test)
regr = linear_model.LinearRegression()
```

```
regr.fit(X_train, y_train)

print('Intercept: \n', regr.intercept_)

print('Coefficients: \n', regr.coef_)

y_pred=regr.predict(X_test)

print(y_pred)

from sklearn.metrics import r2_score

Accuracy=r2_score(y_test,y_pred)*100

print(Accuracy)

import matplotlib.pyplot as plt

plt.scatter(y_test,y_pred);

plt.xlabel('Actual');

plt.ylabel('Predicted');

import seaborn as sns

sns.regplot(x=y_test,y=y_pred,ci=None,color ='red');
```

Q.2 Write a R program to concatenate two given factors.

```
f1 <- factor(sample(LETTERS, size=6, replace=TRUE))

f2 <- factor(sample(LETTERS, size=6, replace=TRUE))

print("Original factors:")

print(f1)

print(f2)

f = factor(c(levels(f1)[f1], levels(f2)[f2]))

print("After concatenate factor becomes:")

print(f)
```

Q.1 Write a script in R to create a list of employees and perform the following:

a. Display names of employees in the list.
b. Add an employee at the end of the list.
c. Remove the third element of the list.

Q.2 Consider following observations/data. And apply simple linear regression and find
out estimated coefficients b1 and b1 Also analyse the performance of the model
(Use sklearn package)
x = np.array([1,2,3,4,5,6,7,8])
y = np.array([7,14,15,18,19,21,26,23])

```
 import numpy as np

from sklearn.linear_model import LinearRegression

x= np.array([1,2,3,4,5,6,7,8]).reshape((-1, 1))

print(x)

y = np.array([7,14,15,18,19,21,26,23])

print(y)

model = LinearRegression()

model.fit(x, y)

x_new = np.array(9).reshape((-1, 1))

y_new_pred = model.predict(x_new)

print(y_new_pred)

print('Slope:- ', model.coef_)
```

Q.1 Write a R program to add, multiply and divide two vectors of integer type. (vector
length should be minimum 4)

```
m1 = matrix(c(1, 2, 3, 4, 5, 6), nrow = 2)
```

```
print("Matrix-1:")

print(m1)

m2 = matrix(c(0, 1, 2, 3, 0, 2), nrow = 2)

print("Matrix-2:")

print(m2)

result = m1 + m2

print("Result of addition")

print(result)

result = m1 - m2

print("Result of subtraction")

print(result)

result = m1 * m2

print("Result of multiplication")

print(result)

result = m1 / m2

print("Result of division:")

print(result)
```

Q.2 Write a Python program build Decision Tree Classifier using Scikit-learn package for diabetes data set (download database from https://www.kaggle.com/uciml/pima-indians-diabetes-database)

Q.1 Write a python program to implement multiple Linear Regression model for a car

dataset. Dataset can be downloaded from: https://www.w3schools.com/python/python_ml_multiple_regression.asp

```
import pandas

from sklearn import linear_model
```

```
df = pandas.read_csv("data.csv")

print(df)

X = df[['Weight', 'Volume']]

print(X)

y = df['CO2']

print(y)
```

Q.2 Write a script in R to create a list of employees and perform the following:

a. Display names of employees in the list.
b. Add an employee at the end of the list.
c. Remove the third element of the list.

Q.1 Write a python program to implement k-means algorithms on a synthetic dataset.

```
import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.datasets import make_blobs

data = make_blobs(n_samples=300, n_features=2, centers=5,

cluster_std=1.8,random_state=101)

data[0].shape

data[1]

plt.scatter(data[0][:,0],data[0][:,1],c=data[1],cmap='brg')

from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=5)

kmeans.fit(data[0])

kmeans.cluster_centers_
kmeans.labels_f, (ax1, ax2) = plt.subplots(1, 2, sharey=True,figsize=(10,6))

ax1.set_title('K Means')

ax1.scatter(data[0][:,0],data[0][:,1],c=kmeans.labels_,cmap='brg')

ax2.set_title("Original")

ax2.scatter(data[0][:,0],data[0][:,1],c=data[1],cmap='brg'
```

Q.2 Write a R program to sort a list of strings in ascending and descending order.

Q.1 Write a R program to reverse a number and also calculate the sum of digits of that number.

```
n = as.integer(readline(prompt = "Enter a number :"))
sum = 0
while (n > 0) {
  r = n %% 10
  sum = sum + r
  n = n %/% 10
}
print(paste("Sum of digit is :", sum))
```

Q.2 Write a python program to implement hierarchical Agglomerative clustering algorithm. (Download Customer.csv dataset from github.com).

```
import numpy as nm

import matplotlib.pyplot as mtp

import pandas as pd

dataset = pd.read_csv('Wholesale customers data.csv')

dataset
```

```
x = dataset.iloc[:, [3, 4]].values
print(x)
import scipy.cluster.hierarchy as shc
dendro = shc.dendrogram(shc.linkage(x, method="ward"))
mtp.title("Dendrogrma Plot")
mtp.ylabel("Euclidean Distances")
mtp.xlabel("Customers")
mtp.show()
from sklearn.cluster import AgglomerativeClustering
hc= AgglomerativeClustering(n_clusters=5, affinity='euclidean', linkage='ward')
y_pred= hc.fit_predict(x)
mtp.scatter(x[y_pred == 0, 0], x[y_pred == 0, 1], s = 100, c = 'blue', label = 'Cluster 1')
mtp.scatter(x[y_pred == 1, 0], x[y_pred == 1, 1], s = 100, c = 'green', label = 'Cluster 2')
mtp.scatter(x[y_pred== 2, 0], x[y_pred == 2, 1], s = 100, c = 'red', label = 'Cluster 3')
mtp.scatter(x[y_pred == 3, 0], x[y_pred == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
mtp.scatter(x[y_pred == 4, 0], x[y_pred == 4, 1], s = 100, c = 'magenta', label = 'Cluster
5')
mtp.title('Clusters of customers')
mtp.xlabel('Milk')
mtp.ylabel('Grocery')
mtp.legend()
mtp.show()
```

Q.1 Write a python program to implement k-means algorithm to build prediction model
(Use Credit Card Dataset CC GENERAL.csv Download from kaggle.com)

```
import numpy as nm
import matplotlib.pyplot as mtp

import pandas as pd

dataset = pd.read_csv('creditcard.csv')

dataset

x = dataset.iloc[:, [3, 4]].values

print(x)

from sklearn.cluster import KMeans

wcss_list= []

for i in range(1, 11):

kmeans = KMeans(n_clusters=i, init='k-means++', random_state= 42)

kmeans.fit(x)

wcss_list.append(kmeans.inertia_)

mtp.plot(range(1, 11), wcss_list)

mtp.title('The Elobw Method Graph')

mtp.xlabel('Number of clusters(k)')

mtp.ylabel('wcss_list')

mtp.show()

kmeans = KMeans(n_clusters=3, init='k-means++', random_state= 42)

y_predict= kmeans.fit_predict(x)

mtp.scatter(x[y_predict == 0, 0], x[y_predict == 0, 1], s = 100, c = 'blue', label =

'Cluster 1') #for first cluster

mtp.scatter(x[y_predict == 1, 0], x[y_predict == 1, 1], s = 100, c = 'green', label =

'Cluster 2') #for second cluster

mtp.scatter(x[y_predict== 2, 0], x[y_predict == 2, 1], s = 100, c = 'red', label =

'Cluster 3') #for third cluster
```

```
mtp.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300,

c = 'yellow', label = 'Centroid')

mtp.title('Clusters of Credit Card')

mtp.xlabel('V3')

mtp.ylabel('V4')

mtp.legend()

mtp.show()
```

Q.2 Write a script in R to create a list of employees and perform the following:
a. Display names of employees in the list.
b. Add an employee at the end of the list.
c. Remove the third element of the list.

Q.1 Write a python program to implement hierarchical clustering algorithm. (Download
Wholesale customers data dataset from github.com).

```
import numpy as nm

import matplotlib.pyplot as mtp

import pandas as pd

dataset = pd.read_csv('Wholesale customers data.csv')

dataset

x = dataset.iloc[:, [3, 4]].values

print(x)

import scipy.cluster.hierarchy as shc

dendro = shc.dendrogram(shc.linkage(x, method="ward"))

mtp.title("Dendrogrma Plot")

mtp.ylabel("Euclidean Distances")

mtp.xlabel("Customers")
mtp.show()

from sklearn.cluster import AgglomerativeClustering

hc= AgglomerativeClustering(n_clusters=5, affinity='euclidean', linkage='ward')

y_pred= hc.fit_predict(x)

mtp.scatter(x[y_pred == 0, 0], x[y_pred == 0, 1], s = 100, c = 'blue', label = 'Cluster 1')

mtp.scatter(x[y_pred == 1, 0], x[y_pred == 1, 1], s = 100, c = 'green', label = 'Cluster 2')

mtp.scatter(x[y_pred== 2, 0], x[y_pred == 2, 1], s = 100, c = 'red', label = 'Cluster 3')

mtp.scatter(x[y_pred == 3, 0], x[y_pred == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')

mtp.scatter(x[y_pred == 4, 0], x[y_pred == 4, 1], s = 100, c = 'magenta', label = 'Cluster
5')

mtp.title('Clusters of customers')

mtp.xlabel('Milk')

mtp.ylabel('Grocery')

mtp.legend()

mtp.show()
```

Q.2 Write a R program to concatenate two given factors.

```
f1 <- factor(sample(LETTERS, size=6, replace=TRUE))

f2 <- factor(sample(LETTERS, size=6, replace=TRUE))

print("Original factors:")

print(f1)

print(f2)

f = factor(c(levels(f1)[f1], levels(f2)[f2]))

print("After concatenate factor becomes:")

print(f)
```