Titanic\_Random Forest

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Read the titanic dataset which divided into train.csv and test.csv from the following website.

train\_data <- read.csv(file="https://raw.githubusercontent.com/agconti/kaggle-titanic/master/data/train.csv",header=T,sep=",")  
test\_data <- read.csv(file="https://raw.githubusercontent.com/agconti/kaggle-titanic/master/data/test.csv",header=T,sep=",")

str(train\_data)

## '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 : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 581 ...  
## $ Sex : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...  
## $ 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 : Factor w/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 ...  
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...  
## $ Cabin : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...  
## $ Embarked : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...

str(test\_data)

## '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 : Factor w/ 418 levels "Abbott, Master. Eugene Joseph",..: 210 409 273 414 182 370 85 58 5 104 ...  
## $ Sex : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...  
## $ 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 : Factor w/ 363 levels "110469","110489",..: 153 222 74 148 139 262 159 85 101 270 ...  
## $ Fare : num 7.83 7 9.69 8.66 12.29 ...  
## $ Cabin : Factor w/ 77 levels "","A11","A18",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ Embarked : Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...

test\_data$Survived <- NA  
Full\_data <- rbind(train\_data, test\_data)  
Full\_data$Survived = as.factor(Full\_data$Survived)  
Full\_data$Pclass = as.factor(Full\_data$Pclass)  
Full\_data$Name = as.character(Full\_data$Name)  
Full\_data$Title <- sapply(Full\_data$Name, FUN=function(x) {strsplit(x, split='[,.]')[[1]][2]})  
Full\_data$Title <- sub(' ', '', Full\_data$Title)  
Full\_data$Title[Full\_data$Title %in% c('Mlle', 'Ms', 'Dona', 'Lady')] <- 'Miss'  
Full\_data$Title[Full\_data$Title %in% c('Mme')] <- 'Mrs'  
Full\_data$Title[Full\_data$Title %in% c('Capt', 'Col', 'Don', 'Dr', 'Jonkheer', 'Rev', 'the Countess', 'Major', 'Sir')] <- 'Officer'  
mean\_mr = mean(Full\_data$Age[Full\_data$Title == 'Mr' & !is.na(Full\_data$Age)])  
Full\_data$Age[Full\_data$Title == 'Mr' & is.na(Full\_data$Age)]=mean\_mr  
mean\_mrs = mean(Full\_data$Age[Full\_data$Title == 'Mrs' & !is.na(Full\_data$Age)])  
Full\_data$Age[Full\_data$Title == 'Mrs' & is.na(Full\_data$Age)]=mean\_mrs  
mean\_master = mean(Full\_data$Age[Full\_data$Title == 'Master' & !is.na(Full\_data$Age)])  
Full\_data$Age[Full\_data$Title == 'Master' & is.na(Full\_data$Age)]=mean\_master  
mean\_miss = mean(Full\_data$Age[Full\_data$Title == 'Miss' & !is.na(Full\_data$Age)])  
Full\_data$Age[Full\_data$Title == 'Miss' & is.na(Full\_data$Age)]=mean\_miss  
mean\_officer = mean(Full\_data$Age[Full\_data$Title == 'Officer' & !is.na(Full\_data$Age)])  
Full\_data$Age[Full\_data$Title == 'Officer' & is.na(Full\_data$Age)]=mean\_officer  
Full\_data$Fare[is.na(Full\_data$Fare)] = mean(Full\_data$Fare[which(Full\_data$Embarked == 'S' & Full\_data$Pclass == '3' & Full\_data$SibSp == 0 & Full\_data$Parch == 0)], na.rm = TRUE)  
Full\_data$Embarked[Full\_data$Embarked == ""] = 'S'  
Full\_data$Title = as.factor(Full\_data$Title)  
str(Full\_data)

## 'data.frame': 1309 obs. of 13 variables:  
## $ PassengerId: int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Survived : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...  
## $ Pclass : Factor w/ 3 levels "1","2","3": 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 : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...  
## $ Age : num 22 38 26 35 35 ...  
## $ 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 : Factor w/ 929 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 ...  
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...  
## $ Cabin : Factor w/ 187 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...  
## $ Embarked : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...  
## $ Title : Factor w/ 5 levels "Master","Miss",..: 3 4 2 4 3 3 3 1 4 4 ...

Consider Survived, Pclass, Sex, Age, Sibsp, Parch, Fare, Embarked and Title

Full\_data <- Full\_data[c(2:3,5:8,10,12:13)]

Splitting the dataset into training set and test set

training\_set <- Full\_data[!is.na(Full\_data$Survived),]  
test\_set <- Full\_data[is.na(Full\_data$Survived),]

Fitting Random Forest to the training test

library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

set.seed(1234)  
classifier = randomForest(Survived ~ ., data = training\_set, ntree = 1000)  
classifier

##   
## Call:  
## randomForest(formula = Survived ~ ., data = training\_set, ntree = 1000)   
## Type of random forest: classification  
## Number of trees: 1000  
## No. of variables tried at each split: 2  
##   
## OOB estimate of error rate: 16.5%  
## Confusion matrix:  
## 0 1 class.error  
## 0 500 49 0.08925319  
## 1 98 244 0.28654971

Predicting the test results

Survived\_pred = predict(classifier, newdata = test\_set)  
Survived\_pred

## 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906   
## 0 0 0 0 1 0 1 0 1 0 0 0 1 0 1   
## 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921   
## 1 0 0 0 1 0 1 1 0 1 0 1 0 0 0   
## 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936   
## 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1   
## 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951   
## 0 0 0 1 1 0 0 1 1 0 0 0 0 0 1   
## 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966   
## 0 0 0 1 1 1 1 0 0 1 1 0 0 0 1   
## 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981   
## 0 0 1 0 1 1 0 0 0 0 0 1 0 1 1   
## 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996   
## 0 0 1 0 0 0 1 0 1 0 1 0 0 0 1   
## 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011   
## 0 0 0 0 0 0 1 1 1 1 0 0 1 0 1   
## 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026   
## 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0   
## 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041   
## 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0   
## 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056   
## 1 0 0 1 0 0 1 0 0 1 1 1 1 0 0   
## 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071   
## 0 0 0 1 0 0 0 0 0 0 1 1 0 1 1   
## 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086   
## 0 0 1 0 1 0 1 0 0 0 0 0 1 0 1   
## 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101   
## 0 1 1 0 0 1 1 1 1 0 0 1 0 1 0   
## 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116   
## 0 0 0 1 0 0 1 0 1 0 1 0 1 0 1   
## 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131   
## 0 0 1 0 0 0 1 0 0 0 0 0 0 1 1   
## 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146   
## 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0   
## 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161   
## 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0   
## 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176   
## 0 0 1 1 0 1 0 0 0 0 0 1 1 0 1   
## 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191   
## 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0   
## 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206   
## 0 0 0 0 1 1 0 1 0 0 0 0 0 1 1   
## 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221   
## 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0   
## 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236   
## 1 0 0 1 0 0 0 0 0 1 0 0 0 1 1   
## 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251   
## 1 0 1 0 1 1 0 0 0 1 0 1 0 0 0   
## 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266   
## 0 1 1 0 1 0 0 0 1 0 0 1 0 0 1   
## 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281   
## 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0   
## 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296   
## 0 1 1 0 0 1 0 1 0 0 1 0 1 0 0   
## 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309   
## 0 0 0 1 1 1 1 0 0 1 0 0 1   
## Levels: 0 1

Saving the prediction result

Prediction <- data.frame(PassengerId = test\_data$PassengerId, Survived = Survived\_pred)  
write.csv(Prediction, file = 'Titanic\_RandomForest.csv', row.names = F)