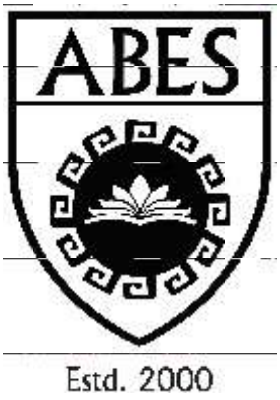
Practical File

Lab Name … **Data Analytics Lab……** ***Lab Code……KIT-651……***

.



Name.....Tanuj Yadav.....

Adm.No...2018BCI1026..... Univ. Roll No...1803211050.....

Course ……B.Tech……..……Branch……CEIT……..……..

Sem……6th…….…… Section……A……..……..

ABES
Engineering College
(College Code-032)
NAAC Accredited, NBA Accredited Branches (CSE, ECE, EN & IT)

ISO 9001:2015
TÜVRheinland CERTIFIED
www.tuv.com ID 9105047739

19th Km. Stone, NH-09, Ghaziabad - 201009 (UP), India
Phone : 0120-7135112, 9999889341 Fax : 0120-7135115 Website : www.abes.ac.in, Email: info@abes.ac.in

ABES Engineering College, Ghaziabad

Student Name:..... Harshul Gupta
Roll No:..... 1803211023

I N D E X

Exp. No	Name of the Experiment	Date of Practical	Assessment by Faculty Member					
		Schedule / Actual	Pre-Lab Writing work * MM:05	Implementation/ Active Participation MM:10	Graph, Results/ Output, Calc. MM: 05	In Time submission of Lab reports/ Viva Voce MM:05	Total MM:25	Signature with Date
1	Understanding of statistics on dataset	4/13/2021						
2	Handling data files in R	20/04/2021						
3	Handling Multi-dimensional Array	22/05/2021						
4	Perform mean, median, mode and standard deviation	13/04/2021						
5	Perform data processing operations	27/04/2021						
6	Perform Dimensionality Reduction	4/5/2021						
7	Perform Simple Linear Regression	24/05/2021						

Teacher's Remarks (if any): **Average Marks:**

Name & Sign. Of Faculty members(s) with date **Name & Sign. Of Lab In charge** **HOD**

Note:
1. *Pre-Lab writing work should include problem statement, objective , Algorithum (if applicable)and Methodology.
2. Faculty members will check pre Lab writing work & Lab work readings/output in each class and sign with date.
3. Please use pen with blue colour Ink only.

Saathi

Date ___/___/_____

Practical-1

Objective : To understand the statistics on data base.

Algorithm:

1. Take a array of numbers from user.
2. Find out minimum, maximum, average sum and square root of each number
3. Use min(), max(), avg(), sum(), sqrt() round() function
4. Display out put

Code:

```
x ← scan()
print(x)
# maximum of an array
max = x[1]
for (i in x)
{
if (i > max) {
max = i
}}
print(paste0("without function=" max))
print(paste0("without function=" max(x))

# minimum of an array
min = x[1]
for (i in x)
{ if (i < min){
min = i
}}
print(paste0("without function=" min))
```

Page No.

Date __/__/____

Saathi

```
print(paste0("with function = ", min(x))
# Average of an array
  j=0
  counter = 0
for(i in x)
{j = j+1
  counter = counter+1
# print(paste0("without function =", sum))
 print(paste0("with function = ", sum(x))

# Square root of a number
  num = x[3]
 print(paste0("without function =", num^0.5))
  print(sqrt(x[3]))

# Round off
  print(round(x[4]))
```

Output

```
  1 2 3 4 5 6 ← Input
[1,2,3,4,5,6]
without function = 6
  with function = 6
without function = 1
  with function = 1
without function = 3.5
  with function = 3.5
without function = 2.1
  with function = 2.1
without function = 1.732050807568 88
  with function = 1.73205 1
```

Page No.

Date __/__/__ Practical - 2

Saathi

Objective : To handle the data in R program

1. Explore the UCI machine learning repository
2. Download the dataset in CSV, xls, .txt file format
3. Use read() function to import the dataset
4. Save the file in data frame & print the header using header function.

Code

```
data ← read csv("file="/home/RITIK/Desktop/prac2.CSV", header = T)
head(data 1)
v ← c(1,2,3,4,5,6,7)
write.csv(v, file = "/home/utsav/Desktop/labcs
data 2 ← read.table(file.choose())
head(data 2)
write.table(v, file = "/home/RITIK/Desktop/prac2.txt"
    append = F, sep = "/", dec = ",", row.name
    = T, col.names = T)
data 3 ← read.xlsx("/home/utsav/desktop/prac2.text"
    sheetname = "sheet")
head(data 3)
write.xlsx(v, file = "/home/utsav/desktop/text.xlsx",
    append = F, sep = "/" dec = ",", row.
    names = T, col.names = T)
```

Output

	Name	Roll.no
1	A	1
2	B	2

Page No.

Date __/__/____

Saathi

3	C	3
4	D	4
5	E	5
6	F	6

write.csv(v, file = "/home/ustav/desktop/lab.csv")

[1] Testing data for reading text file in R
<0 rows> (or 0 length row.names)

	Name	Branch
1	A	CS
2	B	CSIT
3	C	IT
4	D	EC
5	E	ME

Date __/__/__ Practical-3

Objective: To understand the handling of multidimensional array
To get the input matrix from user & perform matrix addition, subtraction, multiplication Inverse, transpose & division operation using vector concept in R

Algorithm: 1. Take a input from user in 2-D arrays of size 3x3, i.e

2. Array1:- 20 30 23 Array-2: 15, 12, 22
14 21 28 19, 23, 30
18 19 17 21, 23, 11

3. Perform addition, subtraction multiplication Inverse, transpose & division operation one

4. Display output

Code:→ Arr1 ← matrix(c(20,30,23,14,21,28,19,18,19,17)

nrow = 3, by row = True

print (Arr1)

Arr2 ← matrix(c(15,12,28,19,25,30,21,23,11),

nrow = 3, by Row = True

print (Arr2)

Addition ← Arr1 + Arr2

print (Addition

SUB ← Arr1 - Arr2

print (sub)

multiplication ← Arr1 %*% Arr2

print (multiplication)

Inverse ← Inv(t(Arr1))

Page No.

Date __/__/____

Saathi

print (Inverse)
Trans ← t(Arr2)
print (Trans)
Div ← Arr1 / Arr2
print (Div)

output :-

20 30 23
14 21 28
18 19 17

15 12 28
19 25 30
21 23 11

5 18 -5
-5 -4 -2
-3 -4 6

1353 1519 1713
1197 1337 1330
988 1082 1261

-0.09191176 0.19970588 -0.05882353
-0.03834034 -0.03886555 0.08403361
0.18750000 -0.02500000 0.00000000

15 19 21
12 25 23
20 30 11

page No.

Saathi

Date __/__/____

1.33333333	2.50000000	0.8214284
0.7368421	0.84000000	0.933333
0.8571429	0.8260870	1.5454545

Date __/__/____ Practical - 04 Saathi

Objective : To find the statistics of data set

Algorithm

* Apply mean, median & standard deviation formula to find statistics of given data set
* Apply mean( ), median( ), mode( ) & sd( ) function
* Find all the parameters of data frame calc.
* Display the output

Code:

```
data <- read csv("E:\\sample.csv")
head(data)
mode <- function(v){
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v,uniqv)))]
print(("mean:", mean(data\[["n"]])))
print(("median:", median(data)[("n")]))
print(("mode:", mode[data][["n\"]]))
print(("Standard deviation", sd(data[("n")])))
```

output → x

```
1 2 3
2 5 6
3 7 8
4 2 3
5 5 6
6 7 9
mean: 53.375
median: 5 6
mode: 2 3
standard deviation 20.3167
```

Page No.

Date __/__/ Practical-5

Saathi

Objective: To understand data cleaning processing

Algorithm

1. Find missing values in data using is.na() function
2. Fill missing value using mean/median/mode
3. Normalise the data using min-max normalization

Code

```
v <- c(3,NA,5,6,8,NA,9,12,10,NA,21,25)
a <- v
b <- v
print(is.na(v))
a[is.na(a)] <- mean(a, na.rm=T)
b[is.na(b)] <- median(b, na.rm=T)
print("using mean")
print(b)
d <- a
for (i in 1:length(a))
{d[i] = (d[i] - min(a)) / (max(a) - min(a))}
print("Normalised data")
print(d)
```

output →

FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE

using mean

3, 11, 5, 6, 8, 11, 9, 12, 10, 11, 21, 25

using median

3, 9, 5, 6, 8, 9, 9, 12, 10, 9, 21, 25

Date __/__/____

Saathi

Normalized data

0.00, 0.36 0.09 0.13, 0.22, 0.36, 0.21, 0.40
0.31, 0.36, 0.81, 1.00

Date __/__/__ Practical-6

Saathi

Objective: To get the precise & importa

Algorithm:

1. Run Process the dataset
2. Built pcachart function for explaratary or data analysis on variance
3. Standardize the data by using scale & apply "pcamp" function
4. Choose the principal components with highest variance
5. visualization of data in the new reduced dimension

Code:

```
data1 ← read.csv ("E:/1 sample csv." header=F)
n ← as.matrix (data1)
print ("original data")
print(n)
a ← scale(n)
c ← princomp (a) $ rotation [1:2]
print ("required principle components")
print (c)
    y ← n %*% c
print ("reduced dimension data")
  print(y)
```

Page No.

Date __/__/____

Saathi

output

original matrix

V_1	V_2	V_3
90	60	90
90	90	30
60	60	60
60	60	90
30	30	30

Required principle components

PC_1	PC_2
-0.7071	0.0000
-0.6660	0.3333
-0.2357	-0.9428

Reduced dimension data

PC_1	PC_2
-124.33	-64.85
-130.71	1.71
-96.56	-36.56
-103.63	-64.85
-48.28	-18.28

Date __/__/__ Practical-07

Saathi

Objective To perform simple linear progression with

· Performing the regression technique on given dataset

Algorithm

1. Load the data & make sure the data meet the assumption :- * Normality * correlation * linearity
2. the formula for linear regression is $y = a_0 + a_1 x$ $y \rightarrow$ predicted value i.e. $a_0 \rightarrow$ value by of y when x is 0 $a_1 \rightarrow$ regression coefficient : $x \rightarrow$ independent variable
3. Display the regression line.

Code;

```
m2 <- matrix(c(95,85,85,95,80,70,70,70,65,60,70)
  .byrow = TRUE)
meanx = mean(m2[1])
meany = mean(m2[2])
print("mean of x:")
print(mean.x)
print("mean of y:")
print(meany)
a1n = 0
a1d = 0
for(i in 1:length(m2[1])){
a1n = a1n+((m2[1][i] - mean.x)*(m2[2][i]-mean
a1d = a1d + (m2[1][i] - mean.x)**2
```

Page No.

Date __/__/____

Saathi

```
a1 = a1n/a1d
a0 = mean y - (a1 * mean x)
print ("Regression line")
print (cat ("y=", a0, "+", a1, "x"))
```

output

```
mean of x:
78
mean of y
77
Regression line;
y = 26.78082 + 0.6458356x.
```

Objective → Perform the clustering technique to categorise iris's data in groups.

Algorithm →

1) load the iris dataset
2) Specify the number of clusters to be created
3) Select randomly K- objects from the dataset as the initial cluster sector or means.
4) Assign each observation to their closest centroid based on the Euclidean distance b/w the object and the centeroid.
5) for each of the K clusters update the cluster centroid by calculating the new mean values of all the data points in cluster.
6) Iteratively minimize the total within sum of square i.e iterate steps 3 & 4 until cluster can't be updated further.
7) Print the cluster formed.

Code →

```
library (datasets)
df ← iris
set.seed (25)
nrows ← sample (narrow(df))
df ← df [nrows, ]
```

```
print("head of iris dataset")
head(df)
df <- df[,2:4]

n <- nrow(df)
cols <- names(df)
df["Cluster"] <- NA
df["dist"] <- 999
  m <- length(cols)
  n-c <- 3
   s <- sample(1:n, n-c)
   smdf <- df[0,]
  for(i in s){
    smdf <- rbind(smdf, df[i,])
  }
  s <- nrow(smdf)
  smdf <- smdf[,-5]
  smdf <- smdf[,-5]
   print("Centroid selected:")
   smdf
   a <- FALSE
   while(a==FALSE){
    for(i in 1:n){
     for(j in 1:5){
          dist <- 0
```

```
for (k in 1:m) {
  dist <- dist + abs(df[i,k] - smdf[j,k])
}

  if (df[i, m+2] > dist) {
    df[i, m+1] <- j
    df[i, m+2] <- dist
  }
}
}

sm <- aggregate(x= df[, 1:5],
                by = list(df$Cluster),
                FUN= mean)

sm <- sm[, -1]
a <- identical(sm, smdf)

smdf <- sm
}
print("Number of clusters found:")
df %>% count(Cluster)
print("Final centroids")
print(smdf)
```

Output → Head of Iris dataset

Sepal Length	Sepal width	Petal Length	Petal Width
6.1	2.6	5.6	1.4
5.1	3.3	1.7	0.5
5.3	3.7	1.5	0.2
6.1	2.8	4.0	1.3
6.5	3.0	5.8	2.2

Centroid selected →

Sepal length	Sepal width	Petal width	Petal width
7.2	3.6	6.1	2.5
6.4	2.8	5.6	2.1
7.2	3.0	5.8	1.6

Number of clusters found.

cluster	n
1	12
2	88
3	50

Final centeroids →

7.291667	3.208333	6.25833	2.233333
5.388636	3.132955	5.798864	0.537818
6.296000	2.888000	4.846000	1.676000

Objective → To perform KNN classification for the given dataset

Algorithm →

1. Load the Dataset.
2. Calculate the Euclidean measure b/w the data points
3. Arrange the Euclidean distance in ascending order
4. Initialize K and take the first K distance from the list.
5. Calculate the No. of datapoints belonging to ith class among K points.
6. If $K_i > K_j \; \forall \; i \neq j$, put x in class i.

Code →

```
df<- read.csv (file="C:/users/Caba.csv", header=TRU
head (df)
ec-dist <- function (x, df) { df["distance"]<-NU
 for (i in 1: nrow (df)) {
    d<-0
    for (j in 1: (ncol (df)-2)) {
       d<- d+(x[j]- df[i,j])** 2
    }
```

```
}
return(df)

KNN <- function(df,k,x){
  df <- ec_dist(x,df)
  df <- df[order(df[,ncol(df)]),]
    m <- 0
    l <- 0
  for(i in 1:k){
    if(df[i,3]=="M"){
      m= m+1
    }
    else{
      l <- l+1
    }
}
class <- function(m,n){ if(m>1){print("M")}
  else{print("L")}}
  print("class is:")
  class(m,l)
}
print("new data to be predicted:")
  weight <- readline(prompt="Enter weight:")
  height <- readline(prompt="Enter Height:")
```

```
x ← c (as.integer(height), as.integer(weight))
KNN(df, 5, x)
```

Outcome → New data point to predict

```
Enter Width : 68
Enter Height: 168
class is : C
```

Roll no. → 1803211024
Admission No. → 2018BC11032

Date __/__/____

Saathi

Practical Name → To Preform Market Basket analysis Using Association Rules

Objective → Perform the association rules for Analysis

Algorithm →

(i) Start with itemset Containing just a Single item Such as {apple}, {pear}
(ii) Determine the support for itemsets Keep the itemsets that meet your Minimum Support threshold and remove the itemset that do not.
(iii) Using the itemsets you have kept from Step.
(iv) Repeat Step 1 & 2 until there are no newer itemsets

Code →

```
df ← read.csv(file = "C:/Users/purchase.csv", header = TRUE)
head(df)
a ← unlist(strsplit(df[1,2], split = ","))
for ( j in 2 = nrow(df)) {
        s ← unlist(strsplit(df[j,2], split = ","))
        a ← unlist(union(a,s))
    }
for ( i in 1 = length(a)) {
        df[a[i]] ← NA
}
```

Page No.

Roll no. → 1803211024
Admission No. → 2018BCI1032

Date __/__/____

```
for (i in 1:nrow(df)){
    s<- unlist(strsplit(df[i,2], split = ","))
    for(j in 1:length(a)){
        if(a[j] %in% s){
            df[i, a[j]] <-1
        }
        else{
            df(i, a[j]) <- 0
        }
    }
sf <- df[, 3:ncol(df)]
k <- 0
m <- length(a)
b <- a
while(k<m){
    for(j in 1:length(a)){
        s <- length(b)
        for(i in 1:s){
            if(colsums(data.frame(sdf[b[i]] *
                sdf[a[j]]) <2
            || (grepl(a[j], b[i])){
        }
        else{
            b <- append(b, paste(b[i], a[j], sep="v"))
            sdf[paste(b[i], b[j], sep=",")]
                = sdf[b[i]] * sdf[b[j]]
        }
    }
    k <- k+1
}
}
```

Roll no. → 18032 11040
Admission No → 20188(I 1057 Saath

Date ___/___/_____

```
print("Rule Formed")
for (i in 1:length(b)){
        p<- colsum(3df(b[i]))
if (p>=.2){
    print(paste(B[i], "→", p))
}
}
```

Output:-

TID	Items
100	1,3,4
200	2,3,5
300	1,2,3,5
400	2,5

Rule formed:

1→2
3→3
2→3
5→3
31→2
23→2
53→2
52→3
235→2