BIG DATA ANALYTICS

A PRACTICAL REPORT

ON

BIG DATA ANALYTICS

SUBMITTED BY
Mr. Mohd Kaif
Roll No: 22001

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.Mohd Kaif** 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

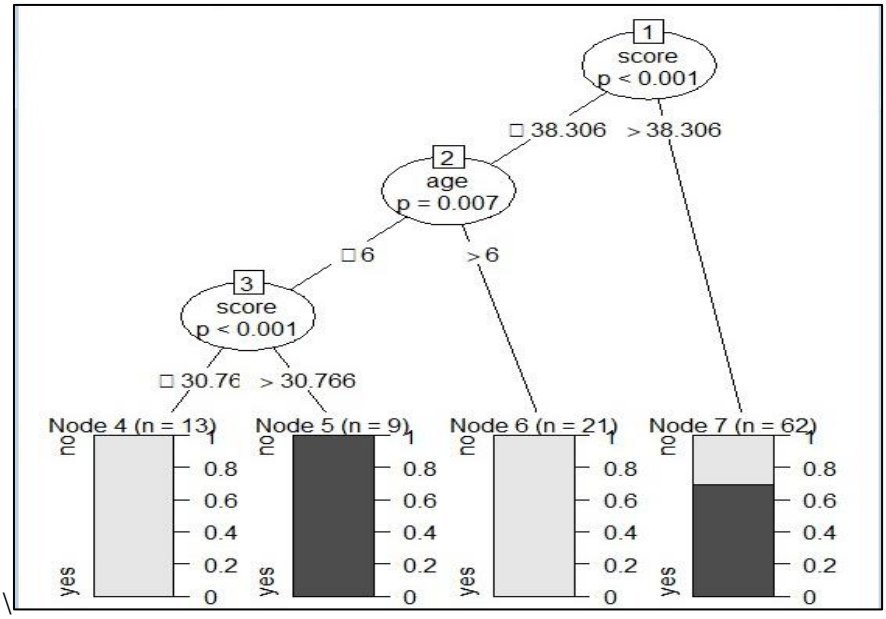
Practical 1

Aim: Implement Decision tree classification technique

```
library(party)
print(head(readingSkills))
input.dat  <- readingSkills[c(1:105),]
output.tree<- ctree(
nativeSpeaker ~ age +shoeSize+ score,
 data = input.dat
)
plot(output.tree)
```

Output:

```
> library(party)
Loading required package: grid
Loading required package: mvtnorm
Loading required package: modeltools
Loading required package: stats4
Loading required package: strucchange
Loading required package: zoo

Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric

Loading required package: sandwich
Warning message:
package 'party' was built under R version 4.2.3
> print(head(readingSkills))
  nativeSpeaker age shoeSize    score
1           yes   5 24.83189 32.29385
2           yes   6 25.95238 36.63105
3            no  11 30.42170 49.60593
4           yes   7 28.66450 40.28456
5           yes  11 31.88207 55.46085
6           yes  10 30.07843 52.83124
> input.dat  <- readingSkills[c(1:105),]
> output.tree <- ctree(
+   nativeSpeaker ~ age +shoeSize+ score,
+   data = input.dat
+ )
```



,

Practical 2

Aim: Implement SVM classification technique

Practical 2

AIM: Implement SVM classification technique

```
install.packages("caret")
library('caret')
heart <- read.csv("C:\\Users\\PC NO 19 IT\\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:

```
> install.packages("caret")
Warning: package 'caret' is in use and will not be installed
> library('caret')
> heart <- read.csv("C:\\Users\\PC NO 19 IT\\Downloads\\heart.csv", sep = ',', header = FALSE)
> str(heart)
'data.frame':   304 obs. of  14 variables:
 $ V1 : chr  "age" "63" "37" "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] 214  14
> dim(testing);
[1] 90 14
> anyNA(heart)
[1] FALSE
> summary(heart)
      V1                 V2                 V3                 V4
 Length:304         Length:304         Length:304         Length:304
 Class :character   Class :character   Class :character   Class :character
```

```
> summary(heart)
      V1                 V2                 V3                 V4
 Length:304         Length:304         Length:304         Length:304
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character
      V5                 V6                 V7                 V8
 Length:304         Length:304         Length:304         Length:304
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character
      V9                V10                V11                V12
 Length:304         Length:304         Length:304         Length:304
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character
     V13                V14
 Length:304         Length:304
 Class :character   Class :character
 Mode  :character   Mode  :character
> 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)
Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut = 10,  :
  These variables have zero variances: V176, V4172, V4192, V5199, V5205, V5284, V5319, V8116, V8184, V8195, V100.3, V101.1, V102.3
Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut = 10,  :
```

```
214 samples
 13 predictor
  3 classes: '0', '1', 'output'

Pre-processing: centered (340), scaled (340)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 194, 193, 193, 193, 193, 193, ...
Resampling results:

  Accuracy   Kappa
  0.7145009  0.4218929

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      0      0      0      0      0      0
[211] 0      0      0      0
Levels: 0 1 output
```

Practical 3

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

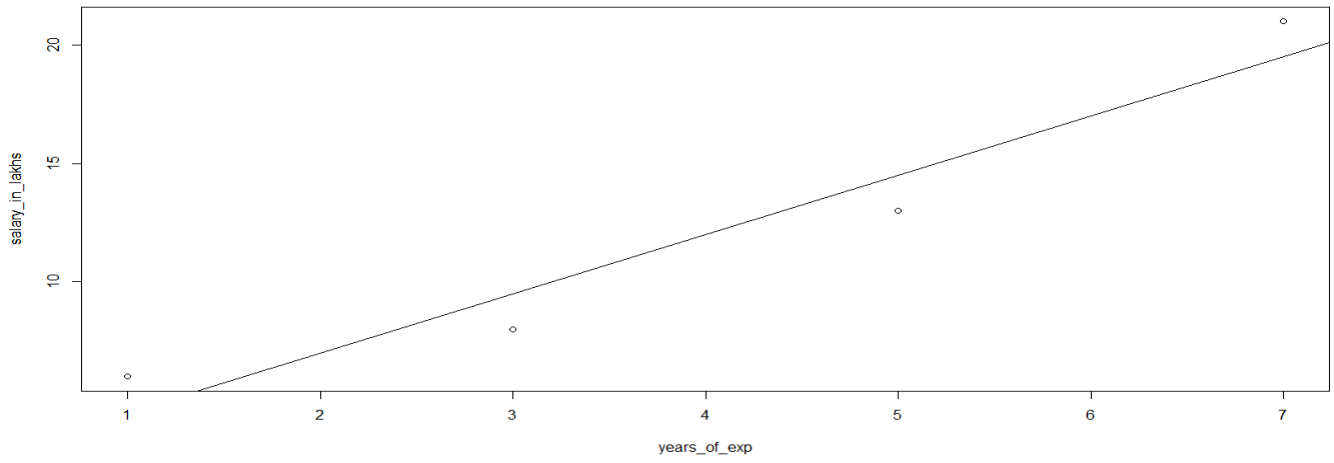
Practical 3

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.

```
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:

```
> 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
  years_of_exp salary_in_lakhs
1            7              21
2            5              13
3            1               6
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.

Practical 4

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

```
mydata<-read.csv("C:\\Users\\RDNC\\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:

```
> mydata<-read.csv("C:\\Users\\RDNC\\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
```

```
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

Practical 5

Aim: Build a Classification Model

```
In [1]: %matplotlib inline
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
```

```
In [2]: fruits = pd.read_table('fruit_data_with_colors.txt')
        fruits.head()
```

Out[2]:

	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

```
In [3]: print(fruits['fruit_name'].unique())
```

```
['apple' 'mandarin' 'orange' 'lemon']
```

```
In [4]: print(fruits.shape)
```

```
(59, 7)
```

Logistic Regression

```
In [23]: from sklearn.linear_model import LogisticRegression

         logreg = LogisticRegression()
         logreg.fit(X_train, y_train)

         print('Accuracy of Logistic regression classifier on training set: {:.2f}'
              .format(logreg.score(X_train, y_train)))
         print('Accuracy of Logistic regression classifier on test set: {:.2f}'
              .format(logreg.score(X_test, y_test)))
```

```
Accuracy of Logistic regression classifier on training set: 0.75
Accuracy of Logistic regression classifier on test set: 0.47
```

Decision Tree

```
In [27]: from sklearn.tree import DecisionTreeClassifier

         clf = DecisionTreeClassifier().fit(X_train, y_train)

         print('Accuracy of Decision Tree classifier on training set: {:.2f}'
              .format(clf.score(X_train, y_train)))
         print('Accuracy of Decision Tree classifier on test set: {:.2f}'
              .format(clf.score(X_test, y_test)))
```

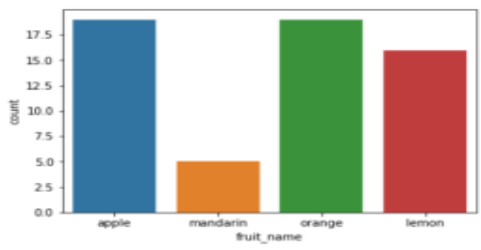
```
Accuracy of Decision Tree classifier on training set: 1.00
Accuracy of Decision Tree classifier on test set: 0.67
```

Fruit type distribution

```
In [147]: print(fruits.groupby('fruit_name').size())
```

```
fruit_name
apple       19
lemon       16
mandarin     5
orange      19
dtype: int64
```

```
In [148]: import seaborn as sns
          sns.countplot(fruits['fruit_name'],label="Count")
          plt.show()
```



Practical 6

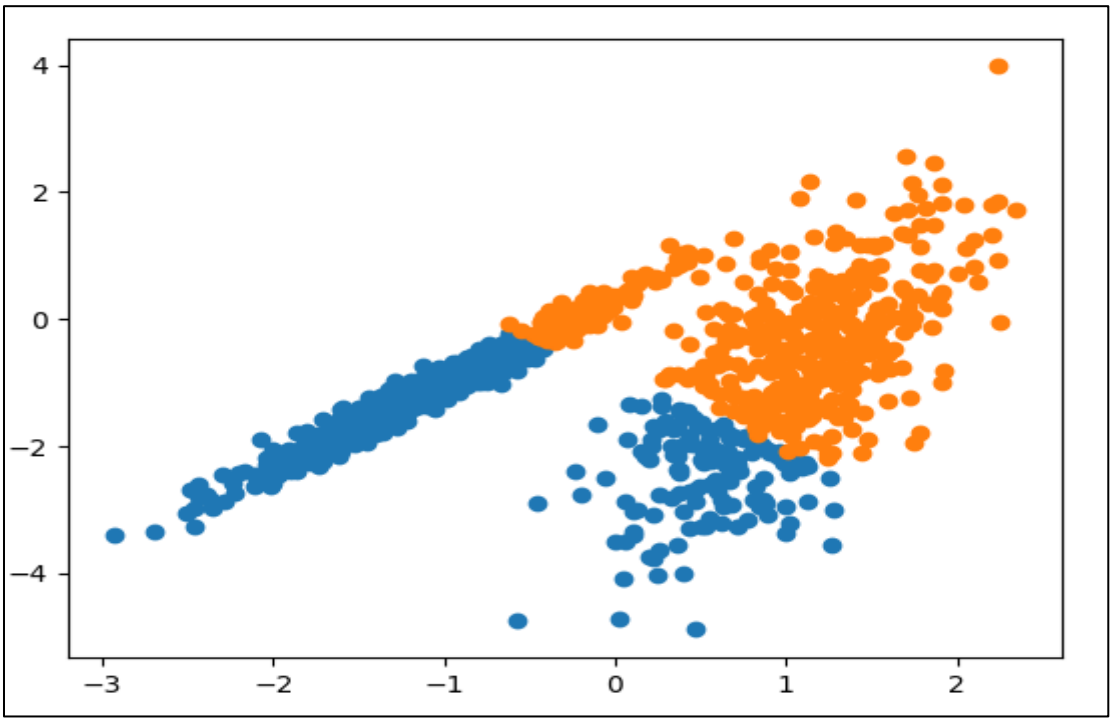
Aim: Build a Clustering Model

Practical 6

Aim: Build a Clustering Model

```
# k-mean clustering
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import KMeans
from matplotlib import pyplot
# define dataset
X, _ =
make_classification(n_samples=1000,n_features=2,n_informative=2,n_redundant=0,n_clusters_per_class=
1,random_state=4)
#define the model
model= KMeans(n_clusters=2)
#fit the model
model.fit(X)
# assign a cluster to each example
yhat=model.predict(X)
#retrieve unique clusters
clusters = unique(yhat)
for cluster in clusters:
row_ix=where(yhat== cluster)
pyplot.scatter(X[row_ix,0],X[row_ix,1])
pyplot.show()
```

Output:



Practical 7

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

Practical 7

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

1. Install Java

– extract and install Java in C:\Java

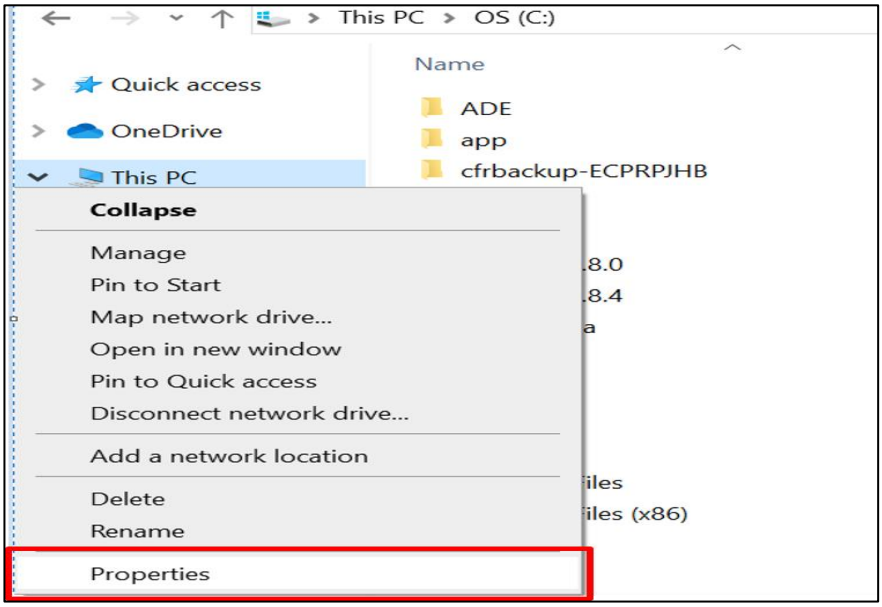
– open cmd and type ->javac -version

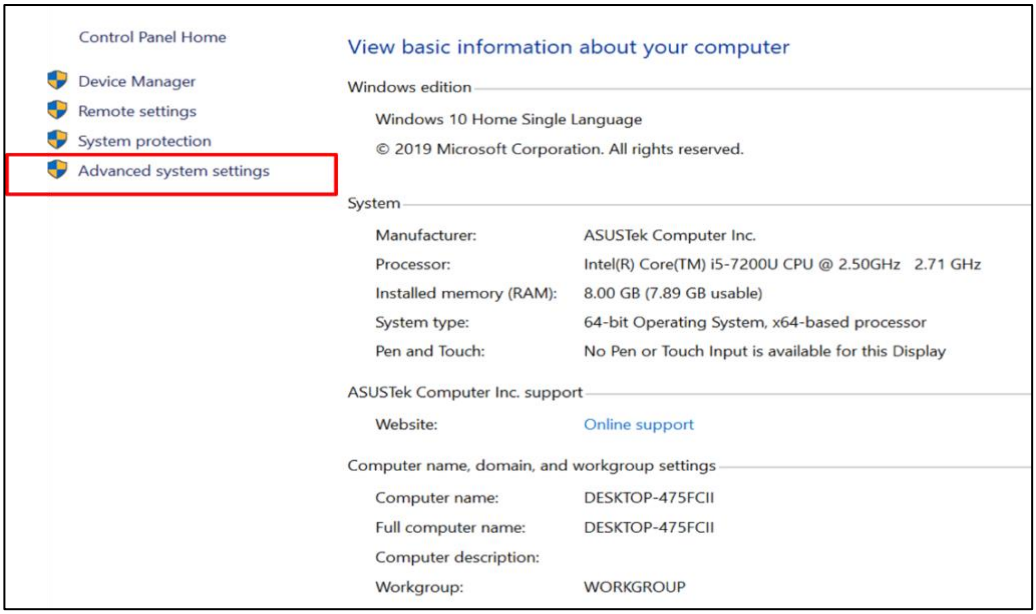
```
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\asus>javac -version
javac 1.8.0_241
```

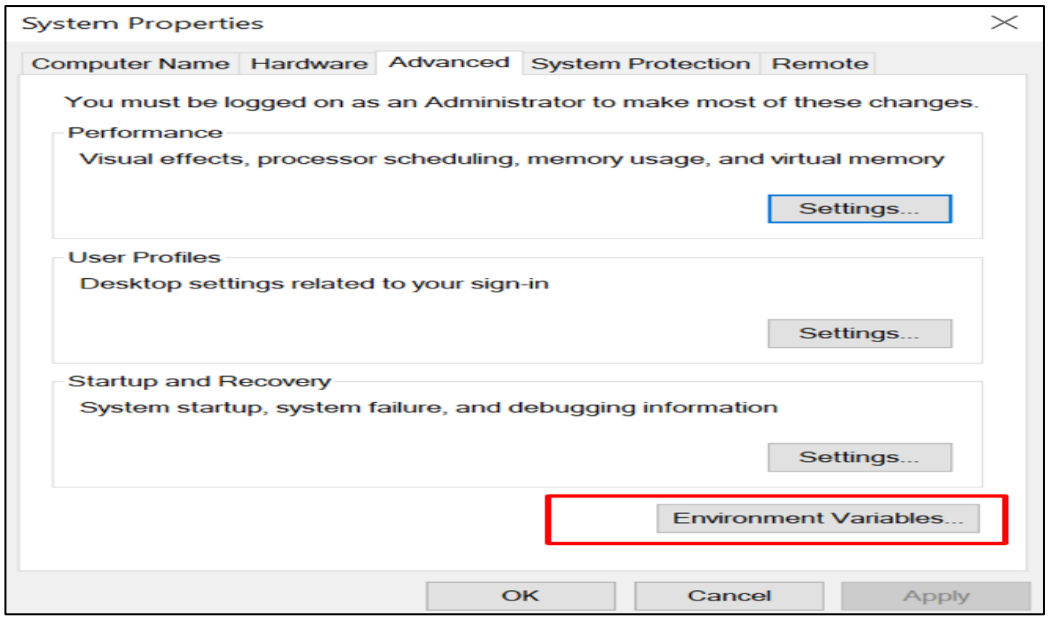
2. Download Hadoop

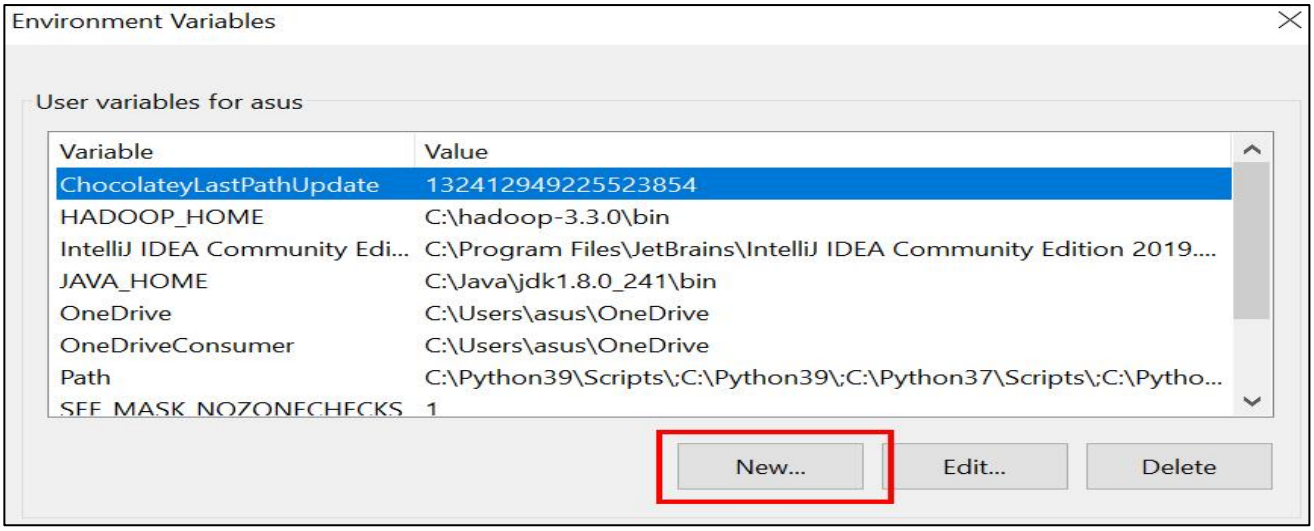
k– extract to C:\Hadoop

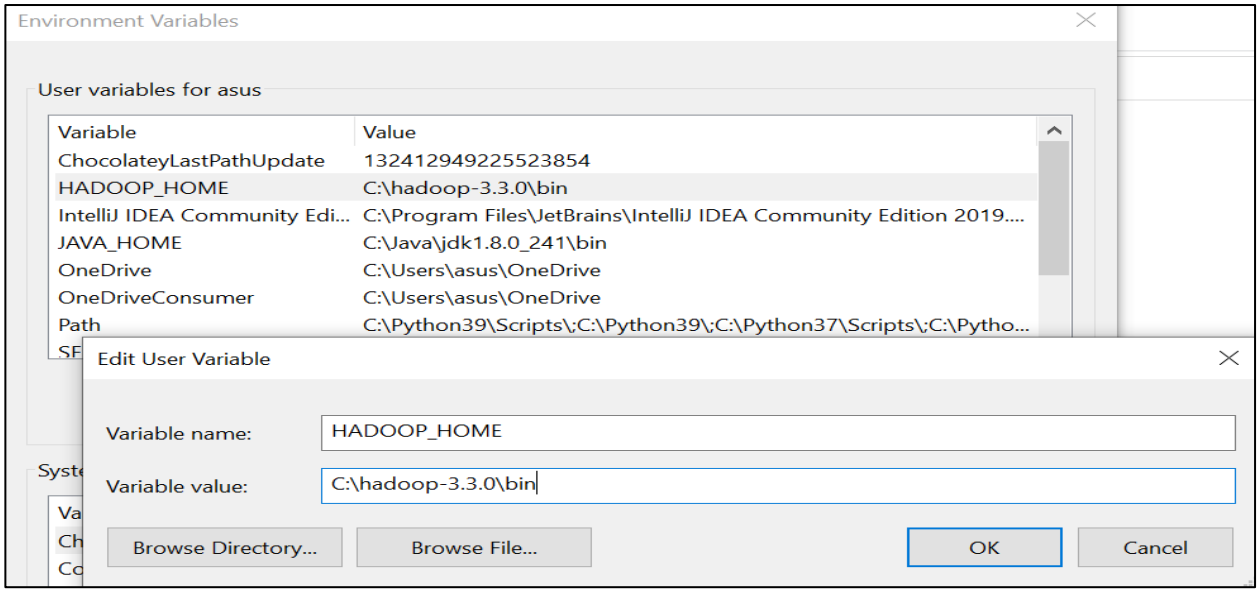
3. Set the path JAVA_HOME Environment variable
4. Set the path HADOOP_HOME Environment variable

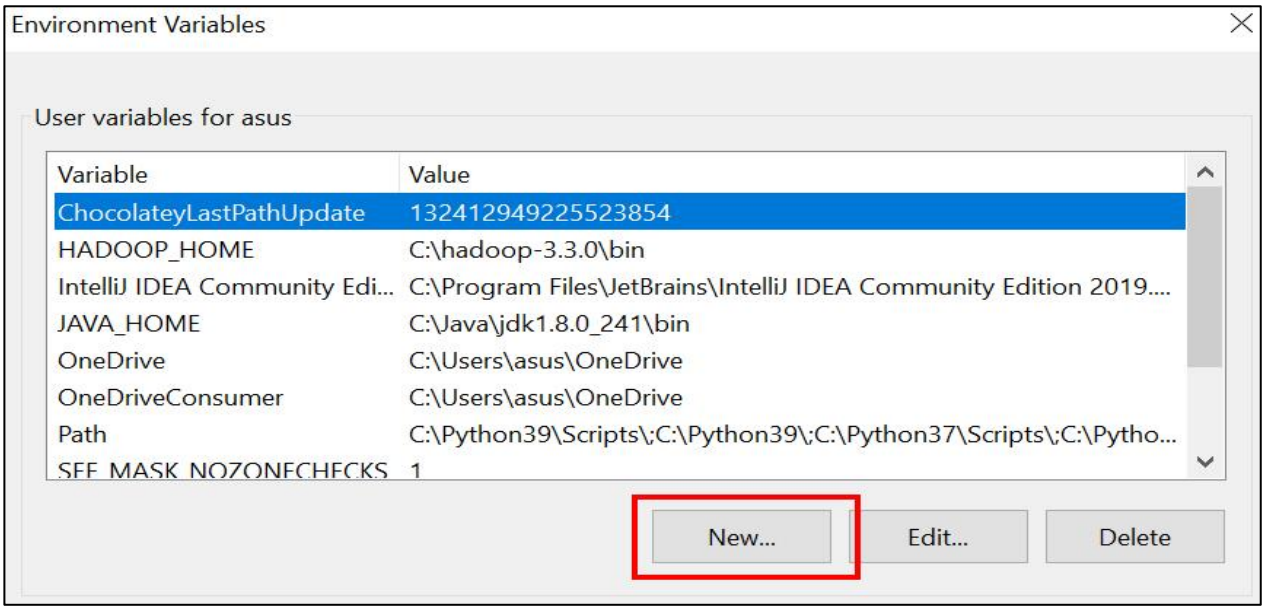


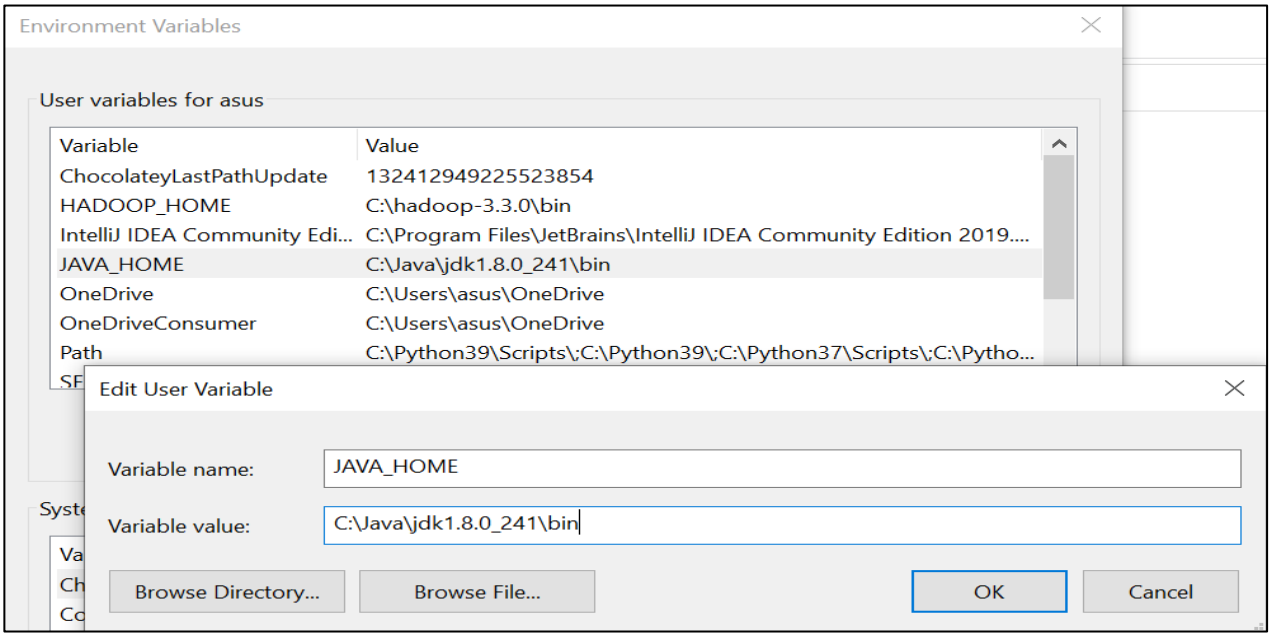

Control Panel Home
Device Manager
Remote settings
System protection
Advanced system settings
View basic information about your computer
Windows edition
Windows 10 Home Single Language
© 2019 Microsoft Corporation. All rights reserved.
System
Manufacturer: ASUSTek Computer Inc.
Processor: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz
Installed memory (RAM): 8.00 GB (7.89 GB usable)
System type: 64-bit Operating System, x64-based processor
Pen and Touch: No Pen or Touch Input is available for this Display
ASUSTek Computer Inc. support
Website: Online support
Computer name, domain, and workgroup settings
Computer name: DESKTOP-475FCII
Full computer name: DESKTOP-475FCII
Computer description:
Workgroup: WORKGROUP

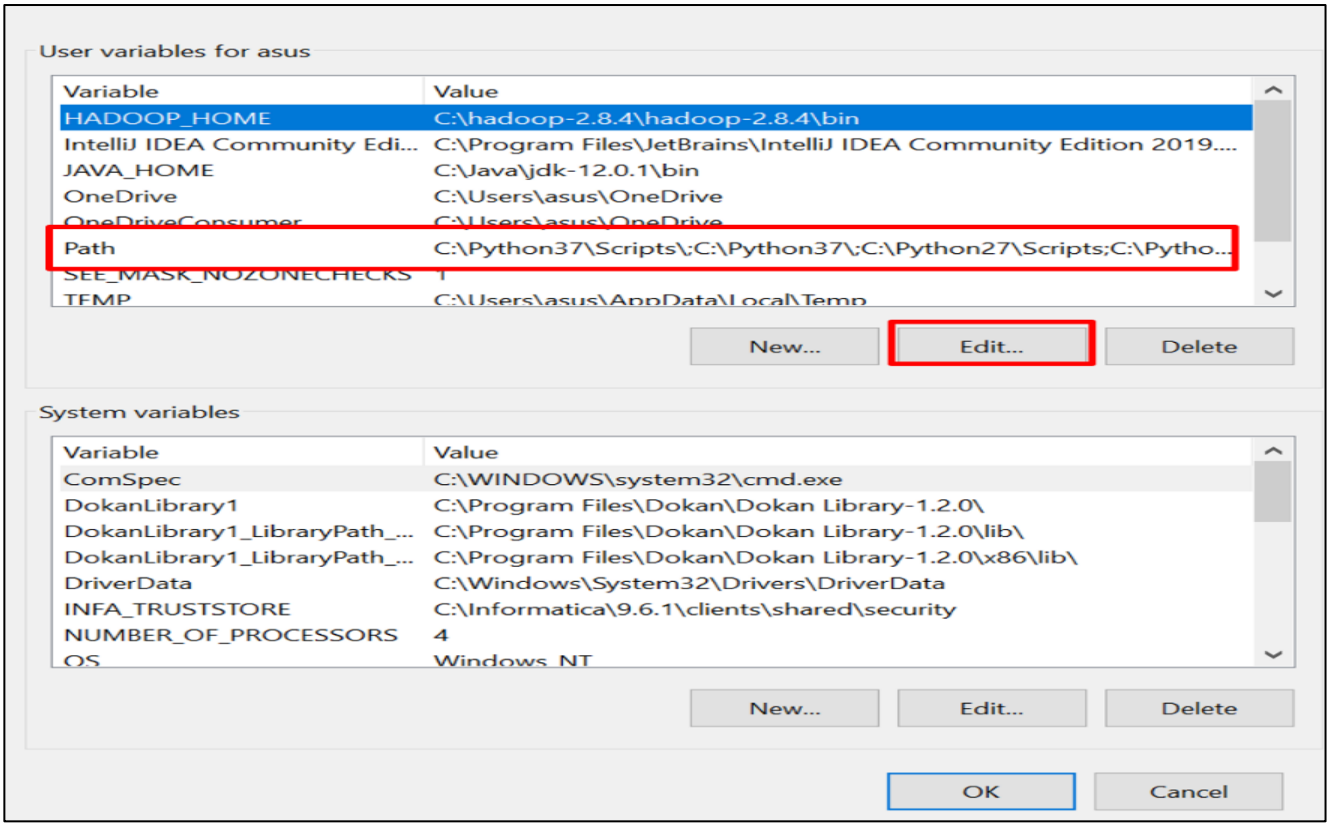

System Properties
Computer Name
Hardware
Advanced
System Protection
Remote
You must be logged on as an Administrator to make most of these changes.
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings...
User Profiles
Desktop settings related to your sign-in
Settings...
Startup and Recovery
System startup, system failure, and debugging information
Settings...
Environment Variables...
OK
Cancel
Apply

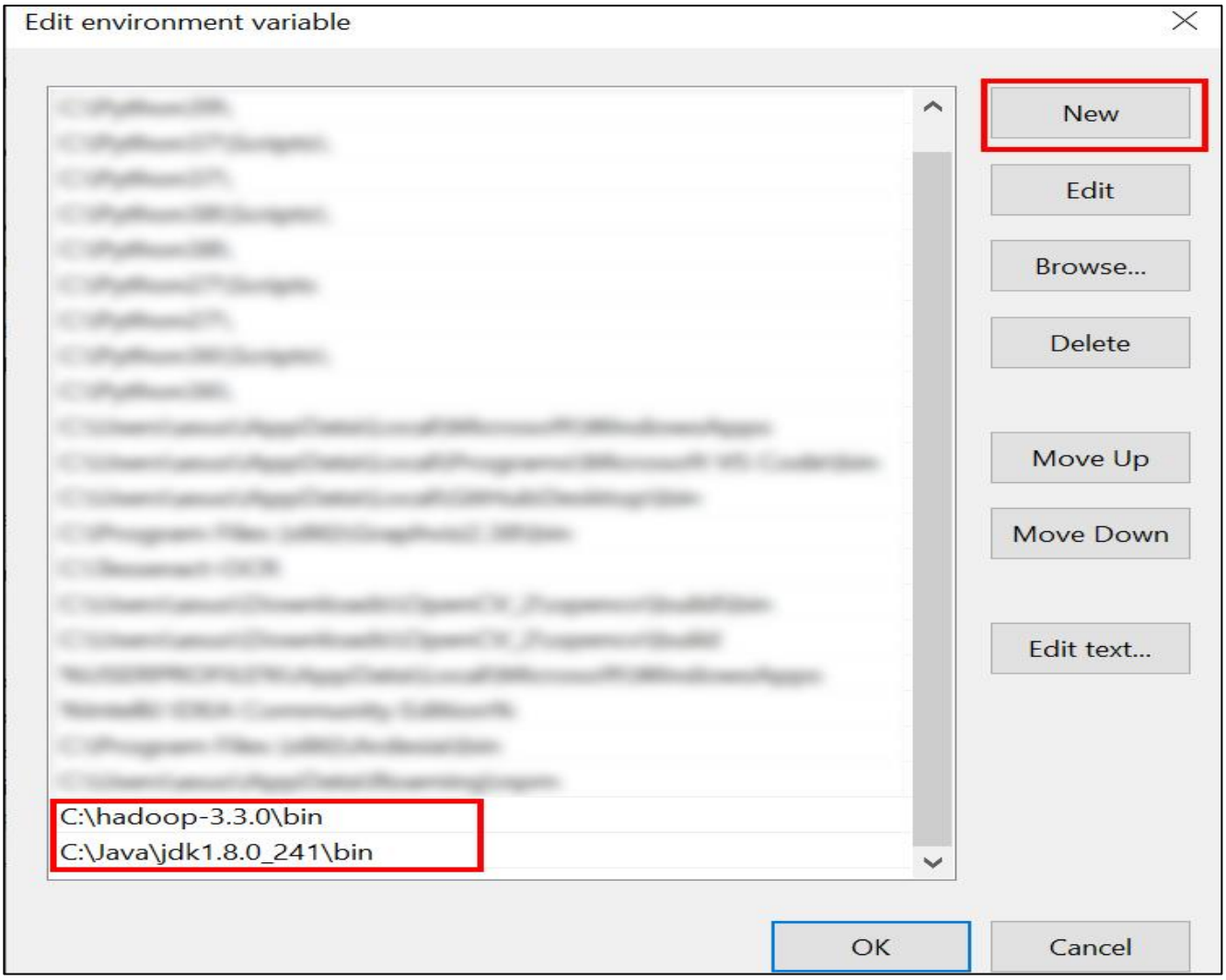

Environment Variables
User variables for asus
Variable
Value
ChocolateyLastPathUpdate
132412949225523854
HADOOP_HOME
C:\hadoop-3.3.0\bin
IntelliJ IDEA Community Edi...
C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2019....
JAVA_HOME
C:\Java\jdk1.8.0_241\bin
OneDrive
C:\Users\asus\OneDrive
OneDriveConsumer
C:\Users\asus\OneDrive
Path
C:\Python39\Scripts\;C:\Python39\;C:\Python37\Scripts\;C:\Pytho...
SEE_MASK_NOZONECHECKS
1
New...
Edit...
Delete


Environment Variables
User variables for asus
Variable
Value
ChocolateyLastPathUpdate
132412949225523854
HADOOP_HOME
C:\hadoop-3.3.0\bin
IntelliJ IDEA Community Edi...
C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2019....
JAVA_HOME
C:\Java\jdk1.8.0_241\bin
OneDrive
C:\Users\asus\OneDrive
OneDriveConsumer
C:\Users\asus\OneDrive
Path
C:\Python39\Scripts\;C:\Python39\;C:\Python37\Scripts\;C:\Pytho...
Edit User Variable
Variable name:
HADOOP_HOME
Variable value:
C:\hadoop-3.3.0\bin
Browse Directory...
Browse File...
OK
Cancel


Environment Variables
User variables for asus
Variable
Value
ChocolateyLastPathUpdate
132412949225523854
HADOOP_HOME
C:\hadoop-3.3.0\bin
IntelliJ IDEA Community Edi...
C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2019....
JAVA_HOME
C:\Java\jdk1.8.0_241\bin
OneDrive
C:\Users\asus\OneDrive
OneDriveConsumer
C:\Users\asus\OneDrive
Path
C:\Python39\Scripts\;C:\Python39\;C:\Python37\Scripts\;C:\Pytho...
SFE_MASK_NOZONECHECKS
1
New...
Edit...
Delete


Environment Variables
User variables for asus
Variable
Value
ChocolateyLastPathUpdate
132412949225523854
HADOOP_HOME
C:\hadoop-3.3.0\bin
IntelliJ IDEA Community Edi...
C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2019....
JAVA_HOME
C:\Java\jdk1.8.0_241\bin
OneDrive
C:\Users\asus\OneDrive
OneDriveConsumer
C:\Users\asus\OneDrive
Path
C:\Python39\Scripts\;C:\Python39\;C:\Python37\Scripts\;C:\Pytho...
Edit User Variable
Variable name:
JAVA_HOME
Variable value:
C:\Java\jdk1.8.0_241\bin
Browse Directory...
Browse File...
OK
Cancel


User variables for asus
Variable
Value
HADOOP_HOME
C:\hadoop-2.8.4\hadoop-2.8.4\bin
IntelliJ IDEA Community Edi...
C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2019....
JAVA_HOME
C:\Java\jdk-12.0.1\bin
OneDrive
C:\Users\asus\OneDrive
Path
C:\Python37\Scripts\;C:\Python37\;C:\Python27\Scripts;C:\Pytho...
New...
Edit...
Delete
System variables
Variable
Value
ComSpec
C:\WINDOWS\system32\cmd.exe
DokanLibrary1
C:\Program Files\Dokan\Dokan Library-1.2.0\
DokanLibrary1_LibraryPath_...
C:\Program Files\Dokan\Dokan Library-1.2.0\lib\
DokanLibrary1_LibraryPath_...
C:\Program Files\Dokan\Dokan Library-1.2.0\x86\lib\
DriverData
C:\Windows\System32\Drivers\DriverData
INFA_TRUSTSTORE
C:\Informatica\9.6.1\clients\shared\security
NUMBER_OF_PROCESSORS
4
New...
Edit...
Delete
OK
Cancel


Edit environment variable
C:\hadoop-3.3.0\bin
C:\Java\jdk1.8.0_241\bin
New
Edit
Browse...
Delete
Move Up
Move Down
Edit text...
OK
Cancel

5. Configurations

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

paste the xml code in folder and save

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

==========

Rename “mapred-site.xml.template” to “mapred-site.xml” and edit this file C:/Hadoop-3.3.0/etc/hadoop/mapred-site.xml, paste xml code and save this file.

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

==========

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,

paste xml code and save this file.

```
<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>
```

==========

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

paste xml code and save this file.

```
 <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>
```

==========

Edit file C:/Hadoop-3.3.0/etc/hadoop/hadoop-env.cmd by closing the command line

"JAVA_HOME=%JAVA_HOME%" instead of set "JAVA_HOME=C:\Java"

==========

6)Hadoop Configurations

– Copy folder bin and replace existing bin folder in

C:\Hadoop-3.3.0\bin

– Format the NameNode

– Open cmd and type command "hdfsnamenode –format"

```
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\hadoop-3.3.0\bin>hdfs namenode -format
```

7. Testing

– Open cmd and change directory to C:\Hadoop-3.3.0\sbin

– type start-all.cmd

```
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\hadoop-3.3.0\sbin>start-all.cmd
```

(Or you can start like this)

– Start namenode and datanode with this command

– type start-dfs.cmd

– Start yarn through this command

– type start-yarn.cmd

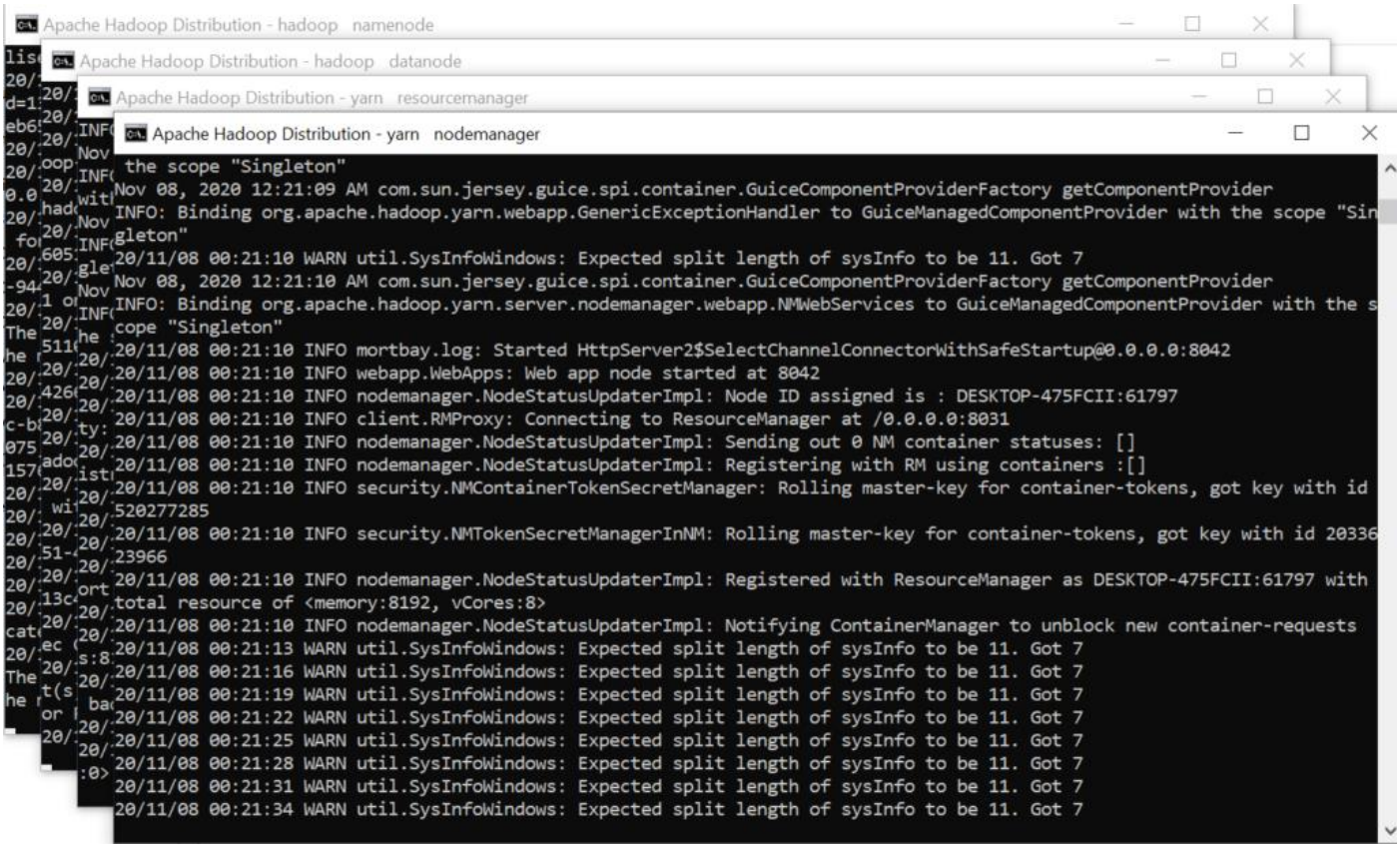
Make sure these apps are running

– Hadoop Namenode

– Hadoop datanode

– YARN Resource Manager

– YARN Node Manager



Open: http://localhost:8088

Practical 8

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

Practical 8

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

1)Insert data

```
from pymongo import MongoClient
client= MongoClient('localhost:27017')
db = client.train
def insert():
   try:
traincsvId =input('Enter traincsv Passenger Id:')
traincsvName =input('Enter Name:')
traincsvAge =input('Enter age:')
traincsvFare =input('Enter Fare:')
traincsvSex =input('Enter Sex:')
traincsvTicket =input('Enter Ticket:')
db.traincsv.insert_one(
        {
           "PassengerId": traincsvId,
              "Name":traincsvName,
           "Age":traincsvAge,
           "Fare":traincsvFare,
           "Sex":traincsvSex,
           "Ticket":traincsvTicket,
           })
      print("\nInserted data successfully\n")
   except Exception as e:
      print(str(e))
insert()
```

Output:

```
_id: ObjectId('64746ba3ada629fcc997b22d')
PassengerId: "22016"
Name: "IQRA"
Age: "21"
Fare: "2345"
Sex: "Female"
Ticket: "234"

_id: ObjectId('64746bfcdae90db5e59a7820')
PassengerId: "22017"
Name: "KANCHAN"
Age: "21"
Fare: "234"
Sex: "Female"
Ticket: "678"
```

2) Find data

```
from pymongo import MongoClient
client= MongoClient('localhost:27017')
db = client.train
def read():
   try:
trainCol=db.traincsv.find()
      print("All data From database")
      for train in trainCol:
         print(train)
   except Exception as e:
      print(str(e))
read()
```

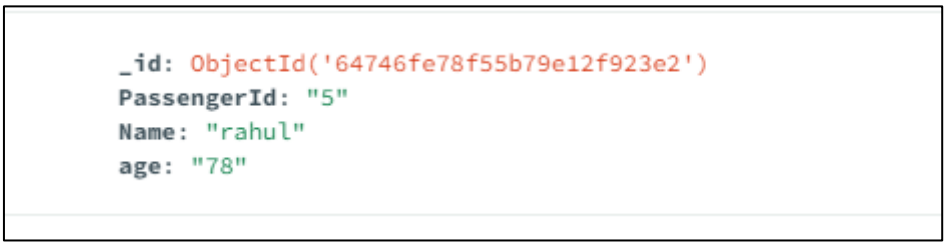
Output:

```
All data From database
{'_id': ObjectId('6465a89ec42f1ef323a5abc7'), 'PassengerId': '22017'}
{'_id': ObjectId('6465a974f7be6de29ddb29c5'), 'PassengerId': '22014'}
{'_id': ObjectId('6465aa334c11d91f7f008a7b'), 'PassengerId': '22014', 'Name': 'iqra', 'age': '34'}
{'_id': ObjectId('6465b43e024f7be17fedbb5e'), 'PassengerId': '22016', 'Name': 'uzma', 'Age': '23', 'Fare': '235', 'Sex': 'female', 'Ticket': '678'}
{'_id': ObjectId('6465b457581a31b75d3fd621'), 'PassengerId': '22019', 'Name': 'ayub', 'Age': '34', 'Fare': '234', 'Sex': 'male', 'Ticket': '345'}
{'_id': ObjectId('6465b472a24537a3018494a5'), 'PassengerId': '22012', 'Name': 'zoheb', 'Age': '35', 'Fare': '234', 'Sex': 'male', 'Ticket': '789'}
{'_id': ObjectId('6465d4eb81d8094d08c22810'), 'PassengerId': '5', 'Name': 'raju', 'age': '89'}
{'_id': ObjectId('6465dae1234c77549ab5d57d'), 'PassengerId': '1', 'Name': 'sham', 'Age': '78', 'Fare': '890', 'Sex': 'male', 'Ticket': '234'}
{'_id': ObjectId('64746ba3ada629fcc997b22d'), 'PassengerId': '22016', 'Name': 'IQRA', 'Age': '21', 'Fare': '2345', 'Sex': 'Female', 'Ticket': '234'}
{'_id': ObjectId('64746bfcdae90db5e59a7820'), 'PassengerId': '22017', 'Name': 'KANCHAN', 'Age': '21', 'Fare': '234', 'Sex': 'Female', 'Ticket': '678'}

[Done] exited with code=0 in 0.814 seconds
```

3) Update data

```
from pymongo import MongoClient
client= MongoClient('localhost:27017')
db = client.train
def update():
   try:
traincsvId =input('Enter Passenger Id to update:')
traincsvName =input('Enter Name to update:')
traincsvAge =input('Enter age to update:')
db.traincsv.insert_one(
        {"PassengerId": traincsvId,
          "Name":traincsvName,
          "age":traincsvAge,
        }

      )
      print("\nUpdated data successfully\n")
   except Exception as e:
      print(str(e))
update()
```

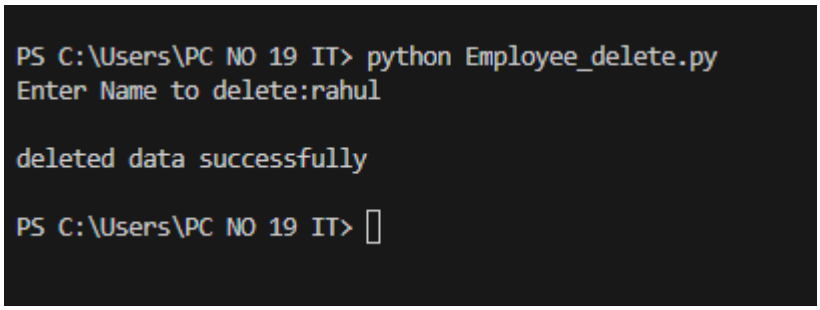
Output:



4) Delete data

```
from pymongo import MongoClient
client= MongoClient('localhost:27017')
db = client.train
def delete():
   try:
      criteria=input("Enter Name to delete:")
db.traincsv.delete_one({"Name":criteria})
      print("\ndeleted data successfully\n")
   except Exception as e:
      print(str(e))
delete()
```

Output:



PRESENTATION

Responsibilities of a data scientist

-Mohd Kaif
M.Sc. I.T. Part 1
22001

Introduction

A Data Scientist is a professional who collects large amounts of data using analytical, statistical, and programmable skills. It is their responsibility to use data to develop solutions tailored to meet the organisation's unique needs.

Organizations are increasingly using more and more data in their everyday operations. A data scientist interprets the raw data and extracts valuable meaning out of it. They then use this information to find patterns and develop solutions that an organization needs to grow and compete.

If we must define a data scientist, it would be someone who makes value out of data. Wondering what exactly constitutes the job of a data scientist? Well, no points for guessing, a day of a data scientist revolves around - as the job title suggests - data and data everywhere.

The data scientist job description involves fetching information from various sources and analyzing it to get a clear understanding of how an organization performs. The scientist uses statistical and analytical methods plus AI tools to automate specific processes within the organization and develop smart solutions to business challenges. After interpreting the data, they present the results in a clear and interesting way. The objective is to help the organization analyze trends to make better decisions. Thus, a good data scientist needs to have the right combination of technical, analytical, and communication skills.

Data Mining

Data mining is the process of extracting useful information from a large volume of data sources such as databases, log files, and social media platforms. Data scientists use statistical and machine learning techniques to analyze these datasets and identify trends, patterns, and relationships. The goal of data mining is to extract valuable insights from data that can be used to drive business decisions.

Data mining involves a range of techniques, including clustering, regression, and classification. Clustering is used to group data points into similar clusters based on their similarities, while regression is used to predict a numerical value based on a set of input variables. Classification involves the creation of a model that can classify new data points into predefined classes or categories.

Data mining has a wide range of applications, including marketing, healthcare, and finance. In marketing, data mining is used to identify customer segments and develop targeted marketing campaigns. In healthcare, data mining is used to analyze patient data and develop predictive models for disease diagnosis and treatment. In finance, data mining is used to detect fraudulent transactions and develop risk models for investments.

Machine Learning

Machine learning is a subfield of artificial intelligence that enables computers to learn from data without being explicitly programmed. Data scientists use machine learning algorithms to select relevant features from large datasets and optimize classifiers to build predictive models. The goal of machine learning is to create models that can learn from data and make accurate predictions on new, unseen data.

Machine learning involves a range of techniques, including supervised learning, unsupervised learning, and reinforcement learning. Supervised learning involves the creation of a model that can learn from labeled data to make predictions on new data. Unsupervised learning involves the creation of a model that can learn from unlabeled data to identify patterns and relationships. Reinforcement learning involves the creation of a model that can learn from feedback to optimize its performance.

Preprocessing

Preprocessing is the process of preparing data for analysis by cleaning, transforming, and normalizing it. Data scientists perform preprocessing on structured and unstructured data to ensure it is clean, complete, and usable for analysis. This involves removing missing data, correcting errors, and transforming data into a format that can be easily analyzed.

Preprocessing involves a range of techniques, including data cleaning, data transformation, and data normalization. Data cleaning involves the removal of irrelevant data, duplicate data, and incomplete data. Data transformation involves the conversion of data into a format that can be easily analyzed, such as converting categorical data into numerical data. Data normalization involves the scaling of data to ensure that it is consistent across different variables.

Data Collection

Data collection involves gathering information from various sources, such as databases, social media, and customer feedback, to support decision-making. Data scientists enhance data collection procedures to include all relevant information for developing analytical systems that can generate insights. They may also work to improve data quality and ensure data is stored securely.

Data collection involves a range of techniques, including surveys, experiments, and observational studies. Surveys involve collecting data from a sample of individuals using questionnaires or interviews. Experiments involve manipulating variables to observe their effects on a particular outcome. Observational studies involve observing and recording data without manipulating variables

Data Processing

Data processing is the process of converting raw data into a format that can be analyzed. Data scientists process, cleanse, and validate the integrity of data to be used for analysis. This involves using data cleaning techniques to remove missing data, correcting errors, and transforming data into a format that can be easily analyzed. Once the data has been processed, it is ready for analysis using statistical and machine learning techniques.

Data processing involves a range of techniques, including data cleaning, data transformation, and data validation. Data cleaning involves the removal of irrelevant data, duplicate data, and incomplete data. Data transformation involves the conversion of data into a format that can be easily analyzed, such as converting categorical data into numerical data. Data validation involves ensuring that the data is accurate and complete and meets the required standards.

Data Analysis

Data analysis is the process of examining large amounts of data to find patterns and insights. Data scientists use statistical and machine learning techniques to analyze data and identify trends, patterns, and relationships. The goal of data analysis is to extract valuable insights from data that can be used to drive business decisions.

Data analysis involves a range of techniques, including statistical analysis, machine learning, and data visualization. Statistical analysis involves using mathematical models to analyze data and identify patterns and relationships. Machine learning involves using algorithms to learn from data and make predictions on new data. Data visualization involves using charts and graphs to present data in a visually appealing way.

Prediction systems and machine learning algorithms

Data scientists develop prediction systems and machine learning algorithms to automate decision-making and improve business outcomes. They use machine learning algorithms to develop predictive models that can be used to forecast future events and identify patterns and trends in large datasets. The goal of prediction systems and machine learning algorithms is to enable businesses to make data-driven decisions that are more accurate and effective.

Prediction systems and machine learning algorithms involve a range of techniques, including regression, classification, and clustering. Regression involves predicting a numerical value based on a set of input variables. Classification involves the creation of a model that can classify new data points into predefined classes or categories. Clustering involves grouping data points into similar clusters based on their similarities.

Results presentation

Data scientists present their results in a clear and concise manner to enable business leaders to make informed decisions. They use data visualization techniques to present complex data in a visually appealing way that is easy to understand. The goal of results presentation is to enable business leaders to make data-driven decisions that are based on accurate and reliable data.

Results presentation involves a range of techniques, including data visualization, data storytelling, and report writing. Data visualization involves using charts and graphs to present data in a visually appealing way. Data storytelling involves presenting data in the form of a narrative that is easy to understand. Report writing involves presenting data in a written report that summarizes key findings and recommendation

Proposing solutions and strategies

Data scientists propose solutions and strategies to tackle business challenges based on their analysis of data. They work closely with business leaders to understand their needs and identify areas where data can be used to drive business outcomes. The goal of proposing solutions and strategies is to enable businesses to make informed decisions that are based on accurate and reliable data.

Proposing solutions and strategies involves a range of techniques, including data analysis, data

THANK YOU