**DAY 4 – LAB ASSESSMENT**

**INDUSEKHAR**

**192124123**

1.Randomly Sample the iris dataset such as 80% data for training and 20% for test and

create Logistics regression with train data, use species as target and petals width and

length as feature variables , Predict the probability of the model using test data,  Create

Confusion matrix for above test model

**CODE:**

library(caret)

data(iris)

set.seed(123)

indexes <- createDataPartition(iris$Species, p = 0.8, list = FALSE)

training <- iris[indexes, ]

testing <- iris[-indexes, ]

model <- train(Species ~ Petal.Length + Petal.Width, data = training, method = "glm", family = "multinomial")

probs <- predict(model, testing, type = "prob")

preds <- predict(model, testing)

cm <- confusionMatrix(preds, testing$Species)

print(cm$table)

**2. (i)Write suitable R code to compute the mean, median ,mode of the following values**

**c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20)**

**(ii) Write R code to find 2nd  highest and 3 rd Lowest value of above problem**.

**Code:**

x <- c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20)

mean(x)

median(x)

mode(x)

x <- c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20)

second\_highest <- sort(unique(x), decreasing = TRUE)[2]

cat("The 2nd highest value is:", second\_highest, "\n")

third\_lowest <- sort(unique(x))[3]

cat("The 3rd lowest value is:", third\_lowest)

**o/p:**

> mean(values)

[1] 60

> median(values)

[1] 70

> mode(values)

[1] 80

The 2nd highest value is: 80

The 3rd lowest value is: 50

2. (i)Get the Summary Statistics of air quality dataset

(ii)Melt airquality data set and display as a long – format data?

(iii)Melt airquality data and specify month and day to be “ID variables”?

 (iv)Cast the molten airquality data set with respect to month and date features

 (v) Use cast function appropriately and compute the average of Ozone, Solar.R , Wind

and temperature per month?

INPUT:

summary(airquality)  
library(reshape2)  
airquality\_melt <- melt(airquality, id.vars = c("Month", "Day"))  
airquality\_melt2 <- melt(airquality, id.vars = c("Month", "Day"))  
airquality\_cast <- dcast(airquality\_melt, Month + Day ~ variable, mean)  
airquality\_mean <- dcast(airquality\_melt, Month ~ variable, mean)

3.(i) Find any missing values(na) in features and drop the missing values if its less than

10%

else replace that with  mean of that feature.

   (ii) Apply a linear regression algorithm using Least Squares Method on “Ozone” and

“Solar.R”

   (iii)Plot Scatter plot between Ozone and Solar and add regression line created by

above model

INPUT:

library(tidyverse)  
data("airquality")  
missing\_values <- airquality %>%   
 is.na() %>%   
 sum()  
if (sum(missing\_values) / nrow(airquality) < 0.1)   
{  
 airquality <- airquality %>%   
 drop\_  
model <- lm(Ozone ~ Solar.R, data = airquality)  
ggplot(airquality, aes(x = Solar.R, y = Ozone)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", formula = model, se = FALSE) +  
 labs(x = "Solar Radiation", y = "Ozone")

4. Load dataset named ChickWeight,

( i).Order the data frame, in ascending order by feature name “weight” grouped by

  feature

“diet” and Extract the last 6 records from order data frame.

 (ii).a Perform melting function based on “Chick&quot;, &quot;Time&quot;, &quot;Diet&quot;   features as ID

variables

 b. Perform cast function to display the mean value of weight grouped by Diet

 c. Perform cast function to display the mode of weight grouped by Diet

INPUT:

library(tidyverse)  
  
data("ChickWeight")  
  
ordered\_df <- ChickWeight %>%   
 group\_by(Diet) %>%   
 arrange(weight) %>%   
 slice\_tail(6)  
melted\_df <- ChickWeight %>%   
 melt(id.vars = c("Chick", "Time", "Diet"))  
mean\_df <- melted\_df %>%   
 cast(Diet ~ variable, mean)  
mode\_df <- melted\_df %>%   
 group\_by(Diet, value) %>%   
 summarize(n = n()) %>%   
 arrange(desc(n)) %>%   
 group\_by(Diet) %>%   
 slice\_head(1) %>%   
 select(Diet, value)

7. a.  Create Box plot for “weight” grouped by “Diet”

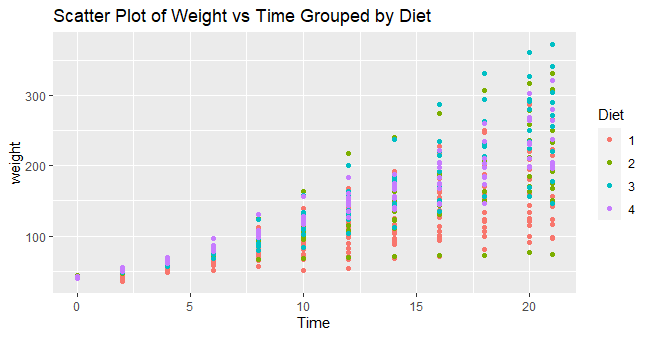
          b. Create a Histogram for “weight” features belong to Diet- 1 category

          c.  Create Scatter plot for “ weight” vs “Time” grouped by Diet

INPUT:

library(ggplot2)  
  
ggplot(ChickWeight, aes(x = Diet, y = weight)) +   
 geom\_boxplot() +   
 ggtitle("Box Plot of Weight Grouped by Diet")  
ggplot(ChickWeight[ChickWeight$Diet == 1, ], aes(x = weight)) +   
 geom\_histogram(fill = "blue", color = "black") +   
 ggtitle("Histogram of Weight for Diet 1")  
ggplot(ChickWeight, aes(x = Time, y = weight, color = Diet)) +   
 geom\_point() +   
 ggtitle("Scatter Plot of Weight vs Time Grouped by Diet")

OUTPUT:



8.a. Create multi regression model to find a weight of the chicken , by “Time” and

“Diet” as as

predictor variables

b. Predict weight for Time=10 a and c. Find the error in model for same

INPUT:

df <- data.frame(Time = c(...), Diet = c(...), Weight = c(...))  
model <- lm(Weight ~ Time + Diet, data = df)  
new\_data <- data.frame(Time = 10, Diet = 1)  
predicted\_weight <- predict(model, new\_data)  
residuals <- residuals(model)  
MSE <- mean(residuals^2)  
RMSE <- sqrt(MSE)93.-

9 .For this exercise, use the (built-in) dataset Titanic.

a. Draw a Bar chart to show details of “Survived” on the Titanic based on passenger

Class

b. Modify the above plot based on gender of people who survived

c. Draw histogram plot to show distribution of feature “Age”

INPUT:

data("Titanic")  
ggplot(Titanic, aes(x = Class, fill = Survived)) +  
 geom\_bar(position = "dodge") +  
 labs(x = "Passenger Class", y = "Number of Survivors", fill = "Survived") +  
 ggtitle("Survived on the Titanic Based on Passenger Class")  
  
ggplot(Titanic, aes(x = Class, fill = Survived, color = Sex)) +  
 geom\_bar(position = "dodge") +  
 labs(x = "Passenger Class", y = "Number of Survivors", fill = "Survived", color = "Gender") +  
 ggtitle("Survived on the Titanic Based on Passenger Class and Gender")  
  
ggplot(Titanic, aes(x = Age)) +  
 geom\_histogram(binwidth = 5) +  
 labs(x = "Age", y = "Frequency") +  
 ggtitle("Distribution of Age on the Titanic")

10. a. Create a data frame based on below table.

b. Create a regression model for that data frame table to show the amount of sales(Sales) based

on the how much the company spends (Spends) in advertising

c. Predict the Sales if Spend=13500

INPUT:

Month <- c(1,2,3,4,5,6,7,8,9,10,11,12)  
Spend <- c(100,0,4000,5000,4500,3000,4000,9000,11000,15000,12000,7000,3000)  
Sales <- c(991,4,4048,7,5432,4,5004,4,3471,9,4255,1,9487,1,11891,4,15848,4,13134,8,7850,4,3628,4)  
  
df <- data.frame(Month, Spend, Sales)  
model <- lm(Sales ~ Spend, data = df)  
summary(model)  
predict(model, data.frame(Spend = 13500))

SET-2

1.(i) Write a R program to extract the five of the levels of factor created from a random

sample from the LETTERS (Part of the base R distribution.)

(ii)Write R function to find the range of given vector. Range=Max-Min

Sample input, C&lt;-(9,8,7,6,5,4,3,2,1),

output=8

(iii)Wirte the R function to find the number of vowels in given string

Sample input c&lt;- “matrix”, output&lt;-2

INPUT:

letters\_sample <- sample(LETTERS, 5)  
letters\_factor <- factor(letters\_sample)  
levels(letters\_factor)  
find\_range <- function(vec) {  
 max\_val <- max(vec)  
 min\_val <- min(vec)  
 range <- max\_val - min\_val  
 return(range)  
}  
C <- c(9,8,7,6,5,4,3,2,1)  
find\_range(C)  
find\_vowels <- function(string) {  
 vowels <- c("a", "e", "i", "o", "u", "A", "E", "I", "O", "U")  
 count <- 0  
 for (i in 1:nchar(string)) {  
 if (string[i] %in% vowels) {  
 count <- count + 1  
 }  
 }  
 return(count)  
}  
string <- "matrix"  
find\_vowels(string)

OUTPUT:

> levels(letters\_factor)

[1] "D" "H" "O" "W" "Y"

> find\_range(C)

[1] 8

> find\_vowels(string)

[1] 0

2.Load inbuild dataset “ChickWeight” in R

(i) Explore the summary of Data set, like number of Features and its type. Fins the number

of records for each features

(ii)Extract last 6 records of dataset

(iii) order the data frame, in ascending order by feature name  “weight”  grouped by

feature “diet”

(iv)Perform melting function based on “Chick&quot;,&quot;Time&quot;,&quot;Diet&quot;   features as ID variables

(v)Perform cast function to display the mean value of weight grouped by Diet

INPUT:

data("ChickWeight")  
str(ChickWeight)  
summary(ChickWeight)  
tail(ChickWeight, 6)  
ChickWeight\_grouped <- group\_by(ChickWeight, Diet)  
ChickWeight\_ordered <- arrange(ChickWeight\_grouped, weight)  
library(reshape2)  
ChickWeight\_melted <- melt(ChickWeight, id.vars=c("Chick", "Time", "Diet"))  
ChickWeight\_cast <- dcast(ChickWeight\_melted, Diet ~ variable, mean)

OUTPUT:

> tail(ChickWeight, 6)

weight Time Chick Diet

573 155 12 50 4

574 175 14 50 4

575 205 16 50 4

576 234 18 50 4

577 264 20 50 4

578 264 21 50 4

3.(i)Get the Statistical  Summary of  “ChickWeight” dataset

(ii)Create Box plot for “weight”  grouped by “Diet”

(iii)Create a Histogram for  “Weight” features  belong to Diet- 1 category

(iv) Create a Histogram for  “Weight” features  belong to Diet- 4 category

(v) Create Scatter plot  for weight vs Time grouped by Diet

INPUT:

summary(ChickWeight)  
library(ggplot2)  
ggplot(ChickWeight, aes(x=Diet, y=weight)) + geom\_boxplot()  
library(ggplot2)  
diet1 <- subset(ChickWeight, Diet == 1)  
ggplot(diet1, aes(x=weight)) + geom\_histogram()  
library(ggplot2)  
diet4 <- subset(ChickWeight, Diet == 4)  
ggplot(diet4, aes(x=weight)) + geom\_histogram()  
library(ggplot2)  
ggplot(ChickWeight, aes(x=Time, y=weight, color=Diet)) + geom\_point()  
  
OUTPUT:

