1. Demonstrate following preprocessing operations using R/Python

```
x=read.csv("C:/Users/HP/Documents/DS/titanic_train.csv")
X
head(x)
library(caTools)
```

Created duplicate dataset with var dup

```
dup=x
View(dup)
```

a. Deleting missing values

na.omit() is used to delete the rows of missing data

```
missing_data<-na.omit(dup)
```

b. Replacing missing values

Replacing the missing value from the age column

```
age=mean(dup$Age, na.rm=TRUE)
dup$Age=replace(dup$Age, is.na(dup$Age),age)
View(dup)
```

c. Imputing missing values

Imputing missing values means using a statistical technique to fill in missing values with estimated values.

In the images please use dup as duplicated data set

#PMM (Predictive Mean Matching) – For numeric variables

```
> im_dt<-mice(y[,c("Age","Pclass","Sex")],method="pmm")

 iter imp variable
  1   1  Age
  1   2  Age
  1   3  Age
  1   4  Age
  1   5  Age
  2   1  Age
  2   2  Age
  2   3  Age
  2   4  Age
  2   5  Age
  3   1  Age
  3   2  Age
  3   3  Age
  3   4  Age
  3   5  Age
  4   1  Age
  4   2  Age
  4   3  Age
  4   4  Age
  4   5  Age
  5   1  Age
  5   2  Age
  5   3  Age
  5   4  Age
  5   5  Age
Warning message:
Number of logged events: 1
> y$Age<-im_dt$imp$Age
Error in `$<-.data.frame`(`*tmp*`, Age, value = list(`1` = c(36, 24, 36,  :
  replacement has 177 rows, data has 891
> View(y)

> y$Age=replace(y$Age, is.na(y$Age),age)
> View(y)
> im_dt<-mice(y[,c("Age","Pclass","Sex")],method="pmm")

 iter imp variable
  1   1
  1   2
  1   3
  1   4
  1   5
  2   1
  2   2
  2   3
  2   4
  2   5
  3   1
  3   2
  3   3
  3   4
  3   5
  4   1
  4   2
  4   3
  4   4
  4   5
  5   1
  5   2
  5   3
  5   4
  5   5
Warning message:
Number of logged events: 1
> y$Age<-im_dt$imp$Age
Error in `$<-.data.frame`(`*tmp*`, Age, value = list(`1` = logical(0),  :
  replacement has 0 rows, data has 891
```

d. Work with categorical variables

If we check the levels of the SEX column from the dataset

Categorical variable we get only on the factor value to distinguishing into gender

dup$Sex<- as.factor(dup$Sex)//this will convert the string to factor variables

```
[639] female male   male   femal
[661] male   male   male   male
[683] male   male   male   male
[705] male   male   female male
[727] female female male   femal
[749] male   male   female male
[771] male   male   female male
[793] female male   male   male
[815] male   male   female male
[837] male   male   male   male
[859] female male   male   male
[881] female male   female male
Levels: female male
```

dup$Sex <- relevel(dup$Sex, ref="male") // the levels will start with male

```
[727] female female male   female
[749] male   male   female male
[771] male   male   female male
[793] female male   male   male
[815] male   male   female male
[837] male   male   male   male
[859] female male   male   male
[881] female male   female male
Levels: male female
```

e. Work with outlier

Outliers are data points that don't fit the pattern of the rest of the data set. The best way to detect the outliers in the given data set is to plot the boxplot of the data set and the point located outside the box in the boxplot are all the outliers in the data set.

boxplot(dup$Age)

boxplot.stats(dup$Age)

```
> boxplot.stats(dup$Age)
$stats
[1]  3.00000 22.00000 29.69912 35.00000 54.00000

$n
[1] 891

$conf
[1] 29.01100 30.38723

$out
 [1]  2.00 58.00 55.00  2.00 66.00 65.00  0.83 59.00 71.00 70.50  2.00 55.
[27] 61.00  2.00 60.00  1.00  1.00 64.00 65.00 56.00  0.75  2.00 63.00 58.
[53] 70.00 60.00 60.00 70.00  0.67 57.00  1.00  0.42  2.00  1.00 62.00  0.
```

Demonstrate Simple Linear Regression model using R/Python

Use Dataset: Use any suitable dataset

a) Define Problem Statement

We want to understand the relationship between the number of cylinders in a car and its fuel efficiency, measured in Miles per Gallon (mpg). We want to build a Simple Linear Regression model to predict the mpg based on the number of cylinders.

b) Define Null Hypothesis

The null hypothesis is that there is no significant linear relationship between the number of cylinders and the fuel efficiency in the mtcars dataset.

c) Perform Pre-processing operations on dataset

```
x=mtcars #import inbuilt datasets mtcars
x
y=is.null() #checking the null values on the mtcars
y
im=mice(y[,c("mpg","cyl","disp","hp","vs","am")])#checking the empty columns
```

For this dataset we can conclude that there is no need to check the missing data due to the data is fulfilled the criteria (all data are completed)

d) Prepare Model

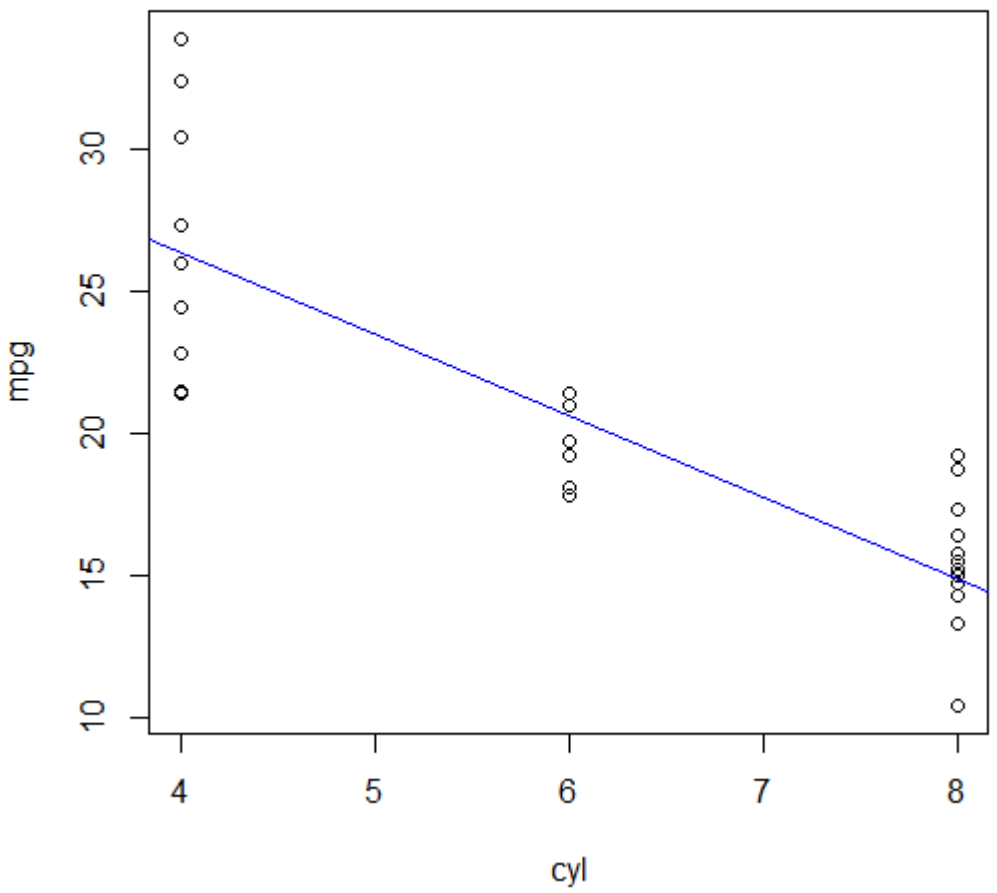
```
model=lm(mpg~cyl,data=x )
model
```

e) Use Model for prediction

```
pd=predict(model)
pd
```

f) Evaluate Model

```
summary(model) pvalue
plot(mpg~cyl, data=x) or plot(x$mpg~x$cyl)
abline(model, col=”blue”)#before this line we have to call the plot()
```


mpg
10
15
20
25
30
4
5
6
7
8
cyl

Demonstrate Multiple Linear Regression in mtcars dataset using R

a) Define Problem Statement

We want to understand the relationship between several variables (cylinders, displacement, horsepower, weight, acceleration, and model year) in a car and its fuel efficiency, measured in Miles per Gallon (mpg). We want to build a Multiple Linear Regression model to predict the mpg based on these variables.

b) Define Null Hypothesis

The null hypothesis is that there is no significant linear relationship between (the column names which u r going to use) and the fuel efficiency in the mtcars dataset.

c) Perform Pre-processing operations on dataset

Same as above

d) Prepare Model

```
model=lm(mpg~cyl+disp+hp+wt,data=x)
model
```

e) Use Model for prediction

```
predict(model)
```

f) Evaluate Model

```
summary(model)
plot(mpg~cyl, data=x) or plot(x$mpg~x$cyl)
abline(model, col=”blue”)#before this line we have to call the plot()
```

Perform following tasks

a) Create any R markdown document implementing any machine learning algorithm of your choice.

1. Open RStudio.
2. Click on File > New File > R Markdown.
3. Choose a title for your document and select the output format (HTML, PDF, or Word).
4. Choose a default template or create your own custom template.
5. Write the content of your document in R Markdown format, including any code chunks for machine learning algorithms.
6. Knit the document to produce the output.

b) Upload it in your RStudio account

1. Click on the "Publish" button in the top-right corner of the RStudio window.
2. Choose "RPubs" as the publishing service.
3. Enter a title and description for your document.
4. Click on "Publish".
5. After the upload is complete, you will receive a link to your published document.

https://rpubs.com/BasantM/linearreg

Demonstrate Logistic Regression Using R

a) Define Problem Statement

We want to determine whether a car is "efficient" or "not efficient" based on its characteristics, such as engine displacement, horsepower, weight, etc

b) Define Null Hypothesis

There is no significant relationship between the car characteristics and its efficiency.

c) Is it classification or prediction problem. Explain.

Classification problem , this problem is classified into efficient and not efficient

d) Perform Pre-processing operations on dataset

```
x=mtcars

x$mpg01 <- ifelse(x$mpg > 20 ,1 ,0)//x$mpg is greater than 20 print 1 or else 0

x

summary(x)
```

e) Prepare Model

```
model<-glm(mpg01~. , data=x)

model
```

f) Use Model for prediction

```
pred=prediction(model,type=”response”)

# type=”response/link”gives same O/P and terms gives the all columns prediction value

pred
```

g) Evaluate Model

```
 pr<-ifelse(pred> 0.5,1,0)

 pr

> ac<-mean(pr==x$mpg01)
```

```
> ac

[1] 0.96875

> table(pr,x$mpg01)

pr   0  1

  0 18  1

  1  0 13

>
```

This above table () gives the result of confusion matrix . So the above programs conclude that there is 18 cars are not efficient and 1 car is predicted efficient but actually wasn't and 13 cars are efficient.

Perform following Hypothesis testing methods using R

a) One sample t-test

x=mtcars

x

t.test(x$mpg)

```
> t.test(x$mpg)

        One Sample t-test

data:  x$mpg
t = 18.857, df = 31, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 17.91768 22.26357
sample estimates:
mean of x
 20.09062
```

b) Two sampled t-test

t.test(x$mpg~x$vs)#2nd parameter always has 2 levels means(0,1)/true/false/yes/no

```
> t.test(x$mpg~x$vs)

        Welch Two Sample t-test

data:  x$mpg by x$vs
t = -4.6671, df = 22.716, p-value = 0.0001098
alternative hypothesis: true difference in means between group 0 an$
95 percent confidence interval:
```

c) Paired sampled t-test

t.test(xmpg, xwt, paired=TRUE)

```
> t.test(x$mpg,x$wt, paired=TRUE)

        Paired t-test

data:  x$mpg and x$wt
t = 13.847, df = 31, p-value = 8.096e-15
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 14.38815 19.35860
sample estimates:
mean difference
       16.87337
```

d) ANOVA (F-TEST)

model<-lm(mpg~cyl , data=x)

anova(model)

```
> model<-lm(mpg~cyl,data=x)
> anova(model)
Analysis of Variance Table

Response: mpg
           Df Sum Sq Mean Sq F value    Pr(>F)
cyl         1 817.71  817.71  79.561 6.113e-10 ***
Residuals 30 308.33   10.28
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Implement PCAusing R/Python

a) What is Dimension reduction?

Dimension reduction is the process of reducing the number of variables or features in a dataset while retaining as much information as possible

b) What are different methods for dimension reduction?

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Independent Component Analysis (ICA)

c) Why Dimension reduction is important?

- It reduces the complexity of the data, making it easier to analyze and interpret.
- It can facilitate data compression and storage.

d) Implement PCA Algorithm on suitable dataset.

```
x=iris[,-5]

cm=cov(x)

cm

ex=eigen(cm)

ex

dp=PCA(x,ncp=3,graph=TRUE)# it will give graph

dp

dp$var

summary(dp)
```

e) How to evaluate the above algorithm?

We will get more information about the principal component using summary(). We can evaluate the effectiveness of PCA by comparing the performance of ml on the original datasets. If the performance of original data is similar to the reduced data after the reduction, then we can conclude that PCA was successfully reduced the iris dataset.

Perform following task using MongoDB

a) Create a suitable database in MongoDB.

use basantdb//created db

b) Create a suitable collection in the database.

db.createCollection(“user”)//

c) Insert 3 documents in the above collection.

([{}])

db.user.insertMany([{

name:”Basant”,

surname:”Mandal”,

mob:9004523829

},

{

name:”Basant2”,

surname:”Mandal”,

mob:7021724649

},

{

name:”Ashish”,

surname:”Mandal”,

mob:7021724649

}

])

d) Perform CRUD operation on documents inserted in collection.

db.user.insertOne({

name:”Basant3”,

surname:”Mandal”,

mob:7021724649

})

db.user.find()//to read

db.user.updateOne({name:”Basant3”},{$set:{name:”Basant ”}})

db.user.find()

db.user.deleteOne({name:”Basant2”})

e) What is use(s) of MongoDB database?

It is used in web applications,mobile applications, IoT platforms,RealTime Analytics, BigData

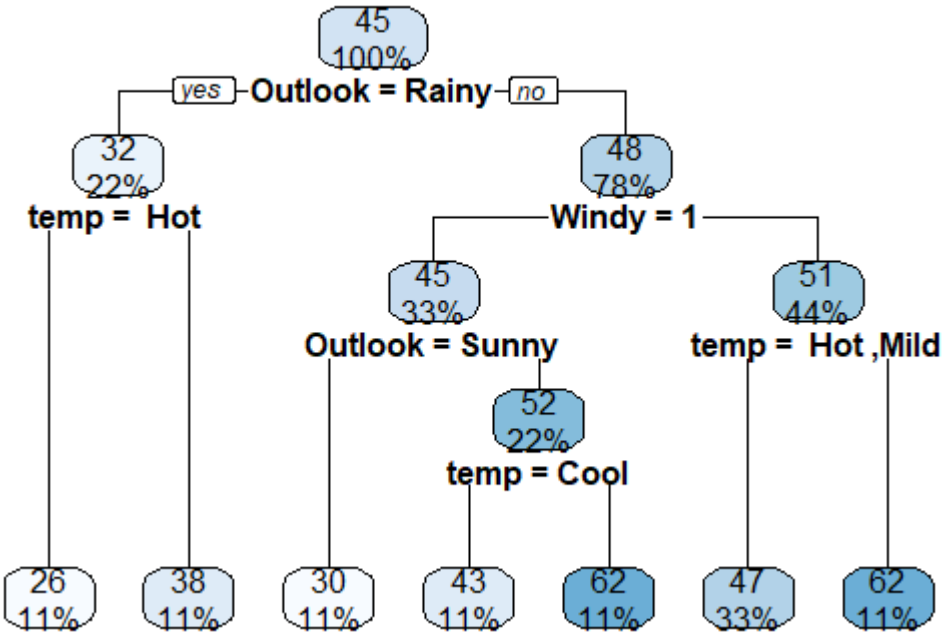
Implement Decision Tree Algorithm (Classifier) using R

a) Define Problem

In this problem, we will use the mtcars dataset to build a decision tree classifier that predicts whether a car has high or low mileage per gallon (mpg) based on its characteristics such as horsepower, weight, and number of cylinders

b) Implement Decision Tree Algorithm on suitable dataset.

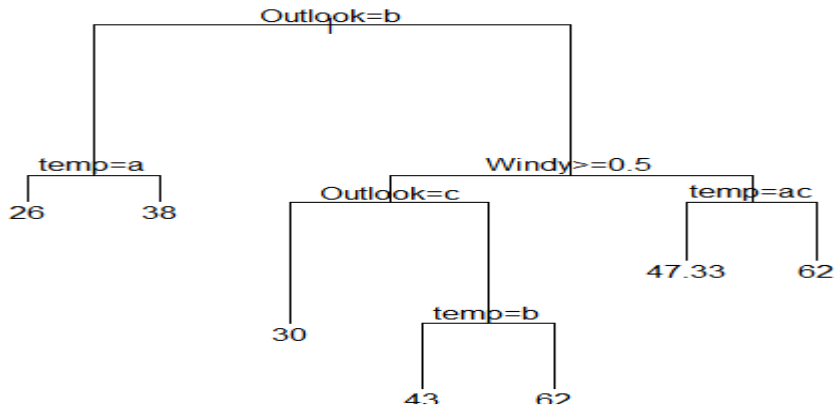
```
library(rpart)

library(rpart.plot)

> x=read.csv("C:/Users/HP/Documents/DS/weather2.csv")

> x

> s=sample(nrow(x),.7*nrow(x))

> tr=x[s,]

> ts=x[-s,]

> ts

> dt=rpart(Hours.Played ~ . ,data=tr)

>rpart.plot(dt)
```



or

> plot(dt)

> text(dt)



c) How to evaluate the above algorithm?

> pred=predict(dt,data=tr)

> pred

> table(pred, tr$Hours.Played)

Implement K-means clustering using R

a) What is clustering?

Clustering is a type of unsupervised learning in which the goal is to group similar data points together based on their characteristics.

b) Write steps of K-means clustering algorithm.

The K-means clustering algorithm involves the following steps:

1. Choose the number of clusters K.
2. Randomly select K points from the dataset to serve as the initial centroids of the clusters.
3. Assign each data point to the cluster whose centroid is closest to it.
4. Recalculate the centroids of each cluster based on the mean of the data points assigned to that cluster.
5. Repeat steps 3 and 4 until convergence is achieved, i.e., the assignments of data points to clusters do not change.

c) How to determine best value of k?

The best value of K for K-means clustering can be determined using techniques such as the elbow method and the silhouette method. Using WCSS (within-cluster sum of squares) we get the K value.

d) Implement K-means clustering on suitable dataset.

```
> x=iris
> boxplot()
> head(x)
> summary(x)
> km<-kmeans(x,centers=3)
> km
```

e) How to evaluate the above algorithm?

```
> km$cluster
> x$Species
> table(km$cluster,x$Species)
```

Implement Time-series forecasting using R

a) What is time-series data? Give example.

Time-series data is a type of data where observations are recorded at equally spaced time intervals. Eg: Weather Data, website traffic data

b) Define the problem.

The problem in time-series forecasting is to predict future values of a variable based on its past values. This is useful in many applications such as finance, economics, and weather forecasting.

c) Implement Time-series forecasting on suitable dataset.

```
> x=table(AirPassengers)

> View(x)

> y=frequency(AirPassengers)

> y

> tsd=ts(AirPassengers,frequency=12)

> tsd

> plot(tsd)

> de=decompose(tsd,"multiplicative")

> plot(de)
```

d) How to evaluate the above algorithm?

```
> model<-arima(AirPassengers)
> model
> accuracy(model)
```