GitHub Gist


tristantao / titanic-R
Last active 18 hours ago

Source code for titanic

titanic-R

```
http://leada.s3.amazonaws.com/titanic_data/train.csv
http://leada.s3.amazonaws.com/titanic_data/test.csv

?setwd
setwd('path-to-folder')
getwd()

trainData <- read.csv("train.csv", header = TRUE, stringsAsFactors = FALSE)
testData <- read.csv("test.csv", header = TRUE, stringsAsFactors = FALSE)


#############
#Train Data#
#############
head(trainData)
tail(trainData)

trainData

counts <- table(trainData$Survived, trainData$Sex)
barplot(counts, xlab = "Gender", ylab = "Number of People", main = "Survival by Sex")
counts[2] / (counts[1] + counts[2])
counts[4] / (counts[3] + counts[4])

trainData$Age
mean_age <- round(mean(trainData$Age,na.rm=T), digits = 3)

for (i in 1:nrow(trainData)) {
  if (is.na(trainData$Age[i])) {
    trainData$Age[i] <- mean_age
  }
}

#############
##Test Data#
#############
head(testData, 10)
tail(testData, 10)
testData


plot(density(testData$Age, na.rm = TRUE), main = "TestData Age Density")
plot(density(trainData$Age, na.rm = TRUE), main = "TrainData Age Density")

test_mean_age <- round(mean(testData$Age, na.rm= T), digits = 3)

for (i in 1:nrow(testData)) {
  if (is.na(testData$Age[i])) {
    testData$Age[i] <- test_mean_age
  }
}

######################
#Classification Tree#
######################
head(trainData, 1)
library('rpart')
tree_model <- rpart(Survived ~ Pclass + Sex + Age, data = trainData, method = "class")

plot(tree_model)
text(tree_model)
test_predictions <- round(predict(tree_model, newdata = testData)[, 2], 0)

model_submission <- cbind(testData$PassengerId, test_predictions)

colnames(model submission) <- c("PassengerId", "Survived")
```

```
66
67  write.csv(model_submission, "mysubmission.csv", row.names = FALSE)
68
69  #################
70  #Improve Scores#
71  #################
72  trainData["Child"] <- NA
73
74  for (i in 1:nrow(trainData)) {
75    if (trainData$Age[i] <= 18) {
76      trainData$Child[i] <- 1
77    } else {
78      trainData$Child[i] <- 2
79    }
80  }
81
82  testData["Child"] <- NA
83  for (i in 1:nrow(testData)) {
84    if (testData$Age[i] <= 18) {
85      testData$Child[i] <- 1
86    } else {
87      testData$Child[i] <- 2
88    }
89  }
90
91
92  ########################
93  ####Additiona Work#####
94  ########################
95
96  master_vector = grep("Master.",trainData$Name, fixed=TRUE)
97  miss_vector = grep("Miss.", trainData$Name, fixed=TRUE)
98  mrs_vector = grep("Mrs.", trainData$Name, fixed=TRUE)
99  mr_vector = grep("Mr.", trainData$Name, fixed=TRUE)
100 dr_vector = grep("Dr.", trainData$Name, fixed=TRUE)
101
102 for(i in master_vector) {
103   trainData$Name[i] = "Master"
104 }
105 for(i in miss_vector) {
106   trainData$Name[i] = "Miss"
107 }
108 for(i in mrs_vector) {
109   trainData$Name[i] = "Mrs"
110 }
111 for(i in mr_vector) {
112   trainData$Name[i] = "Mr"
113 }
114 for(i in dr_vector) {
115   trainData$Name[i] = "Dr"
116 }
117
118 master_age = round(mean(trainData$Age[trainData$Name == "Master"], na.rm = TRUE), digits = 2)
119 miss_age = round(mean(trainData$Age[trainData$Name == "Miss"], na.rm = TRUE), digits =2)
120 mrs_age = round(mean(trainData$Age[trainData$Name == "Mrs"], na.rm = TRUE), digits = 2)
121 mr_age = round(mean(trainData$Age[trainData$Name == "Mr"], na.rm = TRUE), digits = 2)
122 dr_age = round(mean(trainData$Age[trainData$Name == "Dr"], na.rm = TRUE), digits = 2)
123
124 for (i in 1:nrow(trainData)) {
125   if (is.na(trainData[i,5])) {
126     if (trainData$Name[i] == "Master") {
127       trainData$Age[i] = master_age
128     } else if (trainData$Name[i] == "Miss") {
129       trainData$Age[i] = miss_age
130     } else if (trainData$Name[i] == "Mrs") {
131       trainData$Age[i] = mrs_age
132     } else if (trainData$Name[i] == "Mr") {
133       trainData$Age[i] = mr_age
134     } else if (trainData$Name[i] == "Dr") {
135       trainData$Age[i] = dr_age
136     } else {
137       print("Uncaught Title")
138     }
139   }
140 }
141
142 trainData["Family"] = NA
143 for(i in 1:nrow(trainData)) {
144   x = trainData$SibSp[i]
145   y = trainData$Parch[i]
146   trainData$Family[i] = x + y + 1
```

```
  trainData$Family[i] = x + y + 1
}

trainData["Mother"]
for(i in 1:nrow(trainData)) {
  if(trainData$Name[i] == "Mrs" & trainData$Parch[i] > 0) {
    trainData$Mother[i] = 1
  } else {
    trainData$Mother[i] = 2
  }
}

####Don't forget to do the above also for the test data! ######
```

(/ashokbazaarvoice)