BIG DATA ANALYTICS

A PRACTICAL REPORT
ON
BIG DATA ANALYTICS

SUBMITTED BY
Mr. RAHUL KEWAT
Roll No: 22006

UNDER THE GUIDANCE OF
PROF. AKBER KHAN

Submitted in fulfillment of the requirements for qualifying
MSc. IT Part I Semester - II Examination 2022-2023

University of Mumbai
Department of Information Technology

R.D. & S.H National College of Arts, Commerce &
S.W.A. Science College Bandra (West), Mumbai – 400 050

R. D. & S. H. National & S. W. A. Science College

Bandra (W), Mumbai – 400050.

Department of Information Technology
M.Sc. (IT – SEMESTER II)

Certificate

This is to certify that Big Data Analytics Practicals performed at R.D & S.H National & S.W.A. Science College by Mr. **Rahul Kewat** holding Seat No. _____ studying Master of Science in Information Technology Semester – II has been satisfactorily completed as prescribed by the University of Mumbai, during the year 2022– 2023.

Subject In-Charge **Coordinator In-Charge** **External Examiner**

College Stamp

INDEX

Practical 1

Aim: Implement Decision tree classification technique

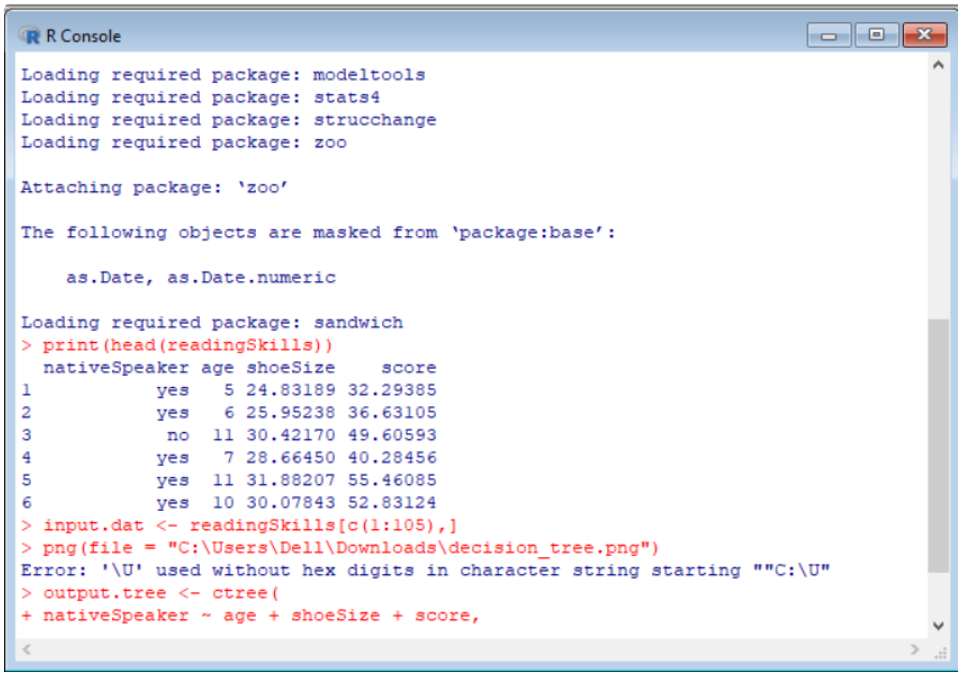
Writeup:

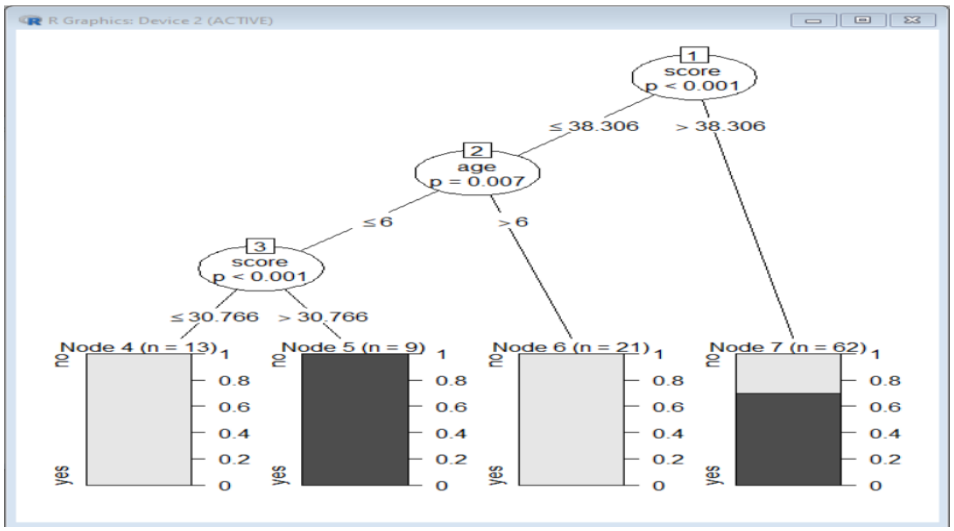
[A]: Implement Decision tree classification technique

Code:

```
library(party)
print(head(readingSkills))
input.dat <- readingSkills[c(1:105),]
png(file = "C:\Users\Dell\Downloads\decision_tree.png")
output.tree <- ctree(
nativeSpeaker ~ age + shoeSize + score,
data = input.dat)
plot(output.tree)
```

Output:





Practical 2

Aim: Implement SVM classification technique

Writeup:

[A]: Implement SVM classification technique
Code:

```
install.packages("caret")
library('caret')
heart <- read.csv("C:\\Users\\Dell\\Downloads\\heart.csv", sep = ',', header =
FALSE)
str(heart)
#split training and test dataset
intrain<- createDataPartition(y = heart$V14, p= 0.7, list = FALSE)
training <- heart[intrain,]
testing <- heart[-intrain,]
dim(training);
dim(testing);
anyNA(heart)
summary(heart)
training[["V14"]] <- factor(training[["V14"]])
trctrl<- trainControl(method = "repeatedcv", number = 10, repeats = 3)
svm_Linear<- train(V14 ~., data = training, method =
"svmLinear",trControl=trctrl,preProcess = c("center", "scale"),tuneLength = 10)
svm_Linear
test_pred<- predict(svm_Linear, newdata = training)
test_pred
```

Output:

```
> str(heart)
'data.frame':   290 obs. of  14 variables:
 $ V1 : chr  "age" "60" "35" "41" ...
 $ V2 : chr  "sex" "1" "1" "0" ...
 $ V3 : chr  "cp" "3" "2" "1" ...
 $ V4 : chr  "trtbps" "145" "130" "130" ...
 $ V5 : chr  "chol" "233" "250" "204" ...
 $ V6 : chr  "fbs" "1" "0" "0" ...
 $ V7 : chr  "restecg" "0" "1" "0" ...
 $ V8 : chr  "thalachh" "150" "187" "172" ...
 $ V9 : chr  "exng" "0" "0" "0" ...
 $ V10: chr  "oldpeak" "2.3" "3.5" "1.4" ...
 $ V11: chr  "slp" "0" "0" "2" ...
 $ V12: chr  "caa" "0" "0" "0" ...
 $ V13: chr  "thall" "1" "2" "2" ...
 $ V14: chr  "output" "1" "1" "1" ...
> #split training and test dataset
> intrain<- createDataPartition(y = heart$V14, p= 0.7, list = FALSE)
Warning message:
In createDataPartition(y = heart$V14, p = 0.7, list = FALSE) :
  Some classes have a single record ( output ) and these will be selected for the sample
> training <- heart[intrain,]
> testing <- heart[-intrain,]
> dim(training);
[1] 204  14
> dim(testing);
[1] 86 14
> anyNA(heart)
[1] FALSE
> summary(heart)
      V1                 V2                 V3                 V4
 Length:290         Length:290         Length:290         Length:290
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character
      V5                 V6                 V7                 V8
 Length:290         Length:290         Length:290         Length:290
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character
      V9                V10                V11                V12
 Length:290         Length:290         Length:290         Length:290
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character
     V13                V14
 Length:290         Length:290
```

```
> svm_Linear
Support Vector Machines with Linear Kernel

204 samples
 13 predictor
  3 classes: '0', '1', 'output'

Pre-processing: centered (345), scaled (345)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 184, 185, 182, 184, 183, 183, ...
Resampling results:

  Accuracy   Kappa
  0.7657044  0.517642

Tuning parameter 'C' was held constant at a value of 1
> test_pred<- predict(svm_Linear, newdata = training)
> test_pred
  [1] output 1      1      1      1      1      1      1      1      1
 [11] 1      1      1      1      1      1      1      1      1      1
 [21] 1      1      1      1      1      1      1      1      1      1
 [31] 1      1      1      1      1      1      1      1      1      1
 [41] 1      1      1      1      1      1      1      1      1      1
 [51] 1      1      1      1      1      1      1      1      1      1
 [61] 1      1      1      1      1      1      1      1      1      1
 [71] 1      1      1      1      1      1      1      1      1      1
 [81] 1      1      1      1      1      1      1      1      1      1
 [91] 1      1      1      1      1      1      1      1      1      1
[101] 1      1      1      1      1      1      1      1      1      1
[111] 1      1      1      1      1      1      1      0      0      0
[121] 0      0      0      0      0      0      0      0      0      0
[131] 0      0      0      0      0      0      0      0      0      0
[141] 0      0      0      0      0      0      0      0      0      0
[151] 0      0      0      0      0      0      0      0      0      0
[161] 0      0      0      0      0      0      0      0      0      0
[171] 0      0      0      0      0      0      0      0      0      0
[181] 0      0      0      0      0      0      0      0      0      0
[191] 0      0      0      0      0      0      0      0      0      0
[201] 0      0      0      0
Levels: 0 1 output
> 
```

Practical 3

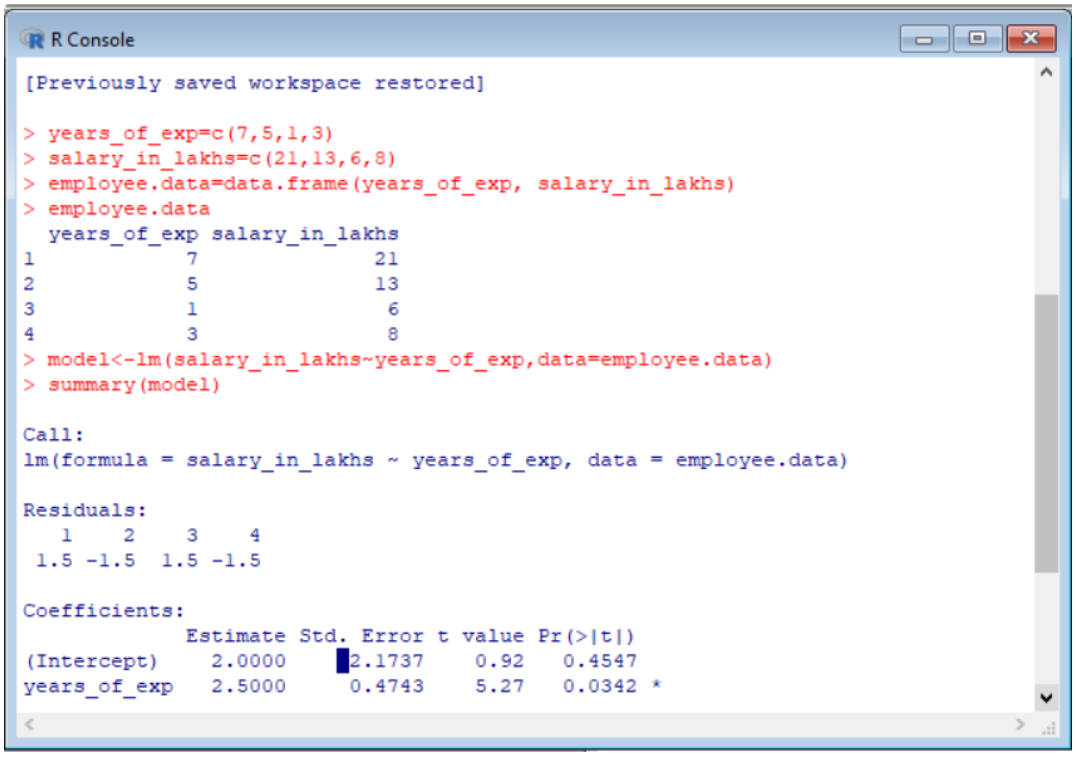
Aim: Implement Regression Model to import a data from web storage. Name the dataset and now do Linear Regression to find out relation between variables. Also check the model is fit or not.
Writeup:

[A]:Implement Regression Model to import a data from web storage. Name the dataset and now do Linear Regression to find out relation between variables. Also check the model is fit or not.

Code:

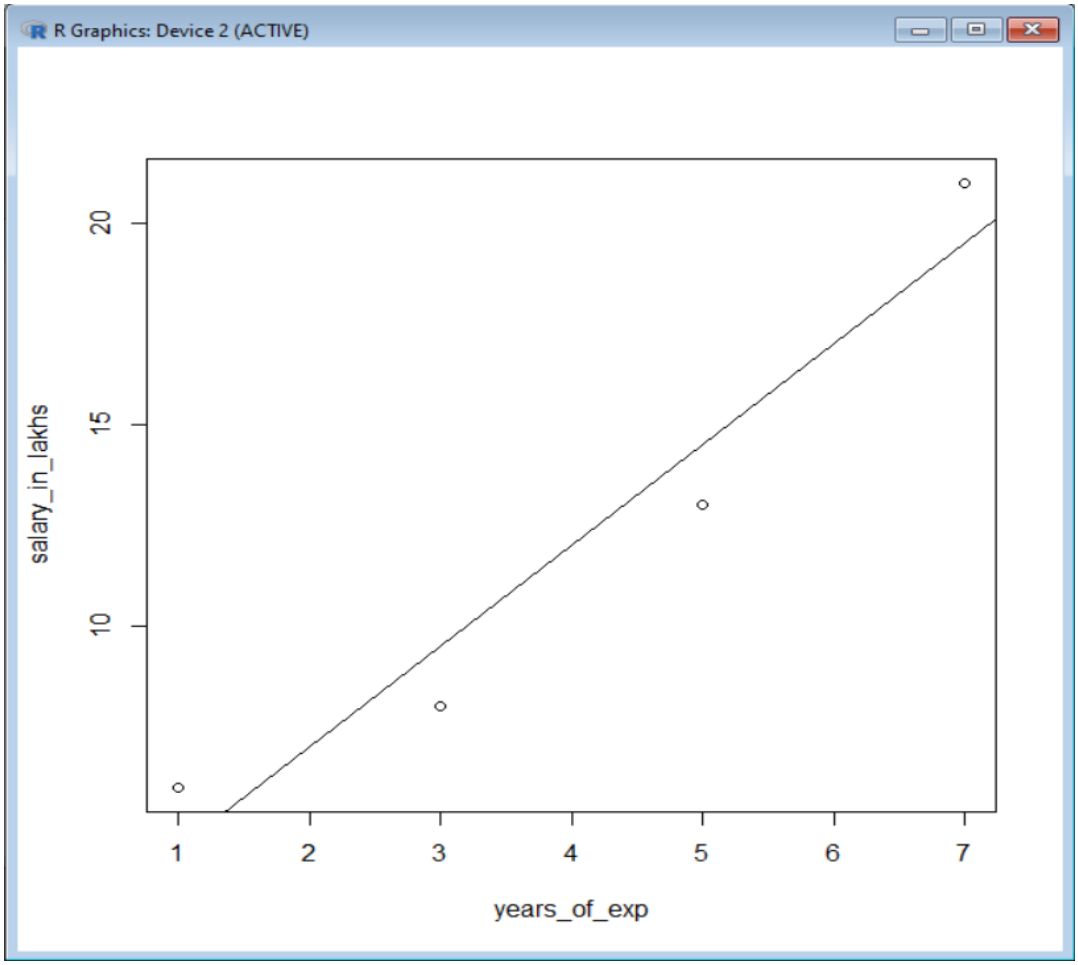
```
years_of_exp=c(7,5,1,3)
salary_in_lakhs=c(21,13,6,8)
employee.data=data.frame(years_of_exp, salary_in_lakhs)
employee.data
model<-lm(salary_in_lakhs~years_of_exp,data=employee.data)
summary(model)
plot(salary_in_lakhs~years_of_exp,data=employee.data)
abline(model)
```

Output:



R Console

```
4            3               8
> model<-lm(salary_in_lakhs~years_of_exp,data=employee.data)
> summary(model)

Call:
lm(formula = salary_in_lakhs ~ years_of_exp, data = employee.data)

Residuals:
   1    2    3    4
 1.5 -1.5  1.5 -1.5

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    2.0000     2.1737    0.92   0.4547
years_of_exp   2.5000     0.4743    5.27   0.0342 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.121 on 2 degrees of freedom
Multiple R-squared:  0.9328,    Adjusted R-squared:  0.8993
F-statistic: 27.78 on 1 and 2 DF,  p-value: 0.03417

> plot(salary_in_lakhs~years_of_exp,data=employee.data)
> abline(model)
>
```



Practical 4

Aim: Apply Multiple Regression on a dataset having a continuous independent variable.

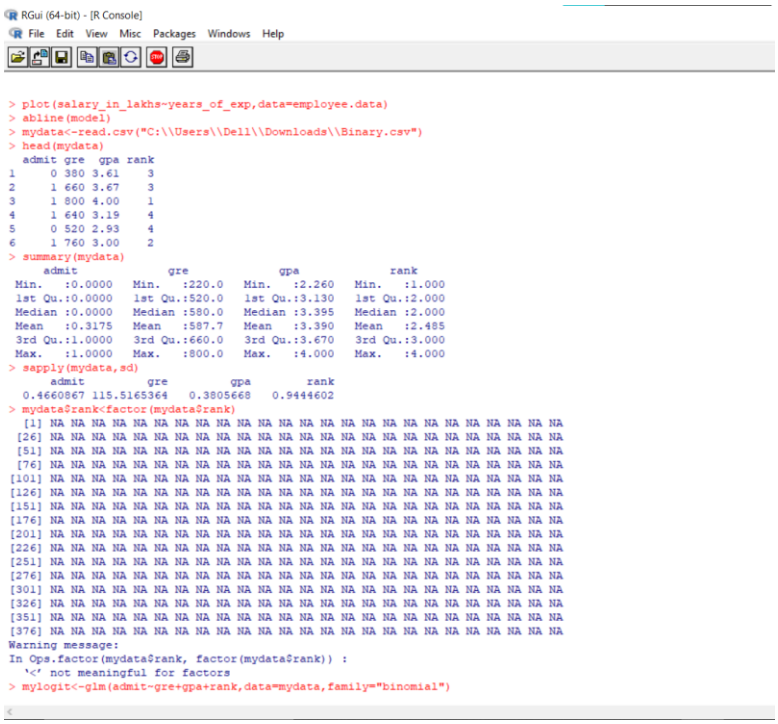
Writeup:

[A]: Apply Multiple Regression on a dataset having a continuous independent variable.

Code:

```
mydata<-read.csv("C:\\Users\\Dell\\Downloads\\Binary.csv")
head(mydata)
summary(mydata)
sapply(mydata,sd)
mydata$rank<factor(mydata$rank)
mylogit<-glm(admit~gre+gpa+rank,data=mydata,family="binomial")
summary(mylogit)
```

Output:



```
RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help

> plot(salary_in_lakhs~years_of_exp,data=employee.data)
> abline(model)
> mydata<-read.csv("C:\\Users\\Dell\\Downloads\\Binary.csv")
> head(mydata)
  admit gre  gpa rank
1     0 380 3.61    3
2     1 660 3.67    3
3     1 800 4.00    1
4     1 640 3.19    4
5     0 520 2.93    4
6     1 760 3.00    2
> summary(mydata)
     admit             gre             gpa             rank
 Min.   :0.0000   Min.   :220.0   Min.   :2.260   Min.   :1.000
 1st Qu.:0.0000   1st Qu.:520.0   1st Qu.:3.130   1st Qu.:2.000
 Median :0.0000   Median :580.0   Median :3.395   Median :2.000
 Mean   :0.3175   Mean   :587.7   Mean   :3.390   Mean   :2.485
 3rd Qu.:1.0000   3rd Qu.:660.0   3rd Qu.:3.670   3rd Qu.:3.000
 Max.   :1.0000   Max.   :800.0   Max.   :4.000   Max.   :4.000
> sapply(mydata,sd)
     admit        gre        gpa       rank
 0.4660867 115.5165364   0.3805668   0.9444602
> mydata$rank<factor(mydata$rank)
  [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
 [26] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
 [51] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
 [76] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[101] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[126] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[151] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[176] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[201] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[226] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[251] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[276] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[301] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[326] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[351] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[376] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
Warning message:
In Ops.factor(mydata$rank, factor(mydata$rank)) :
  '<' not meaningful for factors
> mylogit<-glm(admit~gre+gpa+rank,data=mydata,family="binomial")
```

```
Warning message:
In Ops.factor(mydata$rank, factor(mydata$rank)) :
  '<' not meaningful for factors
> mylogit<-glm(admit~gre+gpa+rank,data=mydata,family="binomial")
> summary(mylogit)

Call:
glm(formula = admit ~ gre + gpa + rank, family = "binomial",
    data = mydata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5802  -0.8848  -0.6382   1.1575   2.1732

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.449548   1.132846  -3.045  0.00233 **
gre          0.002294   0.001092   2.101  0.03564 *
gpa          0.777014   0.327484   2.373  0.01766 *
rank        -0.560031   0.127137  -4.405 1.06e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 499.98  on 399  degrees of freedom
Residual deviance: 459.44  on 396  degrees of freedom
AIC: 467.44

Number of Fisher Scoring iterations: 4

>
```

Practical 5

Aim: Build a Classification Model.

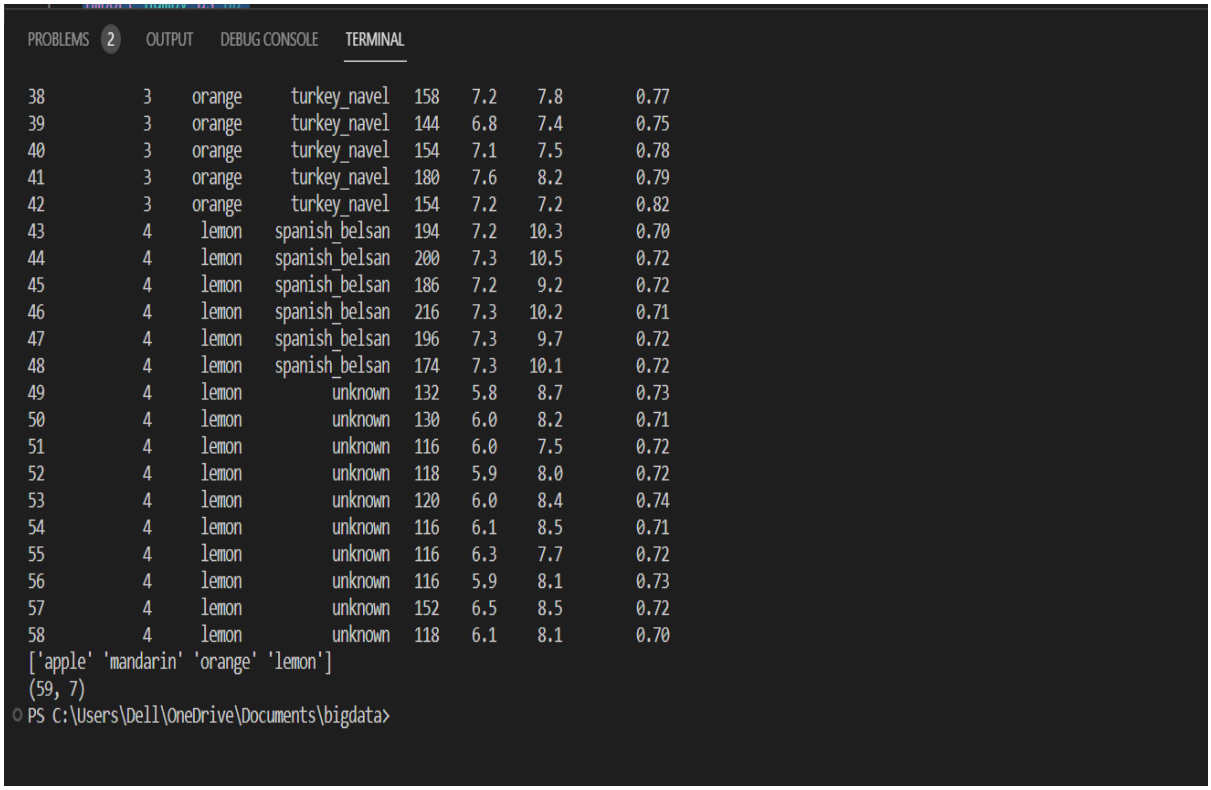
Writeup:

[A]: Build a Classification Model.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

fruits=pd.read_table('C:\\Users\\Dell\OneDrive\\Documents\\Rahul
Kewat\\IPCV\\images\\fruit_data_with_colors.txt')
fruits.head()
print(fruits)
print(fruits['fruit_name'].unique())
print(fruits.shape)
```

Output:

```
PROBLEMS 2  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\Dell\OneDrive\Documents\bigdata> & C:/Users/Dell/AppData/Local/Programs/Python/Python310/python.exe c:/Users/Dell/OneDrive/Documents/bi
gdata/classification.py
    fruit_label fruit_name     fruit_subtype  mass  width  height  color_score
0             1      apple      granny_smith   192    8.4     7.3         0.55
1             1      apple      granny_smith   180    8.0     6.8         0.59
2             1      apple      granny_smith   176    7.4     7.2         0.60
3             2   mandarin          mandarin    86    6.2     4.7         0.80
4             2   mandarin          mandarin    84    6.0     4.6         0.79
5             2   mandarin          mandarin    80    5.8     4.3         0.77
6             2   mandarin          mandarin    80    5.9     4.3         0.81
7             2   mandarin          mandarin    76    5.8     4.0         0.81
8             1      apple          braeburn   178    7.1     7.8         0.92
9             1      apple          braeburn   172    7.4     7.0         0.89
10            1      apple          braeburn   166    6.9     7.3         0.93
11            1      apple          braeburn   172    7.1     7.6         0.92
12            1      apple          braeburn   154    7.0     7.1         0.88
13            1      apple  golden_delicious   164    7.3     7.7         0.70
14            1      apple  golden_delicious   152    7.6     7.3         0.69
15            1      apple  golden_delicious   156    7.7     7.1         0.69
16            1      apple  golden_delicious   156    7.6     7.5         0.67
17            1      apple  golden_delicious   168    7.5     7.6         0.73
18            1      apple       cripps_pink   162    7.5     7.1         0.83
19            1      apple       cripps_pink   162    7.4     7.2         0.85
20            1      apple       cripps_pink   160    7.5     7.5         0.86
21            1      apple       cripps_pink   156    7.4     7.4         0.84
22            1      apple       cripps_pink   140    7.3     7.1         0.87
23            1      apple       cripps_pink   170    7.6     7.9         0.88
24            3     orange     spanish_jumbo   342    9.0     9.4         0.75
25            3     orange     spanish_jumbo   356    9.2     9.2         0.75
26            3     orange     spanish_jumbo   362    9.6     9.2         0.74
27            3     orange  selected_seconds   204    7.5     9.2         0.77
28            3     orange  selected_seconds   140    6.7     7.1         0.72
29            3     orange  selected_seconds   160    7.0     7.4         0.81
30            3     orange  selected_seconds   158    7.1     7.5         0.79
31            3     orange  selected_seconds   210    7.8     8.0         0.82
32            3     orange  selected_seconds   164    7.2     7.0         0.80
33            3     orange      turkey_navel   190    7.5     8.1         0.74
34            3     orange      turkey_navel   142    7.6     7.8         0.75
35            3     orange      turkey_navel   150    7.1     7.9         0.75
36            3     orange      turkey_navel   160    7.1     7.6         0.76
37            3     orange      turkey_navel   154    7.3     7.3         0.79
```



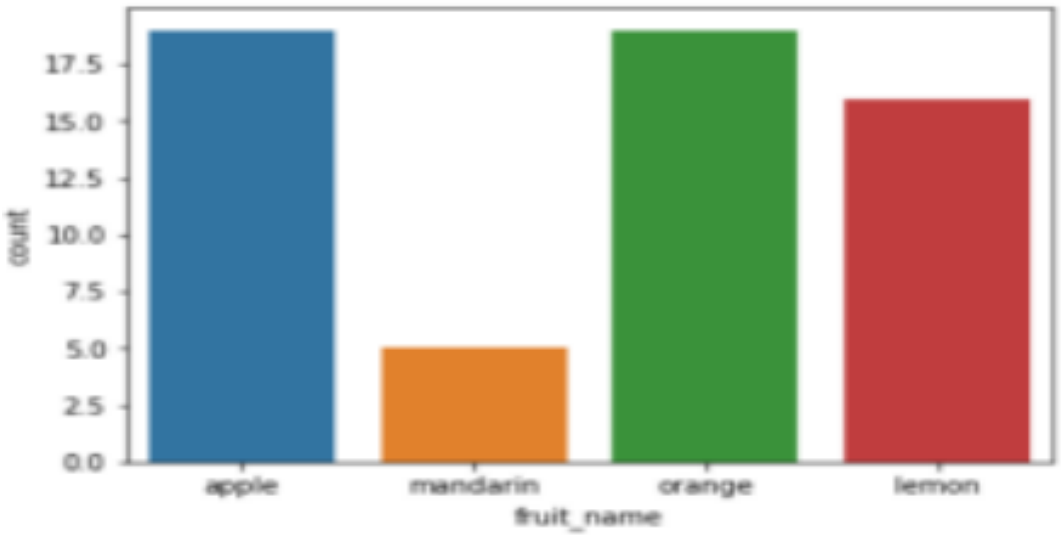
```
PROBLEMS 2   OUTPUT   DEBUG CONSOLE   TERMINAL

38          3    orange      turkey_navel   158   7.2    7.8        0.77
39          3    orange      turkey_navel   144   6.8    7.4        0.75
40          3    orange      turkey_navel   154   7.1    7.5        0.78
41          3    orange      turkey_navel   180   7.6    8.2        0.79
42          3    orange      turkey_navel   154   7.2    7.2        0.82
43          4     lemon    spanish_belsan   194   7.2   10.3        0.70
44          4     lemon    spanish_belsan   200   7.3   10.5        0.72
45          4     lemon    spanish_belsan   186   7.2    9.2        0.72
46          4     lemon    spanish_belsan   216   7.3   10.2        0.71
47          4     lemon    spanish_belsan   196   7.3    9.7        0.72
48          4     lemon    spanish_belsan   174   7.3   10.1        0.72
49          4     lemon           unknown   132   5.8    8.7        0.73
50          4     lemon           unknown   130   6.0    8.2        0.71
51          4     lemon           unknown   116   6.0    7.5        0.72
52          4     lemon           unknown   118   5.9    8.0        0.72
53          4     lemon           unknown   120   6.0    8.4        0.74
54          4     lemon           unknown   116   6.1    8.5        0.71
55          4     lemon           unknown   116   6.3    7.7        0.72
56          4     lemon           unknown   116   5.9    8.1        0.73
57          4     lemon           unknown   152   6.5    8.5        0.72
58          4     lemon           unknown   118   6.1    8.1        0.70
['apple' 'mandarin' 'orange' 'lemon']
(59, 7)
PS C:\Users\Dell\OneDrive\Documents\bigdata>
```

[B]: Fruit Type Distribution

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
fruits = pd.read_table("C:\\Users\\Dell\\OneDrive\\Documents\\Rahul
Kewat\\IPCV\\images\\fruit_data_with_colors.txt")
a=fruits.groupby("fruit_name").size()
print(a)

a["fruit_name"]=a.index
sns.countplot(x="fruit_name",data=fruits)
plt.show
```

Output:

```
PROBLEMS 2  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\Dell\OneDrive\Documents\bigdata> & C:/Users/Dell/AppData/Local/Programs/Python/Python310/python.exe c:/Users/Dell/OneDrive/Documents/bi
gdata/distribustion.py
fruit_name
apple       19
lemon       16
mandarin     5
orange      19
dtype: int64
PS C:\Users\Dell\OneDrive\Documents\bigdata>
```



Practical 6

Aim: Build a Clustering Model

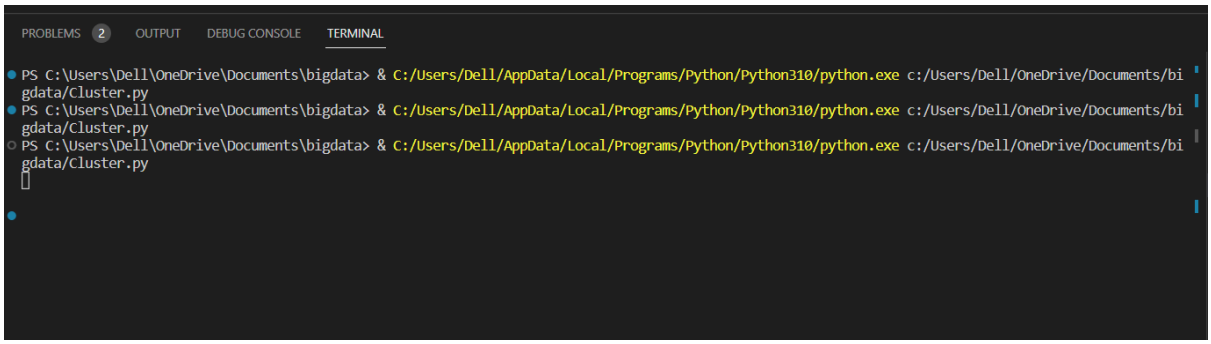
Writeup:

[A]: Build a Clustering Model

Code:

```
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import KMeans
# synthetic classification dataset
from numpy import where
from sklearn.datasets import make_classification
from matplotlib import pyplot
# define dataset
X, y = make_classification(n_samples=1000, n_features=2, n_informative=2,
n_redundant=0, n_clusters_per_class=1, random_state=4)
# create scatter plot for samples from each class
for class_value in range(2):
 # get row indexes for samples with this class
 row_ix = where(y == class_value)
 # create scatter of these samples
 pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
# show the plot
pyplot.show()
```

Output:



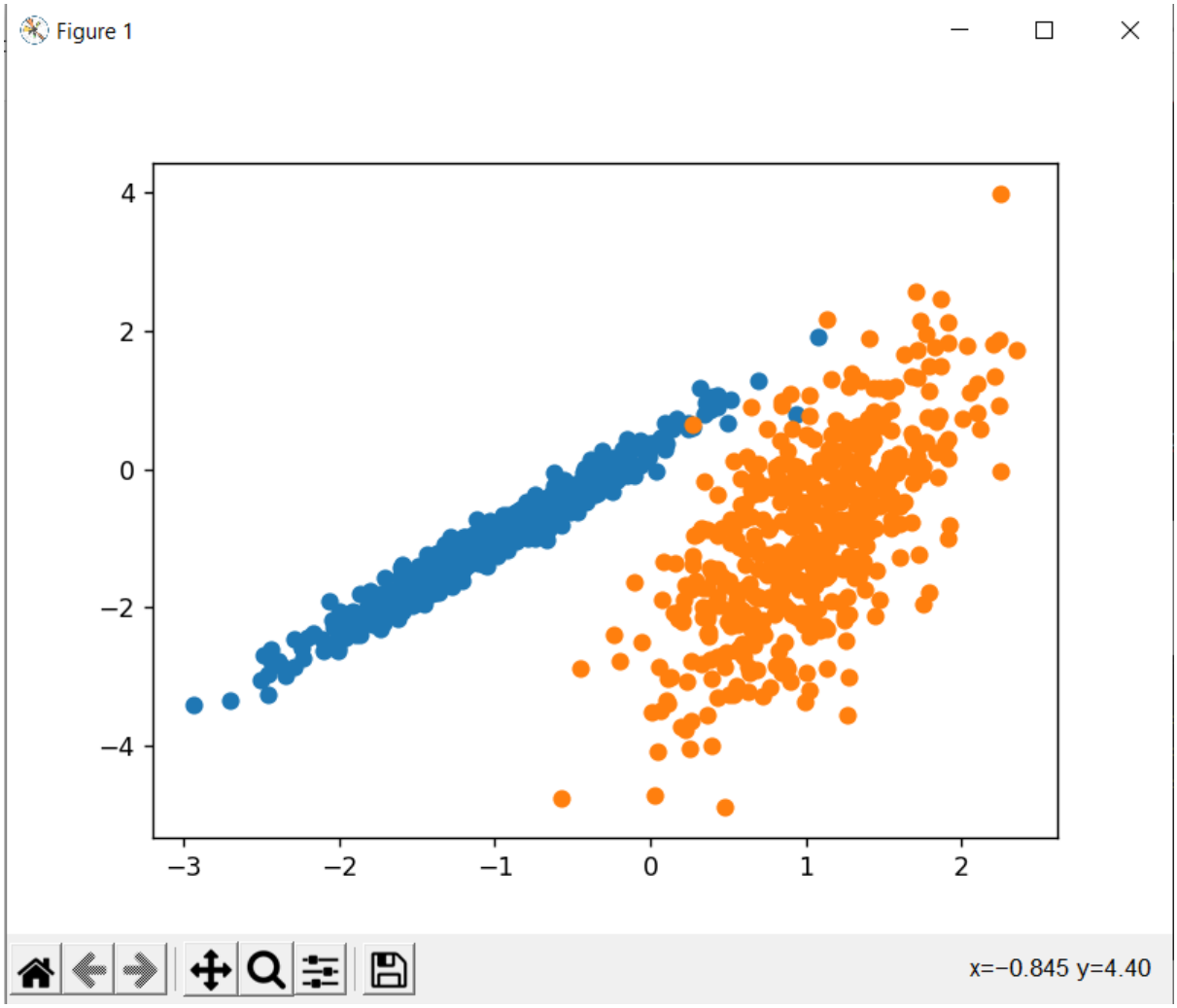


Figure 1
4
2
0
−2
−4
−3
−2
−1
0
1
2
x=−0.845 y=4.40

Practical 7

Aim: Install, configure and run Hadoop and HDFS and explore HDFS

Writeup:

Pre-requisites:

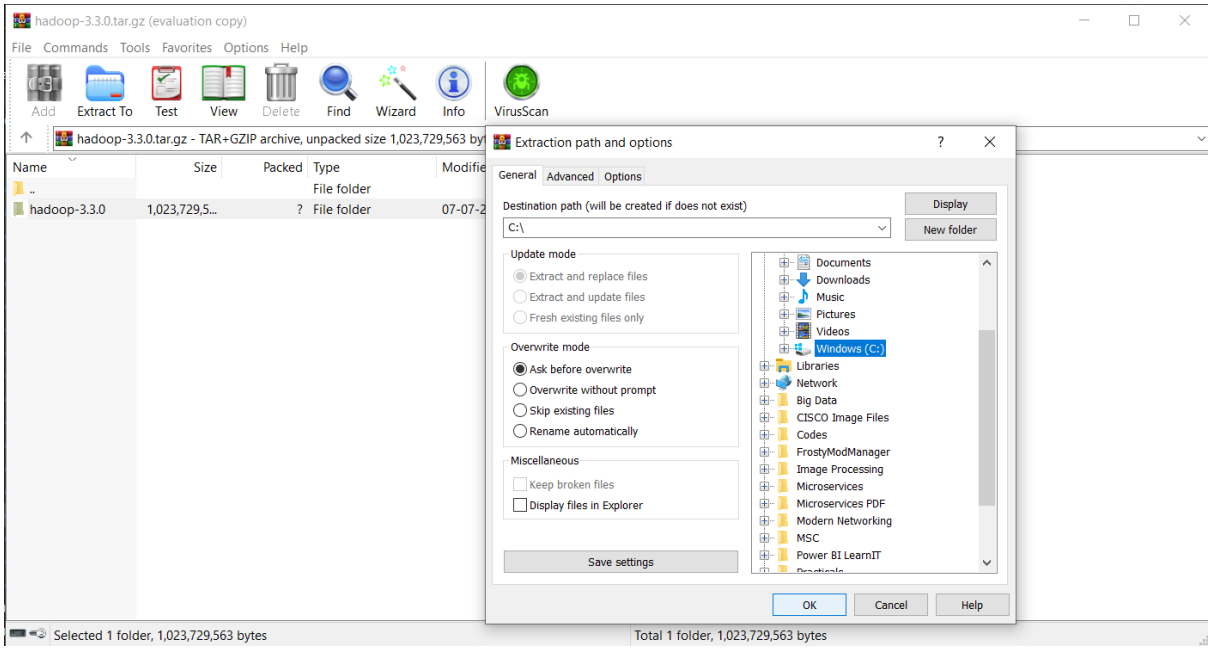
- **Java JDK 8.0**
- **Apache Hadoop 3.3.4 from (https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.3.4/hadoop-3.3.4-src.tar.gz)**

Step 1: Check Java version with command **javac -version.**
Open command prompt and type the above command.

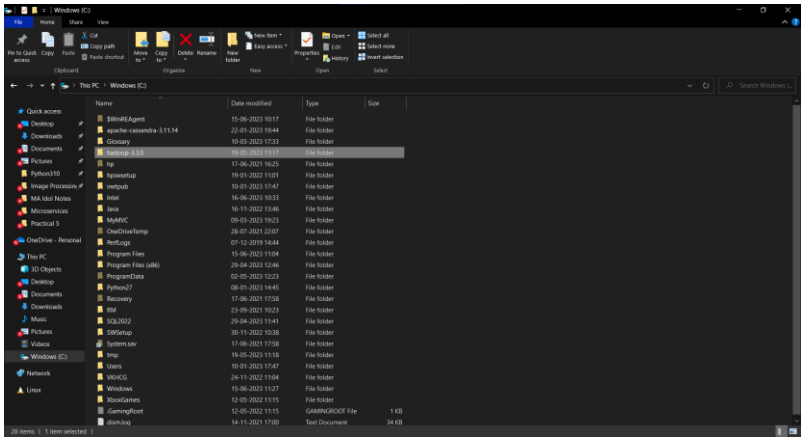
Output:

```
Microsoft Windows [Version 10.0.19045.3086]
(c) Microsoft Corporation. All rights reserved.

C:\Users\raman>javac -version
javac 1.8.0_352

C:\Users\raman>
```

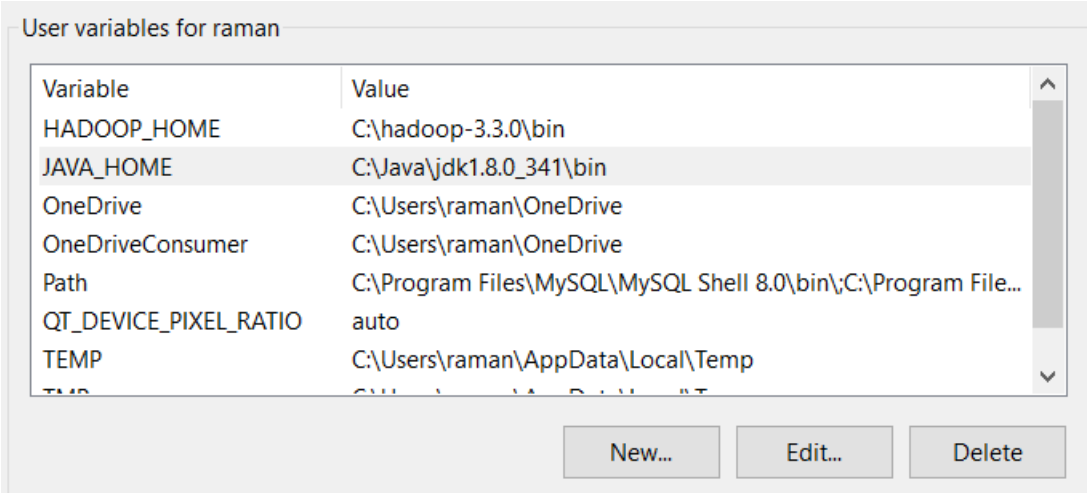
Step 2: Extract the Hadoop files from the compressed folder to the C- Drive Directory



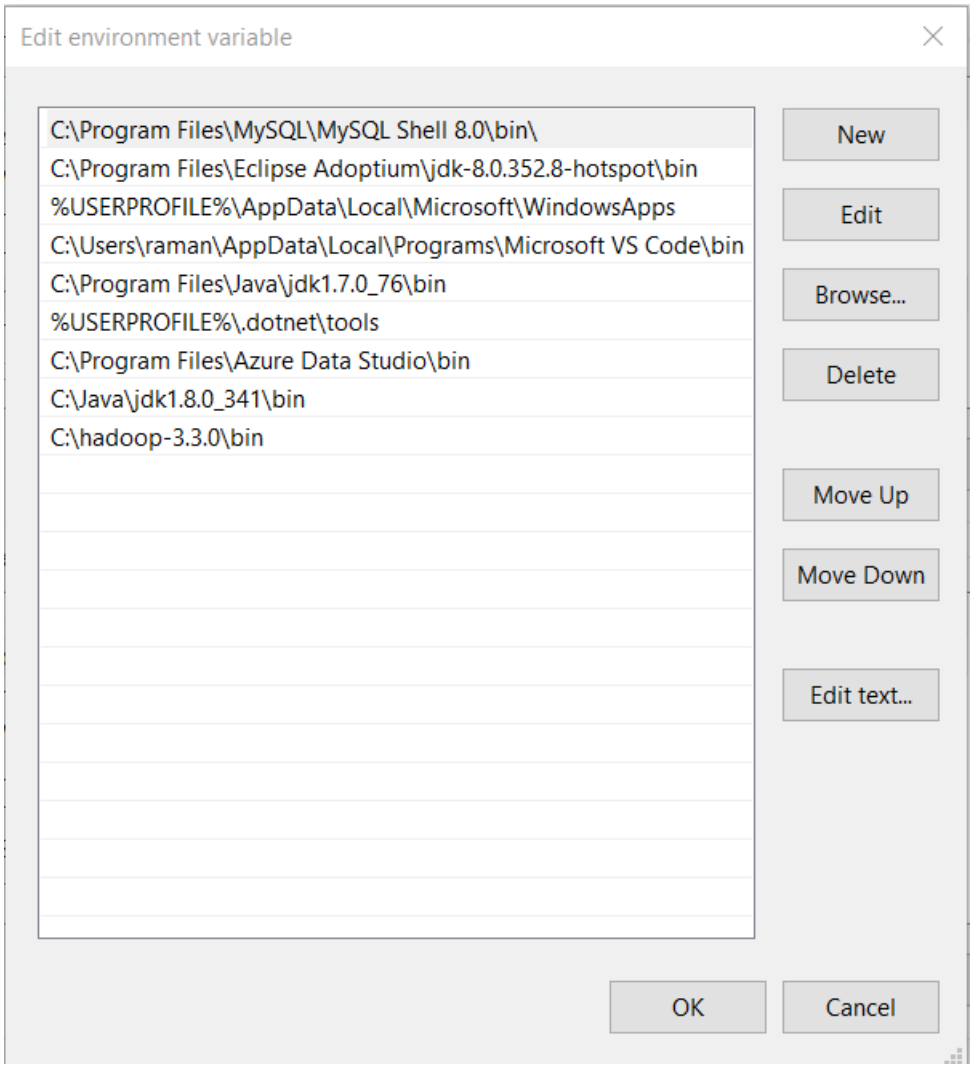
Ensure that both the Hadoop Folder as well as the JAVA folder are in the Main Directory of the C-Drive.



Step 3: Configure the Environment Variables for **JAVA_HOME** and **HADOOP_HOME**.



Also check the Path attribute within the environment variables and create the necessary locations for Hadoop and Java.



Step 4: Configuring Hadoop Files.

- Edit file C:/Hadoop-3.3.0/etc/hadoop/core-site.xml

Code:

```
<configuration>
        <property>
                <name>fs.defaultFS</name>
                <value>hdfs://localhost:9000</value>
        </property>
</configuration>
```

Paste the above code within the configuration file.

- Rename “mapred-site.xml.template” to “mapred-site.xml” and edit this file C:/Hadoop3.3.0/etc/hadoop/mapred-site.xml

Code:

```
<configuration>
        <property>
                <name>mapreduce.framework.name</name>
                <value>yarn</value>
        </property>
</configuration>
```

Paste the above code within the configuration file.

- **Creating Folders:-**
  - Create folder “data” under “C:\Hadoop-3.3.0”
  - Create folder “datanode” under “C:\Hadoop-3.3.0\data”
  - Create folder “namenode” under “C:\Hadoop-3.3.0\data”

- Edit file C:\Hadoop-3.3.0/etc/hadoop/hdfs-site.xml

Code:

```
<configuration>
        <property>
                <name>dfs.replication</name>
                <value>1</value>
        </property>
        <property>
                <name>dfs.namenode.name.dir</name>
                <value>/hadoop-3.3.0/data/namenode</value>
        </property>
        <property>
                 <name>dfs.datanode.data.dir</name>
                 <value>/hadoop-3.3.0/data/datanode</value>
        </property>
</configuration>
```

Paste the above code in the configuration file.

- Edit file C:/Hadoop-3.3.0/etc/hadoop/yarn-site.xml

Code:

```
<configuration>
        <property>
                <name>yarn.nodemanager.aux-services</name>
                <value>mapreduce_shuffle</value>
        </property>
        <property>
                <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
                <value>org.apache.hadoop.mapred.ShuffleHandler</value>
        </property>
```

</configuration>

Paste the above code in the configuration file.

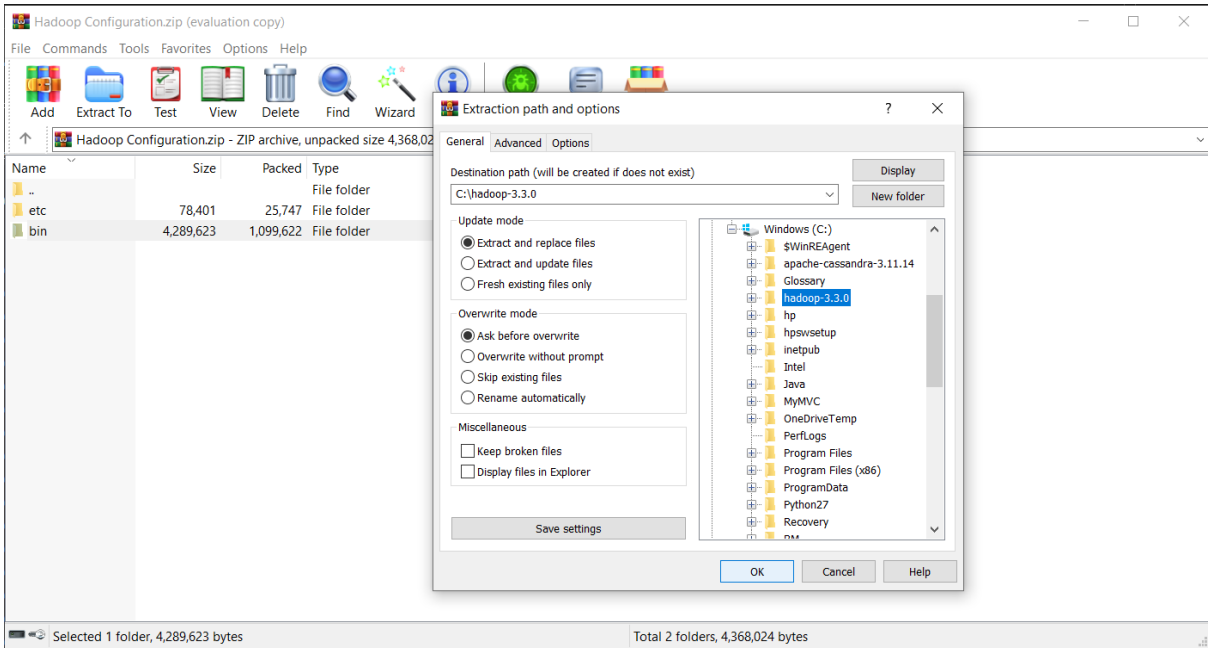
- Edit file C:/Hadoop-3.3.0/etc/hadoop/hadoop-env.cmd

Search for the line:- "**JAVA_HOME=%JAVA_HOME%**" Replace the above line with **set JAVA_HOME = C:\Java\jdk version you have downloaded\"**
It should look like this:

```
@rem The java implementation to use.  Required.
set JAVA_HOME=C:\Java\jdk1.8.0_341\
```

Step 5: Hadoop Configurations

- From the Downloaded Hadoop configuration file extract the **bin** folder and replace it with the **bin** folder in the Hadoop Main Directory.



Step 6: Starting Hadoop

- In the Hadoop File Directory, Open a cmd and type in the command **hdfs namenode -format**This is done to test if the instance is working.

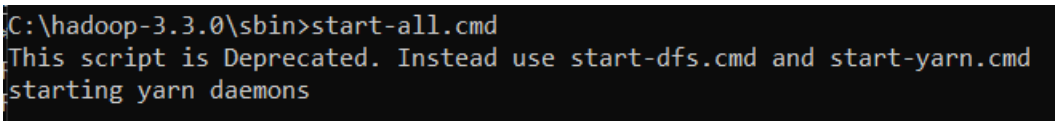
Output:

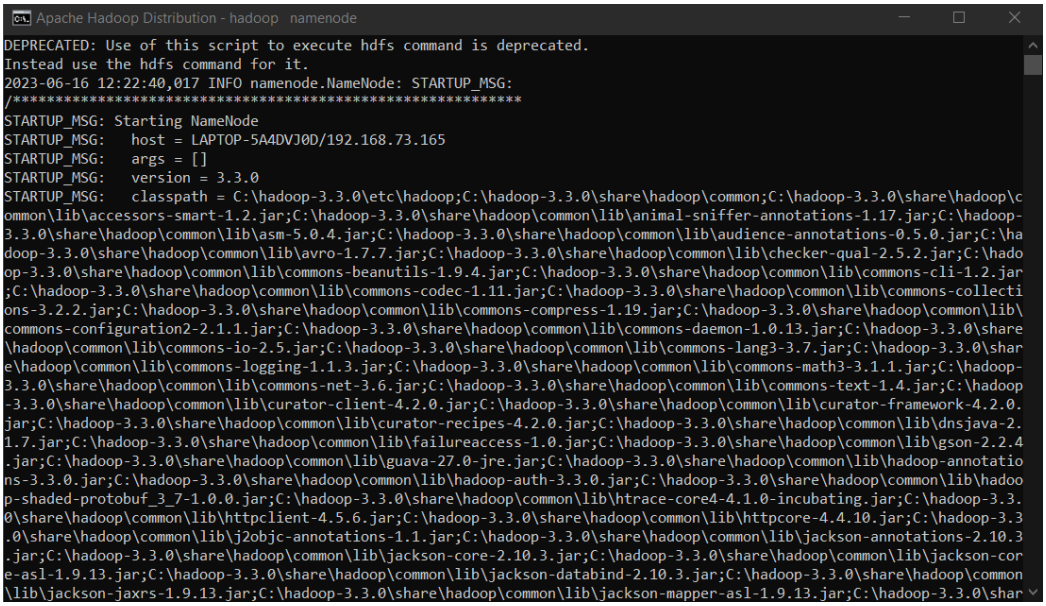
Step 7: Testing Hadoop

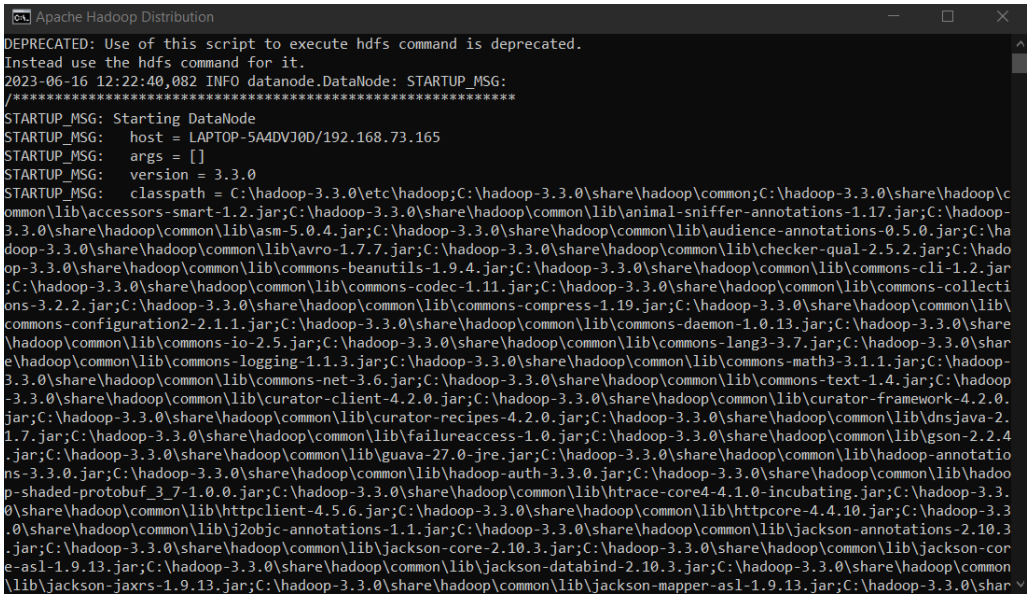
Now within the same cmd created in the previous step change the directory to the **sbin** file within Hadoop.
Type the command: **start-all.cmd**
After execution of the command there should be four instances created:

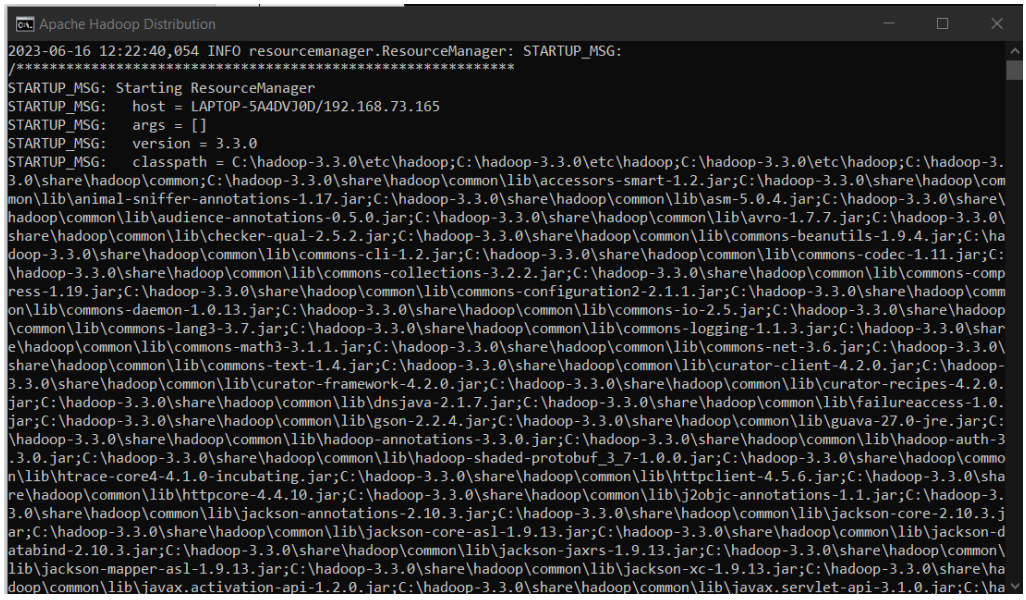
- **Hadoop Namenode**
- **Hadoop datanode**
- **YARN Resource Manager**
- **YARN Node Manager**

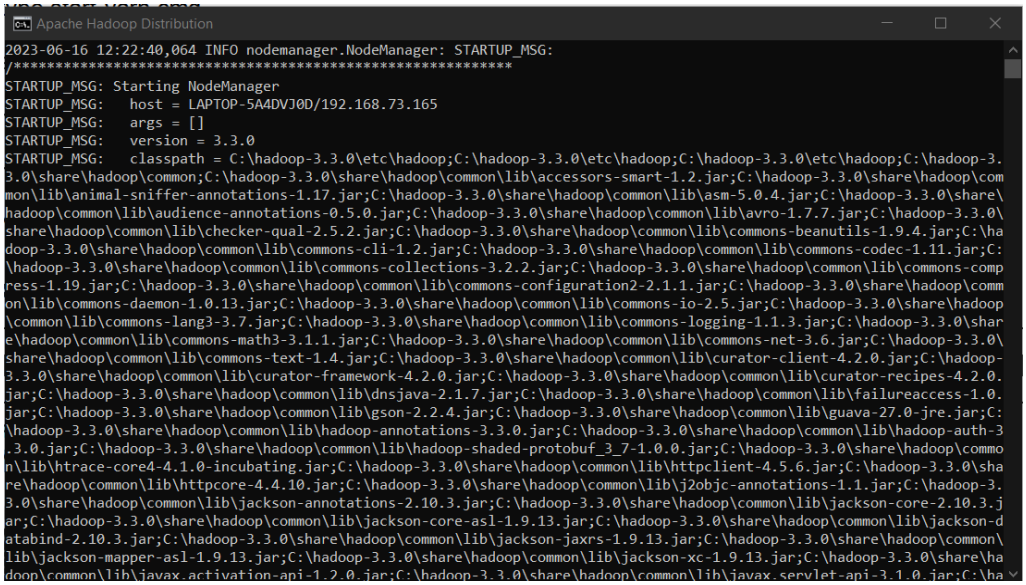
Output:






Apache Hadoop Distribution
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
2023-06-16 12:22:40,082 INFO datanode.DataNode: STARTUP_MSG:
/**
STARTUP_MSG: Starting DataNode
STARTUP_MSG: host = LAPTOP-5A4DVJ0D/192.168.73.165
STARTUP_MSG: args = []
STARTUP_MSG: version = 3.3.0
STARTUP_MSG: classpath = C:\hadoop-3.3.0\etc\hadoop;C:\hadoop-3.3.0\share\hadoop\common;C:\hadoop-3.3.0\share\hadoop\common\lib\accessors-smart-1.2.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\animal-sniffer-annotations-1.17.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\asm-5.0.4.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\audience-annotations-0.5.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\avro-1.7.7.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\checker-qual-2.5.2.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-beanutils-1.9.4.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-cli-1.2.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-codec-1.11.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-collections-3.2.2.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-compress-1.19.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-configuration2-2.1.1.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-daemon-1.0.13.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-io-2.5.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-lang3-3.7.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-logging-1.1.3.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-math3-3.1.1.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-net-3.6.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-text-1.4.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\curator-client-4.2.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\curator-framework-4.2.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\curator-recipes-4.2.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\dnsjava-2.1.7.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\failureaccess-1.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\gson-2.2.4.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\guava-27.0-jre.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\hadoop-annotations-3.3.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\hadoop-auth-3.3.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\hadoop-shaded-protobuf_3_7-1.0.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\htrace-core4-4.1.0-incubating.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\httpclient-4.5.6.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\httpcore-4.4.10.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\j2objc-annotations-1.1.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-annotations-2.10.3.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-core-2.10.3.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-core-asl-1.9.13.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-databind-2.10.3.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-jaxrs-1.9.13.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-mapper-asl-1.9.13.jar;C:\hadoop-3.3.0\shar


Apache Hadoop Distribution
2023-06-16 12:22:40,054 INFO resourcemanager.ResourceManager: STARTUP_MSG:
/**
STARTUP_MSG: Starting ResourceManager
STARTUP_MSG: host = LAPTOP-5A4DVJ0D/192.168.73.165
STARTUP_MSG: args = []
STARTUP_MSG: version = 3.3.0
STARTUP_MSG: classpath = C:\hadoop-3.3.0\etc\hadoop;C:\hadoop-3.3.0\etc\hadoop;C:\hadoop-3.3.0\etc\hadoop;C:\hadoop-3.3.0\share\hadoop\common;C:\hadoop-3.3.0\share\hadoop\common\lib\accessors-smart-1.2.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\animal-sniffer-annotations-1.17.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\asm-5.0.4.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\audience-annotations-0.5.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\avro-1.7.7.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\checker-qual-2.5.2.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-beanutils-1.9.4.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-cli-1.2.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-codec-1.11.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-collections-3.2.2.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-compress-1.19.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-configuration2-2.1.1.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-daemon-1.0.13.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-io-2.5.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-lang3-3.7.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-logging-1.1.3.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-math3-3.1.1.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-net-3.6.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-text-1.4.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\curator-client-4.2.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\curator-framework-4.2.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\curator-recipes-4.2.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\dnsjava-2.1.7.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\failureaccess-1.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\gson-2.2.4.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\guava-27.0-jre.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\hadoop-annotations-3.3.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\hadoop-auth-3.3.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\hadoop-shaded-protobuf_3_7-1.0.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\htrace-core4-4.1.0-incubating.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\httpclient-4.5.6.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\httpcore-4.4.10.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\j2objc-annotations-1.1.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-annotations-2.10.3.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-core-2.10.3.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-core-asl-1.9.13.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-databind-2.10.3.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-jaxrs-1.9.13.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-mapper-asl-1.9.13.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-xc-1.9.13.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\javax.activation-api-1.2.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\javax.servlet-api-3.1.0.jar;C:\ha


Apache Hadoop Distribution
2023-06-16 12:22:40,064 INFO nodemanager.NodeManager: STARTUP_MSG:
/**
STARTUP_MSG: Starting NodeManager
STARTUP_MSG: host = LAPTOP-5A4DVJ0D/192.168.73.165
STARTUP_MSG: args = []
STARTUP_MSG: version = 3.3.0
STARTUP_MSG: classpath = C:\hadoop-3.3.0\etc\hadoop;C:\hadoop-3.3.0\etc\hadoop;C:\hadoop-3.3.0\etc\hadoop;C:\hadoop-3.3.0\share\hadoop\common;C:\hadoop-3.3.0\share\hadoop\common\lib\accessors-smart-1.2.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\animal-sniffer-annotations-1.17.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\asm-5.0.4.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\audience-annotations-0.5.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\avro-1.7.7.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\checker-qual-2.5.2.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-beanutils-1.9.4.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-cli-1.2.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-codec-1.11.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-collections-3.2.2.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-compress-1.19.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-configuration2-2.1.1.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-daemon-1.0.13.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-io-2.5.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-lang3-3.7.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-logging-1.1.3.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-math3-3.1.1.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-net-3.6.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\commons-text-1.4.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\curator-client-4.2.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\curator-framework-4.2.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\curator-recipes-4.2.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\dnsjava-2.1.7.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\failureaccess-1.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\gson-2.2.4.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\guava-27.0-jre.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\hadoop-annotations-3.3.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\hadoop-auth-3.3.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\hadoop-shaded-protobuf_3_7-1.0.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\htrace-core4-4.1.0-incubating.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\httpclient-4.5.6.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\httpcore-4.4.10.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\j2objc-annotations-1.1.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-annotations-2.10.3.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-core-2.10.3.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-core-asl-1.9.13.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-databind-2.10.3.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-jaxrs-1.9.13.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-mapper-asl-1.9.13.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\jackson-xc-1.9.13.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\javax.activation-api-1.2.0.jar;C:\hadoop-3.3.0\share\hadoop\common\lib\javax.servlet-api-3.1.0.jar;C:\ha

Practical 8

Aim: Implement an application that stores big data in MongoDB and manipulate it using Python.

Writeup:

- **Insert data :**

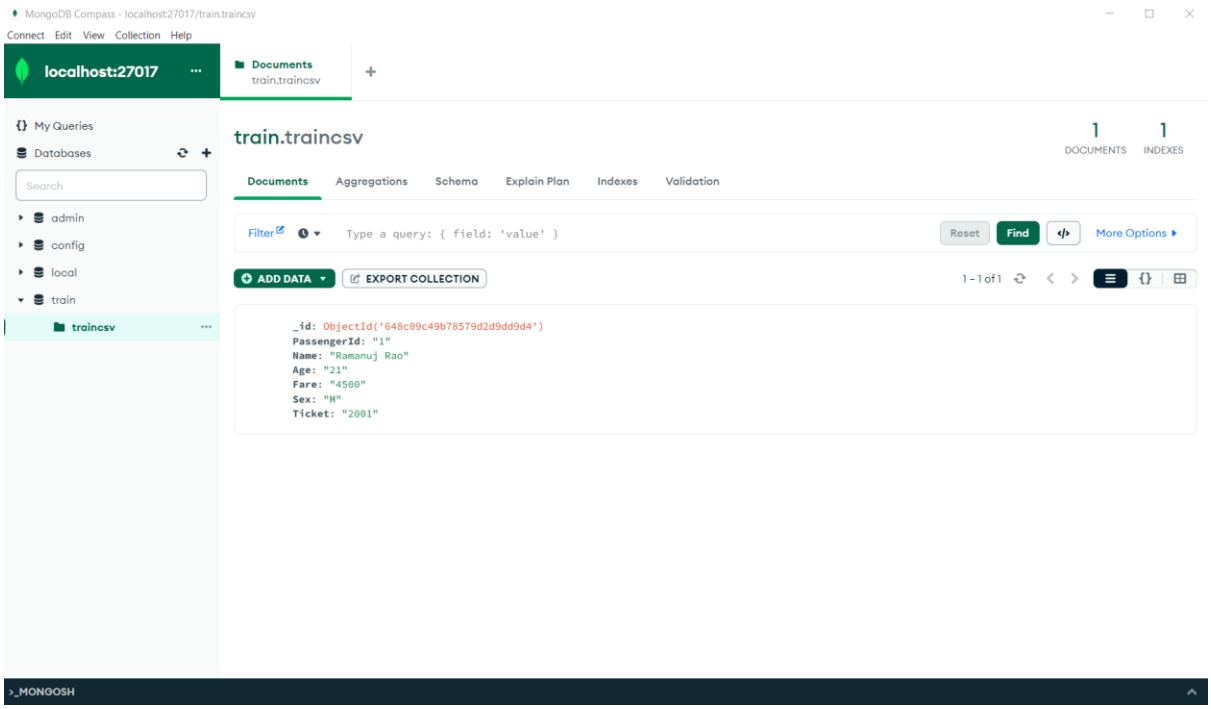
Code:

```
from pymongo import MongoClient
client= MongoClient('localhost:27017')
db = client.train

def insert():
    try:
        Id =input(' Enter traincsv Passenger Id: ')
        Name =input('Enter Name: ')
        Age =input('Enter age: ')
        Fare =input('Enter Fare: ')
        Sex =input('Enter Sex: ')
        Ticket =input('Enter Ticket: ')
        db.traincsv.insert_one(
            {
                "PassengerId": Id,
                "Name":Name,
                "Age":Age,
                "Fare":Fare,
                "Sex":Sex,
                "Ticket":Ticket,
                })
        print("\nInserted data successfully\n")

    except Exception as e:
        print(str(e))

insert()
```

Output:

```
PS C:\Users\raman\BigData> python '.\INSERT Operation.py'
 Enter traincsv Passenger Id: 1
Enter Name: Ramanuj Rao
Enter age: 21
Enter Fare: 4500
Enter Sex: M
Enter Ticket: 2001

Inserted data successfully

PS C:\Users\raman\BigData>
```

Confirm that the data has been inserted by using MongoDB Compass to check whether the data has been inserted into the database.



- **Find Data:**

Code:

```
from pymongo import MongoClient
client= MongoClient('localhost:27017')
db = client.train

def read():
    try:
        TrainCol =db.traincsv.find()
        print("All Data From Train \n")

        for Train in TrainCol:
            print(Train)

    except Exception as e:
        print(str(e))

read()
```

Output:

```
PS C:\Users\raman\BigData> python '.\FIND Operation.py'
All Data From Train

{'_id': ObjectId('648c09c49b78579d2d9dd9d4'), 'PassengerId': '1', 'Name': 'Ramanuj Rao', 'Age': '21', 'Fare': '4500', 'Sex': 'M', 'Ticket': '2001'}
PS C:\Users\raman\BigData>
```

- **Update Data:**

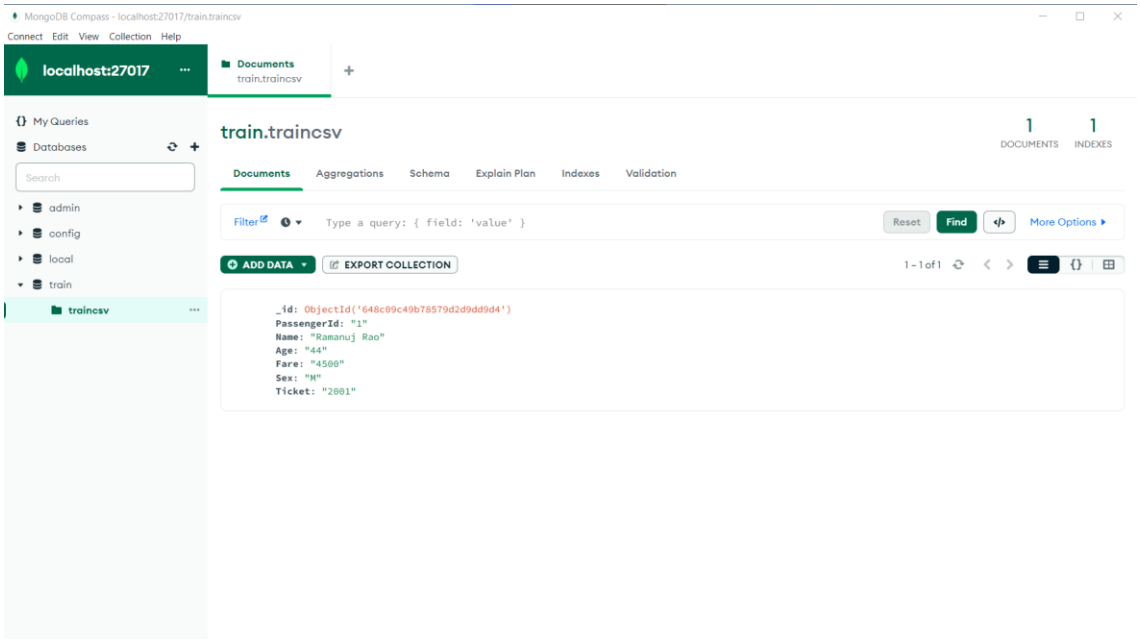
Code:

```
from pymongo import MongoClient
client= MongoClient('localhost:27017')
db = client.train

def update():
    try:
        name = input("Enter the Name to Update: ")
        age = input("Enter the Age to Update: ")

        db.traincsv.update_one({"Name": name},
                               {"$set": {"Age": age} }
                                )

        print("Record has been Updated")

    except Exception as e:
        print(str(e))

update()
```

Output:

```
PS C:\Users\raman\BigData> python '.\UPDATE Operation.py'
Enter the Name to Update: Ramanuj Rao
Enter the Age to Update: 44
Record has been Updated
PS C:\Users\raman\BigData>
```

Check on Mongo Compass as well



- **Delete Data:**

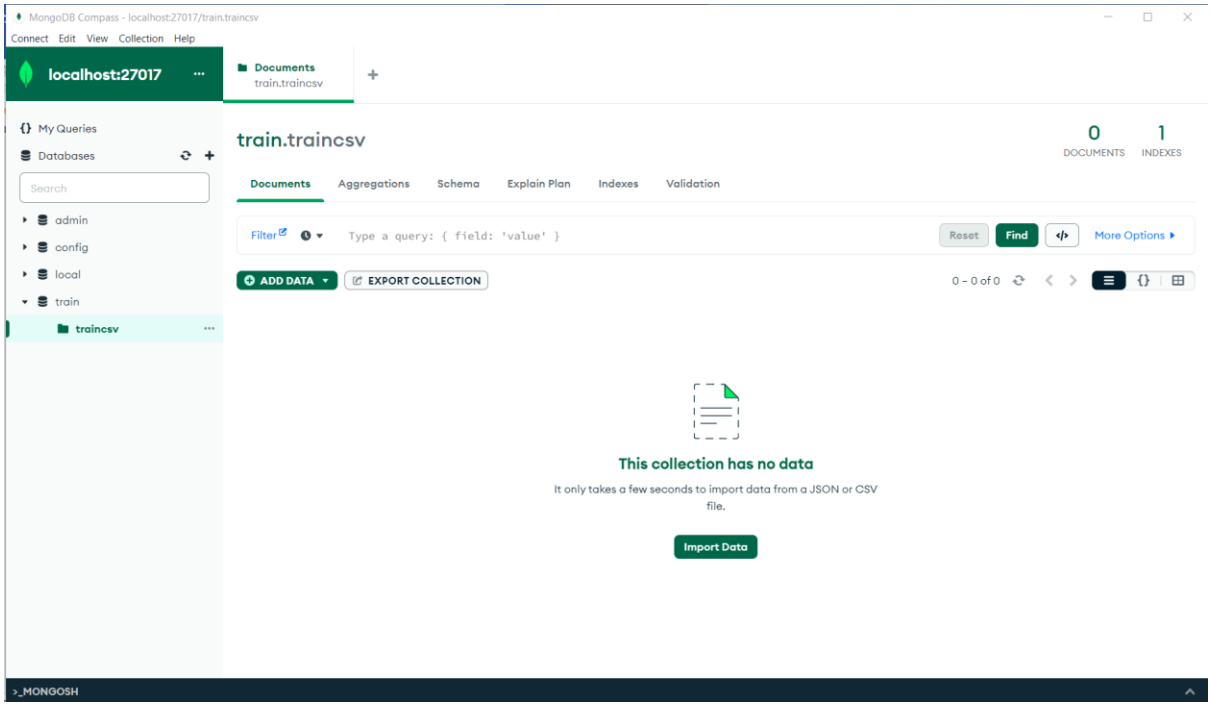
Code:

```
from pymongo import MongoClient

client = MongoClient("localhost:27017")

db = client.train

def delete():
    try:
        value = input("\n Enter the Name to Delete: ")
        db.traincsv.delete_one({"Name":value})
        print("\n DELETION SUCCESSFUL \n")

    except Exception as e:
        print(str(e))

delete()
```

Output:

```
PS C:\Users\raman\BigData> python '.\DELETE Operation.py'
 Enter the Name to Delete: Ramanuj Rao

 DELETION SUCCESSFUL

PS C:\Users\raman\BigData>
```

Check on Mongo Compass as well



PRESENTATION

Logistic Regression

Prepared By: Rahul Kewat
Msc-IT Part-1
Roll No-22006

What is logistic regression?

- Logistic regression is a data analysis technique that uses mathematics to find the relationships between two data factors. It then uses this relationship to predict the value of one of those factors based on the other. The prediction usually has a finite number of outcomes, like yes or no.
- **For example**, let's say you want to guess if your website visitor will click the checkout button in their shopping cart or not. Logistic regression analysis looks at past visitor behavior, such as time spent on the website and the number of items in the cart. It determines that, in the past, if visitors spent more than five minutes on the site and added more than three items to the cart, they clicked the checkout button. Using this information, the logistic regression function can then predict the behavior of a new website visitor.

Why is logistic regression important?

- Logistic regression is an important technique in the field of artificial intelligence and machine learning (AI/ML). ML models are software programs that you can train to perform complex data processing tasks without human intervention.

  Below, we list some benefits of using logistic regression over other ML techniques.

- **Simplicity**

  Logistic regression models are mathematically less complex than other ML methods. Therefore, you can implement them even if no one on your team has in-depth ML expertise.

Why is logistic regression important?

- **Speed**

  Logistic regression models can process large volumes of data at high speed because they require less computational capacity, such as memory and processing power. This makes them ideal for organizations that are starting with ML projects to gain some quick wins.

- **Flexibility**

  You can use logistic regression to find answers to questions that have two or more finite outcomes. You can also use it to preprocess data. For example, you can sort data with a large range of values, such as bank transactions, into a smaller, finite range of values by using logistic regression.

Use Cases

The logistic regression model is applied to a variety of situations in both the public and the private sector.

Some common ways that the logistic regression model is used include the following:

- **Medical:**

  Develop a model to determine the likelihood of a patient's successful response to a specific medical treatment or procedure. Input variables could include age, weight, blood pressure, and cholesterol levels.

Use Cases

- **Finance:**

  Using a loan applicant's credit history and the details on the loan, determine the probability that an applicant will default on the loan. Based on the prediction, the loan can be approved or denied, or the terms can be modified.

- **Marketing:**

  Determine a wireless customer's probability of switching carriers (known as churning) based on age, number of family members on the plan, months remaining on the existing contract, and social network contacts. With such insight, target the high-probability customers with appropriate offers to prevent churn.

Use Cases

- **Engineering:**

  Based on operating conditions and various diagnostic measurements, determine the probability of a mechanical part experiencing a malfunction or failure. With this, probability estimate, schedule the appropriate preventive maintenance activity.

Thank You