

TITLE:

DATE:

PAGE NO:

Week 8:**Regression Model:**

- a) Import a data from web storage. Name the dataset and now do Logistic Regression to find out relation between variables that are affecting the admission of a student in an institute based on his or her GRE score, GPA obtained and rank of the student. Also check the model is fit or not. Require (foreign), require (MASS).
- b) Apply regression Model to predict the data on any dataset.

Program:

```
A) data <- read.csv("D:/5d8_r/Admission_Predict.csv")

head(data,n=3)

summary(data)

data$Admission <- ifelse(data$Chance.of.Admit >= 0.75, 1, 0)

model <- glm(Admission ~ GRE.Score + CGPA, data = data, family = binomial)

summary(model)

pseudo_r2 <- 1 - (model$deviance / model$null.deviance)

cat("Pseudo R-squared:", pseudo_r2, "\n")

predicted_probs <- predict(model, type = "response")

predicted_class <- ifelse(predicted_probs >= 0.5, 1, 0)

conf_matrix <- table(Predicted = predicted_class, Actual = data$Admission)

print(conf_matrix)

accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)

cat("Accuracy:", round(accuracy, 4), "\n")
```

Output:

Serial.No.	GRE.Score	TOEFL.Score	University.Rating	SOP	LOR	CGPA	Research	Chance.of.Admit
1	1	337	118	4 4.5 4.5 9.65	1	0.92		
2	2	324	107	4 4.0 4.5 8.87	1	0.76		
3	3	316	104	3 3.0 3.5 8.00	1	0.72		

TITLE:

DATE:

PAGE NO:

Call:
 glm(formula = Admission ~ GRE.Score + CGPA, family = binomial,
 data = data)

Coefficients:

Estimate Std. Error z value Pr(>|z|)
 (Intercept) -70.82714 8.32195 -8.511 <2e-16 ***
 GRE.Score 0.09199 0.02762 3.330 0.000868 ***
 CGPA 4.78748 0.68169 7.023 2.17e-12 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 550.51 on 399 degrees of freedom
 Residual deviance: 226.07 on 397 degrees of freedom
 AIC: 232.07

Pseudo R-squared: 0.5893459

Actual

Predicted 0 1

0 196 26

1 24 154

Accuracy: 0.875

TITLE:

DATE:

PAGE NO:

```

B) require(MASS)

data(mtcars)

head(mtcars)

linear_model <- lm(mpg ~ hp + wt, data = mtcars)

summary(linear_model)

predicted_mpg <- predict(linear_model, mtcars)

data.frame(Actual = mtcars$mpg, Predicted = round(predicted_mpg, 2))

plot(mtcars$mpg, predicted_mpg,
      xlab = "Actual MPG", ylab = "Predicted MPG",
      main = "Actual vs Predicted MPG", pch = 19, col = "blue")

abline(0, 1, col = "red", lwd = 2)

SSE <- sum((mtcars$mpg - predicted_mpg)^2)

SST <- sum((mtcars$mpg - mean(mtcars$mpg))^2)

R2 <- 1 - (SSE / SST)

cat("Model R-squared:", round(R2, 3))

```

Output:

```

mpg cyl disp hp drat wt qsec vs am gear carb
Mazda RX4      21.0  6 160 110 3.90 2.620 16.46 0 1  4  4
Mazda RX4 Wag  21.0  6 160 110 3.90 2.875 17.02 0 1  4  4
Datsun 710     22.8  4 108  93 3.85 2.320 18.61 1 1  4  1
Hornet 4 Drive  21.4  6 258 110 3.08 3.215 19.44 1 0  3  1
Hornet Sportabout 18.7  8 360 175 3.15 3.440 17.02 0 0  3  2
Valiant        18.1  6 225 105 2.76 3.460 20.22 1 0  3  1
Actual Predicted
Mazda RX4      21.0   23.57
Mazda RX4 Wag  21.0   22.58
Datsun 710     22.8   25.28
Hornet 4 Drive  21.4   21.27
Hornet Sportabout 18.7   18.33
Call:
lm(formula = mpg ~ hp + wt, data = mtcars)
Residuals:
    Min     1Q   Median     3Q    Max
-3.941 -1.600 -0.182  1.050  5.854

```

TITLE:

DATE:

PAGE NO:

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	37.22727	1.59879	23.285	< 2e-16 ***
hp	-0.03177	0.00903	-3.519	0.00145 **
wt	-3.87783	0.63273	-6.129	1.12e-06 ***

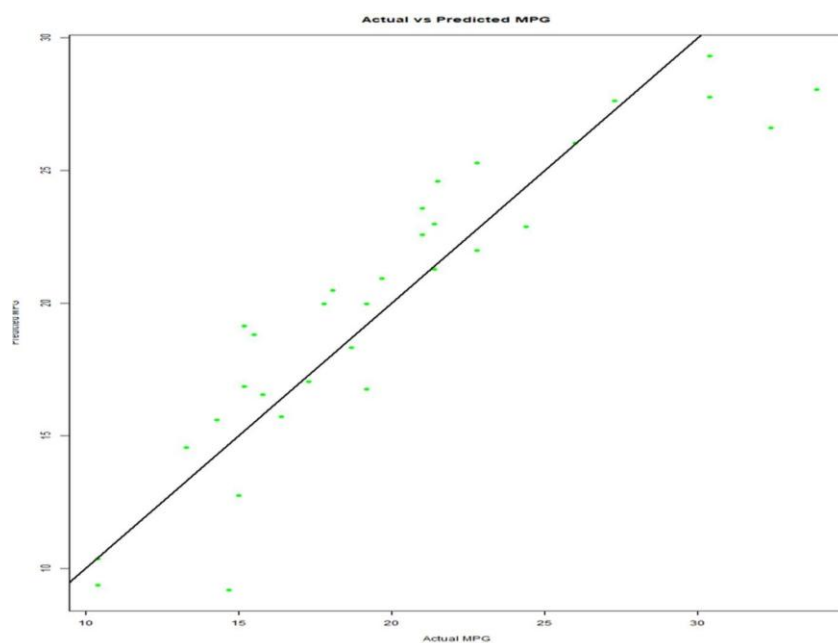
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.593 on 29 degrees of freedom

Multiple R-squared: 0.8268, Adjusted R-squared: 0.8148

F-statistic: 69.21 on 2 and 29 DF, p-value: 9.109e-12

Model R-squared: 0.827



TITLE:

DATE:

PAGE NO:

Week 9:**Multiple Regression Model:**

- a) Develop Multiple Regression model for any dataset.
- b) Apply Multiple Regression Model to predict the data on any dataset.

Program:

```

A) library(MASS)    # Contains Boston

dataset library(caret) # For data partitioning

library(ggplot2) # For visualization

data("Boston")

head(Boston)

set.seed(123)

trainIndex <- createDataPartition(Boston$medv, p = 0.8, list = FALSE)

trainData <- Boston[trainIndex, ]

testData <- Boston[-trainIndex, ]

lm_model <- lm(medv ~ ., data = trainData)

summary(lm_model)

predictions <- predict(lm_model, newdata = testData)

rmse <- sqrt(mean((predictions - testData$medv)^2))

cat("RMSE: ", rmse)

ggplot(data.frame(Actual = testData$medv, Predicted = predictions), aes(x = Actual, y = Predicted)) +
  geom_point(color = "blue") +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Actual vs Predicted: Boston Housing Prices",
       x = "Actual Median Value of Homes ($1000s)",
       y = "Predicted Median Value of Homes ($1000s)")

par(mfrow = c(2, 2))

plot(lm_model)

```

TITLE:

DATE:

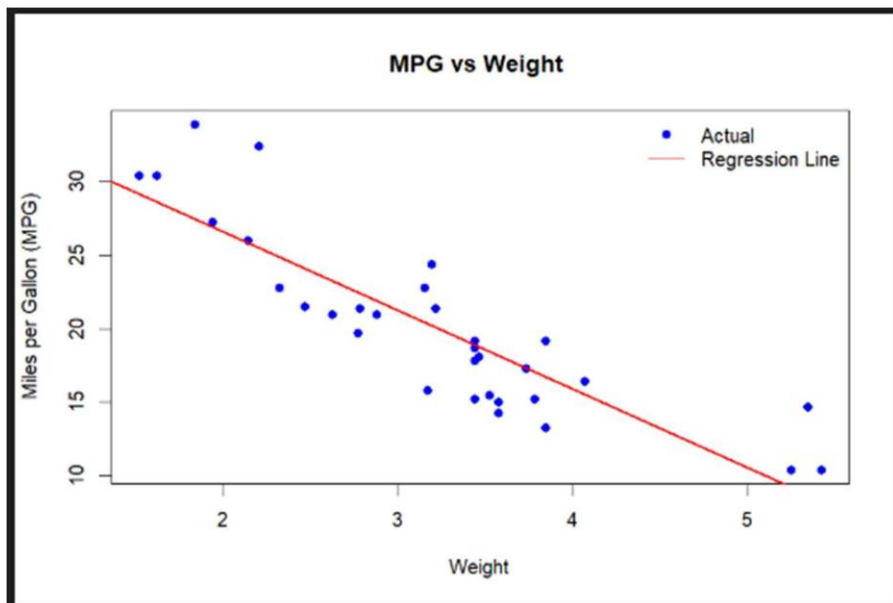
PAGE NO:

Output:

```

crim zn indus chas nox  rm  age  dis rad tax ptratio  black lstat medv
1 0.00632 18 2.31  0 0.538 6.575 65.2 4.0900  1 296  15.3 396.90  4.98 24.0
2 0.02731  0 7.07  0 0.469 6.421 78.9 4.9671  2 242  17.8 396.90  9.14 21.6
3 0.02729  0 7.07  0 0.469 7.185 61.1 4.9671  2 242  17.8 392.83  4.03 34.7
4 0.03237  0 2.18  0 0.458 6.998 45.8 6.0622  3 222  18.7 394.63  2.94 33.4
5 0.06905  0 2.18  0 0.458 7.147 54.2 6.0622  3 222  18.7 396.90  5.33 36.2
6 0.02985  0 2.18  0 0.458 6.430 58.7 6.0622  3 222  18.7 394.12  5.21 28.7
RMSE: 4.588948

```



TITLE:**DATE:****PAGE NO:**

```
B) install.packages("caret")
install.packages("ggplot2")
library(caret)
library(ggplot2)
data("mtcars")
head(mtcars)
set.seed(123)
trainIndex <- createDataPartition(mtcars$mpg, p = 0.8, list = FALSE)
trainData <- mtcars[trainIndex, ]
testData <- mtcars[-trainIndex, ]
cat("Dimensions of trainData:", dim(trainData), "\n")
cat("Dimensions of testData:", dim(testData), "\n")
lm_model_mtcars <- lm(mpg ~ cyl + disp + hp + wt + qsec + drat + vs + am + gear + carb, data = trainData)
summary(lm_model_mtcars)
predictions_mtcars <- predict(lm_model_mtcars, newdata = testData)
cat("Length of predictions:", length(predictions_mtcars), "\n")
cat("Length of testData$mpg:", length(testData$mpg), "\n")
rmse_mtcars <- sqrt(mean((predictions_mtcars - testData$mpg)^2))
cat("RMSE: ", rmse_mtcars, "\n")
comparison_data <- data.frame(Actual = testData$mpg, Predicted = predictions_mtcars)
```

TITLE:**DATE:****PAGE NO:**

OUTPUT:

```
      mpg cyl disp  hp drat   wt  qsec vs am gear carb
Mazda RX4          21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag      21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710          22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive      21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout  18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant             18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
RMSE: 4.808981>
```

TITLE:**DATE:****PAGE NO:****Week 10:****Classification Model:****a) Install relevant package for classification.****b) Implement and evaluate the performance of various classifiers with different datasets.****PROGRAMS:**a) `install.packages("caret")` # For building and evaluating classification models`install.packages("e1071")` # For SVM and Naive Bayes`install.packages("randomForest")` # For Random Forest classifier`install.packages("nnet")` # For Neural Network classifier`install.packages("rpart")` # For Decision Tree classifier`install.packages("ggplot2")` # For data visualization`library(caret)``library(e1071)``library(randomForest)``library(nnet)``library(rpart)``library(ggplot2)``data(iris)``set.seed(123)``trainIndex <- createDataPartition(iris$Species, p = 0.8, list = FALSE)``trainData <- iris[trainIndex,]``testData <- iris[-trainIndex,]``model <- randomForest(Species ~ ., data = trainData, ntree = 100)``predictions <- predict(model, testData)``confusionMatrix(predictions, testData$Species)`

TITLE:**DATE:****PAGE NO:****OUTPUT:**

Confusion Matrix and Statistics

Reference
Prediction setosa versicolor virginica
setosa 10 0 0
versicolor 0 10 1
virginica 0 0 9

Overall Statistics

Accuracy : 0.9667
95% CI : (0.8278, 0.9992)
No Information Rate : 0.3333
P-Value [Acc > NIR] : 2.963e-13

Kappa : 0.95

Mcnemar's Test P-Value : NA

Statistics by Class:

	Class: setosa	Class: versicolor	Class: virginica
Sensitivity	1.0000	1.0000	0.9000
Specificity	1.0000	0.9500	1.0000
Pos Pred Value	1.0000	0.9091	1.0000
Neg Pred Value	1.0000	1.0000	0.9524
Prevalence	0.3333	0.3333	0.3333
Detection Rate	0.3333	0.3333	0.3000
Detection Prevalence	0.3333	0.3667	0.3000
Balanced Accuracy	1.0000	0.9750	0.9500

TITLE:**DATE:****PAGE NO:**

B)

library(caret)

library(e1071)

data(iris)

set.seed(123)

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

trainData <- iris[trainIndex,]

testData <- iris[-trainIndex,]

svm_model <- svm(Species ~ ., data = trainData, kernel = "linear") # Linear kernel

svm_predictions <- predict(svm_model, testData)

confusionMatrix(svm_predictions, testData\$Species)

OUTPUT:

Confusion Matrix and Statistics

	Reference		
Prediction	setosa	versicolor	virginica
setosa	10	0	0
versicolor	0	10	1
virginica	0	0	9

Overall Statistics

Accuracy : 0.9667

95% CI : (0.8278, 0.9992)

No Information Rate : 0.3333

P-Value [Acc > NIR] : 2.963e-13

Kappa : 0.95

Mcnemar's Test P-Value : NA

Statistics by Class:

	Class: setosa	Class: versicolor	Class: virginica
Sensitivity	1.0000	1.0000	0.9000
Specificity	1.0000	0.9500	1.0000
Pos Pred Value	1.0000	0.9091	1.0000
Neg Pred Value	1.0000	1.0000	0.9524
Prevalence	0.3333	0.3333	0.3333
Detection Rate	0.3333	0.3333	0.3000
Detection Prevalence	0.3333	0.3667	0.3000
Balanced Accuracy	1.0000	0.9750	0.9500

TITLE:

TITLE:**DATE:****PAGE NO:****Week 11:****Clustering Model:****a) Clustering algorithms for unsupervised classification.****b) Plot the cluster data using R visualizations.**

a) Clustering algorithms for unsupervised classification.

library(caret)

library(ggplot2)

library(dplyr)

data <- iris

ggplot(data, aes(Sepal.Length, Sepal.Width, color = Species)) +

geom_point() +

ggtitle("Original Iris Data (Sepal Length vs Sepal Width)") +

theme_minimal()

data <- data[, -5]

scal <- scale(data)

maximum <- 10

wss <- sapply(1:maximum, function(k) {

kmeans(scal, k, nstart = 50, iter.max = 15)\$tot.withinss

}))

plot(1:maximum, wss, type = "b", xlab = "Number of Clusters (k)", ylab = "Within-cluster Sum of Squares (WSS)")

abline(v = 3, col = "red", lty = 2) # The elbow at k = 3 is usually optimal

km <- kmeans(scal, centers = 3, iter.max = 50)

print(km)

data\$cluster <- as.factor(km\$cluster)

ggplot(data, aes(Sepal.Length, Sepal.Width, color = cluster)) +

geom_point() +

ggtitle("K-Means Clustering (k = 3)") +

TITLE:

DATE:

PAGE NO:

HIERARICAL CLUSTERING

```
library(ggplot2)

library(dplyr)

library(cluster)

data("trees")

head(trees)

tree_data <- trees[, -3]

scaled_data <- scale(tree_data)

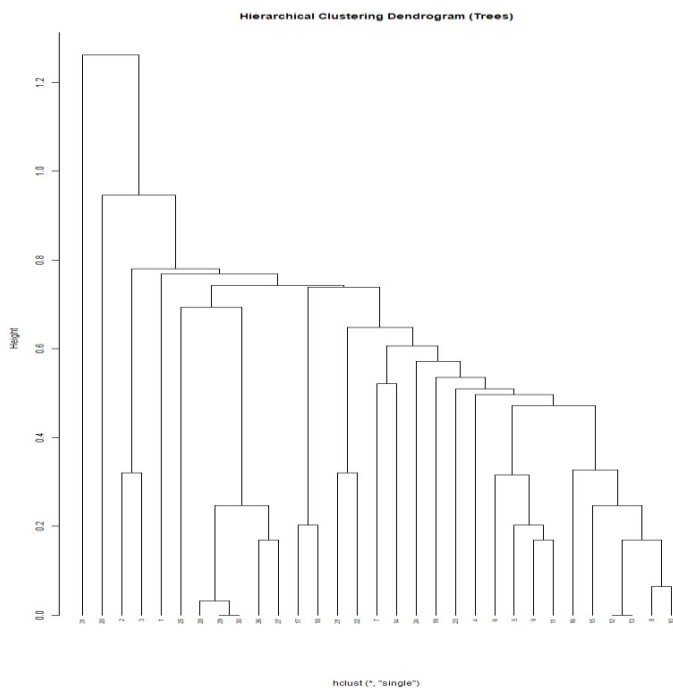
dist_matrix <- dist(scaled_data, method = "euclidean")

hclust_result <- hclust(dist_matrix, method = "single")

plot(hclust_result, main = "Dendrogram using Single Linkage (Tree Dataset)",

     xlab = "", ylab = "Height", sub = "",

     cex = 0.7, hang = -1) # 'cex' for label size and 'hang' for clarity of tree
```

Output

b) Plot the cluster data using r visualization

```
library(ggplot2)

library(dplyr)

data(iris)

iris_data <- iris[, -5]

scaled_data <- scale(iris_data)

set.seed(123) # For reproducibility

kmeans_result <- kmeans(scaled_data, centers = 3, nstart = 25)

iris$Cluster <- as.factor(kmeans_result$cluster)

ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Cluster)) +

  geom_point(size = 4) + ggtitle("Clustering of Iris Dataset using K-means") + theme_minimal() + xlab("Sepal Length") + ylab("Sepal Width") + scale_color_manual(values = c("red", "blue", "green")) # Customize colors

cluster_centers <- as.data.frame(kmeans_result$centers)

cluster_centers$Cluster <- as.factor(1:3)

ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Cluster)) + geom_point(size = 4) + geom_point(data = cluster_centers, aes(x = Sepal.Length, y = Sepal.Width, color = Cluster), shape = 8, size = 6, show.legend = FALSE) + ggtitle("Clustering of Iris Dataset with K-means (Cluster Centers)") + theme_minimal() + xlab("Sepal Length") + ylab("Sepal Width") + scale_color_manual(values = c("red", "blue", "green"))
```

output:

setosa versicolor virginica

1	50	0	0
2	0	48	14
3	0	2	36

