

DATA SCIENCE



Vidya Vikas Education Society's

VIKAS COLLEGE OF ARTS, SCIENCE & COMMERCE

Affiliated to University of Mumbai

RE-ACCREDITED 'A' GRADE BY NAAC

ISO 9001: 2008 CERTIFIED

Vikas High School Marg, Kannamwar Nagar No 2, Vikhroli (E), Mumbai – 400083

Dr. R. K. Patra
Principal

Hon' ble: **Shri P. M. Raut**
Chairman. V. V. Edu. Society

This is to certify that,

Student of T.Y.B.Sc. (Computer Science) (Sem-VI) with college enrolled Roll no. _____ has satisfactorily completed the practical work for the Subject Ethical Hacking in the program of Computer Science from the UNIVERSITY OF MUMBAI for the academic year 2022-2023.

Guided By

Head of Department

Internal Examiner

External Examiner

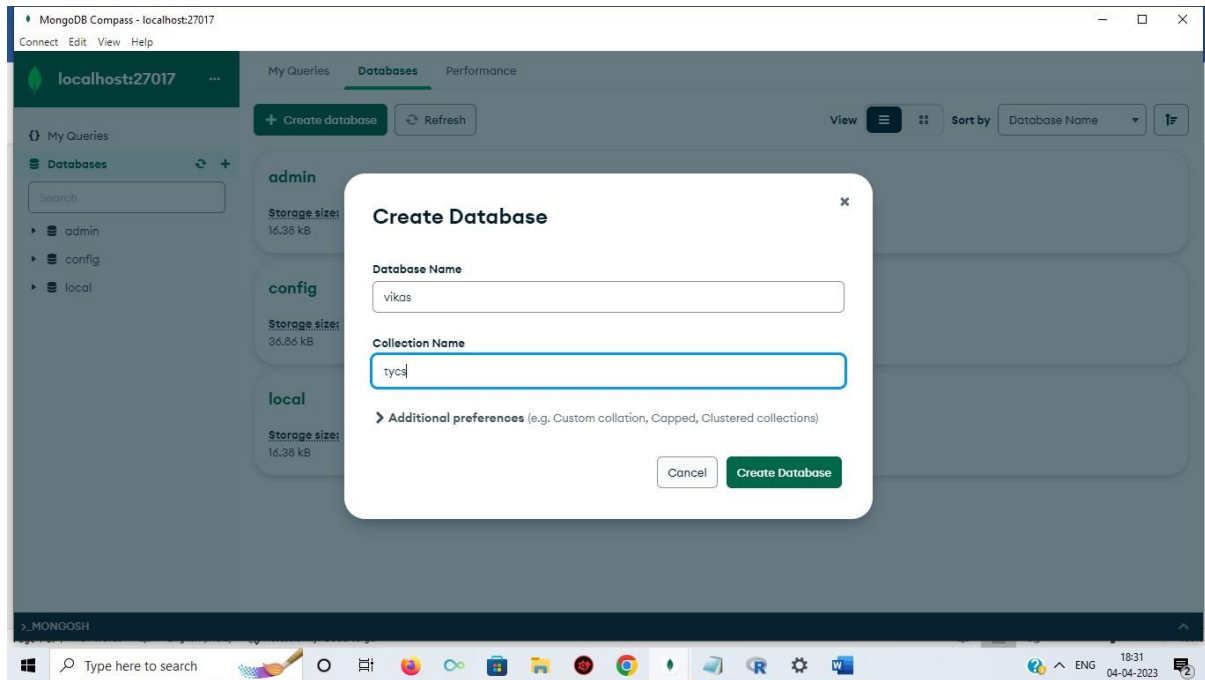
INDEX

SR.NO	PRACTICALS	DATE	SIGN
1	Practical of Data collection, Data curation and management for Unstructured data (NoSQL)		
2	Practical of Data collection, Data curation and management for Large-scale Data system		
3	To perform practical of Principal Component Analysis(PCA).		
4	To perform practical of Clustering.		
5	To perform practical of Time-series forecasting		
6	To perform practical of Simple/Multiple Linear Regression.		
7	To perform practical of Logistics Regression.		
8	To perform practical of Hypothesis testing.		
9	To perform practical of Analysis of Variance .		
10	To perform practical of Decision Tree .		

Practical No: 1

Aim: Practical of Data collection, Data curation and management for Unstructured data (NoSQL)

Step 1: Create database and collection in mongodb 4.0 by using MongoDB Compass.



Step 2: Run the following code in R or Rstudio

```
install.packages('mongolite')
```

```
# Load the mongolite package
```

```
library(mongolite)
```

```
# Connect to the MongoDB database vikas
```

```
conn <- mongo(collection = "tycs", url = "mongodb://localhost:27017/vikas")
```

```
# Insert the document into the collection
```

```
doc <- '{"name": "Aniket", "age": 20}'
```

```
conn$insert(doc)
```

```
# Insert the document into the collection
```

```
doc <- '{"name": "Mayur", "age": 20}'
```

```
conn$insert(doc)
```

```
# Insert the document into the collection
```

```
doc <- '{"name": "Kirti", "age": 20}'
```

```
conn$insert(doc)
```

```
# Confirm that the records have been inserted
```

```
result <- conn$find()
```

```
print(result)
```

```
# Find the record of aniket in the collection
```

```
result <- conn$find('{"name": "Aniket"}')
```

```
print(result)
```

```
#update the age of Aniket to 21
```

```
conn$update('{"name": "Aniket"}', '{"$set": {"age": 21}}')
```

```
# Print the result
```

```
result <- conn$find()
```

```
print(result)
```

```
#delete the record of Aniket
```

```
conn$remove('{"name": "Aniket"}')
```

```
# Print the result
```

```
result <- conn$find()
```

```
print(result)
```

R GUI (64-bit)

File Edit View Misc Packages Windows Help

R Console

```
> #update the age of Aniket to 21
> conn$update('{"name": "Aniket"}', '{"$set": {"age": 21}}')
List of 3
 $ modifiedCount: int 1
 $ matchedCount : int 1
 $ upsertedCount: int 0
> # Print the result
> result <- conn$find()
> print(result)
  name age
1 Aniket  21
2 Mayur   20
3 Kirti   20
>
>
> #delete the record of Aniket
> conn$remove('{"name": "Aniket"}')
>
> # Print the result
> result <- conn$find()
> print(result)
  name age
1 Mayur   20
2 Kirti   20
>
```

Type here to search

18:46 04-04-2023

MongoDB Compass - localhost:27017/vikas.tyos

Connect Edit View Collection Help

localhost:27017 Documents vikas.tyos

My Queries Databases

Search

admin config local vikas tyos

vikas.tyos

Documents Aggregations Schema Explain Plan Indexes Validation

Filter Type a query: { field: 'value' } Reset Find More Options

ADD DATA EXPORT COLLECTION

1 - 2 of 2

```
{ "_id": ObjectId('642c231834dd61274608a3d2'), "name": "Mayur", "age": 20 }
```

```
{ "_id": ObjectId('642c231834dd61274608a3d3'), "name": "Kirti", "age": 20 }
```

>_MONGOSH

Type here to search

18:50 04-04-2023

Practical No:2

Aim: Practical of Data collection, Data curation and management for Large-scale Data system

use tycs

```
db.dropDatabase()
```

```
db.collection1.insert({id:001, Name:"Rajat"})
```

```
show collections;
```

```
db.collection1.insert({ id: 002, name:"Raj", course:[{name:"CS", duration:7}, {name:"Java", duration:5}]})
```

```
var ins = [ {"StudentID" : 100, "Name" : "Mayur"}, {"StudentID" : 101,"Name" : "kirti"}];
```

```
db.collections1.insert(ins);
```

```
db.collection1.find()
```

```
db.collection1.find().pretty()
```

```
db.collection1.find({"id":{"$gt:1}}).pretty()
```

```
db.collection1.find({"id":{"$lt:2}}).pretty()
```

```
db.collection1.update({"Name":"Rajat"},{$set:{"Name":"Aniket"}})
```

```
db.collection1.find({},{"name":1})
```

```
db.collection1.find()
```

```
db.collection1.find().sort({id:-1})
```

Command Prompt - mongo

```
---
> use tycs;
switched to db tycs
> db.dropDatabase()
{ "ok" : 1 }
> db.collection1.insert({id:001, Name:"Rajat"})
2023-04-04T19:01:32.355+0530 E QUERY [js] SyntaxError: illegal character @(shell):1:36
> show collections
> show collections;
> db.collection1.insert({id:001, Name:"Rajat"})
WriteResult({ "nInserted" : 1 })
> show collections;
collection1
> db.collection1.insert({ id: 002, name:"Raj", course:[{name:"CS", duration:7}, {name:"Java", duration:5}]})
WriteResult({ "nInserted" : 1 })
> var ins = [ {"StudentID" : 100, "Name" : "Rajaa"}, {"StudentID" : 101, "Name" : "Raju"}];
>
> db.collections1.insert(ins);
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 2,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
```

Command Prompt - mongo

```
> db.collection1.find()
{ "_id" : ObjectId("642c26ff6b92d8fcf8971a4c"), "id" : 1, "Name" : "Rajat" }
{ "_id" : ObjectId("642c275e6b92d8fcf8971a4d"), "id" : 2, "name" : "Raj", "course" : [ { "name" : "CS", "duration" : 7 },
{ "name" : "Java", "duration" : 5 } ] }
> db.collection1.find().pretty()
{ "_id" : ObjectId("642c26ff6b92d8fcf8971a4c"), "id" : 1, "Name" : "Rajat" }
{
  "_id" : ObjectId("642c275e6b92d8fcf8971a4d"),
  "id" : 2,
  "name" : "Raj",
  "course" : [
    {
      "name" : "CS",
      "duration" : 7
    },
    {
      "name" : "Java",
      "duration" : 5
    }
  ]
}
> db.collection1.find({"id":{"$gt:1}}).pretty()
{
  "_id" : ObjectId("642c275e6b92d8fcf8971a4d"),
  "id" : 2,
  "name" : "Raj",
  "course" : [
    {
      "name" : "CS",
      "duration" : 7
    },
    {
      "name" : "Java",
      "duration" : 5
    }
  ]
}
```



```

> db.collection1.find({"id":{"$lt:2}}).pretty()
{ "_id" : ObjectId("642c26ff6b92d8fcf8971a4c"), "id" : 1, "Name" : "Rajat" }
> db.collection1.update({"Name":"Rajat"},{$set:{"Name":"Anikey"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.collection1.find({},{"name":1})
{ "_id" : ObjectId("642c26ff6b92d8fcf8971a4c") }
{ "_id" : ObjectId("642c275e6b92d8fcf8971a4d"), "name" : "Raj" }
> db.collection1.find()
{ "_id" : ObjectId("642c26ff6b92d8fcf8971a4c"), "id" : 1, "Name" : "Anikey" }
{ "_id" : ObjectId("642c275e6b92d8fcf8971a4d"), "id" : 2, "name" : "Raj", "course" : [ { "name" : "CS", "duration" : 7 }, { "name" : "Java", "duration" : 5 } ] }
> db.collection1.find().sort({id:-1})
{ "_id" : ObjectId("642c275e6b92d8fcf8971a4d"), "id" : 2, "name" : "Raj", "course" : [ { "name" : "CS", "duration" : 7 }, { "name" : "Java", "duration" : 5 } ] }
{ "_id" : ObjectId("642c26ff6b92d8fcf8971a4c"), "id" : 1, "Name" : "Anikey" }
>

```

Practical No. - 3

Aim: To perform practical of Principal Component Analysis(PCA).

Program Code:

```
data_iris <- iris[1:4]
cov_data <- cov(data_iris)
Eigen_data <- eigen(cov_data)
PCA_data <- princomp(data_iris,cor = "False")
Eigen_data$values

PCA_data$sdev^2

PCA_data$loadings[,1:4]

Eigen_data$vectors

summary(PCA_data)

biplot(PCA_data)
```

```
screepplot(PCA_data,type = 'lines')
model2 = PCA_data$loadings[,1]
model2_scores <- as.matrix(data_iris)%*%model2
```

```
library(class)
install.packages("e1071")
library(e1071)
```

```
mod1 <- naiveBayes(iris[,1:4],iris[,5])
mod2 <- naiveBayes(model2_scores,iris[,5])
```

```
table(predict(mod1,iris[,1:4]),iris[,5])
```

```
table(predict(mod2,model2_scores),iris[,5])
```

Conclusion: Practical of Principal Component Analysis(PCA) has been executed successfully.

Practical No. - 4

Aim: To perform practical of Clustering.

Program Code:

```
install.packages("ggplot2")
library(ggplot2)

scatter <- ggplot(data=iris,aes(x=Sepal.Length,y=Sepal.Width))
scatter + geom_point(aes(color=Species,shape=Species))+
theme_bw()+
xlab("Sepal Length")+ylab("Sepal Width")+
ggtitle("Sepal Length-Width")

ggplot(data=iris,aes(Sepal.Length,fill=Species))+
theme_bw()+
geom_density(alpha=0.25)+
labs(x="Sepal.Length",title="Species vs Sepal Length")

vol <- ggplot(data=iris,aes(x=Sepal.Length))
vol + stat_density(aes(ymax=..density..,ymin=-
..density..,fill=Species,color=Species),geom="ribbon",position="identity")+
facet_grid(~Species)+coord_flip()+theme_bw()+labs(x="Sepal
Length",title="Species vs Sepal Length")
```

```

vol <- ggplot(data=iris,aes(x=Sepal.Width))
vol + stat_density(aes(ymax=..density..,ymin=-
..density..,fill=Species,color=Species),geom="ribbon",position="identity")+
facet_grid(.~Species)+coord_flip()+theme_bw()+labs(x="Sepal Width",title="Species
vs Sepal Width")

```

```

irisData <- iris[,1:4]
totalwSS<-c()
for(i in 1:15)
{ clusterIRIS<- kmeans(irisData,centers = i)
totalwSS[i] <-clusterIRIS$tot.withinss}

```

```

plot(x=1:15,y=totalwSS,type="b",xlab="Number of Clusters",ylab="Within groups
sum-of-squares")

```

```

install.packages("NbClust")
library(NbClust)
par(mar=c(2,2,2,2))
nb<-NbClust(irisData,method="kmeans")

```

```

hist(nb$Best.nc[1,],breaks=15,main="Histogram for Number of Clusters")

```

```
install.packages("vegan")  
library(vegan)
```

```
modelData<-cascadeKM(irisData,1,10,iter=100)  
plot(modelData,sortg=TRUE)
```

```
modelData$results[2,]
```

```
which.max(modelData$results[2,])
```

```
library(cluster)  
cl<-kmeans(iris[,-5],2)  
dis<-dist(iris[,-5])^2  
sil=silhouette(cl$cluster,dis)  
plot(sil,main="Clustering Data with silhoutte plot using 2  
Clusters",col=c("cyan","blue"))
```

```
library(cluster)  
cl<-kmeans(iris[,-5],8)  
dis<-dist(iris[,-5])^2  
sil=silhouette(cl$cluster,dis)  
plot(sil,main="Clustering Data with silhoutte plot using 8  
Clusters",col=c("cyan","blue","orange","yellow","red","gray","green","maroon"))
```

```
install.packages("factoextra")
library(factoextra)
install.packages("clustertend")
library(clustertend)

genx<-function(x){
  runif(length(x),min(x),(max(x)))}
random_df<-apply(iris[,-5],2,genx)
random_df<-as.data.frame(random_df)
iris[,-5]<-scale(iris[,-5])
random_df<-scale(random_df)
res<-get_clust_tendency(iris[,-5],n=nrow(iris)-1,graph=FALSE)
res$hopkins_stat

hopkins(iris[,-5],n=nrow(iris)-1)

res<-get_clust_tendency(random_df,n=nrow(random_df)-1,graph=FALSE)
res$hopkins_stat
```

Conclusion: Practical of Clustering has been executed successfully.

Practical No. - 5

Aim: To perform practical of Time-series forecasting.

Program Code:

```
data(AirPassengers)
class(AirPassengers)

start(AirPassengers)

end(AirPassengers)

frequency(AirPassengers)

summary(AirPassengers)


plot(AirPassengers)
abline(reg=lm(AirPassengers~time(AirPassengers)))


cycle(AirPassengers)
```



```
plot(aggregate(AirPassengers,FUN=mean))
```

```
boxplot(AirPassengers~cycle(AirPassengers))
```

```
acf(log(AirPassengers))
```

```
(fit<-arima(log(AirPassengers),c(0,1,1),seasonal=list(order=c(0,1,1),period=12)))
```

```
pred<-predict(fit,n.ahead=10*12)
```

```
ts.plot(AirPassengers,2.718^pred$pred,log="y",lty=c(1,3))
```

Conclusion: Practical of Time-series forecasting has been executed successfully.

Practical No. - 6

Aim: To perform practical of Simple/Multiple Linear Regression.

Program Code:

```
lsfit(iris$Petal.Length,iris$Petal.Width)$coefficients
```

```
plot(iris$Petal.Length,iris$Petal.Width,pch=21,bg=c("red","green3","blue")[unclass(iris$Species)],main="Iris Data",xlab="Petal length",ylab="Petal width")  
abline(lsfit(iris$Petal.Length,iris$Petal.Width)$coefficients,col="black")
```

```
lm(Petal.Width~Petal.Length,data=iris)$coefficients
```

```
plot(iris$Petal.Length,iris$Petal.Width,pch=21,bg=c("red","green3","blue")[unclass(iris$Species)],main="Iris Data",xlab="Petal length",ylab="Petal width")  
abline(lm(Petal.Width~Petal.Length,data=iris)$coefficients,col="black")
```

```
summary(lm(Petal.Width~Petal.Length,data=iris))
```

```
plot(iris$Sepal.Width,iris$Sepal.Length,pch=21,bg=c("red","green3","blue")[unclass(iris$Species)],main="Iris Data",xlab="Sepal Width",ylab="Sepal Length")  
abline(lm(Sepal.Length~Sepal.Width,data=iris)$coefficients,col="black")
```

```
summary(lm(Sepal.Length~Sepal.Width,data=iris))
```

```
plot(iris$Sepal.Width,iris$Sepal.Length,pch=21,bg=c("red","green3","blue")[unclass(
iris$Species)],main="Iris Data",xlab="Petal length",ylab="Sepal length")
```

```
abline(lm(Sepal.Length~Sepal.Width,data=iris)$coefficients,col="black")
abline(lm(Sepal.Length~Sepal.Width,
  data=iris[which(iris$Species=="setosa"),])$coefficients,col="red")
abline(lm(Sepal.Length~Sepal.Width
  data=iris[which(iris$Species=="versicolor"),])$coefficients,col="green3")
abline(lm(Sepal.Length~Sepal.Width,
  data=iris[which(iris$Species=="virginica"),])$coefficients,col="blue")
```

```
lm(Sepal.Length~Sepal.Width,data=iris[which(iris$Species=="setosa"),])$coefficients
```

```
lm(Sepal.Length~Sepal.Width,data=iris[which(iris$Species=="versicolor"),])$coefficients
```

```
lm(Sepal.Length~Sepal.Width,data=iris[which(iris$Species=="virginica"),])$coefficients
```

```
lm(Sepal.Length~Sepal.Width:Species+Species-1,data=iris)$coefficients
```

```
summary(lm(Sepal.Length~Sepal.Width:Species+Species-1,data=iris))
```

```
summary(step(lm(Sepal.Length~Sepal.Width*species,data=iris)))
```

```
lm(Sepal.Length~Sepal.Width:Species+Species-1,data=iris)$coefficients
```

```
lm(Sepal.Length~Sepal.Width:Species+Species,data=iris)$coefficients
```

Conclusion: Practical of Simple/Multiple Linear Regression has been executed successfully.

Practical No. - 7

Aim: To perform practical of Logistics Regression.

Program Code:

```
library(datasets)
ir_data<-iris
head(ir_data)

str(ir_data)

levels(ir_data$species)

sum(is.na(ir_data))

ir_data<-ir_data[1:100,]
set.seed(100)
samp<-sample(1:100,80)
ir_test<-ir_data[samp,]
ir_ctrl<-ir_data[-samp,]
install.packages("ggplot2")
library(ggplot2)
install.packages("GGally")
library(GGally)
ggpairs(ir_test)
```

```
y<-ir_test$Species;x<-ir_test$Sepal.Length  
glfit<-glm(y~x,family='binomial')  
summary(glfit)
```

```
newdata<-data.frame(x=ir_ctrl$Sepal.Length)  
predicted_val<-predict(glfit,newdata,type="response")  
prediction<-data.frame(ir_ctrl$Sepal.Length,ir_ctrl$Species,predicted_val)  
prediction
```

```
qplot(prediction[,1],round(prediction[,3]),col=prediction[,2],xlab='Sepal  
Length',ylab='prediction using Logistic Reg.')
```

Conclusion: Practical of Logistics Regression has been executed successfully.

Practical No. - 8

Aim: To perform practical of Hypothesis testing.

Program Code:

```
x=c(6.2,6.6,7.1,7.4,7.6,7.9,8,8.3,8.4,8.5,8.6,  
8.8,8.8,9.1,9.2,9.4,9.4,9.7,9.9,10.2,10.4,10.8,11.3,11.9)  
t.test(x-9,alternative ="two.sided",conf.level = 0.95)
```

```
x=c(418,421,421,422,425,427,431,434,437,439,446,447,448,453,454,463,465)  
y=c(429,430,430,431,36,437,440,441,445,446,447)
```

```
test2<-t.test(x,y,alternative = "two.sided",mu=0,var.equal=F,conf.level=0.95)  
test2
```

Conclusion: Practical of Hypothesis testing has been executed successfully.

Practical No. - 9

Aim: To perform practical of Analysis of Variance .

Program Code:

```
y1=c(18.2,20.1,17.6,16.8,18.8,19.7,19.1)
y2=c(17.4,18.7,19.1,16.4,15.9,18.4,17.7)
y3=c(15.2,18.8,17.7,16.5,15.9,17.1,16.7)
y=c(y1,y2,y3)
n=rep(7,3)
n
```

```
group =rep(1:3,n)
group
```

```
tmp=tapply(y,group,stem)
```

```
stem(y)
```

```
tmpfn=function(x){sum=sum(x),mean=mean(x),var=var(x),n=length(x)}  
tapply(y,group,tmpfn)
```

```
tmpfn(y)
```

```
data=data.frame(y=y,group=factor(group))  
fit=lm(y~group,data)  
anova(fit)
```

```
df=anova(fit)[,"Df"]  
names(df)=c("trt","err")  
df
```

```
alpha=c(0.05,0.01)  
qf(alpha,df["trt"],df["err"],lower.tail=FALSE)
```

```
anova(fit)["Residuals","Sum Sq"]
```

```
anova(fit)["Residuals","Sum Sq"]/qchisq(c(0.025,0.975),18,lower.tail=FALSE)
```

Conclusion: Practical of Analysis of Variance has been executed successfully.

Practical No. - 10

Aim: To perform practical of Decision Tree .

Program Code:

```
mydata<-data.frame(iris)
attach(mydata)

install.packages("rpart")
library(rpart)
model<-
rpart(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata,method="class")
plot(model)
text(model,use.n=TRUE,all=TRUE,cex=0.8)
```

```
install.packages("tree")
library(tree)
model1<-
tree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata,method="class",split="gini")
plot(model1)
text(model1,all=TRUE,cex=0.6)
```

```
install.packages("party")
library(party)
model2<-
ctree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata)
plot(model2)
```

```
library(tree)
mydata<-data.frame(iris)
attach
```

```
modell<-
tree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata,met
hod="class",control=tree.control(nobs=150,mincut=10))
plot(modell)
text(modell,all=TRUE,cex=0.6)
```

```
predict(modell,iris)(mydata)
```

```
model2<-
ctree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata,co
ntrols=ctree_control(maxdepth=2))
plot(model2)
```

Conclusion: Practical of Decision tree has been executed successfully.