22306A1012

22306A

Sr. No.	Title/Aim of The Practical	Page No.	Date	Signature
1	a) K-Means Clustering: Clustering algorithms for unsupervised classification. Read a datafile grades_km_input.csv and apply k-means clustering. Plot the cluster data using R visualizations.	1	21/02/2023	
	b) Apriori Algorithm (PBL): Implement Apriori Algorithm Recommending grocery items to a customer that is most frequently bought together, given a data set of transactions by customers of a store, using Ariory algorithm using Market_Basket_Optimisation.csv file.	5		
2	a) Regression Model: Import data from web storage – binary.csv. Name the dataset and do Logistic Regression to find out relation between variables that are affecting the admission of a student in an institute based on his or her GRE score, GPA obtained and rank of the student. Also check the model is fit or not.	10	10/03/2023	
	b) MULTIPLE REGRESSION MODEL: Apply multiple regressions, if data have a continuous independent variable. Apply on above dataset – binary.csv.	12		
	c) Design a Simple Linear Regression Model using the above dataset. (HINT: consider GRE Score or GPA score as independent variable.	14		
3	a) Decision Tree: Implement Decision Tree classification technique using Social_Network_Ads.csv dataset.	18	17/03/2023	
	b) SVM Classification: Implement SVM Classification technique using Social_Network_Ads.csv dataset. Evaluate the performance of classifier.	21		

4	a) Naïve Bayes Classification: Implement Naïve Bayes Classification technique using Social_Network_Ads.csv dataset. Evaluate the performance of classifier. b) Text Analysis (PBL): Find the confusion matrix to find restaurant review based of sentiment analysis of Natural Language processing. Use Resaurentreviews.tsv file for your study.	26 28	10/04/2023	
5	Comparative Study of various machine learning models (Newly added): Take the inbuilt data file: iris and perform classification on that data using various classification models – Decision Tree, K Nearest Neighbour and Support Vector Machine. Find the confusion matrix for all three models and evaluate them by finding their accuracy. Find the algorithm which performs best on the given data file, out of all these three models.	31	12/04/2023	
6	Install, configure and run Hadoop and HDFS and explore HDFS on Windows	36	17/04/2023	
7	Implement word count / frequency programs using MapReduce.	43	17/04/2023	
8	Implement an application that stores big data in Hbase / MongoDB and manipulate it using R / Python.	50	19/04/2023	

21/02/23

Practical 1

1a] Aim: clustering algorithm for unsupervised classification. Read data file grades-km input.csv. and apply kmeans clustering

Describe K means clustering

1] K means algorithm is clustering technique used to partition a collection of m object into k distinct cluster

Step 1: Choose the value of k and k initial guesses for centroids

step 2: Calculate distance from each data point to each centroid

step 3: Compute centroid of each cluster and update centroid accordingly

step 4: Repeat step 2 and 3 until algorithm converges ie until assignments of data points to cluster no longer change

Important function

1] wcss = vector(): This is used to create empty vector 'wcss'

2] for(i in 1:10) wcss[i] = sum (kmeans dataset)

- This is used to implement elbow method
- It runs loop from 1 to 10 and computes sum of squared distance of data

3] kmeans = kmeans (x = dataset, centers = 5): This is used to perform k means clustering with 5 clusters on 'dataset'

1B

Aim Implement aripri algorithm recommending grocery items to customer that most frequently bought together, given data set of transaction by customer of store using ariori algorithm

Describe ariori algorithm in detail

1] Aprion algorithm is an association rule mining algorithm used to find frequent itemsets in database

2] It generates candidates itemsets of size k and prunes those that do not meet minimum support threshold

3] It iteratively increses the size of itemsets based on apriori property of reduce search space

4] The output is set of frequent itemsets that can be used to generate association rules for market basket analysis recommendation system, cross selling statergies

10/3/23

Practical 2a

Aim : Regression model : Import data from web storage. Name dataset and do logistic regression to find out relation between variable

Describe logistic regression

1] Logistic regression is a statistical method used to analyze relationship beween a binary outcome variable and one or more predictor variables

2] It is type of regression analysis that models probablity of binary outcome as a function of predictor variable.

3] The logistic function is used to transform output to range between 0 & 1, representing propiablity of outcome

4] Logistic regression is commonly used in fields such as healthcare, finance and marketing to predict likelihood of an event occurring based on various input factors

Practical 2 b

Aim: Apply multiple regression if data have continous independent variable Apply on above dataset

— Explain multiple regression in detail

1] Multiple regression is statistical technique used to analyze relationship between a dependent variable and two or more independant variables

2] It extenas simple linear regression to include multiple predictors, allowing for the examination of how predictors jointly affect outcome

3] The goal is to estimate the strength and direction of relationship between the independent and dependent variable as well as to predict value of dependent variable based on values of independent variable

Practical 2 c

Aim : Design simple linear regression model using above dataset

1] Linear regression is a statistical method used to analyze relationship between a dependent variable and one independent variable

2] It assumes a linear relationship between variables, meaning that change in dependent variable

3] The goal of linear regression is to find best-fit line that represent relationship between variables

4] This line is determined by minimizing the sum of squared error between observed values and predicted values

5] The resulting equation can be used to predict dependent variable based on values of independent variables

17/3/23

Practical 3

30] Aim: implement Decision tree classification technique

* Describe Decision tree classification in detail

Decision tree is popular classification algorithm that recursively splits the data into smaller subsets based on most important features

Important function

1] rpart (formula, data) : This is function is used to fit decision tree model to the training set. The 'formula' argument is formula of form 'dependent variable ~ independent variables'

2] predict (object, newdata, type) : This function is used to predict the test set result. It takes the decision tree model fitted on training set as object

3] table (x, y) : This function creates a contigency table of x and y variables. It is used to evaluate accuracy of model

Practical 3B

Aim: Implement SVM classification technique

Explain SVM classification in detail

1] Support Vector machine (SVM) is common classification method that combines linear models with instance based learning techniques

2] SVM select small number of critical boundry instances called support vectors from each class and build linear decision function that seperates them widely as possible

3] SVM chooses the extreme points that help in creating the hyperplane These extreme cases are called a support vectors

4] Consider diagram in which there are two different categories that are classified using decision boundry

Practical 4 10/4/23

4a aim] Naive bayes classification : Implement Naive bayes classification technique using social network ads.csv dataset

* Describe Naive bayes classification in detail

Bayes Theorem

1] Bayes theorem gives the relationship between probabilities of C and A and conditional probabilities P(C|A) and P(A|C)

2] P(C|A) can be expressed using the formula given below

$$P(C|A) = \frac{P(A \cap C)}{P(A)}$$

$$P(C|A) = \frac{P(A|C) \cdot P(C)}{P(A)}$$

3] A more general form of bayes theorem assigns classified lable to object with multiple attribute

$$P(C_i|A) = \frac{P(a_1, a_2 \ldots a_m|C_i) \cdot P(C_i)}{P(a_1, a_2, a_m)}$$

Practical 4b

Aim : Find the confusion matrix to find restorent review based on sentiment analysis of Natural Language processing

Explain all stages of text analysis in short

1] Collect raw text Data science team monitors website and scrap pages

2] Represent text - Each review is converted into a suitable document representation with proper indices and corpus is built on these indicated reviews

3] Compute - The usefulness of each word in reviews is computed using methods such as TFIDF

4] Categorize document by topics :- Achived through topic models such as Latent Dirichilet allocation.

5] Determine sentiment of reviews :- Identify wheather reviews are positive or negative

6] Review the results and gain greater insights :- Marketing others result, analyze them reports findings using visualization techinques.

Practical 5 12/04/2023

Aim: Comparitive study of various machine learning algorithm and models: Take data file iris and perform classification on data using various classification models. Find the algorithm which performs best in given data file out of this 3 models

Comparitive study

1] In this practical, three different models are built on training set: Decision tree, k-Nearest neighbour, Support vector machine

2] The test set results are predicted for each model and accuracy of each mode is calculated using confusion metrix

3] Finally, the data.frame() function create a data frame called models with two coloumns:- Techinque and Accurracy-percentage

4] The c() function is used to create vectors for each coloumn. where the technique names and corrosponding accurracy percentage are listed

5] Finally models data frame is printed to console which looks like

Techinque	Accurracy percentage
Decision tree	88.8889
KNN	94.6444
SVM	94.6444

6) This provides comparision of accuracy of each model on test set allowing user to determine which model performed the best

Practical 6 17/4/23

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

1] Hadoop is an open source distributed computed framework that enables the processing of large data set across cluster of computers using simple programming models

2] It provides highly scalable and fault tolerent ~~level~~ environment for processing and storing large data sets

3] Hadoop distributed file system (HDFS) is distributed file system that stores data across multiple machine in a hadoop cluster

4] HDFS is highly ~~fau~~ fault tolerent and provides high throughput access to application data.

Practical 7 17/4/23

Aim: Implement word count / Frequency program using MapReduce

1] MapReduce is programming model used for processing and generating large data sets in parellel and distributed fashion

2] It divedes a large dataset into smaller chunkes, distributs them accross cluster, and processes them in parelled

3] The MapReduce process consists of two phases: The Map phase and The Reduce phase

4] In the Map phase, data is transformed in to key value pairs, and in reduced phase, this key value pairs are aggregated to generate final output

5] The wordcount program is a classic example of MapReduce that counts the occurances of words in large datasets

6] It involves mapping each word to key value pair and then reducing counts of some word

Practical 8 19/4/23

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

1] HBase is distributed NoSQL database that is built on top of the Hadoop distributed file system (HDFS)

2] It provides real time ~~and~~ read and write access to large dataset, and it is designed to store and manage structured data.

3] HBase is widely used for storing and processing large amount of data in a scalable and fault tolerent manner.

4] MongoDB is popoular NoSQL Database that uses a doccament oriented data model

5] It stores data in JSON-like documents and supports dynamic schema design

6] MongoDB provides high performance, scalability and availability, and it is often used for building morden web applications and managing large database

Practical No: 1 Ninad Karlekar 22306A1012

Practical 1 a) Date: 21/02/2023

K-Means Clustering:

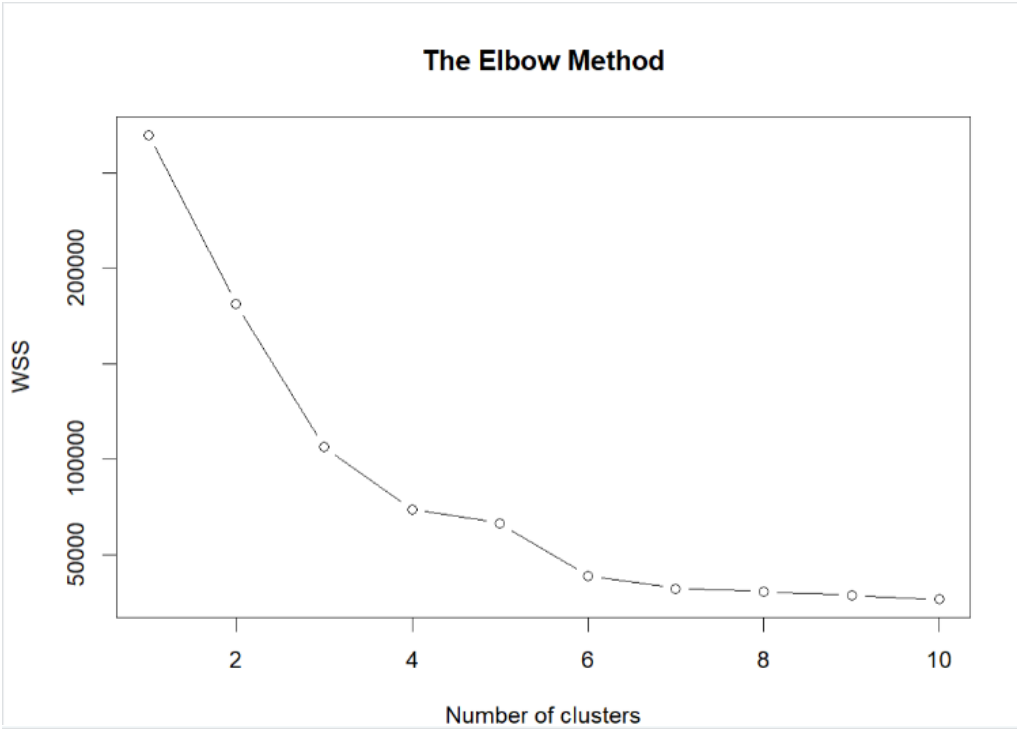
Aim: Clustering algorithms for unsupervised classification. Read a data file grades_km_input.csv and apply k-means clustering. Plot the cluster data using R visualizations.

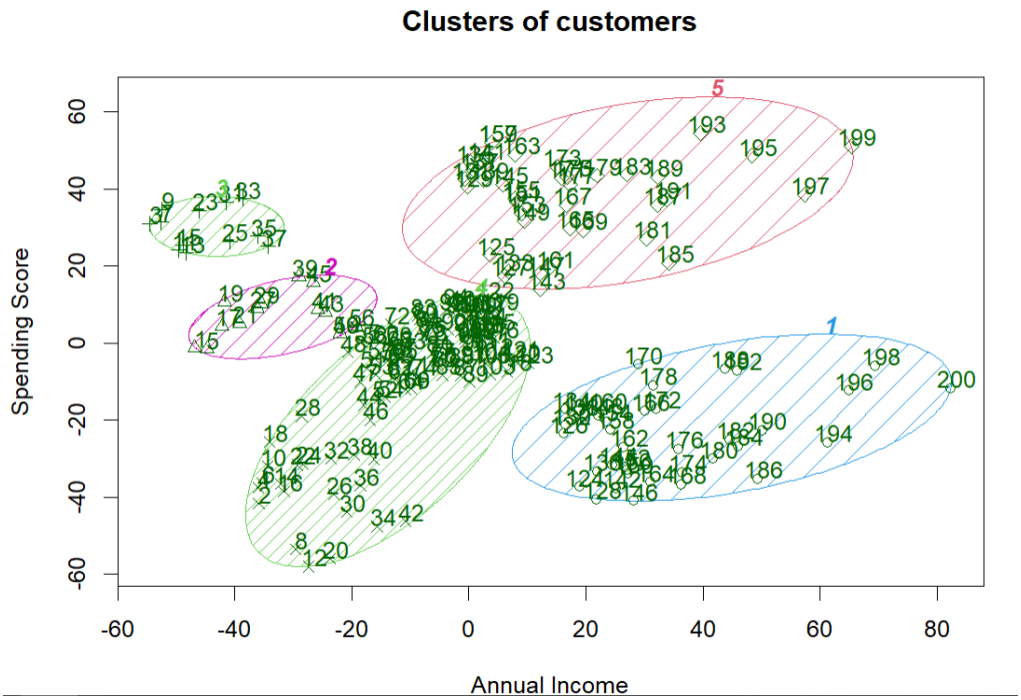
Describe K-Means Clustering algorithm in detail.

Code

```
# K-Means Clustering
# Importing the dataset
dataset <- read.csv('F:\\GitHub\\Practical_BscIT_MscIT_Ninad\\MscIT\\Semester 2\\BigDataAnalytics\\Dataset\\Mall_Customers.csv')
head(dataset)
dataset <- dataset[4:5]
head(dataset)
# Compute the Within Cluster Sum of Squares (WCSS) for different number of clusters
wcss <- vector()
for (i in 1:10) {
  wcss[i] <- sum(kmeans(dataset, i)$withinss)
}
# Plot the WCSS values
plot(1:10, wcss, type = 'b', main = paste('The Elbow Method'),
    xlab = 'Number of clusters', ylab = 'WSS')
# Fit K-Means to the dataset with 5 clusters
kmeans_model <- kmeans(x = dataset, centers = 5)
y_kmeans <- kmeans_model$cluster
# Visualize the clusters
library("cluster")
clusplot(dataset, y_kmeans, lines = 0, shade = TRUE, color = TRUE, labels = 2,
        main = paste('Clusters of customers'),
        xlab = "Annual Income",
        ylab = "Spending Score")
```

output


The Elbow Method
WSS
50000
100000
200000
2
4
6
8
10
Number of clusters


Clusters of customers
Spending Score
-60
-40
-20
0
20
40
60
Annual Income
-60
-40
-20
0
20
40
60
80
1
2
3
4
5

Practical 1 b) Date: 21/02/2023

Apriori Algorithm (PBL):

Aim: Implement Apriori Algorithm Recommending grocery items to a customer that is most frequently bought together, given a data set of transactions by customers of a store, using Ariory algorithm using Market_Basket_Optimisation.csv file.

Describe Apriori Algorithm in detail.

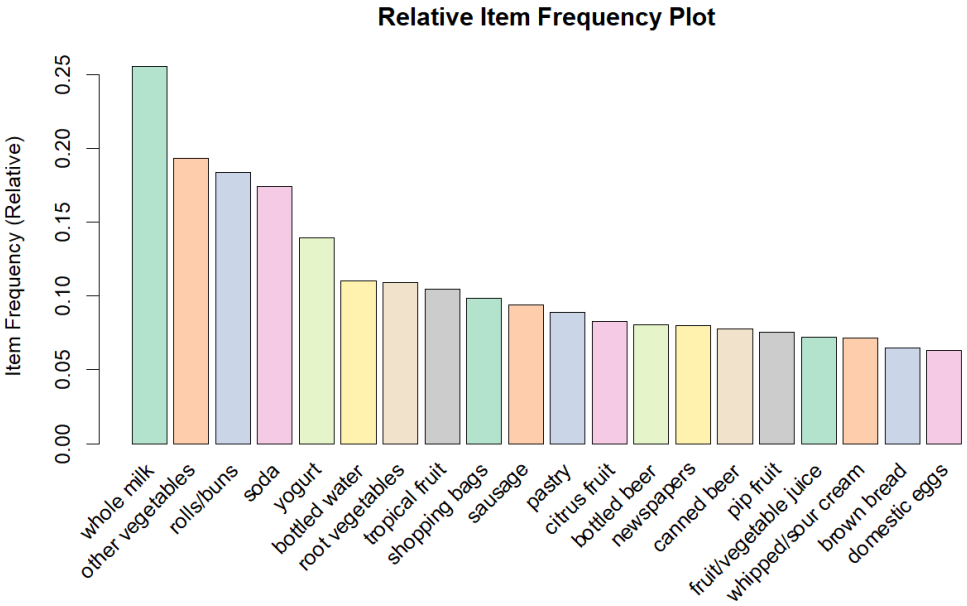
Code

```
install.packages("arules")
install.packages("arulesViz")
install.packages("RColorBrewer")
# Loading Libraries
library(arules)
library(arulesViz)
library(RColorBrewer)
# import dataset
data(Groceries)
Groceries
summary(Groceries)
class(Groceries)
# using apriori() function
rules = apriori(Groceries, parameter = list(supp = 0.02, conf = 0.2))
summary (rules)
# using inspect() function
inspect(rules[1:10])
# using itemFrequencyPlot() function
arules::itemFrequencyPlot(Groceries, topN = 20,
                          col = brewer.pal(8, 'Pastel2'),
                          main = 'Relative Item Frequency Plot',
                          type = "relative",
                          ylab = "Item Frequency (Relative)")
itemsets = apriori(Groceries, parameter = list(minlen=2, maxlen=2,support=0.02,
target="frequent itemsets"))
summary(itemsets)
# using inspect() function
inspect(itemsets[1:10])
itemsets_3 = apriori(Groceries, parameter = list(minlen=3, maxlen=3,support=0.02,
target="frequent itemsets"))
summary(itemsets_3)
# using inspect() function
inspect(itemsets_3)
```

output

```
> inspect(rules[1:10])
     lhs                    rhs                  support    confidence coverage   lift     count
[1]  {}                  => {whole milk}       0.25551601 0.2555160  1.00000000 1.0000000 2513
[2]  {frozen vegetables} => {whole milk}       0.02043721 0.4249471  0.04809354 1.6630940  201
[3]  {beef}              => {whole milk}       0.02125064 0.4050388  0.05246568 1.5851795  209
[4]  {curd}              => {whole milk}       0.02613116 0.4904580  0.05327911 1.9194805  257
[5]  {pork}              => {other vegetables} 0.02165735 0.3756614  0.05765125 1.9414764  213
[6]  {pork}              => {whole milk}       0.02216573 0.3844797  0.05765125 1.5047187  218
[7]  {frankfurter}       => {whole milk}       0.02053889 0.3482759  0.05897306 1.3630295  202
[8]  {bottled beer}      => {whole milk}       0.02043721 0.2537879  0.08052872 0.9932367  201
[9]  {brown bread}       => {whole milk}       0.02521607 0.3887147  0.06487036 1.5212930  248
[10] {margarine}         => {whole milk}       0.02419929 0.4131944  0.05856634 1.6170980  238
```

```
> inspect(itemsets_3)
     items                                                 support    count
[1] {root vegetables, other vegetables, whole milk} 0.02318251 228
[2] {other vegetables, whole milk, yogurt}          0.02226741 219
>
```



Practical No: 2A Ninad Karlekar 22306A1012 BDA Date: 14/03/2023

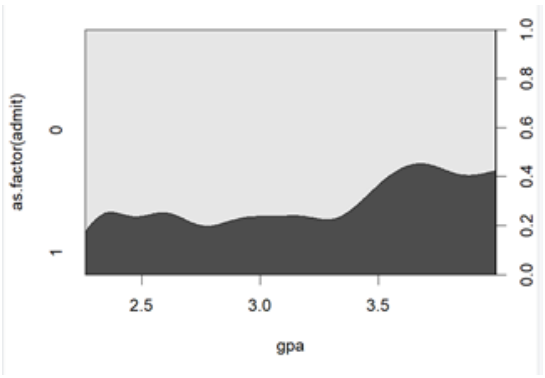
Logistic Regression

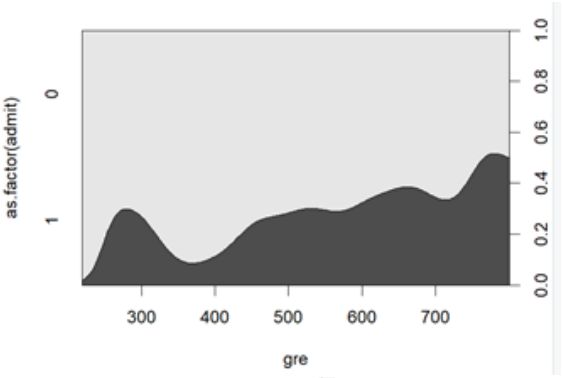
Aim: Regression Model: Import data from web storage. Name the dataset and do Logistic Regression to find out relation between variables that are affecting the admission of a student in an institute based on his or her GRE score, GPA obtained and rank of the student. Also check the model is fit or not
Describe Logistic Regression in detail.
Code

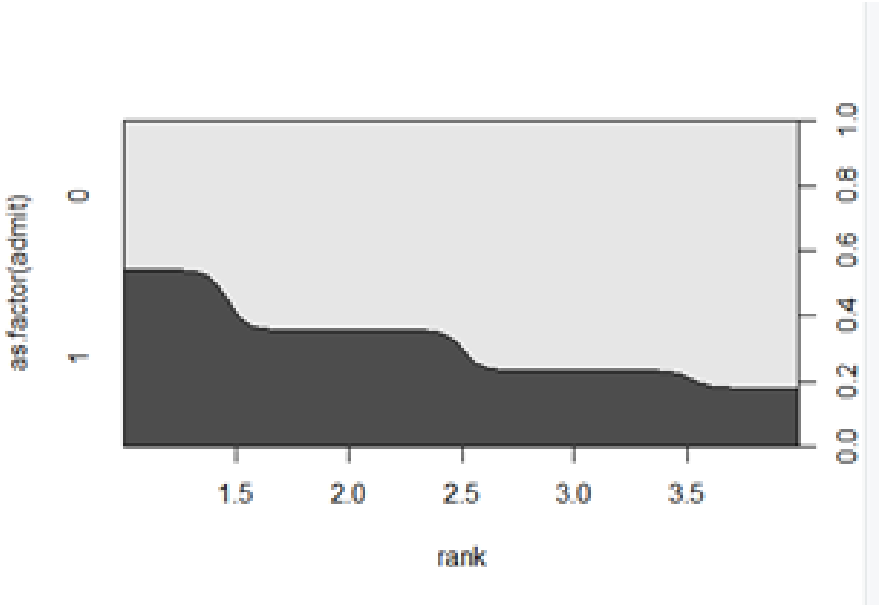
```
college <-
read.csv("https://raw.githubusercontent.com/ropensci/datapack/main/inst/extdata/pkg-example/binary.csv")
head(college)
nrow(college)
install.packages("caTools")
library(caTools)
split <- sample.split(college, SplitRatio = 0.75)
split
training_reg <- subset(college, split == "TRUE")
test_reg <- subset(college, split == "FALSE")
fit_logistic_model <- glm(admit ~ .,data = training_reg,family = "binomial")
coef(fit)["gre"]
coef(fit)["gpa"]
coef(fit)["rank"]
predict_reg <- predict(fit_logistic_model,test_reg, type = "response")
predict_reg
cdplot(as.factor(admit)~ gpa, data=college)
cdplot(as.factor(admit)~ gre, data=college)
cdplot(as.factor(admit)~ rank, data=college)
predict_reg <- ifelse(predict_reg >0.5,1,0)
predict_reg
table(test_reg$admit, predict_reg)
```

Output

```
>
> table(test_reg$admit, predict_reg)
   predict_reg
     0  1
  0 70  2
  1 21  7
```


as.factor(admit)
0
1
0.0
0.2
0.4
0.6
0.8
1.0
2.5
3.0
3.5
gpa


as.factor(admit)
0
1
0.0
0.2
0.4
0.6
0.8
1.0
300
400
500
600
700
gre


as.factor(admit)
0
1
0.0
0.2
0.4
0.6
0.8
1.0
1.5
2.0
2.5
3.0
3.5
rank

Practical 2 b) Date: 14/03/2023

Multiple Regression

MULTIPLE REGRESSION MODEL: Apply multiple regressions, if data have a continuous independent variable. Apply on above dataset.

Explain Multiple regression in detail.

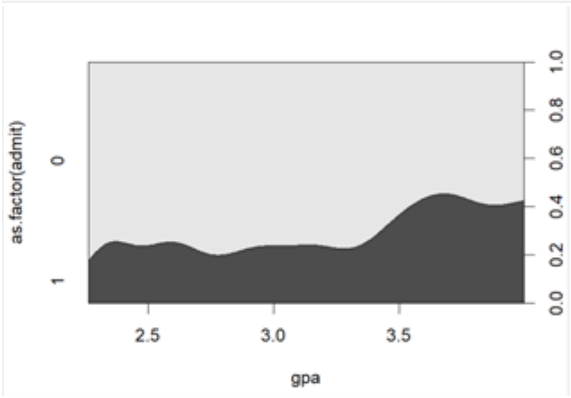
Code

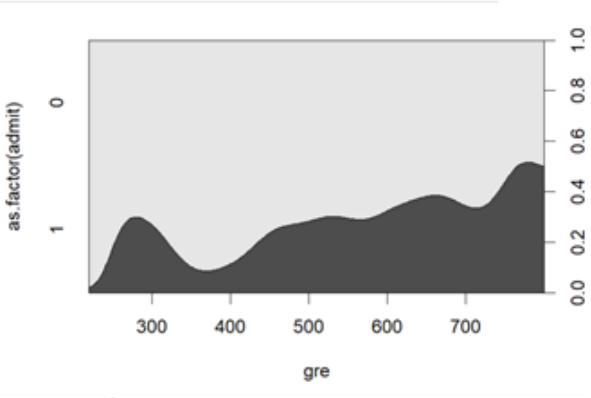
```
college <- read.csv("https://raw.githubusercontent.com/csquared/udacity-dlnd/master/nn/binary.csv")
head(college)
nrow(college)
install.packages("caTools")
library(caTools)
split <- sample.split(college, SplitRatio = 0.75)
split
training_reg <- subset(college, split == "TRUE")
test_reg <- subset(college, split == "FALSE")
fit_MRegressor_model <- lm(formula = admit ~ gre+gpa+rank, data = training_reg)
predict_reg <- predict(fit_MRegressor_model,newdata = test_reg)
predict_reg
cdplot(as.factor(admit)~ gpa, data=college)
cdplot(as.factor(admit)~ gre, data=college)
cdplot(as.factor(admit)~ rank, data=college)
predict_reg <- ifelse(predict_reg >0.5,1,0)
predict_reg
table(test_reg$admit, predict_reg)
```

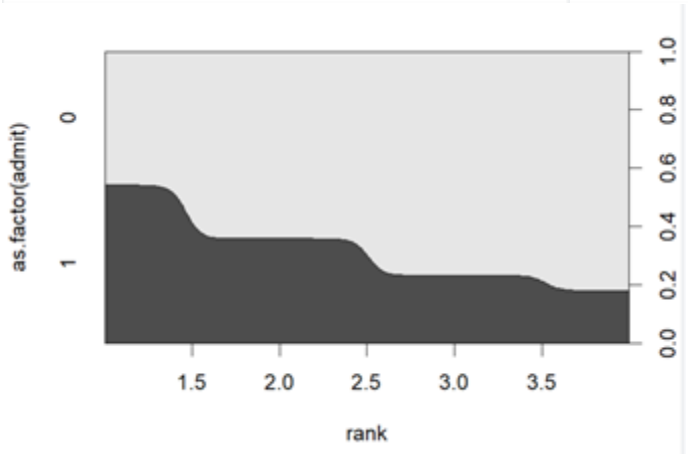
Output

```
n/binary.csv")
> head(college)
  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
> nrow(college)
[1] 400
>
```

```
> predict_reg
  1   5   9  13  17  21  25  29  33  37  41  45  49  53  57  61  65  69  73  77  81  85  89  93  97 101
  0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   1   1   0   0
105 109 113 117 121 125 129 133 137 141 145 149 153 157 161 165 169 173 177 181 185 189 193 197 201 205
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1
209 213 217 221 225 229 233 237 241 245 249 253 257 261 265 269 273 277 281 285 289 293 297 301 305 309
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
313 317 321 325 329 333 337 341 345 349 353 357 361 365 369 373 377 381 385 389 393 397
  0   0   0   0   0   0   0   0   0   0   0   1   0   1   0   0   0   0   0   0   0   0
> table(test_reg$admit, predict_reg)
   predict_reg
      0  1
  0 70  4
  1 23  3
>
```







Take Home Task

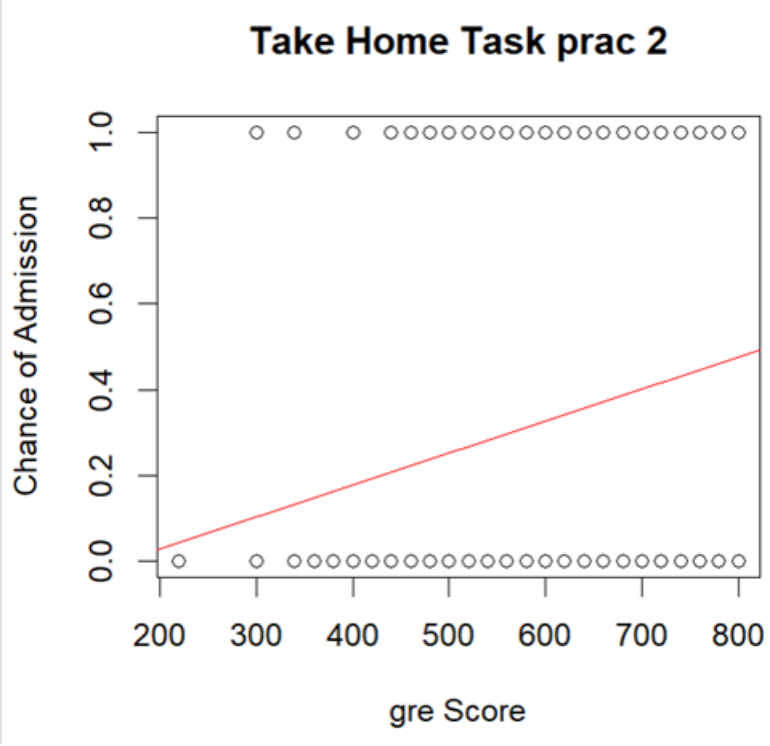
Design a Simple Linear Regression Model using the above dataset. (HINT: consider GRE Score or GPA score as independent variable.

CODE:

```
# Load the dataset
data <- read.csv("https://raw.githubusercontent.com/csquared/udacity-dlnd/master/nn/binary.csv")

# Plot the relationship between gre and chance of admission
plot(data$gre, data$admit, xlab = "gre Score", ylab = "Chance of Admission", main = "Take Home Task prac 2" )

# Fit a simple linear regression model
model <- lm(admit ~ gre, data = data)

 # Print the summary of the model
summary(model)

# Plot the regression line
abline(model, col = "red")

 # Make a prediction using the model
new_data <- data.frame(gre = 3.5)
prediction <- predict(model, newdata = new_data)
prediction
```

OUTPUT:

```
> prediction <- p
> prediction
          1
-0.1172362
```



Practical No: 3 Ninad Karlekar 22306A1012

Practical 3 a) Date: 17/03/2023

Decision Tree

Aim: Implement Decision Tree classification technique.

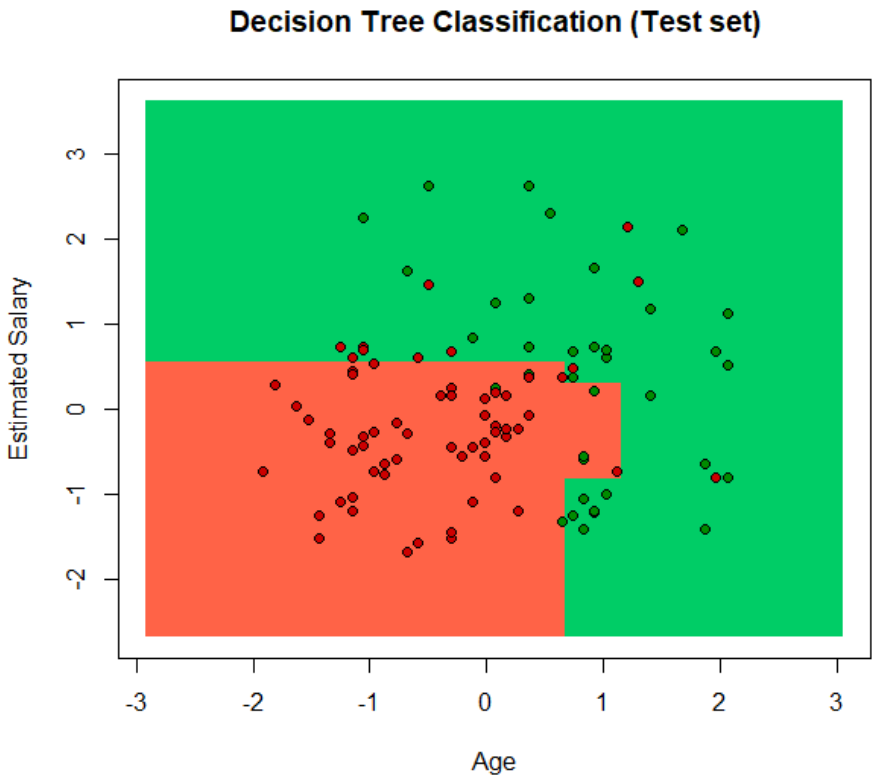
Describe Decision Tree classification in detail.

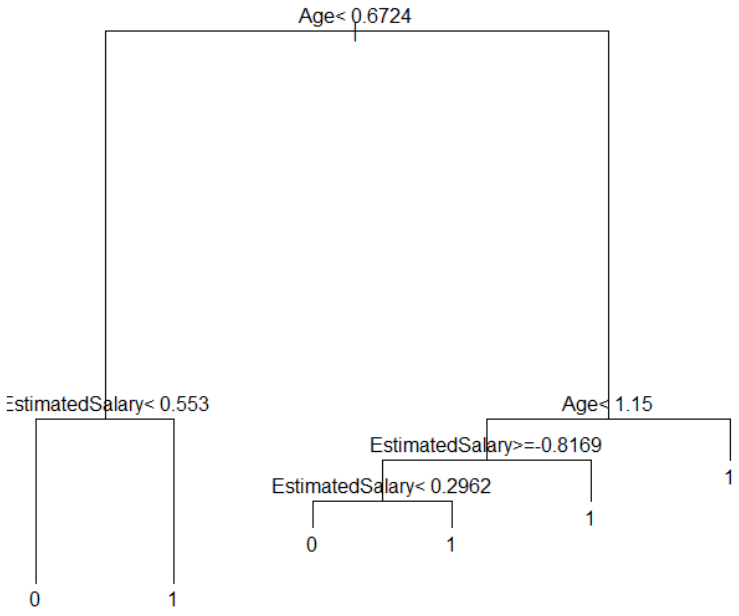
Code

```
# Decision Tree Classification
dataset = read.csv('D:\\nk\\OneDrive_1_3-17-2023\\Social_Network_Ads.csv')
dataset = dataset[3:5]
print(dataset)
# Encoding the target feature as factor
dataset$Purchased = factor(dataset$Purchased, levels = c(0, 1))
# Splitting the dataset into the Training set and Test set
install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset$Purchased, SplitRatio = 0.75)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)# Feature Scaling
training_set[-3] = scale(training_set[-3])
test_set[-3] = scale(test_set[-3])
#install.packages('rpart')
library(rpart)
classifier = rpart(formula = Purchased ~ .,
                   data = training_set)
y_pred = predict(classifier, newdata = test_set[-3], type = 'class')
# Making the Confusion Matrix
cm = table(test_set[, 3], y_pred)
# Visualising the Training set results
install.packages("ElemStatLearn")
library(ElemStatLearn)
set = training_set
X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)
X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)
grid_set = expand.grid(X1, X2)
colnames(grid_set) = c('Age', 'EstimatedSalary')
y_grid = predict(classifier, newdata = grid_set, type = 'class')
plot(set[, -3],
     main = 'Decision Tree Classification (Training set)',
     xlab = 'Age', ylab = 'Estimated Salary',
```

```
    xlim = range(X1), ylim = range(X2))
contour(X1, X2, matrix(as.numeric(y_grid), length(X1), length(X2)), add = TRUE)
points(grid_set, pch = '.', col = ifelse(y_grid == 1, 'springgreen3', 'tomato'))
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))
library(ElemStatLearn)
set = test_set
X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)
X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)
grid_set = expand.grid(X1, X2)
colnames(grid_set) = c('Age', 'EstimatedSalary')
y_grid = predict(classifier, newdata = grid_set, type = 'class')
plot(set[, -3], main = 'Decision Tree Classification (Test set)',
    xlab = 'Age', ylab = 'Estimated Salary',
    xlim = range(X1), ylim = range(X2))
contour(X1, X2, matrix(as.numeric(y_grid), length(X1), length(X2)), add = TRUE)
points(grid_set, pch = '.', col = ifelse(y_grid == 1, 'springgreen3', 'tomato'))
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))
plot(classifier)
text(classifier)
```

Output


Decision Tree Classification (Test set)
Estimated Salary
Age
-3
-2
-1
0
1
2
3


Age< 0.6724
EstimatedSalary< 0.553
Age< 1.15
EstimatedSalary>=-0.8169
EstimatedSalary< 0.2962
0
1
1
1
0
1

Practical 3 b) **Date: 17/03/2023**

SVM Classification:

Implement SVM Classification technique

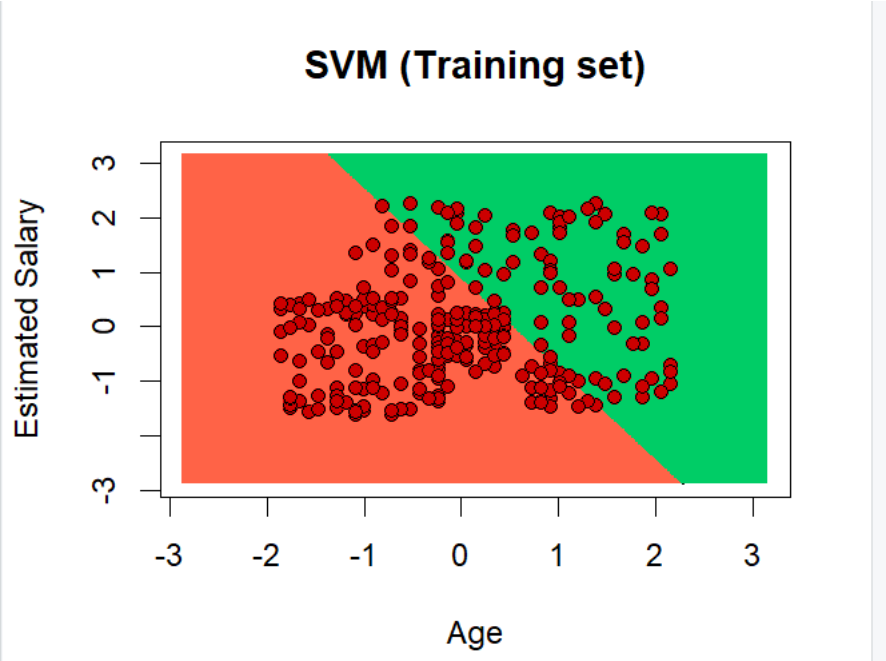
Explain SVM classification in detail.

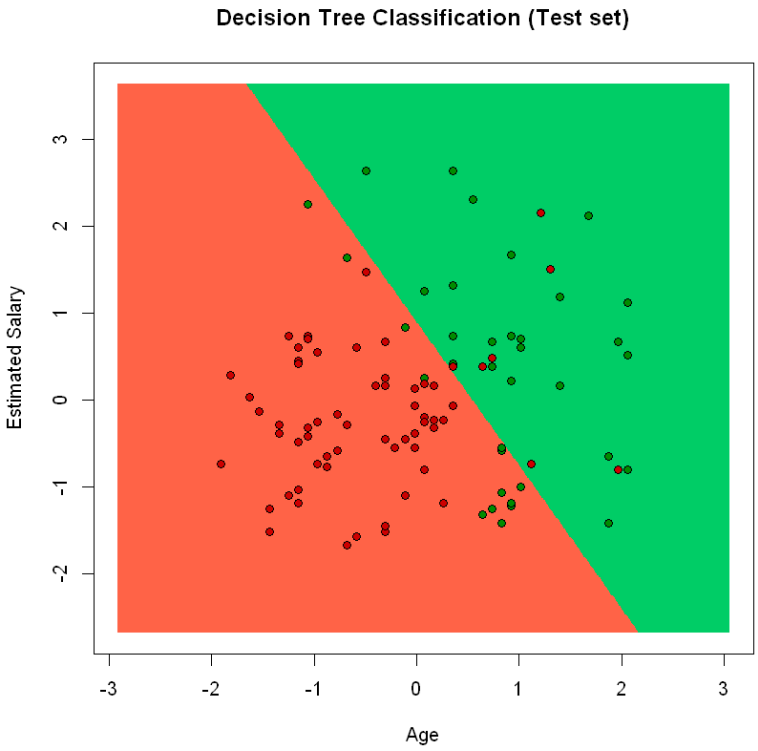
Code

```
# Support vector machine
# Importing the dataset
dataset = read.csv('D:\\nk\\OneDrive_1_3-17-2023\\Social_Network_Ads.csv')
dataset = dataset[3:5]
print(dataset)
# Encoding the target feature as factor
dataset$Purchased = factor(dataset$Purchased, levels = c(0, 1))
# Splitting the dataset into the Training set and Test set
install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset$Purchased, SplitRatio = 0.75)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)# Feature Scaling
training_set[-3] = scale(training_set[-3])
test_set[-3] = scale(test_set[-3])
# Fitting SVM
install.packages('e1071')
library(e1071)
classifier = svm(formula = Purchased ~ .,
                 data = training_set,
                 type = 'C-classification'
                 kernel = 'linear')
print(classifier)
# Predicting the Test set results
y_pred = predict(classifier, newdata = test_set[-3])
# Making the Confusion Matrix
cm = table(test_set[, 3], y_pred)
# Visualising the Training set results
install.packages("ElemStatLearn")
library(ElemStatLearn)
set = training_set
X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)
```

```
X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)
grid_set = expand.grid(X1, X2)
colnames(grid_set) = c('Age', 'EstimatedSalary')
y_grid = predict(classifier, newdata = grid_set, type = 'class')
plot(set[, -3],
     main = 'SVM (Training set)',
     xlab = 'Age', ylab = 'Estimated Salary',
     xlim = range(X1), ylim = range(X2))
contour(X1, X2, matrix(as.numeric(y_grid), length(X1), length(X2)), add = TRUE)
points(grid_set, pch = '.', col = ifelse(y_grid == 1, 'springgreen3', 'tomato'))
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))
# Visualising the Test set results
library(ElemStatLearn)
set = test_set
X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)
X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)
grid_set = expand.grid(X1, X2)
colnames(grid_set) = c('Age', 'EstimatedSalary')
y_grid = predict(classifier, newdata = grid_set, type = 'class')
plot(set[, -3], main = 'Decision Tree Classification (Test set)',
     xlab = 'Age', ylab = 'Estimated Salary',
     xlim = range(X1), ylim = range(X2))
contour(X1, X2, matrix(as.numeric(y_grid), length(X1), length(X2)), add = TRUE)
points(grid_set, pch = '.', col = ifelse(y_grid == 1, 'springgreen3', 'tomato'))
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))
# Plotting the tree
#plot(classifier)
#text(classifier)
```

Output


SVM (Training set)
Estimated Salary
Age
-3
-2
-1
0
1
2
3


Decision Tree Classification (Test set)
Estimated Salary
Age
-3
-2
-1
0
1
2
3

Practical No: 4

Practical 4 a) Date: 10/04/2023

Naïve Bayes Classification:

Aim: Naïve Bayes Classification: Implement Naïve Bayes Classification technique using Social_Network_Ads.csv dataset. Evaluate the performance of classifier.

Describe Naïve Bayes Classification in detail.

Code

```
# Naive Bayes
# Importing the dataset
dataset <- read.csv("F:\\GitHub\\Practical_BscIT_MscIT_Ninad\\MscIT\\Semester
2\\BigDataAnalytics\\Dataset\\Social_Network_Ads.csv")
dataset <- dataset[3:5]
head(dataset)
# Encoding the target feature as factor
dataset$Purchased <- factor(dataset$Purchased, levels = c(0, 1))
# Splitting the dataset into the Training set and Test set
library(caTools)
set.seed(123)
split <- sample.split(dataset$Purchased, SplitRatio = 0.75)
training_set <- subset(dataset, split == TRUE)
test_set <- subset(dataset, split == FALSE)
# Feature Scaling
training_set[-3] <- scale(training_set[-3])
test_set[-3] <- scale(test_set[-3])
# Fitting Naive Bayes to the Training set
library(e1071)
classifier <- naiveBayes(x = training_set[-3], y = training_set$Purchased)
# Predicting the Test set results
y_pred <- predict(classifier, newdata = test_set[-3])
# Making the Confusion Matrix
cm <- table(test_set[, 3], y_pred)
print(cm)
```

output

```
> head(dataset)
  Age EstimatedSalary Purchased
1  19           19000         0
2  35           20000         0
3  26           43000         0
4  27           57000         0
5  19           76000         0
6  27           58000         0
>
> cm <- table(t
> print(cm)
   y_pred
     0  1
  0 57  7
  1  7 29
>
```

Practical 4 b) Date: 10/04/2023

Text Analysis (PBL):

Aim: Find the confusion matrix to find restaurant review based of sentiment analysis of Natural Language processing. Use Resaurentreviews.tsv file for your study.

Explain all stages of Text Analysis in short.

Code

```
# Read in the data
dataset_original <-
read.delim("F:\\GitHub\\Practical_BscIT_MscIT_Ninad\\MscIT\\Semester
2\\BigDataAnalytics\\Dataset\\Restaurant_Reviews.tsv", quote = "", stringsAsFactors =
FALSE)
head(dataset_original)
# Install and load required packages
install.packages('tm')
install.packages('SnowballC')
install.packages('randomForest')
library(tm)
library(SnowballC)
library(caTools)
library(randomForest)
# Create a corpus
corpus <- VCorpus(VectorSource(dataset_original$Review))
corpus <- tm_map(corpus, content_transformer(tolower))
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, removeWords, stopwords())
corpus <- tm_map(corpus, stemDocument)
corpus <- tm_map(corpus, stripWhitespace)
# Create a document term matrix
dtm <- DocumentTermMatrix(corpus)
dtm <- removeSparseTerms(dtm, 0.999)
# Convert the dtm to a data frame
dataset <- as.data.frame(as.matrix(dtm))
dataset$Liked <- dataset_original$Liked
dataset$Liked <- factor(dataset$Liked, levels = c(0,1))
# Split the data into training and test sets
set.seed(123)
split <- sample.split(dataset$Liked, SplitRatio = 0.8)
training_set <- subset(dataset, split == TRUE)
test_set <- subset(dataset, split == FALSE)
# Train a random forest classifier
classifier <- randomForest(x = training_set[-692], y = training_set$Liked, ntree = 10)
```

# Make predictions on the test set and create a confusion matrix
y_pred <- predict(classifier, newdata = test_set[-692])
cm <- table(test_set[,692], y_pred)
print(cm)

output

```
> print(cm)
   y_pred
      0  1
   0 82 18
   1 23 77
```

```
  [1] 1 0 0 1 1 0 0 0 1 1 1 0 0 1 0 0 1 0 0 0 0 1 1 1 1 1 0 1 0 0 1 0 1 0 1 1 1
 [38] 0 1 0 1 0 0 1 0 1 0 1 1 1 1 1 1 0 1 1 0 0 1 0 0 1 1 1 1 1 1 1 0 1 1 1 0 0
 [75] 0 0 0 1 1 0 0 0 0 1 0 1 0 1 1 1 0 1 0 1 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0
[112] 0 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 1 0 1 0 1 1 0 1 1 1 1 0 1 0 0 0 0 1 1 0 0
[149] 0 0 1 1 0 0 1 1 1 1 1 0 0 1 1 0 1 1 1 0 0 1 0 1 1 1 1 0 0 1 1 0 0 0 0 0 1
[186] 1 0 1 1 1 1 1 1 0 1 0 1 0 0 1 1 1 1 0 1 1 1 0 0 0 1 0 0 1 0 1 1 0 1 0 1 0 0
[223] 0 0 0 1 1 1 0 1 1 0 1 0 1 0 0 1 0 1 0 1 0 0 0 0 1 1 1 0 1 0 1 0 1 1 1 1 0 1
[260] 0 1 0 1 1 1 1 0 1 1 0 1 1 1 1 1 1 0 1 1 0 0 1 0 0 0 1 1 0 0 1 0 0 0 1 0 1 1
[297] 0 1 0 1 1 0 0 0 1 0 0 0 1 1 1 0 1 0 1 0 0 1 1 1 0 0 1 1 1 1 1 1 1 0 0 0 1 1
[334] 0 1 1 0 0 1 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 1 1 0 1 1 1 0 1 1 0 1 0 0 1 1 1
[371] 0 0 1 1 0 1 0 1 0 0 0 1 1 0 0 0 1 0 0 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 1 1 0 1
[408] 1 1 0 1 1 0 1 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 0 1 1 1 0 1 0 1 1 1 1 1 0 1 1 0
[445] 1 1 0 0 1 1 0 1 0 0 0 0 1 1 1 1 0 1 1 0 1 1 0 0 1 1 1 1 0 1 0 0 0 1 1 1 1 0
[482] 1 0 0 1 1 1 0 0 1 1 1 0 1 1 0 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0 1 1 1 1 1 1 0 1
[519] 0 1 1 1 0 0 1 0 0 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1 0 1 0 0 1 1 0 0 1 1 1 1 1 0 0
[556] 0 0 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 0 1 1 0 0 1 1 1 0 0 0 1 0 1 1 0 1 0 1 1
[593] 1 0 0 1 0 1 1 0 1 0 1 1 1 1 1 1 0 1 0 0 0 0 0 1 1 0 0 1 0 0 0 1 0 1 1 0 0 0
[630] 1 1 1 1 0 1 0 1 0 1 0 1 1 0 1 0 0 0 0 0 1 0 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1
[667] 1 1 1 1 0 0 1 0 1 1 1 1 1 0 1 0 1 0 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0 0 1
[704] 1 1 0 1 0 1 0 1 0 1 0 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 0 0 1 1 1 0 0 1 1 1
[741] 1 0 0 0 0 1 1 1 0 1 1 1 1 1 1 0 1 0 1 1 0 1 0 0 0 1 0 1 1 1 1 0 1 0 0 1 0 1
[778] 0 0 0 1 1 1 0 0 1 0 1 1 1 1 1 0 0 1 0 1 1 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0 0 0 0
[815] 1 1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 1 1 0 0 0 0 1 0 0 1
[852] 0 1 1 0 0 1 1 0 0 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 0 0 0 0 0
[889] 1 0 1 1 0 0 1 1 1 1 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
[926] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[963] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[1000] 0
Levels: 0 1
```

Practical No: 5 Ninad Karlekar 22306A1012

Date: 12/04/2023

Comparative study

Aim: Comparative Study of various machine learning models (Newly added): Take the inbuilt data file: iris and perform classification on that data using various classification models – Decision Tree, K Nearest Neighbour and Support Vector Machine. Find the confusion matrix for all three models and evaluate them by finding their accuracy. Find the algorithm which performs best on the given data file, out of all these three models.

Code

```
install.packages('rpart')
 install.packages('rpart.plot')
 install.packages('gmodels')
 install.packages('e1071')
 library(rpart)
 library(rpart.plot)
 library(gmodels)
 library(e1071)
data(iris)
 summary(iris)
 #normalize the continuous variables before performing any analysis on the dataset
 temp= as.data.frame(scale(iris[,1:4]))
 temp$Species = iris$Species # levels: setosa versicolor virginica
 summary(temp)
 # Splitting the dataset into the Training set and Test set
 install.packages('caTools')
 library(caTools)
 set.seed(123)
 split = sample.split(temp$Species, SplitRatio = 0.75)
 train = subset(temp, split == TRUE)
 test = subset(temp, split == FALSE)
 nrow(train)
 nrow(test)
 #1. Decision Trees
 dt_classifier = rpart(formula = Species ~ ., data = train)
# Predicting the Test set results
 dt_y_pred = predict(dt_classifier, newdata = test, type = 'class')
 print(dt_y_pred)
# Making the Confusion Matrix for Decision Tree
 cm = table(test$Species, dt_y_pred)
 print(cm)
```

```
#accuracy of DT model
 DTaccu = ((12+9+11)/nrow(test))*100 #true positive nos of 3*3 confusion matrix
 DTaccu
#2. k-Nearest Neighbours
 install.packages('class')
 library(class)
cl = train$Species
 set.seed(1234)
 knn_y_pred = knn(train[,1:4],test[,1:4],cl,k=5)
# cm of k-Nearest Neighbours
 cm = table(test$Species, knn_y_pred)
 print(cm)
#accuracy of KNN model
 KNNaccu = ((12+11+11)/nrow(test))*100 #true positive nos of 3*3 confusion matrix
 KNNaccu
#3. Support Vector Machine(SVM)
svmclassifier = svm(Species ~ . ,data = train)
 svm_y_pred = predict(svmclassifier,newdata = test)
cm = table(test$Species, svm_y_pred)
print(cm)
#accuracy of SVM model
SVMaccu = ((12+11+11)/nrow(test))*100
SVMaccuwhich(dt_y_pred != knn_y_pred)
which(dt_y_pred != svm_y_pred)
#svm vs kNN
which(svm_y_pred != knn_y_pred) #both are equal
#comparison of the accuracy of different models on testing dataset
models = data.frame(Technique = c("Decision Tree","KNN","SVM"),Accuracy_Percentage =
c(DTaccu,KNNaccu,SVMaccu))
 models
print("Hence KNN and SVM are better than decision tree")
```

output

```
> # Making the Confusion Matrix for Decision Tree
> cm = table(test$Species, dt_y_pred)
> print(cm)
            dt_y_pred
             setosa versicolor virginica
  setosa         12          0         0
  versicolor      0          9         3
  virginica       0          1        11
> #accuracy of DT model
> DTaccu = ((12+9+11)/nrow(test))*100 #true positive nos of 3*3 confusion matrix
> DTaccu
[1] 88.88889
> #2. k-Nearest Neighbours
> install.packages("class")
Error in install.packages : Updating loaded packages
> library(class)
> install.packages("class")
```

```
>
> cm = table(test$Species, svm_y_pred)
> print(cm)
            svm_y_pred
             setosa versicolor virginica
  setosa         12          0         0
  versicolor      0         11         1
  virginica       0          1        11
> #accuracy of SVM model
> SVMaccu = ((12+11+11)/nrow(test))*100
> SVMaccu
[1] 94.44444
```

```
> print(cm)
            knn_y_pred
             setosa versicolor virginica
  setosa         12          0         0
  versicolor      0         11         1
  virginica       0          1        11
> #accuracy of SVM model
> SVMaccu = ((12+11+11)/nrow(test))*100
> SVMaccu
[1] 94.44444
> which(dt_y_pred != knn_y_pred)
[1] 13 19
>
> which(dt_y_pred != knn_y_pred)
[1] 13 19
>
>
> which(dt_y_pred != svm_y_pred)
[1] 13 19
> which(svm_y_pred != knn_y_pred) #both are equal
integer(0)
```

```
> models = data.frame(Technique = c("Decision Tree","KNN","SVM"),Accuracy
> models
      Technique Accuracy_Percentage
1 Decision Tree             88.88889
2           KNN             94.44444
3           SVM             94.44444
> print("Hence KNN and SVM are better than decision tree")
[1] "Hence KNN and SVM are better than decision tree"
>
```

Practical No: 6

Date: 17/04/2023

Hadoop Installation:

Aim: Install, configure e4tr `and run Hadoop and HDFS and explore HDFS on Windows

Steps:

Steps to Install Hadoop

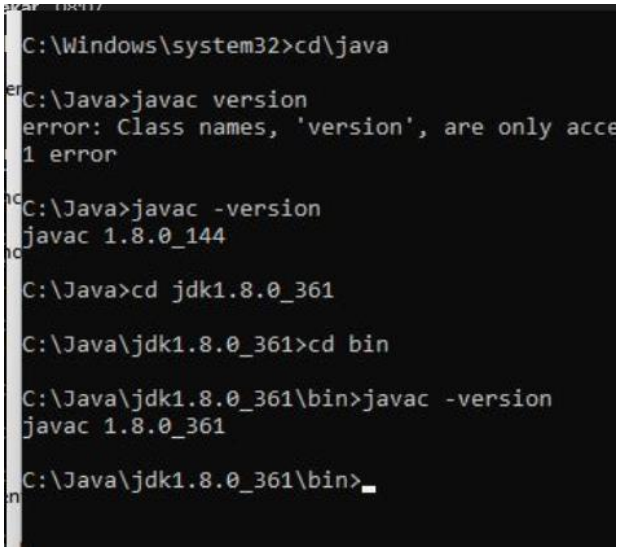
1. Install Java JDK 1.8
2. Download Hadoop and extract and place under C drive
3. Set Path in Environment Variables
4. Config files under Hadoop directory
5. Create folder datanode and namenode under data directory
6. Edit HDFS and YARN files
7. Set Java Home environment in Hadoop environment
8. Setup Complete. Test by executing start-all.cmd

There are two ways to install Hadoop, i.e.

9. Single node
10. Multi node

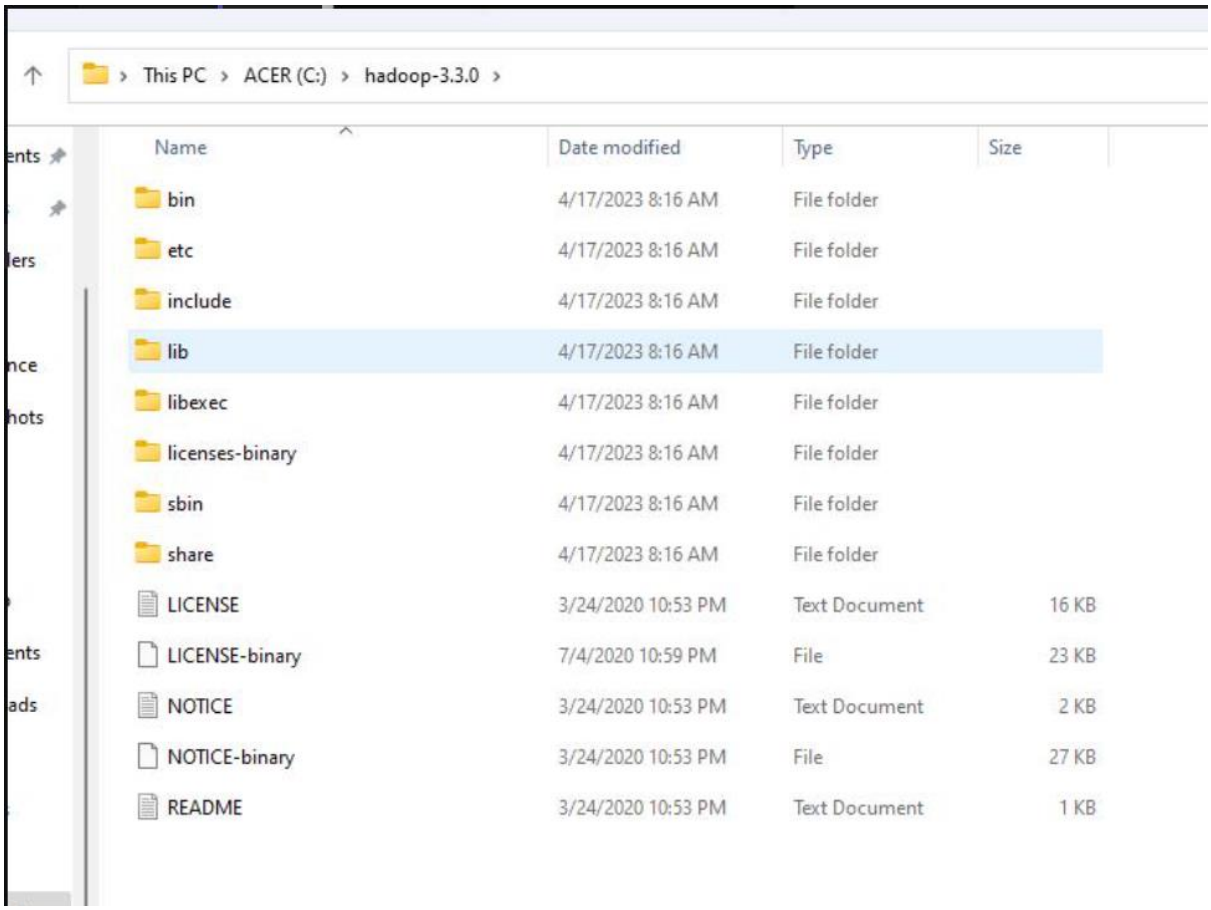
Here, we use multi node cluster.

1. **Install Java**
   - – Java JDK Link to download
     - https://www.oracle.com/java/technologies/javase-jdk8-downloads.html
   - – extract and install Java in C:\Java
   - – open cmd and type -> javac -version



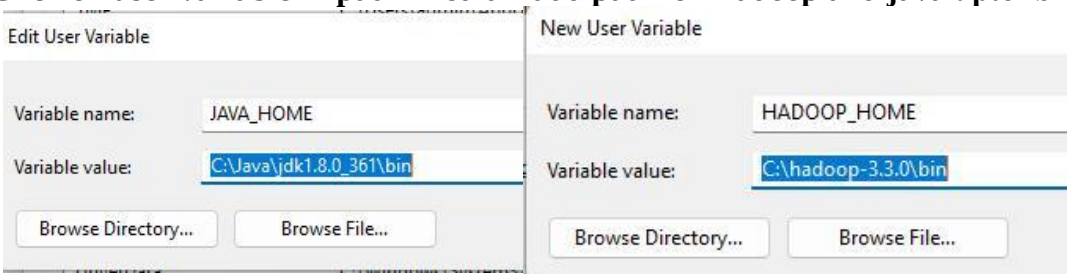
2. **Download Hadoop**
   https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz

- right click .rar.gz file -> show more options -> 7-zip->and extract to C:\Hadoop-3.3.0\



3. Set the path JAVA_HOME Environment variable

4. Set the path HADOOP_HOME Environment variable
Click on New to both user variables and system variables.
Click on user variable -> path -> edit-> add path for Hadoop and java upto 'bin'



Click Ok, Ok, Ok.

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>
  <property>
                <name>yarn.resourcemanager.address</name>
                <value>127.0.0.1:8032</value>
  </property>
```

```
  <property>
                <name>yarn.resourcemanager.scheduler.address</name>
                <value>127.0.0.1:8030</value>
   </property>
   <property>
                <name>yarn.resourcemanager.resource-tracker.address</name>
                <value>127.0.0.1:8031</value>
    </property>
</configuration>
```

==========

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

Find "JAVA_HOME=%JAVA_HOME%" and replace it as
set JAVA_HOME="C:\Java\jdk1.8.0_361"

==========

7. **Download "redistributable" package**

Download and run VC_redist.x64.exe

8. **Hadoop Configurations**

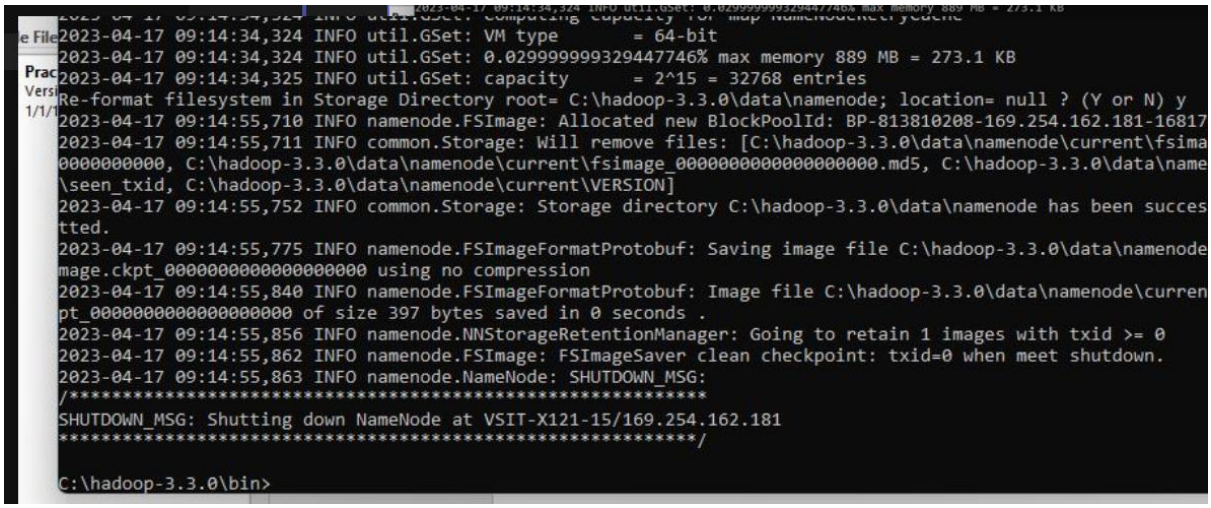
Download bin folder from https://github.com/s911415/apache-hadoop-3.1.0-winutils

– Copy the bin folder to c:\hadoop-3.3.0. Replace the existing bin folder.

9. **copy "hadoop-yarn-server-timelineservice-3.0.3.jar" from ~\hadoop-3.0.3\share\hadoop\yarn\timelineservice to ~\hadoop-3.0.3\share\hadoop\yarn folder.**

10. **Format the NameNode**

– Open cmd 'Run as Administrator' and type command "hdfs namenode –format"



11. Testing

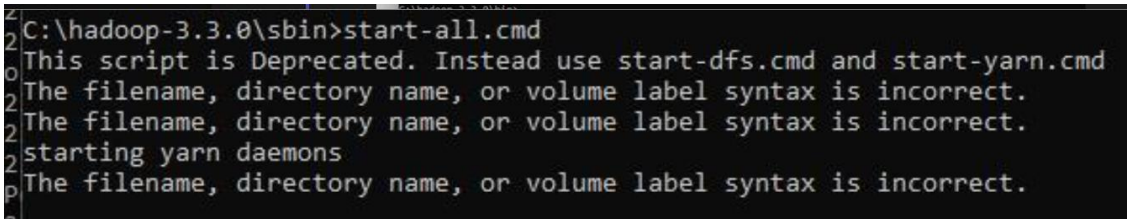
– Open cmd 'Run as Administrator' and change directory to C:\Hadoop-3.3.0\sbin

– type start-all.cmd

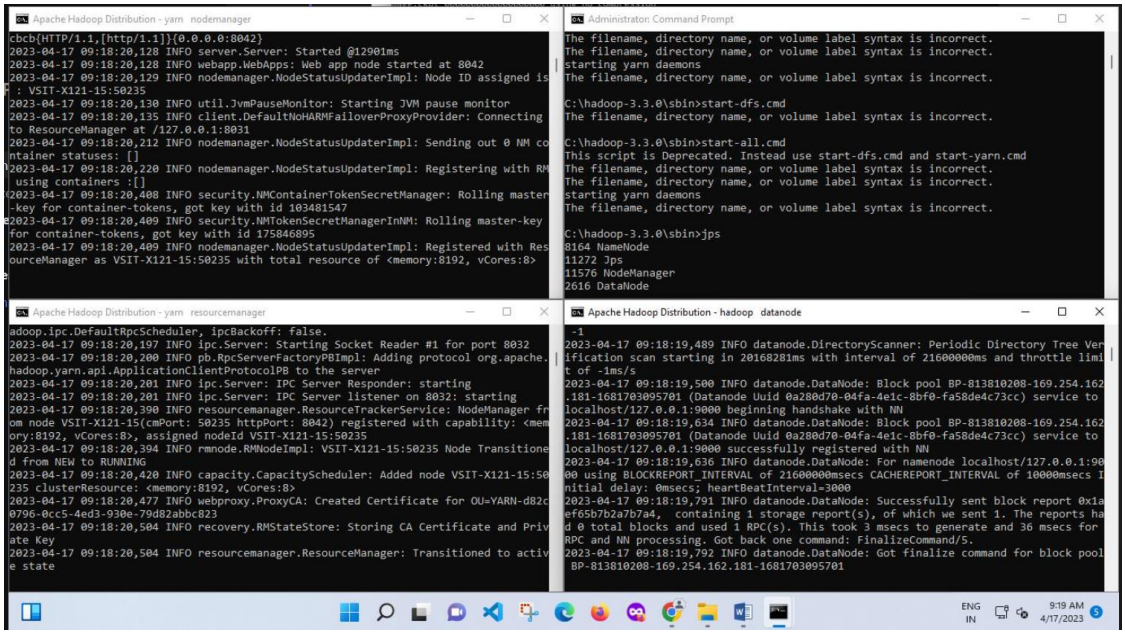
OR

- type start-dfs.cmd

– type start-yarn.cmd

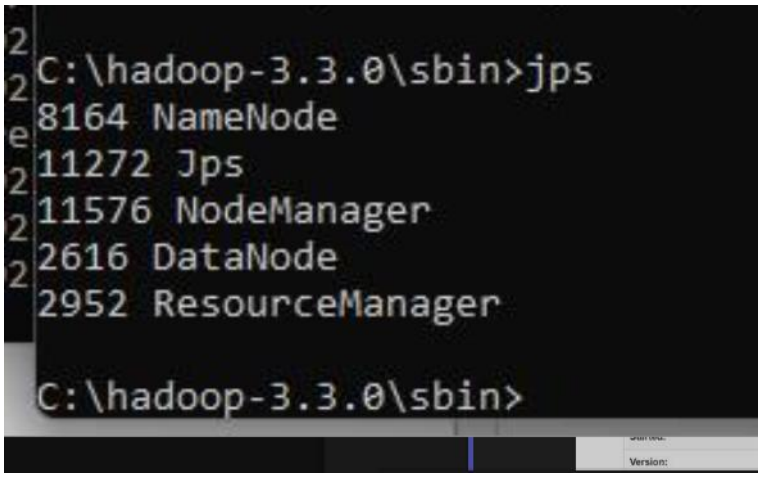


– You will get 4 more running threads for Datanode, namenode, resouce manager and node manager

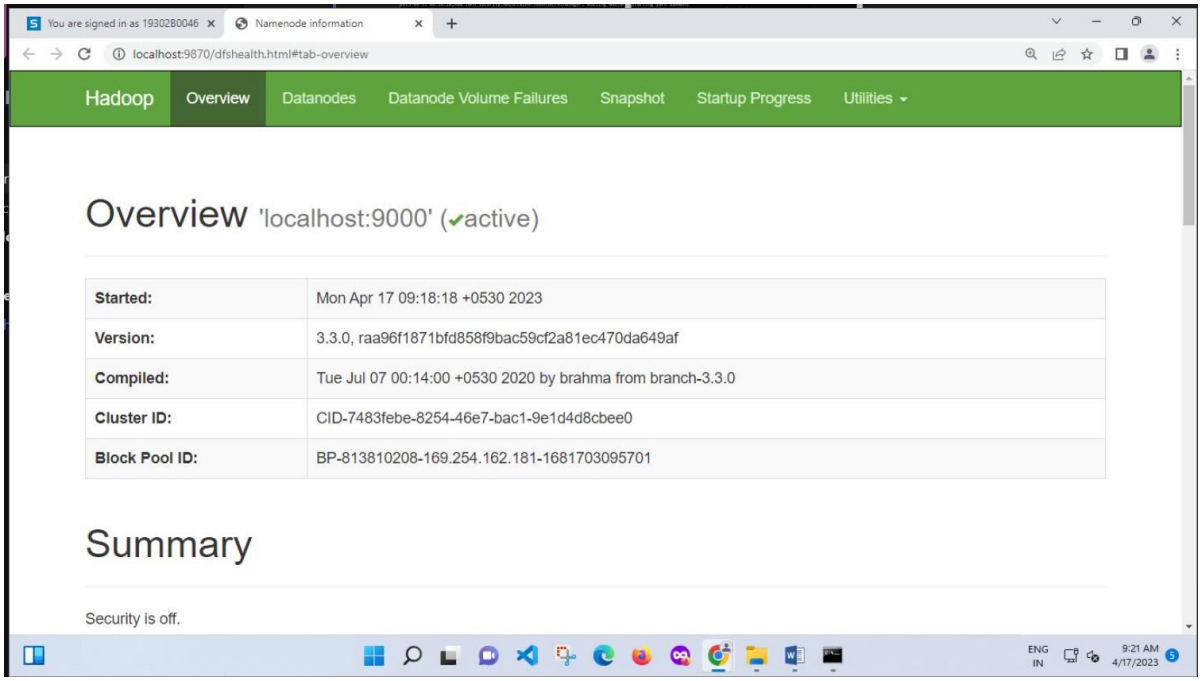


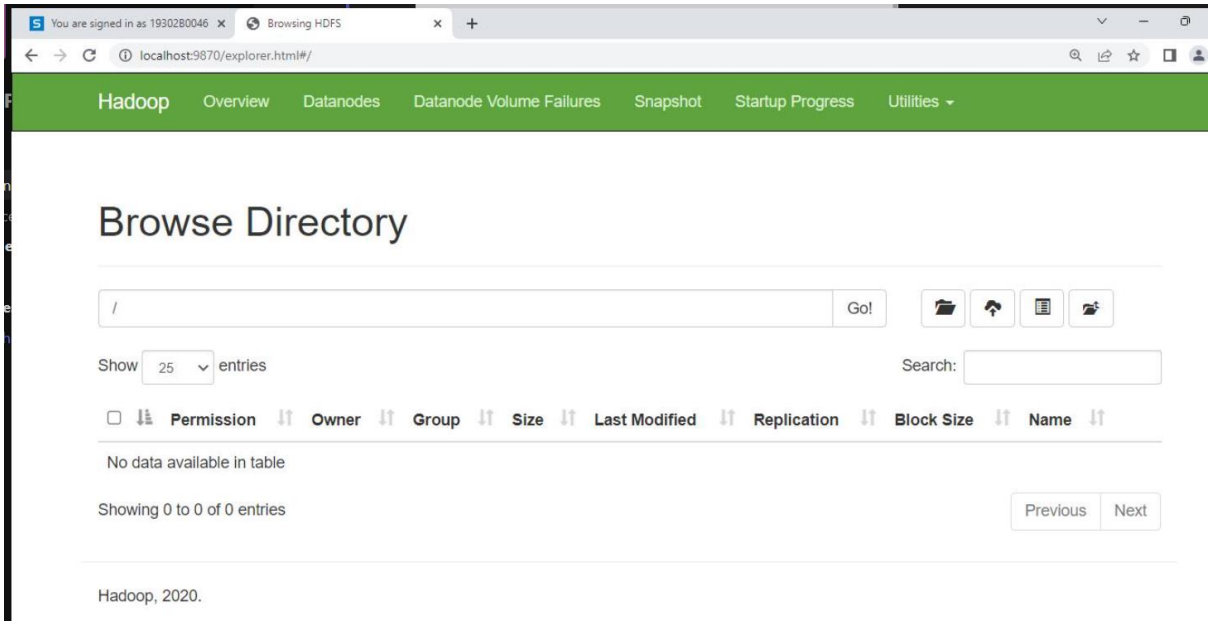
Output:

12. Type JPS command to start-all.cmd command prompt, you will get following output.



13. Run http://localhost:9870/ from any browser


You are signed in as 1930280046
Namenode information
localhost:9870/dfshealth.html#tab-overview
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Overview 'localhost:9000' (active)
Started: Mon Apr 17 09:18:18 +0530 2023
Version: 3.3.0, raa96f1871bfd858f9bac59cf2a81ec470da649af
Compiled: Tue Jul 07 00:14:00 +0530 2020 by brahma from branch-3.3.0
Cluster ID: CID-7483febe-8254-46e7-bac1-9e1d4d8cbee0
Block Pool ID: BP-813810208-169.254.162.181-1681703095701
Summary
Security is off.


You are signed in as 1930280046
Browsing HDFS
localhost:9870/explorer.html#/
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse Directory
/
Go!
Show 25 entries
Search:
Permission
Owner
Group
Size
Last Modified
Replication
Block Size
Name
No data available in table
Showing 0 to 0 of 0 entries
Previous
Next
Hadoop, 2020.

Practical No: 7

Date: 17/04/2023

Hadoop Installation:

Aim: Implement word count / frequency programs using MapReduce.

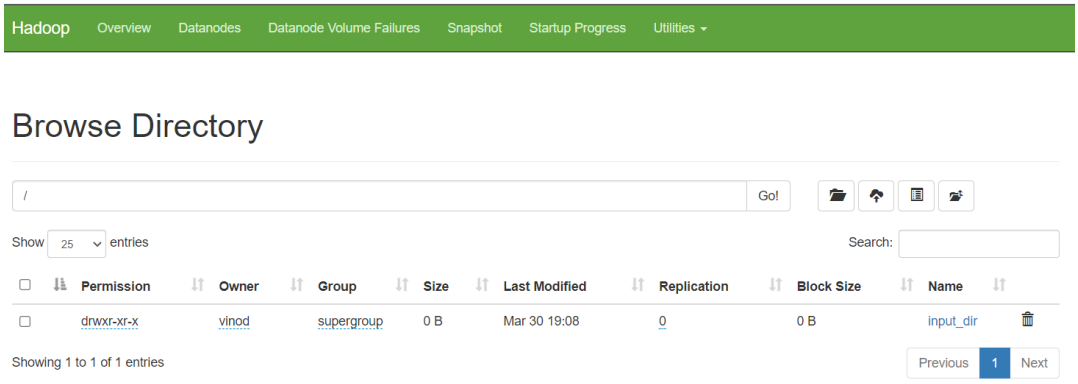
Solution:

C:\hadoop-3.3.0\sbin>start-dfs.cmd
C:\hadoop-3.3.0\sbin>start-yarn.cmd
Open a command prompt as administrator and run the following command to create an input and output folder on the Hadoop file system, to which we will be moving the sample.txt file for our analysis.
C:\hadoop-3.3.0\bin>cd\
C:\>hadoop dfsadmin -safemode leave
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
Safe mode is OFF
C:\>hadoop fs -mkdir /input_dir

Check it by giving the following URL at browser

http://localhost:9870

Utilities -> browse the file system

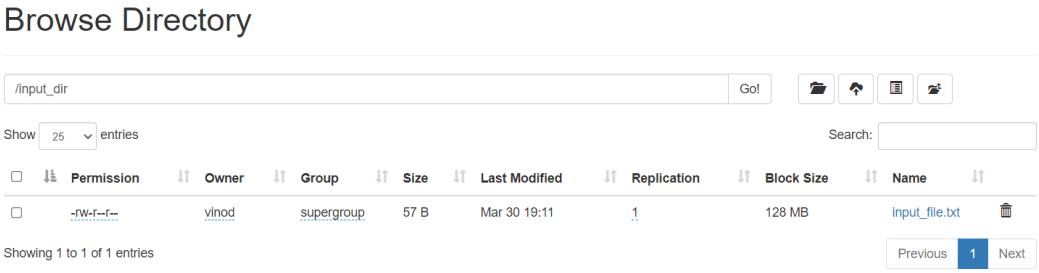


Copy the input text file named input_file.txt in the input directory (input_dir) of HDFS.

Make a file in c:\input_file.txt and write following content in it.

Hadoop Window version is easy compared to Ubuntu version

Now apply the following command at c:\>
C:\> hadoop fs -put C:/input_file.txt /input_dir



Verify input_file.txt available in HDFS input directory (input_dir).

C:\>Hadoop fs -ls /input_dir/

```
C:\>hadoop fs -put C:/input_file.txt /input_dir

C:\>hadoop fs -ls /input_dir/
Found 1 items
-rw-r--r--   1 vinod supergroup         57 2023-03-30 19:11 /input_dir/input_file.txt

C:\>
```

Verify content of the copied file
C:\>hadoop dfs -cat /input_dir/input_file.txt

You can see the file content displayed on the CMD.

```
C:\>hadoop dfs -cat /input_dir/input_file.txt
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
Hadoop Window version is easy compared to Ubuntu version.
C:\>
```

Run MapReduceClient.jar and also provide input and out directories.

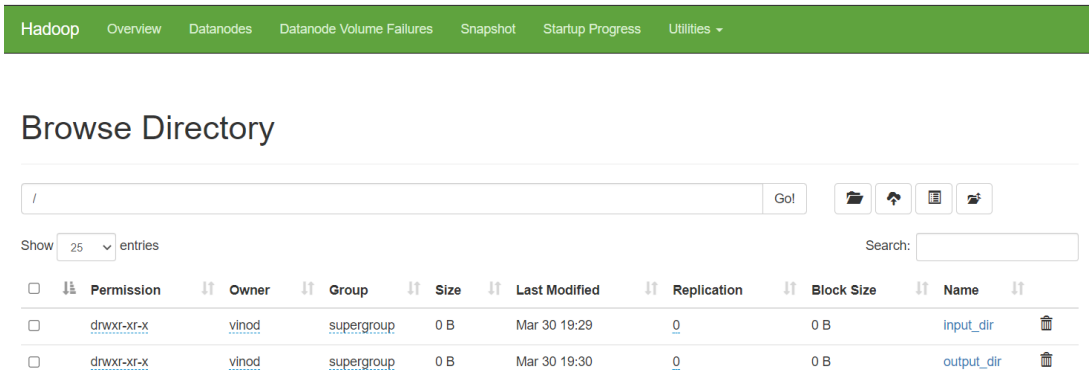
C:\>hadoop jar C:/hadoop-3.3.0/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.0.jar wordcount /input_dir /output_dir

```
                Reduce input groups=8
                Reduce shuffle bytes=103
                Reduce input records=8
                Reduce output records=8
                Spilled Records=16
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=70
                CPU time spent (ms)=219
                Physical memory (bytes) snapshot=517128192
                Virtual memory (bytes) snapshot=792633344
                Total committed heap usage (bytes)=392691712
                Peak Map Physical memory (bytes)=314761216
                Peak Map Virtual memory (bytes)=465485824
                Peak Reduce Physical memory (bytes)=202366976
                Peak Reduce Virtual memory (bytes)=327180288
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=56
        File Output Format Counters
                Bytes Written=65

:\Windows\System32>_
```

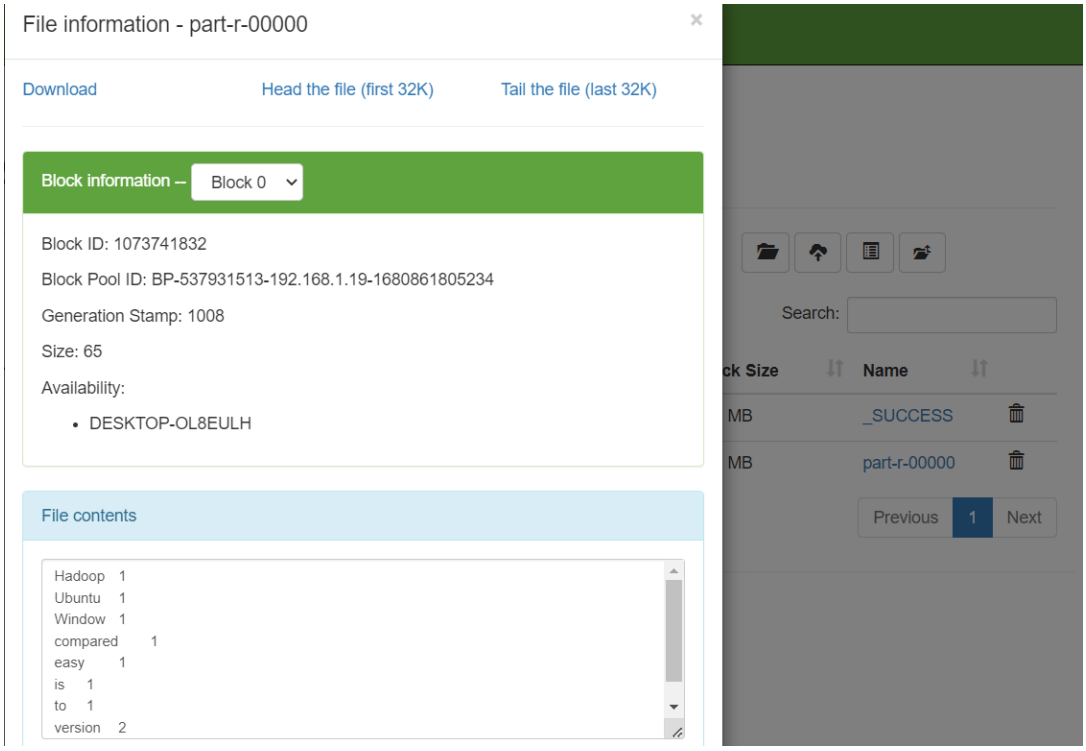
In case, there is some error in executing then copy the file MapReduceClient.jar in C:\ and run the program with the jar file using existing MapReduceClient.jar file as:

C:\> hadoop jar C:/MapReduceClient.jar wordcount /input_dir /output_dir

Now, check the output_dir on browser as follows:



Click on output_dir → part-r-00000 → Head the file (first 32 K) and check the file content as the output.



Alternatively, you may type the following command on CMD window as:

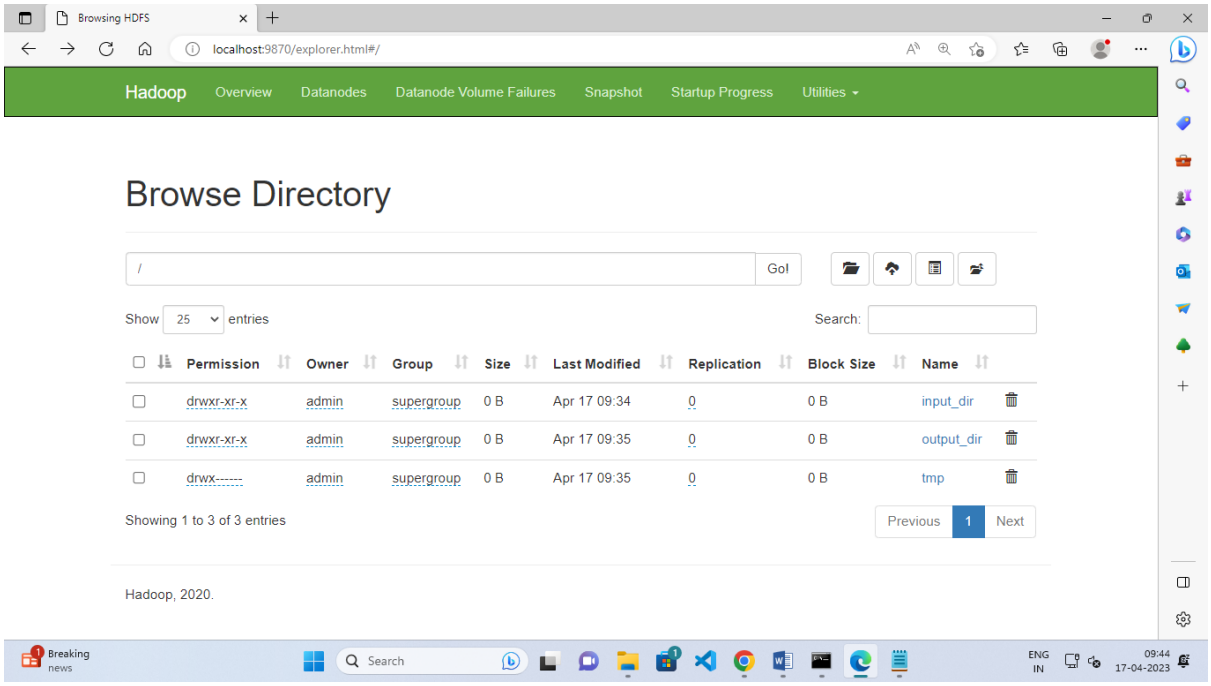
C:\> hadoop dfs -cat /output_dir/*

You can get the following output

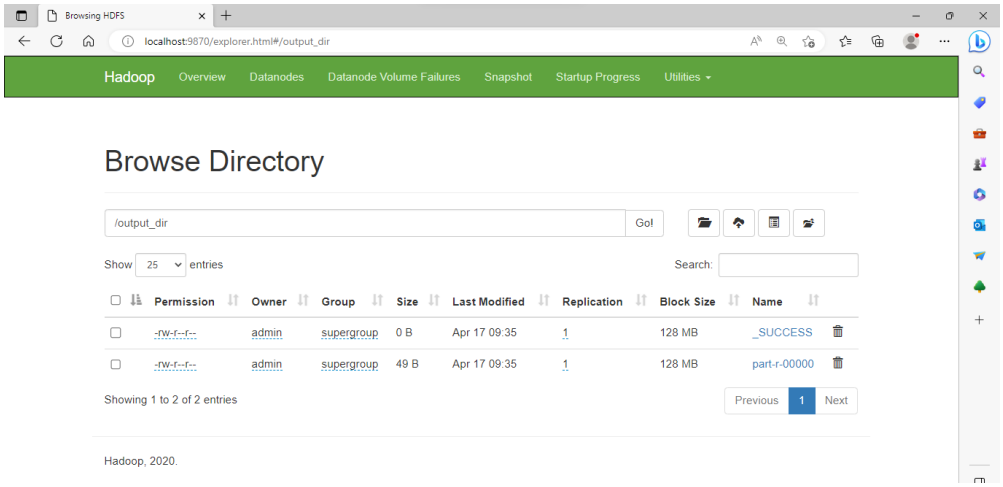
```
C:\Windows\System32>hadoop dfs -cat /output_dir/*
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
Hadoop  1
Ubuntu  1
Window  1
compared        1
easy    1
is      1
to      1
version 2

C:\Windows\System32>
```

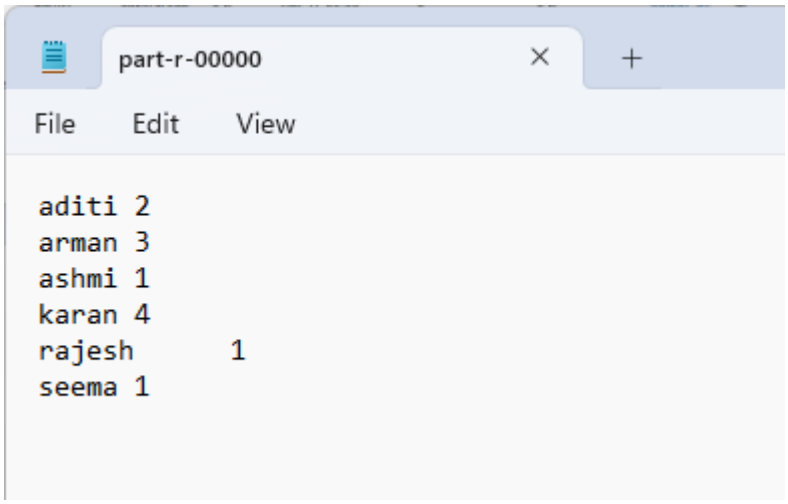
Output:


```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd C:\Hadoop-3.3.0\sbin

C:\hadoop-3.3.0\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\hadoop-3.3.0\sbin>jps
1584 DataNode
11028 Jps
6952 NodeManager
6476 NameNode
9308 ResourceManager

C:\hadoop-3.3.0\sbin>cd\

C:\>hadoop dfsadmin -safemode leave
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
Safe mode is OFF

C:\>hadoop fs -mkdir /input_dir

C:\>hadoop fs -put C:/input_file.txt /input_dir
put: `/input_file.txt': No such file or directory

C:\>hadoop fs -put C:/aditi.txt /input_dir

C:\>hadoop jar C:/hadoop-3.3.0/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.0.jar wordcount /input_dir /output_dir
2023-04-17 09:35:32,570 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /127.0.0.1:8032
```


Browsing HDFS
localhost:9870/explorer.html#/
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse Directory
/
Go!
Show 25 entries
Search:
Permission
Owner
Group
Size
Last Modified
Replication
Block Size
Name
drwxr-xr-x admin supergroup 0 B Apr 17 09:34 0 0 B input_dir
drwxr-xr-x admin supergroup 0 B Apr 17 09:35 0 0 B output_dir
drwx------ admin supergroup 0 B Apr 17 09:35 0 0 B tmp
Showing 1 to 3 of 3 entries
Previous
1
Next
Hadoop, 2020.

Output:


Browsing HDFS
localhost:9870/explorer.html#/output_dir
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse Directory
/output_dir
Go!
Show 25 entries
Search:
Permission
Owner
Group
Size
Last Modified
Replication
Block Size
Name
-rw-r--r-- admin supergroup 0 B Apr 17 09:35 1 128 MB _SUCCESS
-rw-r--r-- admin supergroup 49 B Apr 17 09:35 1 128 MB part-r-00000
Showing 1 to 2 of 2 entries
Previous
1
Next
Hadoop, 2020.


part-r-00000
File
Edit
View
aditi 2
arman 3
ashmi 1
karan 4
rajesh 1
seema 1

Practical No: 8

Date: 19/04/2023

Hadoop Installation:

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

Requirements

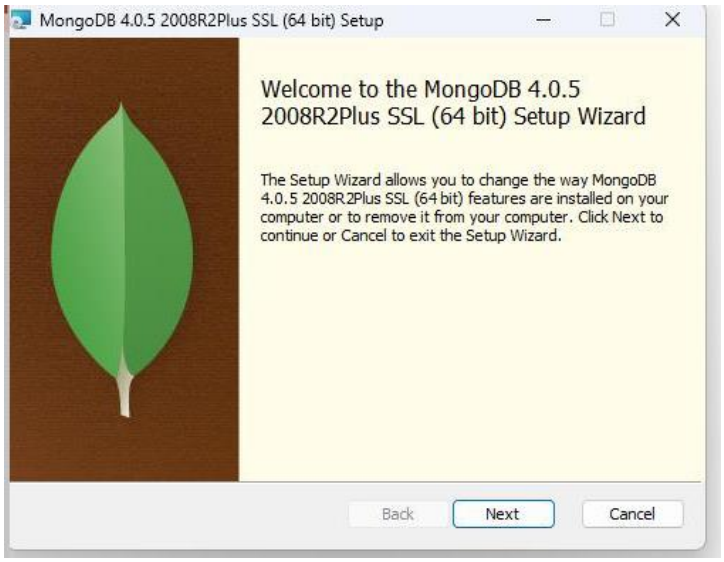
a. PyMongo

b. Mongo Database

Step A: Install Mongo database

Step 1) Go to (https://www.mongodb.com/download-center/community) and Download MongoDB Community Server. We will install the 64-bit version for Windows.

Step 2) Once download is complete open the msi file. Click Next in the start up screen



Step 3)

1. Accept the End-User License Agreement

2. Click Next

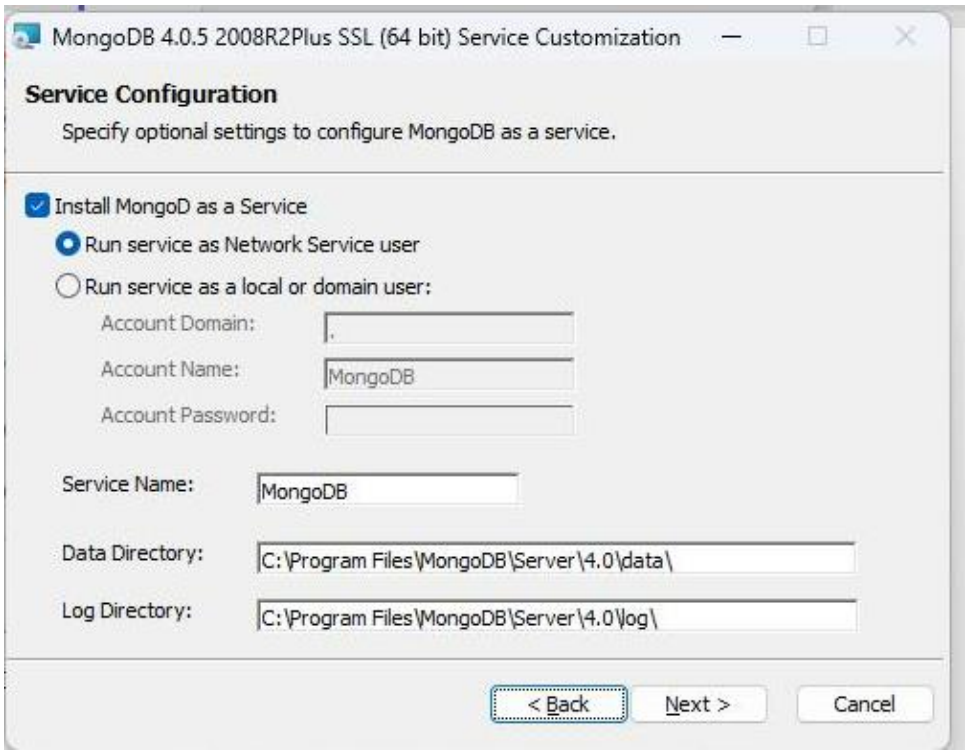
Step 4) Click on the "complete" button to install all of the components. The custom option can be used to install selective components or if you want to change the location of the installation.

Step 5)

1. Select “Run service as Network Service user”. make a note of the data directory,

we”ll need this later.

2. Click Next



Step 6) Click on the Install button to start the installation.

Step 7) Installation begins. Click Next once completed.

Step 8) Click on the Finish button to complete the installation.

Test Mongodb

Step 1) Go to " C:\Program Files\MongoDB\Server\4.0\bin" and double click on **mongo.exe.** Alternatively, you can also click on the MongoDB desktop icon.

Create the directory where MongoDB will store its files.

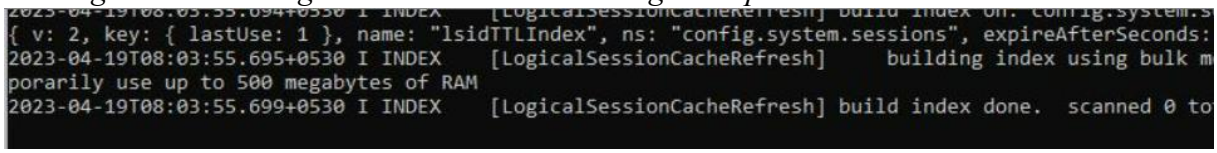
Open command prompt window and apply following commands

C:\users\admin> cd\

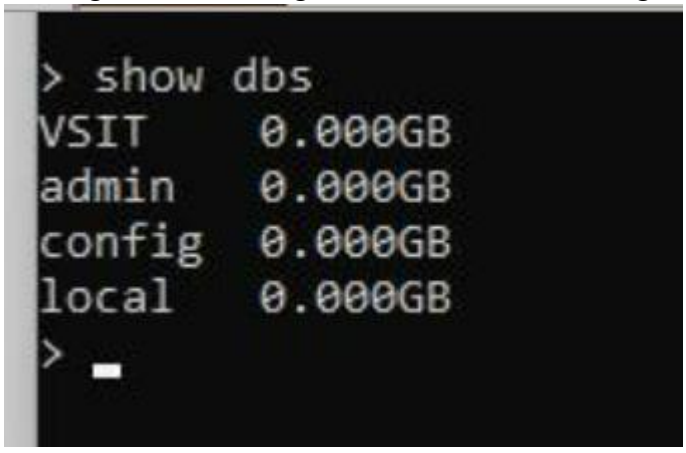
C:\>md data\db

Step 2) Execute mongodb

Open another command prompt window.
C:\> cd C:\Program Files\MongoDB\Server\4.0\bin
C:\Program Files\MongoDB\Server\4.0\bin> mongod
In case if it gives an error then run the following command:
C:\Program Files\MongoDB\Server\4.0\bin> mongod –repair



Step 3) Connect to MongoDB using the Mongo shell
Let the MongoDB daemon to run.
Open another command prompt window and run the following commands:
C:\users\admin> cd C:\Program Files\MongoDB\Server\4.0\bin
C:\Program Files\MongoDB\Server\4.0\bin>mongo



Step 4) Install PyMongo

Open another command prompt window and run the following commands:

Check the python version on your desktop / laptop and copy that path from window explorer

C:\users\admin>cd C:\Program Files\Python311\Scripts
C:\Program Files\<Python38>\Scripts > python -m pip install pymongo

```
C:\Windows\System32>cd C:\Users\admin\AppData\Local\Programs\Python\Python311\Scripts

C:\Users\admin\AppData\Local\Programs\Python\Python311\Scripts>python -m pip install pymongo
Requirement already satisfied: pymongo in c:\users\admin\appdata\local\packages\pythonsoftwarefoundation.pytho
5n2kfra8p0\localcache\local-packages\python310\site-packages (4.2.0)

C:\Users\admin\AppData\Local\Programs\Python\Python311\Scripts>
```

Note: # **-m** option is for <module-name>

Now you have downloaded and installed a mongoDB driver.

Step 5) Test PyMongo

Run the following command from python command prompt

import pymongo

Now, either create a file in Python IDLE or run all commands one by one in sequence on Python cell

Program 1: Creating a Database: create_dp.py

```
import pymongo
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mybigdata"]
print(myclient.list_database_names())
```

```
========== RESTART: D:/NK.py ==========
['VSIT', 'admin', 'config', 'local', 'mybigdata', 'mybigdataNK']
['student']
```

Progam 2: Creating a Collection: create_collection.py

```
import pymongo
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mybigdata"]
mycol=mydb["student"]
print(mydb.list_collection_names())
```

Progam 3: Insert into Collection: insert_into_collection.py

```
import pymongo
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
```

```
mydb = myclient["mybigdata"]
mycol=mydb["student"]
mydict={"name":"Beena", "address":"Mumbai"}
x=mycol.insert_one(mydict) # insert_one(containing the name(s) and value(s) of each field
```

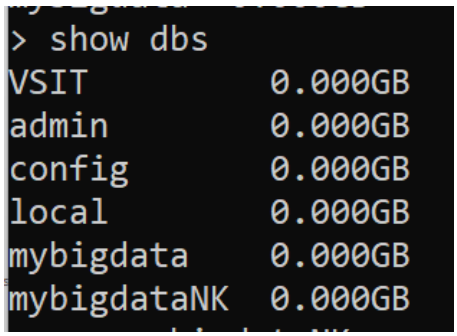
Program 4: Insert Multiple data into Collection: insert_many.py

```
import pymongo
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mybigdataNK"]
print(myclient.list_database_names())
mycol=mydb["student"]
print(mydb.list_collection_names())
mydict={"name":"Ninad", "address":"Mumbai"}
x=mycol.insert_one(mydict) # insert_one(containing the name(s) and value(s) of each field
 mylist=[{"name":"Vighnesh", "address":"Mumbai"}, {"name":"Sarthak",
"address":"Mumbai"},
{"name":"Nidhi", "address":"Pune"}, {"name":"Komal", "address":"Pune"},]
x=mycol.insert_many(mylist)
```

Step 6) Test in Mongodb to check database and data inserted in collection

a. If you want to check your database list, use the command show dbs in mongo command prompt

```
> show dbs
```



b. If you want to use a database with name mybigdata, then use database

statement would be as follow:

> use mybigdataNK

```
> use mybigdataNK
switched to db mybigdataNK
> show collections
```

c. If you want to check collection in mongodb use the command show collections

> show collections

d. If you want to display the first row from collection: db.collection_name.find()

> db.student.findOne()

e. If you want to display all the data from collection: db.collection_name.find()

> db.student.find()

f. count number of rows in a collection

> db.student.count()

```
> show collections
student
> db.student.findOne()
{
        "_id" : ObjectId("643f57fd927b349355491923"),
        "name" : "Ninad",
        "address" : "Mumbai"
}
> db.student.count()
5
```

```
> db.student.find()
{ "_id" : ObjectId("643f57fd927b349355491923"), "name" : "Ninad", "address" : "Mumbai" }
{ "_id" : ObjectId("643f57fd927b349355491924"), "name" : "Vighnesh", "address" : "Mumbai" }
{ "_id" : ObjectId("643f57fd927b349355491925"), "name" : "Sarthak", "address" : "Mumbai" }
{ "_id" : ObjectId("643f57fd927b349355491926"), "name" : "Nidhi", "address" : "Pune" }
{ "_id" : ObjectId("643f57fd927b349355491927"), "name" : "Komal", "address" : "Pune" }
>
```