**CSE3506 Essentials of Data Analytics**

Name : **Sparsh Raj**

Reg. No. : **19BPS1028**

Lab Exercise: 5 – Logistic Regression Modelling

(Dataset: Titanic - Machine Learning from Disaster)

**Objective:**

Perform logistic regression, build the model from the training dataset and use it to predict on the testing set.

**Methods:**

Loading training and test data sets onto R

train<-read.csv("train.csv")

test<-read.csv("test.csv")

str(train)

## 'data.frame': 891 obs. of 12 variables:

## $ PassengerId: int 1 2 3 4 5 6 7 8 9 10 ...

## $ Survived : int 0 1 1 1 0 0 0 0 1 1 ...

## $ Pclass : int 3 1 3 1 3 3 1 3 3 2 ...

## $ Name : chr "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)" ...

## $ Sex : chr "male" "female" "female" "female" ...

## $ Age : num 22 38 26 35 35 NA 54 2 27 14 ...

## $ SibSp : int 1 1 0 1 0 0 0 3 0 1 ...

## $ Parch : int 0 0 0 0 0 0 0 1 2 0 ...

## $ Ticket : chr "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...

## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...

## $ Cabin : chr "" "C85" "" "C123" ...

## $ Embarked : chr "S" "C" "S" "S" ...

str(test)

## 'data.frame': 418 obs. of 11 variables:

## $ PassengerId: int 892 893 894 895 896 897 898 899 900 901 ...

## $ Pclass : int 3 3 2 3 3 3 3 2 3 3 ...

## $ Name : chr "Kelly, Mr. James" "Wilkes, Mrs. James (Ellen Needs)" "Myles, Mr. Thomas Francis" "Wirz, Mr. Albert" ...

## $ Sex : chr "male" "female" "male" "male" ...

## $ Age : num 34.5 47 62 27 22 14 30 26 18 21 ...

## $ SibSp : int 0 1 0 0 1 0 0 1 0 2 ...

## $ Parch : int 0 0 0 0 1 0 0 1 0 0 ...

## $ Ticket : chr "330911" "363272" "240276" "315154" ...

## $ Fare : num 7.83 7 9.69 8.66 12.29 ...

## $ Cabin : chr "" "" "" "" ...

## $ Embarked : chr "Q" "S" "Q" "S" ...

colnames(train) <- tolower(colnames(train))

colnames(test) <- tolower(colnames(test))

Data Cleaning

clean\_variable <- **function**(df, variable, group, lookup\_table){

df[variable] <- apply(df[, c(variable, group)], 1, FUN=**function**(x) {**if** (is.na(x[1]) | x[1]==0) **return**(lookup\_table[lookup\_table[group]==x[2], variable]) **else** **return**(x[1])})

**return**(df)

}

fare\_pclass <- aggregate(fare ~ pclass, data=train, FUN=mean)

train <- clean\_variable(train, "fare", "pclass", fare\_pclass)

test <- clean\_variable(test, "fare", "pclass", fare\_pclass)

Missing data imputation: Filling missing age values with median value

age\_sex <- aggregate(age ~ sex, data=train, FUN=median)

train <- clean\_variable(train, "age", "sex", age\_sex)

test <- clean\_variable(test, "age", "sex", age\_sex)

Subsetting cross validation set from training set.

train$age <- as.numeric(train$age)

train$fare <- as.numeric(train$fare)

train$pclass <- as.factor(train$pclass)

size\_train <- nrow(train)

sample\_index <- sample.int(size\_train, size = floor(0.4\*size\_train))

cv <- train[sample\_index,]

train\_new <- train[setdiff(seq(1:size\_train), sample\_index), ]

Modelling the training set.

**library**(aod)

## Warning: package 'aod' was built under R version 4.1.2

mylogit <- glm(survived ~ pclass + sex + age + sibsp + parch + fare, family=binomial(link=logit), data=train\_new)

Check data fitting for training set and cross validation set.

predict\_survive <- **function**(mylogit, df, prob){

predict\_s <- predict(mylogit, newdata=df, type="response")

**return**(sapply(predict\_s, FUN=**function**(x){**if** (x>prob) **return**(1) **else** **return**(0)}))

}

threshold\_parameter <- **function**(mylogit, df, parameter\_set){

predict\_err <- parameter\_set

k=0

**for** (par **in** parameter\_set){

k <- k + 1

predict\_df <- predict\_survive(mylogit, df, par)

predict\_err[k] <- sum((predict\_df - df$survived)^2)

}

index <- order(predict\_err)[1]

**return**(parameter\_set[index])

}

par\_set <- seq(0.1,0.9,0.05)

best\_prob <- threshold\_parameter(mylogit, cv, par\_set)

predict\_cv <- predict\_survive(mylogit, cv, best\_prob)

predict\_cv\_error <- sum((predict\_cv-cv$survived)^2)/nrow(cv)

predict\_train <- predict\_survive(mylogit, train\_new, best\_prob)

predict\_train\_error <- sum((predict\_train-train\_new$survived)^2)/nrow(train\_new)

Prediction based on test set.

test$age <- as.numeric(test$age)

test$fare <- as.numeric(test$fare)

test$pclass <- as.factor(test$pclass)

test$survived <- predict\_survive(mylogit, test, best\_prob)

pred <- test[c("passengerid", "survived")]

colnames(pred) <- c("PassengerId", "Survived")

pred

## PassengerId Survived

## 1 892 0

## 2 893 0

## 3 894 0

## 4 895 0

## 5 896 1

## 6 897 0

## 7 898 1

## 8 899 0

## 9 900 1

## 10 901 0

## 11 902 0

## 12 903 0

## 13 904 1

## 14 905 0

## 15 906 1

## 16 907 1

## 17 908 0

## 18 909 0

## 19 910 1

## 20 911 1

## 21 912 0

## 22 913 0

## 23 914 1

## 24 915 1

## 25 916 1

## 26 917 0

## 27 918 1

## 28 919 0

## 29 920 0

## 30 921 0

## 31 922 0

## 32 923 0

## 33 924 0

## 34 925 1

## 35 926 0

## 36 927 0

## 37 928 1

## 38 929 1

## 39 930 0

## 40 931 0

## 41 932 0

## 42 933 1

## 43 934 0

## 44 935 1

## 45 936 1

## 46 937 0

## 47 938 0

## 48 939 0

## 49 940 1

## 50 941 1

## 51 942 1

## 52 943 0

## 53 944 1

## 54 945 1

## 55 946 0

## 56 947 0

## 57 948 0

## 58 949 0

## 59 950 0

## 60 951 1

## 61 952 0

## 62 953 0

## 63 954 0

## 64 955 1

## 65 956 0

## 66 957 1

## 67 958 1

## 68 959 0

## 69 960 1

## 70 961 1

## 71 962 1

## 72 963 0

## 73 964 1

## 74 965 1

## 75 966 1

## 76 967 1

## 77 968 0

## 78 969 1

## 79 970 0

## 80 971 1

## 81 972 0

## 82 973 0

## 83 974 0

## 84 975 0

## 85 976 0

## 86 977 0

## 87 978 1

## 88 979 1

## 89 980 1

## 90 981 0

## 91 982 1

## 92 983 0

## 93 984 1

## 94 985 0

## 95 986 1

## 96 987 0

## 97 988 1

## 98 989 0

## 99 990 1

## 100 991 0

## 101 992 1

## 102 993 0

## 103 994 0

## 104 995 0

## 105 996 1

## 106 997 0

## 107 998 0

## 108 999 0

## 109 1000 0

## 110 1001 0

## 111 1002 0

## 112 1003 1

## 113 1004 1

## 114 1005 1

## 115 1006 1

## 116 1007 0

## 117 1008 0

## 118 1009 1

## 119 1010 1

## 120 1011 1

## 121 1012 1

## 122 1013 0

## 123 1014 1

## 124 1015 0

## 125 1016 0

## 126 1017 1

## 127 1018 0

## 128 1019 1

## 129 1020 0

## 130 1021 0

## 131 1022 0

## 132 1023 0

## 133 1024 1

## 134 1025 0

## 135 1026 0

## 136 1027 0

## 137 1028 0

## 138 1029 0

## 139 1030 1

## 140 1031 0

## 141 1032 0

## 142 1033 1

## 143 1034 0

## 144 1035 0

## 145 1036 0

## 146 1037 0

## 147 1038 1

## 148 1039 0

## 149 1040 1

## 150 1041 0

## 151 1042 1

## 152 1043 0

## 153 1044 0

## 154 1045 1

## 155 1046 0

## 156 1047 0

## 157 1048 1

## 158 1049 1

## 159 1050 0

## 160 1051 1

## 161 1052 1

## 162 1053 0

## 163 1054 1

## 164 1055 0

## 165 1056 0

## 166 1057 1

## 167 1058 0

## 168 1059 0

## 169 1060 1

## 170 1061 1

## 171 1062 0

## 172 1063 0

## 173 1064 0

## 174 1065 0

## 175 1066 0

## 176 1067 1

## 177 1068 1

## 178 1069 0

## 179 1070 1

## 180 1071 1

## 181 1072 0

## 182 1073 0

## 183 1074 1

## 184 1075 0

## 185 1076 1

## 186 1077 0

## 187 1078 1

## 188 1079 0

## 189 1080 0

## 190 1081 0

## 191 1082 0

## 192 1083 1

## 193 1084 0

## 194 1085 0

## 195 1086 0

## 196 1087 0

## 197 1088 1

## 198 1089 1

## 199 1090 0

## 200 1091 1

## 201 1092 1

## 202 1093 0

## 203 1094 0

## 204 1095 1

## 205 1096 0

## 206 1097 1

## 207 1098 1

## 208 1099 0

## 209 1100 1

## 210 1101 0

## 211 1102 0

## 212 1103 0

## 213 1104 0

## 214 1105 1

## 215 1106 0

## 216 1107 0

## 217 1108 1

## 218 1109 0

## 219 1110 1

## 220 1111 0

## 221 1112 1

## 222 1113 0

## 223 1114 1

## 224 1115 0

## 225 1116 1

## 226 1117 1

## 227 1118 0

## 228 1119 1

## 229 1120 0

## 230 1121 0

## 231 1122 0

## 232 1123 1

## 233 1124 0

## 234 1125 0

## 235 1126 0

## 236 1127 0

## 237 1128 0

## 238 1129 0

## 239 1130 1

## 240 1131 1

## 241 1132 1

## 242 1133 1

## 243 1134 0

## 244 1135 0

## 245 1136 0

## 246 1137 0

## 247 1138 1

## 248 1139 0

## 249 1140 1

## 250 1141 1

## 251 1142 1

## 252 1143 0

## 253 1144 1

## 254 1145 0

## 255 1146 0

## 256 1147 0

## 257 1148 0

## 258 1149 0

## 259 1150 1

## 260 1151 0

## 261 1152 0

## 262 1153 0

## 263 1154 1

## 264 1155 1

## 265 1156 0

## 266 1157 0

## 267 1158 1

## 268 1159 0

## 269 1160 1

## 270 1161 0

## 271 1162 0

## 272 1163 0

## 273 1164 1

## 274 1165 1

## 275 1166 0

## 276 1167 1

## 277 1168 0

## 278 1169 0

## 279 1170 0

## 280 1171 0

## 281 1172 1

## 282 1173 0

## 283 1174 1

## 284 1175 1

## 285 1176 1

## 286 1177 0

## 287 1178 0

## 288 1179 1

## 289 1180 0

## 290 1181 0

## 291 1182 1

## 292 1183 1

## 293 1184 0

## 294 1185 0

## 295 1186 0

## 296 1187 0

## 297 1188 1

## 298 1189 0

## 299 1190 1

## 300 1191 0

## 301 1192 0

## 302 1193 0

## 303 1194 0

## 304 1195 0

## 305 1196 1

## 306 1197 1

## 307 1198 0

## 308 1199 0

## 309 1200 0

## 310 1201 0

## 311 1202 0

## 312 1203 0

## 313 1204 0

## 314 1205 1

## 315 1206 1

## 316 1207 1

## 317 1208 0

## 318 1209 0

## 319 1210 0

## 320 1211 0

## 321 1212 0

## 322 1213 0

## 323 1214 0

## 324 1215 1

## 325 1216 1

## 326 1217 0

## 327 1218 1

## 328 1219 0

## 329 1220 0

## 330 1221 0

## 331 1222 1

## 332 1223 0

## 333 1224 0

## 334 1225 1

## 335 1226 0

## 336 1227 1

## 337 1228 0

## 338 1229 0

## 339 1230 0

## 340 1231 0

## 341 1232 0

## 342 1233 0

## 343 1234 0

## 344 1235 1

## 345 1236 0

## 346 1237 1

## 347 1238 0

## 348 1239 1

## 349 1240 0

## 350 1241 1

## 351 1242 1

## 352 1243 0

## 353 1244 0

## 354 1245 0

## 355 1246 1

## 356 1247 0

## 357 1248 1

## 358 1249 0

## 359 1250 0

## 360 1251 1

## 361 1252 0

## 362 1253 1

## 363 1254 1

## 364 1255 0

## 365 1256 1

## 366 1257 0

## 367 1258 0

## 368 1259 1

## 369 1260 1

## 370 1261 0

## 371 1262 0

## 372 1263 1

## 373 1264 0

## 374 1265 0

## 375 1266 1

## 376 1267 1

## 377 1268 1

## 378 1269 0

## 379 1270 0

## 380 1271 0

## 381 1272 0

## 382 1273 0

## 383 1274 1

## 384 1275 1

## 385 1276 0

## 386 1277 1

## 387 1278 0

## 388 1279 0

## 389 1280 0

## 390 1281 0

## 391 1282 1

## 392 1283 1

## 393 1284 0

## 394 1285 0

## 395 1286 0

## 396 1287 1

## 397 1288 0

## 398 1289 1

## 399 1290 0

## 400 1291 0

## 401 1292 1

## 402 1293 0

## 403 1294 1

## 404 1295 1

## 405 1296 0

## 406 1297 0

## 407 1298 0

## 408 1299 0

## 409 1300 1

## 410 1301 1

## 411 1302 1

## 412 1303 1

## 413 1304 1

## 414 1305 0

## 415 1306 1

## 416 1307 0

## 417 1308 0

## 418 1309 0