**Lab 11: Multinomial Logistic Regression**

**Problem statement:**

Task1: Refer attached glass\_multiclass.CSV dataset. Develop a multinomial logistic regression classification model as:

* Class variable: type
* First convert class variable into categorical variable
* Independent variable: remaining all
* With summary command observe the results
* Partitioned the data set into train and test data
* Observe the dimension of each data set
* Display and observe the summary of model
* Calculate and display confusion matrix for test dataset
* Calculate and display confusion matrix for train dataset
* Calculate the accuracy of model i.e. against both data set (train and test)
* Calculate the error rate of model i.e. against both data set (train and test)

Task-2: Develop four multinomial logistic regression model as:

* model-1: independent variables (RI,Na, Mg)
* model-2: independent variables (RI,Na, Mg, AI, SI, K)
* model-3: independent variables (RI,Na, Mg, AI, SI, K, Ca, Ba, Fe)
* model-3: independent variables (SI, K, Ca, Ba, Fe)
* Find out best model with justifications

**Source Code:**

#Author: Ashish Upadhyay

#Branch: Computer Science and Engineering

#Semester: 6th

#Dr. SP Mukherjee International Institute of Information Technology, Naya Raipur

#Subject: Machine Learning Lab 11

#Task: Multinomial Logistic Regression Implementation

#Task I

setwd("C:/Users/Ashish Upadhyay/Documents/Semester6/MachineLearning/Lab Programs")

getwd()

data\_set <- read.csv("glass\_multiclass.csv")

head(data\_set)

nrow(data\_set)

names(data\_set)

dim(data\_set)

summary(data\_set)

data\_set$Type <- as.factor(data\_set$Type)

library(caTools)

set.seed(88)

split <- sample.split(data\_set$Type, SplitRatio = 0.75)

#get training and test data

train <- subset(data\_set, split == TRUE)

test <- subset(data\_set, split == FALSE)

dim(test)

dim(train)

table(train$Type)

table(test$Type)

#install.packages('nnet')

library("nnet")

#Multinomial Model

model <- multinom(Type ~ ., data = train, maxit = 1000)

summary(model)

predictions\_train <- predict(model, train)

con\_mat\_train\_model <- table(predicted = predictions\_train, actual = train$Type)

con\_mat\_train\_model

accuracy\_train\_model <- sum(diag(con\_mat\_train\_model)) / sum(con\_mat\_train\_model)

accuracy\_train\_model

error\_rate\_train\_model <- 1 - accuracy\_train\_model

error\_rate\_train\_model

predictions\_test <- predict(model, test)

con\_mat\_test\_model <- table(predicted = predictions\_test, actual = test$Type)

con\_mat\_test\_model

accuracy\_test\_model <- sum(diag(con\_mat\_test\_model)) / sum(con\_mat\_test\_model)

accuracy\_test\_model

error\_rate\_test\_model <- 1 - accuracy\_train\_model

error\_rate\_test\_model

#Task II

#Model-1

model1 <- multinom(Type ~ RI + Na + Mg, data = train, maxit = 1000)

summary(model1)

predictions\_train1 <- predict(model1, train)

con\_mat\_train\_model1 <- table(predicted = predictions\_train1, actual = train$Type)

con\_mat\_train\_model1

accuracy\_train\_model1 <- sum(diag(con\_mat\_train\_model1)) / sum(con\_mat\_train\_model1)

accuracy\_train\_model1

error\_rate\_train\_model1 <- 1 - accuracy\_train\_model1

error\_rate\_train\_model1

predictions\_test1 <- predict(model1, test)

con\_mat\_test\_model1 <- table(predicted = predictions\_test1, actual = test$Type)

con\_mat\_test\_model1

accuracy\_test\_model1 <- sum(diag(con\_mat\_test\_model1)) / sum(con\_mat\_test\_model1)

accuracy\_test\_model1

error\_rate\_test\_model1 <- 1 - accuracy\_train\_model1

error\_rate\_test\_model1

#Model-2

model2 <- multinom(Type ~ RI + Na + Mg + Al + Si + K, data = train, maxit = 1000)

summary(model2)

predictions\_train2 <- predict(model2, train)

con\_mat\_train\_model2 <- table(predicted = predictions\_train2, actual = train$Type)

con\_mat\_train\_model2

accuracy\_train\_model2 <- sum(diag(con\_mat\_train\_model2)) / sum(con\_mat\_train\_model2)

accuracy\_train\_model2

error\_rate\_train\_model2 <- 1 - accuracy\_train\_model2

error\_rate\_train\_model2

predictions\_test2 <- predict(model2, test)

con\_mat\_test\_model2 <- table(predicted = predictions\_test2, actual = test$Type)

con\_mat\_test\_model2

accuracy\_test\_model2 <- sum(diag(con\_mat\_test\_model2)) / sum(con\_mat\_test\_model2)

accuracy\_test\_model2

error\_rate\_test\_model2 <- 1 - accuracy\_train\_model2

error\_rate\_test\_model2

#Model-3

model3 <- multinom(Type ~ RI + Na + Mg + Al + Si + K + Ca + Ba +Fe, data = train, maxit = 1000)

summary(model3)

predictions\_train3 <- predict(model3, train)

con\_mat\_train\_model3 <- table(predicted = predictions\_train3, actual = train$Type)

con\_mat\_train\_model3

accuracy\_train\_model3 <- sum(diag(con\_mat\_train\_model3)) / sum(con\_mat\_train\_model3)

accuracy\_train\_model3

error\_rate\_train\_model3 <- 1 - accuracy\_train\_model3

error\_rate\_train\_model3

predictions\_test3 <- predict(model3, test)

con\_mat\_test\_model3 <- table(predicted = predictions\_test3, actual = test$Type)

con\_mat\_test\_model3

accuracy\_test\_model3 <- sum(diag(con\_mat\_test\_model3)) / sum(con\_mat\_test\_model3)

accuracy\_test\_model3

error\_rate\_test\_model3 <- 1 - accuracy\_train\_model3

error\_rate\_test\_model3

#Model-4

model4 <- multinom(Type ~ Si + K + Ca + Ba + Fe, data = train, maxit = 1000)

summary(model4)

predictions\_train4 <- predict(model4, train)

con\_mat\_train\_model4 <- table(predicted = predictions\_train4, actual = train$Type)

#con\_mat\_train\_model4

accuracy\_train\_model4 <- sum(diag(con\_mat\_train\_model4)) / sum(con\_mat\_train\_model4)

accuracy\_train\_model4

error\_rate\_train\_model4 <- 1 - accuracy\_train\_model4

error\_rate\_train\_model4

predictions\_test4 <- predict(model4, test)

con\_mat\_test\_model4 <- table(predicted = predictions\_test4, actual = test$Type)

#con\_mat\_test\_model4

accuracy\_test\_model4 <- sum(diag(con\_mat\_test\_model4)) / sum(con\_mat\_test\_model4)

accuracy\_test\_model4

error\_rate\_test\_model4 <- 1 - accuracy\_train\_model4

error\_rate\_test\_model4

**Output:**

> #Author: Ashish Upadhyay

> #Branch: Computer Science and Engineering

> #Semester: 6th

> #Dr. SP Mukherjee International Institute of Information Technology, Naya Raipur

> #Subject: Machine Learning Lab 11

> #Task: Multinomial Logistic Regression Implementation

>

>

>

> #Task I

>

> setwd("C:/Users/Ashish Upadhyay/Documents/Semester6/MachineLearning/Lab Programs")

> getwd()

[1] "C:/Users/Ashish Upadhyay/Documents/Semester6/MachineLearning/Lab Programs"

>

> data\_set <- read.csv("glass\_multiclass.csv")

> head(data\_set)

RI Na Mg Al Si K Ca Ba Fe Type

1 1.52101 13.64 4.49 1.10 71.78 0.06 8.75 0 0.00 1

2 1.51761 13.89 3.60 1.36 72.73 0.48 7.83 0 0.00 1

3 1.51618 13.53 3.55 1.54 72.99 0.39 7.78 0 0.00 1

4 1.51766 13.21 3.69 1.29 72.61 0.57 8.22 0 0.00 1

5 1.51742 13.27 3.62 1.24 73.08 0.55 8.07 0 0.00 1

6 1.51596 12.79 3.61 1.62 72.97 0.64 8.07 0 0.26 1

> nrow(data\_set)

[1] 214

>

> names(data\_set)

[1] "RI" "Na" "Mg" "Al" "Si" "K" "Ca" "Ba" "Fe" "Type"

> dim(data\_set)

[1] 214 10

>

> summary(data\_set)

RI Na Mg Al Si

Min. :1.511 Min. :10.73 Min. :0.000 Min. :0.290 Min. :69.81

1st Qu.:1.517 1st Qu.:12.91 1st Qu.:2.115 1st Qu.:1.190 1st Qu.:72.28

Median :1.518 Median :13.30 Median :3.480 Median :1.360 Median :72.79

Mean :1.518 Mean :13.41 Mean :2.685 Mean :1.445 Mean :72.65

3rd Qu.:1.519 3rd Qu.:13.82 3rd Qu.:3.600 3rd Qu.:1.630 3rd Qu.:73.09

Max. :1.534 Max. :17.38 Max. :4.490 Max. :3.500 Max. :75.41

K Ca Ba Fe Type

Min. :0.0000 Min. : 5.430 Min. :0.000 Min. :0.00000 Min. :1.00

1st Qu.:0.1225 1st Qu.: 8.240 1st Qu.:0.000 1st Qu.:0.00000 1st Qu.:1.00

Median :0.5550 Median : 8.600 Median :0.000 Median :0.00000 Median :2.00

Mean :0.4971 Mean : 8.957 Mean :0.175 Mean :0.05701 Mean :2.78

3rd Qu.:0.6100 3rd Qu.: 9.172 3rd Qu.:0.000 3rd Qu.:0.10000 3rd Qu.:3.00

Max. :6.2100 Max. :16.190 Max. :3.150 Max. :0.51000 Max. :7.00

>

> data\_set$Type <- as.factor(data\_set$Type)

>

> library(caTools)

> set.seed(88)

> split <- sample.split(data\_set$Type, SplitRatio = 0.75)

>

> #get training and test data

> train <- subset(data\_set, split == TRUE)

> test <- subset(data\_set, split == FALSE)

>

> dim(test)

[1] 53 10

> dim(train)

[1] 161 10

>

> table(train$Type)

1 2 3 5 6 7

52 57 13 10 7 22

> table(test$Type)

1 2 3 5 6 7

18 19 4 3 2 7

>

> #install.packages('nnet')

> library("nnet")

>

> model <- multinom(Type ~ ., data = train, maxit = 1000)

# weights: 66 (50 variable)

initial value 288.473275

iter 10 value 176.893686

iter 20 value 124.738149

iter 30 value 111.008787

iter 40 value 105.664700

iter 50 value 102.298369

iter 60 value 100.195435

iter 70 value 99.807182

iter 80 value 99.646948

iter 90 value 99.256888

iter 100 value 98.987365

iter 110 value 98.965893

iter 120 value 98.858676

iter 130 value 98.812843

iter 140 value 97.690144

iter 150 value 95.919478

iter 160 value 95.024378

iter 170 value 95.011113

iter 180 value 94.129473

iter 190 value 93.815132

iter 200 value 93.699533

iter 210 value 93.642554

iter 220 value 93.623661

iter 230 value 93.574819

iter 240 value 93.531863

iter 250 value 93.474273

iter 260 value 93.302727

iter 270 value 93.180463

iter 280 value 93.160441

iter 290 value 92.970575

iter 300 value 92.835629

iter 310 value 92.782585

iter 320 value 92.775068

iter 330 value 92.757020

iter 340 value 92.674846

iter 350 value 92.570052

iter 360 value 92.334816

iter 370 value 92.321175

iter 380 value 92.302224

iter 390 value 91.971135

iter 400 value 90.833979

iter 410 value 89.977295

iter 420 value 89.941450

iter 430 value 89.928643

iter 440 value 89.919600

iter 450 value 89.911005

iter 460 value 89.828482

iter 470 value 89.602676

iter 480 value 89.470807

iter 490 value 89.454347

iter 500 value 89.360322

iter 510 value 89.277129

iter 520 value 89.272248

iter 530 value 89.267633

iter 540 value 89.257607

iter 550 value 89.192259

iter 560 value 89.152085

iter 570 value 89.077939

iter 580 value 88.812195

iter 590 value 88.751789

iter 600 value 88.731080

final value 88.680339

converged

> summary(model)

Call:

multinom(formula = Type ~ ., data = train, maxit = 1000)

Coefficients:

(Intercept) RI Na Mg Al Si K

2 203.026613 171.42504 -4.099733 -5.800582 -0.06659444 -4.795857 -3.128619

3 848.442724 -786.05540 5.433639 4.239372 4.84957130 2.743805 4.141046

5 5.733115 12.99851 -81.581835 -77.628788 222.12921154 6.134533 254.091839

6 -14.058868 -23.50412 -15.226861 -27.438127 9.75009541 8.231017 -368.668129

7 -143.968089 933.95243 -11.264131 -18.845955 -12.60991726 -12.313294 -10.288273

Ca Ba Fe

2 -4.495158 -6.789358 0.6437630

3 5.646475 2.223870 0.8239636

5 16.994975 -169.437118 272.7346400

6 -28.089356 -227.566859 -319.8861733

7 -17.173076 -6.244567 -95.8431176

Std. Errors:

(Intercept) RI Na Mg Al Si K

2 0.03920231 0.06701881 0.5768557 0.85545632 1.65179176 0.1511549 2.289055e+00

3 0.06114337 0.09642064 0.8058941 1.12917704 1.76232222 0.2059976 2.938445e+00

5 0.06245625 0.08769877 12.2913630 8.40708139 1.89682777 2.6617805 2.304166e+00

6 0.01613113 0.02450860 0.2228231 0.03882437 0.01984018 1.1730231 4.111865e-10

7 0.10335390 0.16375551 2.0999998 2.96043069 6.87912454 0.7338250 5.774488e+00

Ca Ba Fe

2 0.5632691 2.036229e+00 2.320576e+00

3 0.6197758 4.502377e+00 3.847483e+00

5 5.3353436 7.067090e+00 1.955795e-02

6 0.1572593 3.993370e-10 2.896739e-22

7 2.6269938 5.059777e+00 1.303330e+00

Residual Deviance: 177.3607

AIC: 277.3607

>

> predictions\_train <- predict(model, train)

> con\_mat\_train\_model <- table(predicted = predictions\_train, actual = train$Type)

> con\_mat\_train\_model

actual

predicted 1 2 3 5 6 7

1 34 13 7 0 0 0

2 15 44 4 0 0 1

3 3 0 2 0 0 0

5 0 0 0 10 0 0

6 0 0 0 0 7 0

7 0 0 0 0 0 21

> accuracy\_train\_model <- sum(diag(con\_mat\_train\_model)) / sum(con\_mat\_train\_model)

> accuracy\_train\_model

[1] 0.7329193

> error\_rate\_train\_model <- 1 - accuracy\_train\_model

> error\_rate\_train\_model

[1] 0.2670807

>

> predictions\_test <- predict(model, test)

> con\_mat\_test\_model <- table(predicted = predictions\_test, actual = test$Type)

> con\_mat\_test\_model

actual

predicted 1 2 3 5 6 7

1 13 4 2 0 0 0

2 4 12 2 1 0 1

3 1 0 0 0 0 0

5 0 2 0 2 0 2

6 0 1 0 0 2 0

7 0 0 0 0 0 4

> accuracy\_test\_model <- sum(diag(con\_mat\_test\_model)) / sum(con\_mat\_test\_model)

> accuracy\_test\_model

[1] 0.6226415

> error\_rate\_test\_model <- 1 - accuracy\_train\_model

> error\_rate\_test\_model

[1] 0.2670807

>

> #Task II

>

> model1 <- multinom(Type ~ RI + Na + Mg, data = train, maxit = 1000)

# weights: 30 (20 variable)

initial value 288.473275

iter 10 value 194.980593

iter 20 value 167.160823

iter 30 value 164.649443

iter 40 value 164.541471

iter 50 value 164.530243

iter 60 value 164.515760

iter 70 value 162.209446

iter 80 value 160.494303

iter 90 value 160.466967

iter 100 value 160.364315

iter 110 value 160.327626

iter 120 value 158.513344

iter 130 value 158.119465

iter 140 value 158.089116

iter 150 value 158.057444

iter 160 value 157.957919

iter 170 value 157.170662

iter 180 value 156.788012

iter 190 value 156.751352

iter 200 value 156.741216

iter 210 value 156.370516

iter 220 value 156.156771

iter 230 value 156.153140

iter 240 value 156.143920

iter 250 value 155.737327

iter 260 value 155.498724

iter 270 value 155.482735

iter 280 value 155.476436

iter 290 value 155.414348

iter 300 value 155.397862

iter 310 value 155.369762

iter 320 value 155.368009

iter 330 value 155.227592

iter 340 value 155.162822

iter 350 value 155.134074

iter 360 value 155.131872

iter 370 value 155.090479

iter 380 value 155.066985

iter 390 value 155.051411

iter 400 value 155.049556

iter 410 value 155.020009

iter 420 value 154.988330

iter 430 value 154.969417

iter 440 value 154.968105

iter 450 value 154.962323

iter 460 value 154.927968

iter 470 value 154.915351

iter 480 value 154.912895

iter 490 value 154.904885

iter 500 value 154.884565

iter 510 value 154.874404

iter 520 value 154.871606

iter 530 value 154.867561

iter 540 value 154.845921

iter 550 value 154.830774

iter 560 value 154.828572

iter 570 value 154.827326

iter 580 value 154.813451

iter 590 value 154.806541

iter 600 value 154.804931

iter 610 value 154.802746

iter 620 value 154.783254

iter 630 value 154.772970

iter 640 value 154.771725

iter 650 value 154.769581

iter 660 value 154.731171

iter 670 value 154.713668

iter 680 value 154.710243

iter 690 value 154.709176

iter 700 value 154.682573

iter 710 value 154.664084

iter 720 value 154.660278

iter 730 value 154.654326

iter 740 value 154.374744

iter 750 value 154.272857

iter 760 value 154.193204

iter 770 value 154.191000

iter 780 value 154.188690

iter 780 value 154.188690

final value 154.188690

converged

> summary(model1)

Call:

multinom(formula = Type ~ RI + Na + Mg, data = train, maxit = 1000)

Coefficients:

(Intercept) RI Na Mg

2 293.7176 -192.9106 0.3743614 -1.6682382

3 312.8025 -221.8155 1.7941323 -0.3902145

5 942.8657 -612.5714 -0.3696066 -3.4544177

6 692.9573 -489.6930 4.0657805 -2.4007213

7 804.4952 -554.1998 3.2497388 -3.0097623

Std. Errors:

(Intercept) RI Na Mg

2 1.796424 2.598848 0.4639528 0.5105102

3 2.806285 4.160141 0.6650216 0.9503897

5 2.396639 3.735993 0.6462813 0.5914569

6 4.128280 6.282890 0.9807628 0.6592936

7 3.704468 5.669132 0.9015172 0.6242835

Residual Deviance: 308.3774

AIC: 348.3774

>

> predictions\_train1 <- predict(model1, train)

> con\_mat\_train\_model1 <- table(predicted = predictions\_train1, actual = train$Type)

> con\_mat\_train\_model1

actual

predicted 1 2 3 5 6 7

1 37 15 6 0 0 1

2 14 38 7 4 2 2

3 1 0 0 0 0 0

5 0 1 0 5 0 0

6 0 1 0 0 1 1

7 0 2 0 1 4 18

> accuracy\_train\_model1 <- sum(diag(con\_mat\_train\_model1)) / sum(con\_mat\_train\_model1)

> accuracy\_train\_model1

[1] 0.6149068

> error\_rate\_train\_model1 <- 1 - accuracy\_train\_model1

> error\_rate\_train\_model1

[1] 0.3850932

>

> predictions\_test1 <- predict(model1, test)

> con\_mat\_test\_model1 <- table(predicted = predictions\_test1, actual = test$Type)

> con\_mat\_test\_model1

actual

predicted 1 2 3 5 6 7

1 13 5 2 0 0 0

2 5 12 2 1 1 0

3 0 0 0 0 0 0

5 0 2 0 2 0 1

6 0 0 0 0 0 1

7 0 0 0 0 1 5

> accuracy\_test\_model1 <- sum(diag(con\_mat\_test\_model1)) / sum(con\_mat\_test\_model1)

> accuracy\_test\_model1

[1] 0.6037736

> error\_rate\_test\_model1 <- 1 - accuracy\_train\_model1

> error\_rate\_test\_model1

[1] 0.3850932

>

>

> model2 <- multinom(Type ~ RI + Na + Mg + Al + Si + K, data = train, maxit = 1000)

# weights: 48 (35 variable)

initial value 288.473275

iter 10 value 188.517305

iter 20 value 135.361327

iter 30 value 128.644801

iter 40 value 121.811953

iter 50 value 121.356234

iter 60 value 120.873503

iter 70 value 120.287512

iter 80 value 120.268778

iter 90 value 120.149085

iter 100 value 119.779554

iter 110 value 119.735794

iter 120 value 119.081482

iter 130 value 118.860893

iter 140 value 118.629484

final value 118.627486

converged

> summary(model2)

Call:

multinom(formula = Type ~ RI + Na + Mg + Al + Si + K, data = train,

maxit = 1000)

Coefficients:

(Intercept) RI Na Mg Al Si K

2 25.98450 17.27879 0.07518246 -1.4743160 3.9717069 -0.7444438 1.6374824

3 83.43849 -50.87758 1.32275797 -0.4900864 0.3782597 -0.3346317 0.6921906

5 -56.50829 -67.52222 -0.90700945 -4.3176835 10.2027255 2.2149760 6.2701371

6 -75.05828 -117.38643 6.05473558 -0.8943887 2.6219146 2.3445152 -138.3949282

7 -97.40926 -95.56172 4.62460585 -2.7900264 8.8675225 2.3607118 3.3995930

Std. Errors:

(Intercept) RI Na Mg Al Si K

2 9.58391996 16.9738175 0.6698951 0.5640709 1.297138 0.4380523 1.98893192

3 3.04300834 5.4808904 0.7190405 0.8414779 1.560872 0.1928353 2.47934650

5 1.65510933 2.8672799 1.1272471 0.9762632 2.536777 0.2283417 3.51310366

6 0.07125727 0.1225657 2.0507160 1.4377365 3.892942 0.4584804 0.09775972

7 1.23496880 2.2092538 0.8977652 0.7951621 2.328618 0.1734089 2.68208189

Residual Deviance: 237.255

AIC: 307.255

>

> predictions\_train2 <- predict(model2, train)

> con\_mat\_train\_model2 <- table(predicted = predictions\_train2, actual = train$Type)

> con\_mat\_train\_model2

actual

predicted 1 2 3 5 6 7

1 36 16 7 0 0 0

2 16 38 6 1 0 2

3 0 0 0 0 0 0

5 0 0 0 9 0 0

6 0 0 0 0 7 1

7 0 3 0 0 0 19

> accuracy\_train\_model2 <- sum(diag(con\_mat\_train\_model2)) / sum(con\_mat\_train\_model2)

> accuracy\_train\_model2

[1] 0.6770186

> error\_rate\_train\_model2 <- 1 - accuracy\_train\_model2

> error\_rate\_train\_model2

[1] 0.3229814

>

> predictions\_test2 <- predict(model2, test)

> con\_mat\_test\_model2 <- table(predicted = predictions\_test2, actual = test$Type)

> con\_mat\_test\_model2

actual

predicted 1 2 3 5 6 7

1 14 7 2 0 0 0

2 4 10 2 1 0 1

3 0 0 0 0 0 0

5 0 1 0 2 0 1

6 0 1 0 0 1 0

7 0 0 0 0 1 5

> accuracy\_test\_model2 <- sum(diag(con\_mat\_test\_model2)) / sum(con\_mat\_test\_model2)

> accuracy\_test\_model2

[1] 0.6037736

> error\_rate\_test\_model2 <- 1 - accuracy\_train\_model2

> error\_rate\_test\_model2

[1] 0.3229814

>

>

> model3 <- multinom(Type ~ RI + Na + Mg + Al + Si + K + Ca + Ba +Fe, data = train, maxit = 1000)

# weights: 66 (50 variable)

initial value 288.473275

iter 10 value 176.893686

iter 20 value 124.738149

iter 30 value 111.008787

iter 40 value 105.664700

iter 50 value 102.298369

iter 60 value 100.195435

iter 70 value 99.807182

iter 80 value 99.646948

iter 90 value 99.256888

iter 100 value 98.987365

iter 110 value 98.965893

iter 120 value 98.858676

iter 130 value 98.812843

iter 140 value 97.690144

iter 150 value 95.919478

iter 160 value 95.024378

iter 170 value 95.011113

iter 180 value 94.129473

iter 190 value 93.815132

iter 200 value 93.699533

iter 210 value 93.642554

iter 220 value 93.623661

iter 230 value 93.574819

iter 240 value 93.531863

iter 250 value 93.474273

iter 260 value 93.302727

iter 270 value 93.180463

iter 280 value 93.160441

iter 290 value 92.970575

iter 300 value 92.835629

iter 310 value 92.782585

iter 320 value 92.775068

iter 330 value 92.757020

iter 340 value 92.674846

iter 350 value 92.570052

iter 360 value 92.334816

iter 370 value 92.321175

iter 380 value 92.302224

iter 390 value 91.971135

iter 400 value 90.833979

iter 410 value 89.977295

iter 420 value 89.941450

iter 430 value 89.928643

iter 440 value 89.919600

iter 450 value 89.911005

iter 460 value 89.828482

iter 470 value 89.602676

iter 480 value 89.470807

iter 490 value 89.454347

iter 500 value 89.360322

iter 510 value 89.277129

iter 520 value 89.272248

iter 530 value 89.267633

iter 540 value 89.257607

iter 550 value 89.192259

iter 560 value 89.152085

iter 570 value 89.077939

iter 580 value 88.812195

iter 590 value 88.751789

iter 600 value 88.731080

final value 88.680339

converged

> summary(model3)

Call:

multinom(formula = Type ~ RI + Na + Mg + Al + Si + K + Ca + Ba +

Fe, data = train, maxit = 1000)

Coefficients:

(Intercept) RI Na Mg Al Si K

2 203.026613 171.42504 -4.099733 -5.800582 -0.06659444 -4.795857 -3.128619

3 848.442724 -786.05540 5.433639 4.239372 4.84957130 2.743805 4.141046

5 5.733115 12.99851 -81.581835 -77.628788 222.12921154 6.134533 254.091839

6 -14.058868 -23.50412 -15.226861 -27.438127 9.75009541 8.231017 -368.668129

7 -143.968089 933.95243 -11.264131 -18.845955 -12.60991726 -12.313294 -10.288273

Ca Ba Fe

2 -4.495158 -6.789358 0.6437630

3 5.646475 2.223870 0.8239636

5 16.994975 -169.437118 272.7346400

6 -28.089356 -227.566859 -319.8861733

7 -17.173076 -6.244567 -95.8431176

Std. Errors:

(Intercept) RI Na Mg Al Si K

2 0.03920231 0.06701881 0.5768557 0.85545632 1.65179176 0.1511549 2.289055e+00

3 0.06114337 0.09642064 0.8058941 1.12917704 1.76232222 0.2059976 2.938445e+00

5 0.06245625 0.08769877 12.2913630 8.40708139 1.89682777 2.6617805 2.304166e+00

6 0.01613113 0.02450860 0.2228231 0.03882437 0.01984018 1.1730231 4.111865e-10

7 0.10335390 0.16375551 2.0999998 2.96043069 6.87912454 0.7338250 5.774488e+00

Ca Ba Fe

2 0.5632691 2.036229e+00 2.320576e+00

3 0.6197758 4.502377e+00 3.847483e+00

5 5.3353436 7.067090e+00 1.955795e-02

6 0.1572593 3.993370e-10 2.896739e-22

7 2.6269938 5.059777e+00 1.303330e+00

Residual Deviance: 177.3607

AIC: 277.3607

>

> predictions\_train3 <- predict(model3, train)

> con\_mat\_train\_model3 <- table(predicted = predictions\_train3, actual = train$Type)

> con\_mat\_train\_model3

actual

predicted 1 2 3 5 6 7

1 34 13 7 0 0 0

2 15 44 4 0 0 1

3 3 0 2 0 0 0

5 0 0 0 10 0 0

6 0 0 0 0 7 0

7 0 0 0 0 0 21

> accuracy\_train\_model3 <- sum(diag(con\_mat\_train\_model3)) / sum(con\_mat\_train\_model3)

> accuracy\_train\_model3

[1] 0.7329193

> error\_rate\_train\_model3 <- 1 - accuracy\_train\_model3

> error\_rate\_train\_model3

[1] 0.2670807

>

> predictions\_test3 <- predict(model3, test)

> con\_mat\_test\_model3 <- table(predicted = predictions\_test3, actual = test$Type)

> con\_mat\_test\_model3

actual

predicted 1 2 3 5 6 7

1 13 4 2 0 0 0

2 4 12 2 1 0 1

3 1 0 0 0 0 0

5 0 2 0 2 0 2

6 0 1 0 0 2 0

7 0 0 0 0 0 4

> accuracy\_test\_model3 <- sum(diag(con\_mat\_test\_model3)) / sum(con\_mat\_test\_model3)

> accuracy\_test\_model3

[1] 0.6226415

> error\_rate\_test\_model3 <- 1 - accuracy\_train\_model3

> error\_rate\_test\_model3

[1] 0.2670807

>

>

> model4 <- multinom(Type ~ Si + K + Ca + Ba + Fe, data = train, maxit = 1000)

# weights: 42 (30 variable)

initial value 288.473275

iter 10 value 205.965189

iter 20 value 155.373994

iter 30 value 150.953991

iter 40 value 145.643947

iter 50 value 143.170483

iter 60 value 142.438705

iter 70 value 141.691530

iter 80 value 141.064968

iter 90 value 140.177861

iter 100 value 138.642205

iter 110 value 138.399477

iter 120 value 136.553486

iter 130 value 136.513584

iter 140 value 136.300029

iter 150 value 135.732249

iter 160 value 135.716372

iter 170 value 135.311145

iter 180 value 135.104716

final value 135.083329

converged

> summary(model4)

Call:

multinom(formula = Type ~ Si + K + Ca + Ba + Fe, data = train,

maxit = 1000)

Coefficients:

(Intercept) Si K Ca Ba Fe

2 5.918628 -0.1850184 4.4373011 0.6122962 1.2768883 -0.3208501

3 46.376637 -0.6368082 -0.3132222 -0.1581747 -1.0258372 -1.2687962

5 -83.639168 0.6612129 17.9719726 2.5082157 -0.1476698 3.6152259

6 -261.603409 3.8580477 -167.6912440 -1.5795150 -34.5116136 -82.6154120

7 -321.544481 4.4026375 0.1117102 -0.3228178 10.4514431 -34.7591492

Std. Errors:

(Intercept) Si K Ca Ba Fe

2 0.181144854 0.03732824 1.432751530 0.2513852 2.203916e+00 1.996694e+00

3 0.033493868 0.05670796 1.742530391 0.4103424 3.760468e+00 3.395902e+00

5 0.088578009 0.10633811 4.062301615 0.5524033 2.428180e+00 4.237475e+00

6 0.009440163 0.22062122 0.005634213 1.6293712 1.514334e-06 2.760710e-07

7 0.061870446 0.10700022 2.844427034 0.7900601 2.126609e+00 1.150811e-01

Residual Deviance: 270.1667

AIC: 330.1667

>

> predictions\_train4 <- predict(model4, train)

> con\_mat\_train\_model4 <- table(predicted = predictions\_train4, actual = train$Type)

> #con\_mat\_train\_model4

> accuracy\_train\_model4 <- sum(diag(con\_mat\_train\_model4)) / sum(con\_mat\_train\_model4)

> accuracy\_train\_model4

[1] 0.5900621

> error\_rate\_train\_model4 <- 1 - accuracy\_train\_model4

> error\_rate\_train\_model4

[1] 0.4099379

>

> predictions\_test4 <- predict(model4, test)

> con\_mat\_test\_model4 <- table(predicted = predictions\_test4, actual = test$Type)

> #con\_mat\_test\_model4

> accuracy\_test\_model4 <- sum(diag(con\_mat\_test\_model4)) / sum(con\_mat\_test\_model4)

> accuracy\_test\_model4

[1] 0.5471698

> error\_rate\_test\_model4 <- 1 - accuracy\_train\_model4

> error\_rate\_test\_model4

[1] 0.4099379

>

**Note:** The best model is Model-4, because it is having the lowest AIC value.