Hypothesis testing Methods

1] # One sample t-test in R
```
data <- c( . . . . . )
t.test (data, mu = 30)
```

2] # Two sample t-test
```
sample1 <- c ( . . . . )
sample 2 <- c ( . . . . . )
t.test (sample1 , sample2)
```

3] # Paired sample test
```
before <- c ( . . . . . )
after <- c ( . . . . . )
t.test (before, after, paired = TRUE)
```

4] # ANOVA (F-Test)
```
group1 = c ( . . . . . )
group2= c( . . . . . )
group3 = c( . . . . . )
cg= data.frame (cbind (group1, group2, group3))
cg
boxplot (cg)
Stacked-g = stack(cg)
stacked-g
av= aov (values ~ ind , data = stacked-g)
summary (av)
```

30

★ R Markdown

Implementing decision Tree Classifier

```
library(rpart)
library(datasets)
data("iris")
set.seed(123)
trainIndex <- sample(nrow(iris), 0.7 * nrow(iris))
trainData <- iris[trainIndex, ]
testData <- iris[-trainIndex, ]
model <- rpart(Species ~ ., data = trainData, method = "class")
predictions <- predict(model, testData, type = "class")
accuracy <- sum(predictions == testData$Species) / nrow(testData)
cat("Accuracy:", accuracy)
```

★ Uses of Mongodb

Mongodb is a NoSQL document-oriented database that is flexible, scalable and high performing.

1] Web Apps :- mostly for apps that require to store and process large volumes of unstructured data eg. social media, e-commerce.

2] Big Data :- Its distributed architecture and automatic sharding makes it easy to scale horizontally as data volume increases.

3] Real-Time Analytics :- Mongodb's aggegration framework allows developers to perform complex data analysis in real time.

Decision Tree Classifier

Define Problem:

- We want to build a decision tree classifier that can predict the species of an iris flower based on its sepal length, sepal width, petal length and petal width.

- How to evaluate the algorithm?
- We can evaluate the performance of the decision tree classifier using metrics such as accuracy, precision, recall and F1 score.
- You can also use confusion matrix for evaluation and rpart.plot function for visualization.

1] What is dimension reduction?

→ Dimension reduction is a technique used to reduce the number of features or variables in a dataset while retaining the most important information. It involves transforming high-dimensional data into a lower dimensional space while preserving the structure and relationships between the data points.

2] What are different methods for dimension reduction?

a) PCA
b) LDA - Linear discriminant Analysis
c) t-distributed Matrix Factorization (NMF)
d) Independent Component Analysis (ICA)

3] Why dimension reduction is important?

→ It is important because high-dimensional datasets can be difficult to work with and may suffer from the curse of dimensionality.

This can result in overfitting, increased computational complexity and decreased model performance.

PRINCIPAL COMPONENT ANALYSIS

click on packages and set cran mirror (other → USA IN)
packages → install packages → install package FactoMineR

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

create excel sheet (Math, Eng, Art each 5 entries) as student.csv.

```
x=read.csv("students.csv")
x
cov-mat = cov(x)
cov-mat
ex=eigen(cov-mat)
ex
datapca = PCA(x, ncp=3, graph=TRUE)
datapca $eig
datapca $var
datapca $var $coord
fviz-screenplot(datapca, addlabels=TRUE, ylim=c(0,50))
install.packages('Factoextra', repos="http://cran.us.r-project.org")
library("factoextra")
fviz-screenplot(datapca, addlabels=TRUE, ylim=c(0,50))
head(iris)
x=iris[,-5]
x
cov-iris = cov(x)
cov-iris
irispca=PCA(x, ncp=3, graph=TRUE)
irispca
summary(irispca)
```

K-MEANS CLUSTERING -

```
df= read.csv("AGE.csv")
df
plot(df)
boxplot(df)
set.seed(20)
cl = kmeans(df[,1:2], 3)
cl
iris
view(iris)
head(iris)
summary(iris)
plot(iris)
plot(iris[,3:4])
kmeans cl = kmeans(iris[,3:4], 3)
kmeans cl
table(kmeans cl$cluster, iris$Species)
boxplot(iris)
```

1] what is clustering?
, clustering is a technique used in unsupervised learning to group similar data points together based on their features or characteristics. The goal of clustering is to find patterns or structures in the data that may not be immediately obvious and to identify natural groupings of data points.

2] steps :-
1] Initialize the no. of cluster (k) and randomly assign each data point to a cluster.
2] Calculate the centroid of each cluster.
3] Assign each data point to the cluster with closest centroid.
4] Recalculate the centroid of each cluster based on new assignment.
5] Repeat 3-4 until cluster assignments no longer change.

3] How to determine the value of k?
The best value of k can be determined using the elbow method.

1] What is time-series data? Give example.

Time-series data is a type of data that is collected over time at regular interval. It consists of a sequence of data points, where each points represents a measurement or observation taken at a specific time. Time-series data can be used to analyze trends, patterns and behaviour that occur over time.

Eg: stock price, weather data

2] Problem:-

The problem is to forecast future values of a time-series based on past observations.

Time-series forecasting involes using statistical or machine learning methods to analyze patterns and trends in the data and make predictions about future values.

3] Evaluate

We can evalute the performance of the time-series forecasting algorithm using metrics such as the mean absolute error (MAE) and root mean squared error (RMSE).

TIME SERIES

```
library(forecast)
data(AirPassengers)
trainData <- window(AirPassengers, end = c(1959, 12))
testData <- window(AirPassengers, start = c(1960, 1))
model <- auto.arima(trainData)
prediction <- forecast(model, h = length(testData))
plot(AirPassengers, type = "l", xlim = c(1949, 1961), ylim =
     c(0, 700)) lines(predictions$mean, col = "red")
accuracy <- accuracy(predictions, testData)
print(accuracy)
```

a) #Deleting
```
titanic <- read.csv("titanic_train.csv")
clean_titanic <- na.omit(titanic)
```
b) # Replacing
```
titanic <- read.csv("titanic_train.csv")
titanic$Age <- replace(titanic$Age, is.na(titanic$Age),0)
```
c) Imputing missing values:
```
library(Hmisc)
titanic <- read.csv("titanic_train.csv")
titanic$Age <- impute(titanic$Age, mean)
```
d) Categorical variables:
```
titanic <- read.csv("titanic_train.csv")
titanic$Sex <- factor(titanic$sex)
```
e) Working with outliers.
```
titanic <- read.csv("titanic_train.csv")
boxplot(titanic$Age)
outliers <- boxplot.stats(titatnic$Age)$out
titanic$Age <- titanic$Age[!titanic$Age %in% outliers]
```

Multiple Linear Regression

a) Problem - predict salary of individual based on their years of experience and level of education.

b) Null hypothesis - There is no significant relatn betw salary and years of expereience and level of education

c) salary_data <- read.csv("salary_data.csv")
sum(is.na(salary_data))
str(salary_data)
salary_data$education_level <- as.factor(salary_data$education_level)
summary(salary_data)

d) model <- lm(salary ~ years_experience + education_level, data = salary_d
summary(model)

Simple Linear Regression :

a) Problem statement -
We use Simple linear regression to predict the relationship between no. of hours studied and exam scores of students

b) Null hypothesis -
There is no significant linear relationship between the no. of hours studied and exam scores.

c)
```
data <- read.csv("study_data_raw.csv")
sum(is.na(data$Hours))                    # missing val
sum(is.na(data$score))
data$Hours[is.na(data$Hours)] <- mean(data$Hours, na.rm=TRUE)
" $score[ $score $score )
data$Gender <- as.numeric(factor(data$Gender, levels=c("F","M"))
                                                  labels
```

d)
```
model <- lm(score ~ Hours, data=data)
```

e)
```
new_data <- data.frame(Hours = c(5,7,9))
predictions <- predict(model, newdata = new_data)
print(predictions)
```

f)
```
summary(model)
plot(data$Hours, data$Score)
abline(model, col="red")
```