# BDA LAB EXTERNAL EXAM

1. Write a program to Implement Applications on python, NumPy and pandas

* import pandas as pd

dataset1 = pd.read\_csv("crime.csv")

dataset1

* dataset1.head()
* type(dataset1)
* dataset1.isnull().tail()
* dataset1.notnull().tail()
* dataset1.isnull().sum()
* dataset1[dataset1.Robbery.isnull()]
* dataset1['Robbery'].value\_counts()
* for col in dataset1.columns:

display(dataset1[col].value\_counts())

* dataset\_length=len(dataset1) dataset\_length
* dataset\_col=len(dataset1.columns)

dataset\_col

* dataset1.describe()
* dataset1.Murder.describe()
* dataset1.skew()
* dataset1.var()
* dataset1.kurtosis()
* print(dataset1.dtypes)
* import numpy as np

arr = np.array([1, 2, 3, 4, 5])

print(arr)

* import numpy as np

print(np.\_\_version\_\_)

* import numpyas np

arr = np.array([1, 2, 3, 4, 5])

print(arr)

print(type(arr)))

* import numpy as np

a = np.array(42)

b = np.array([1, 2, 3, 4, 5])

c = np.array([[1, 2, 3], [4, 5, 6]])

d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])

print(a.ndim)

print(b.ndim)

print(c.ndim)

* print(d.ndim)

1. Write a program to create an application that takes the Visualize Data Using Basic Plotting Techniques in Python.

* import pandas as pb

import matplotlib.pyplot as plt

import seaborn as sns

crime=pb.read\_csv('crime.csv')

crime

* plt.plot(crime.Murder,crime.Assault);
* sns.scatterplot(x=’Murder’,y=’Assault’,data=crime,s=100)
* plt.figure(figsize=(12,6))

plt.title(‘Murder Vs Assualt’)

sns.scatterploat(x=’Murder’,y=’Assault’,data=crime,s=100)

* plt.title.(‘Histogram for Robbery’)

plt.hist(crime.Robbery)

* plt.bar(crime.index,crime.Robbery)
* sns.barplot(x=’Robbery’,x=’year’,data=crime)
* import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

data=pd.read\_csv(‘crime.csv’)

x=data.popluation

y=data.carTheft

plt.scatter(x,y)

plt.xlable(‘popluation’)

plt.ylabel(‘carTheft’)

plt.title(‘popluation VS carTheft’)

plt.show();

1. Write a program to Implement NoSQL Database Operations: CRUD operations, Arrays using MongoDB

* mongod –version
* mongosh
* use newmrcetdb
* db.student.insert({name:"RAVI",reg\_num:101,address:{city:"HYDERABAD",pin:"500100",state:"Telangana"}})
* db.student.update({"reg\_num":"101"}, {$set:{"name":"BASHA"}})
* db.student.insert({name:"Rahamat",reg\_num:102,address:{city:"Vijayawada",pin:"500102",state:"Andhra Pradesh"}})
* db.student.find().sort({reg\_num:1}).pretty();

1. Write a program to create function operations for sort, limit, skip and aggregate

db.restaurants.insertMany( [ { "\_id" : 1, "name" : "Central Park Cafe", "borough" : "Manhattan"},

{ "\_id" : 2, "name" : "Rock A Feller Bar and Grill", "borough" : "Queens"},

{ "\_id" : 3, "name" : "Empire State Pub", "borough" : "Brooklyn"},

{ "\_id" : 4, "name" : "Stan's Pizzaria", "borough" : "Manhattan"},

{ "\_id" : 5, "name" : "Jane's Deli", "borough" : "Brooklyn"}, ] )

COUNT:- db.restaurants.aggregate( [ { $group : { \_id : '$ borough ', count : {$sum : 1} } } ] )

SORT:-db.restaurants.aggregate( [ { $sort : { borough : 1 } } ] )

LIMIT:- db.restaurants.aggregate( [ { $sort : { borough : 1, \_id: 1 } } ] )

SKIP:- db.restaurants.aggregate( [{ $skip : 5 }] )

1. Write short note on following MongoDB

i) count ii) sort iii) limit iv) skip v) aggregate

ans:-

i)Count:The count operation in MongoDB is used to retrieve the number of documents that match a specific query within a collection. It takes a query as a parameter and returns the count of documents that satisfy the query criteria. This operation is useful when you want to know how many documents meet a certain condition without retrieving the actual data.

Example:- db.collection("users").count({ age: { $gt: 25 } });

ii) Sort:The sort operation in MongoDB is used to sort the documents in a collection based on one or more fields. It takes one or more field-value pairs as parameters and arranges the documents in ascending or descending order of the specified fields. Sorting can be essential when you need to present data in a specific order for better readability or analysis.Example:- db.collection("products").find().sort({ price: 1 });

iii) Limit:The limit operation in MongoDB is used to restrict the number of documents returned by a query. It takes a single numeric value as a parameter and limits the result set to that number of documents. This is often used to implement pagination or to retrieve only a subset of results to improve query performance.Example:- db.collection("posts").find().limit(10);

iv) Skip:The skip operation in MongoDB is used to skip a specified number of documents and retrieve the rest of the documents from a query result. It is typically used in combination with the limit operation to implement paging and retrieve subsequent pages of data.Example:- db.collection("orders").find().skip(20).limit(10);

v) Aggregate:The aggregate operation in MongoDB is a powerful tool for performing complex data manipulations and transformations. It allows you to process data using a pipeline of stages, where each stage performs a specific operation like filtering, grouping, sorting, and more. The aggregation framework is especially useful for tasks that go beyond simple querying.Example:-

db.collection("sales").aggregate([

{ $match: { date: { $gte: new Date("2023-01-01") } } },

{ $group: { \_id: "$product", totalSales: { $sum: "$amount" } } },

{ $sort: { totalSales: -1 } },

{ $limit: 5 }

]);