**Introduction**

Here, I come Kaggle! Seeing Jack getting frozen in the cold sea and Rose just seeing him losing his breaths one by one made me wonder how many other ill-fated travellers actually survived the biggest ship mishap in the history of mankind. I have gone through a few more R scripts created by my fellow expert kagglers to learn the trick of the art and be able to make my predictions as close to the actual as I can. I will first explore the datasets, visualize the datasets, and then will try to build a model with the help of randomForest algorithm to predict survivals from the ill-fated maiden voyage of Titanic. Please don’t hesitate to give your precious feedbacks (I would love positive ones and appreciate negative ones) to help me better my Machine Learning curve.

My script will be basically divided into the following parts:

* Data exploration
  + Loading and checking data
  + Feature engineering
  + Missing value imputation
  + Data visualization
* Model Building
* Evaluation of model
* Predicting survivals

**Data Exploration**

I first began my project by loading all the required packages that I would need to finally be able to create an efficient algorithm to help me predict survivals more accurately. I will be using ‘data.table’ package to read my dataset fast. I will also be using ‘dplyr’ and ‘sqldf’ packages to manipulate my data. I will be using the very efficient data visualization packages, ‘ggplot2’, ‘ggthemes’, and ‘scales’ to present my findings in an interactive way. I will also be using ‘mice’ and ‘Hmisc’ packages to help me impute missing values efficiently. Finally, I will be using ‘caret’ and ‘randomForest’ packages to help me select the most important variables, create the model, predict the survivals, and evaluate the model.

#Loading packages

library(randomForest) #classification algorithm

library(ggplot2) #visualization

library(dplyr) #data manipulation

library(ggthemes) #data

**Loading the dataset**.

#loading data set

train <- read.csv('train1.csv', stringsAsFactors = FALSE)

test <- read.csv('test1.csv', stringsAsFactors = FALSE)

#checking head of train and test data

head(train)

head (test)

#combining both datasets to see complete data set

complete <- bind\_rows(train, test)

str(complete)

Checking dataset

1.2 Finding missing values and summary of dataset

#summary of complete dataset

summary (complete)

#finding columns with missing values

sapply(complete, function(x) sum(is.na(x)))

PassengerId Survived Pclass Name Sex

Min. : 1 Min. :0.0000 Min. :1.00 Length:1309 Length:1309

1st Qu.: 328 1st Qu.:0.000 1st Qu.:2.00 Class :character Class :character

Median : 655 Median :0.00 Median :3.00 Mode :character Mode :character

Mean : 655 Mean :0.3838 Mean :2.295

3rd Qu.: 982 3rd Qu.:1.0000 3rd Qu.:3.000

Max. :1309 Max. :1.0000 Max. :3.000

NA's :418

Age SibSp Parch Ticket Fare

Min. : 0.17 Min. :0.0000 Min. :0.000 Length:1309 Min. : 0.000

1st Qu.:21.00 1st Qu.:0.0000 1st Qu.:0.000 Class :character 1st Qu.: 7.896

Median :28.00 Median :0.000 Median :0.000 Mode :character Median : 14.454

Mean :29.88 Mean :0.4989 Mean :0.385 Mean : 33.295

3rd Qu.:39.00 3rd Qu.:1.0000 3rd Qu.:0.000 3rd Qu.: 31.275

Max. :80.00 Max. :8.0000 Max. :9.000 Max. :512.329

NA's :263 NA's :1

Cabin Embarked

Length:1309 Length:1309

Class :character Class :character

Mode :character Mode :character

PassengerId Survived Pclass Name Sex Age SibSp Parch

0 418 0 0 0 263 0 0

Ticket Fare Cabin Embarked

0 1 1014 2

There is a total of 1309 observations with 12 variables, divided into 2 datasets:

Train :- 891 observations with 12 variables

Test :- 418 observations with 11 variable

Variable names and description:

|  |  |
| --- | --- |
| **Variable Name** | **Description** |
| PassengerId | Passenger ID number |
| Survived | Survived (1) or Not Survived (0) |
| Pclass | Passenger’s class |
| Name | Passenger’s name |
| Sex | Male (1) or Female (0) |
| Age | Passenger’s age |
| SibSp | Number of siblings/spouses aboard |
| Parch | Number of parents/children aboard |
| Ticket | Ticket number |
| Fare | Exact fare |
| Cabin | Cabin number |
| Embarked | Port of embarkation |

**Exploratory Analysis**:

Relationship between Age and Survived

ggplot(complete[1:891,], aes(Age, fill = factor(Survived))) +

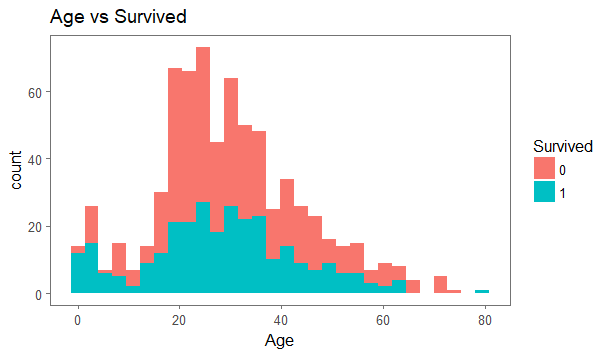
geom\_histogram(bins=30) +

theme\_few() +

xlab("Age") +

scale\_fill\_discrete(name = "Survived") +

ggtitle("Age vs Survived")



Inference: Passengers age between 20 and 40 survived the most

Relationship between sex and survived

ggplot(complete[1:891,], aes(Sex, fill = factor(Survived))) +

geom\_bar(stat = "count", position = 'dodge')+

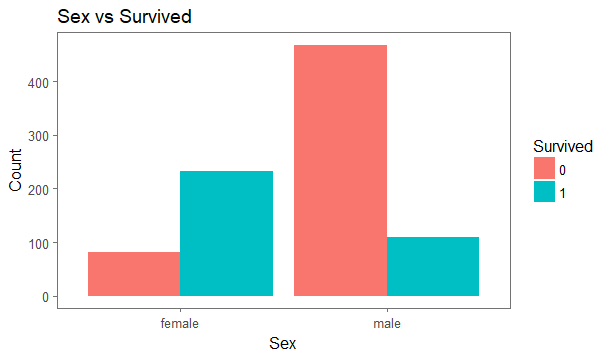
theme\_few() +

xlab("Sex") +

ylab("Count") +

scale\_fill\_discrete(name = "Survived") +

ggtitle("Sex vs Survived")



Inference: Ratio wise more female survived than male

tapply(complete[1:891,]$Survived, complete[1:891,]$Sex,mean)

female male

0.7420382 0.1889081

Meaning 74% of the female passengers survived versus 19% male passengers.

Sex/Age/Survived

ggplot(complete [1:891,], aes(Age, fill = factor(Survived))) +

geom\_histogram(bins=30) +

theme\_few() +

xlab("Age") +

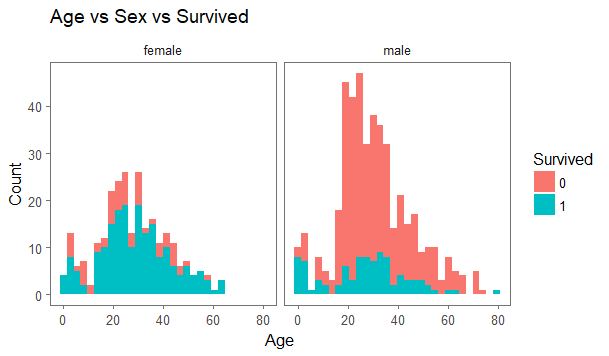
ylab("Count") +

facet\_grid(.~Sex)+

scale\_fill\_discrete(name = "Survived") +

theme\_few()+

ggtitle("Age vs Sex vs Survived")



PClass versus Survived

# Pclass vs Survived

tapply(complete[1:891,]$Survived,complete[1:891,]$Pclass,mean)

ggplot(complete [1:891,], aes(Pclass, fill = factor(Survived))) +

geom\_bar(stat = "count")+

theme\_few() +

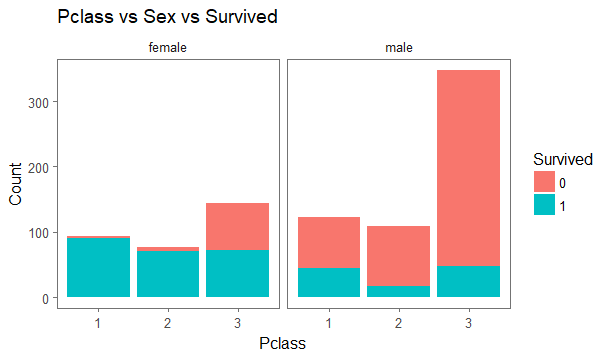
xlab("Pclass") +

facet\_grid(.~Sex)+

ylab("Count") +

scale\_fill\_discrete(name = "Survived") +

ggtitle("Pclass vs Sex vs Survived")



**Feature Engineering:** Basic feature engineering of Titanic data will involve breaking down of given variables to extract some more useful information like getting titles, converting rare titles to more general titles, and getting family name. Feature engineering will also involve getting further information about family size and structures as well as converting cabins into factors and creation of different age groups to analyse our dataset better.

Extracting Title of each passenger and tabulating them by sex

# Extracting Title from Complete dataset

complete$Title <- gsub('(.\*, )|(\\..\*)', '', complete$Name)

#Tabulate Title count by Sex

table(complete$Title, complete$Sex)

|  |  |  |
| --- | --- | --- |
|  | female | male |
| Capt | 0 | 1 |
| Col | 0 | 4 |
| Don | 0 | 1 |
| Dona | 1 | 0 |
| Dr | 1 | 7 |
| Jonkheer | 0 | 1 |
| Lady | 1 | 0 |
| Major | 0 | 2 |
| Master | 0 | 61 |
| Miss | 260 | 0 |
| Mlle | 2 | 0 |
| Mme | 1 | 0 |
| Mr | 0 | 757 |
| Mrs | 197 | 0 |
| Ms | 2 | 0 |
| Rev | 0 | 8 |
| Sir | 0 | 1 |
| the Countess | 1 | 0 |

The most common titles found are:

Master : male (61)

Miss : female (260)

Mr : male (757)

Mrs : female (197)

Also, Mlle and Ms also means Miss only so I will try to combine them into ‘Miss’ only. ‘Mme’ will be combined to ‘Mrs’.

The other less frequent titles, I will be segmenting into Salutation table and thus the I will get the following Title table by Sex.

#Reassigning other less frequent titles to salut\_title

salut\_title <- c("Capt", "Col", "Don", "Dona", "Dr",

"Jonkheer", "Lady", "Major", "Rev", "Sir", "the Countess")

#The French titles are combined to popular English title

complete$Title[complete$Title == 'Mlle'] <- 'Miss'

complete$Title[complete$Title == 'Ms'] <- 'Miss'

complete$Title[complete$Title == 'Mme'] <- 'Mrs'

#Assigning salut\_title to the complete dataset

complete$Title [complete$Title %in% salut\_title] <- "Salutation Title"

#Tabulate Sex by Title

table(complete$Title, complete$Sex)

|  |  |  |
| --- | --- | --- |
|  | female | male |
| Master | 0 | 61 |
| Miss | 264 | 0 |
| Mr | 0 | 757 |
| Mrs | 198 | 0 |
| Salutation | 4 | 25 |

Title vs Survived

ggplot(complete [1:891,], aes(Title, fill = factor(Survived))) +

geom\_bar(stat = "count")+

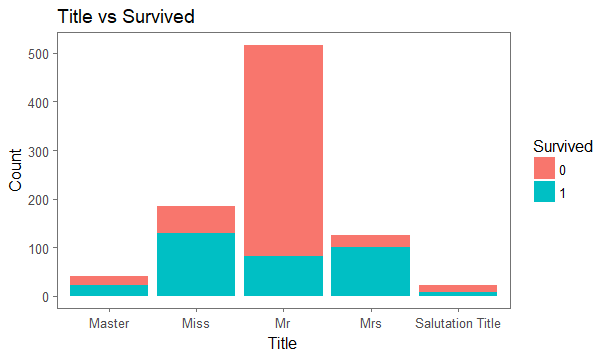
theme\_few() +

xlab("Title") +

ylab("Count") +

scale\_fill\_discrete(name = "Survived") +

ggtitle("Title vs Survived")



Extracting family name

#Extracting family name

complete$Family\_Name <- sapply (complete$Name, function (x)

strsplit(x, split = "[, .]")[[1]][1])

# Create a family size variable including the passenger themselves

complete$Fsize <- complete$SibSp + complete$Parch + 1

# Create a family variable

complete$Family <- paste(complete$Family\_Name, complete$Fsize, sep='-')

complete$Family

Family Size and Survival

Now, we will try to find out if there was a relationship between the survival of a family depending on the family size

# Use ggplot2 to visualize the relationship between family size & survival

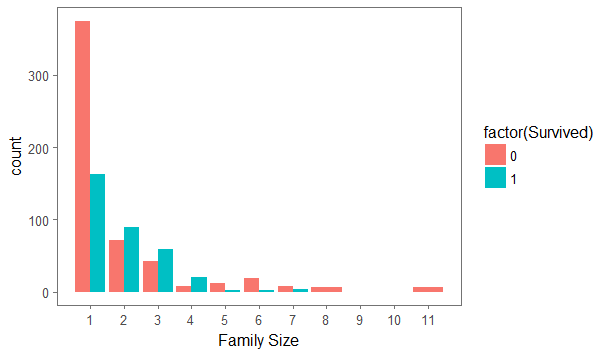
ggplot(complete [1:891,], aes(x = Fsize, fill = factor(Survived))) +

geom\_bar(stat='count', position='dodge') +

scale\_x\_continuous(breaks=c(1:11)) +

labs(x = 'Family Size') +

theme\_few()



A general inference from the above plot can be made that the family size of 2, 3, 4 had more survival ratio than the family size of 1 or the family size greater than 4. Thus, it gives us the option the divide family size into 3 categories, namely, ‘Single’, ‘Small’, and ‘Big’

complete$FsizeD [complete$Fsize==1] -> 'Single'

complete$FsizeD [complete$Fsize<=4 & complete$F\_size>1] -> 'Small'

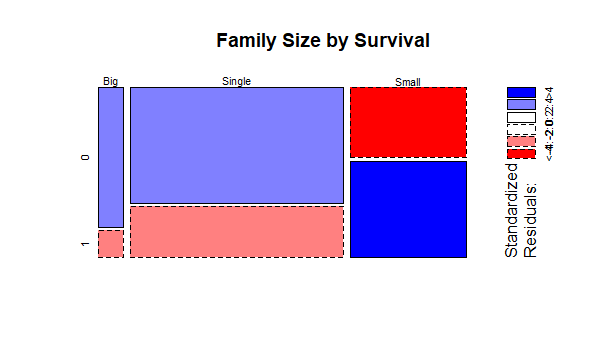
complete$FsizeD [complete$Fsize>4] -> 'Big'

Plotting Family size by survival:

table (complete$FsizeD)

mosaicplot (table (complete$FsizeD, complete$Survived),

+ main='Family Size by Survival', shade=TRUE)



**Big Single Small**

**82 790 437**

**Missing Value Imputation**

#Age = 263

#Fare = 1

#Embarked = 2

#Cabin = 1014

Embarked: Since Embarked has only 2 missing variables, we will impute it with the median.

median(complete$Embarked, na.rm = TRUE)

[1] "S"

Cabin: Since Cabin has just too many variables, but is still a very important variable to predict survival, we will create a dummy variable, “In\_Cabin” with values 0 (no) and 1 (yes).

complete$Cabin <- lapply(complete$Cabin, as.character)

complete$In\_Cabin <- ifelse(is.na (complete$Cabin), 0, 1)

table(complete$In\_Cabin)

0 1

1014 295

Fare : We need to take out the row containing missing data for fare to analyse it.

which(is.na(complete$Fare))

complete[1044, ]

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked | Title | Family\_Name | Fsize | Family | FsizeD | In\_Cabin |
| 1044 | NA | 3 | Storey, Mr. Thomas | male | 60.5 | 0 | 0 | 3701 | NA | NA | S | Mr | Storey | 1 | Storey-1 | Single | 0 |

Since Passenger ID 1044 was PClass 3, we can impute the Fare with the median of PClass 3.

tapply(complete$Fare, complete$Pclass,median, na.rm=TRUE)

complete$Fare[1044] <- median(complete[complete$Pclass == '3', ]$Fare, na.rm = TRUE)

complete$Fare[1044]

Age Imputation:

tapply(complete$Age, complete$Pclass,median, na.rm=TRUE)

tapply(complete$Age, complete$Title,median, na.rm=TRUE)

PClass/Age:

1 2 3

39 29 24

Title/Age:

Master Miss Mr Mrs Salutation Title 4.0 22.0 29.0 35.0 47.5

Create a new variable title.age:

title.age <- aggregate(complete$Age, by = list(complete$Title), FUN = function(x) median(x, na.rm = T))

complete[is.na(complete$Age), "Age"] <- apply(complete[is.na(complete$Age), ]

, 1, function(x) title.age[title.age[, 1]==x["Title"], 2])

#NA value count

sum(is.na(complete$Age)

c

Distribute the passengers into 2 categories, called Child and Adult, based on Age

#Distribute Age into 2 categories, Child and Adult

complete$Age\_Cat <- ifelse(complete$Age < 18, "Child", "Adult")

# Find total numbers of Child and Adult passengers

table(complete$Age\_Cat)

Adult Child

1147 162

Age Category versus Survived

# Age Category versus Survived

table(complete$Age\_Cat, complete$Survived)

0 1

Adult 495 279

Child 54 63

**Data Prediction:**

Creating training subset of important variables

#Making a subset of complete dataset

complete\_sub <- complete [, c(2, 3, 5, 6, 10, 12, 13, 17, 18, 19)]

str(complete)

#converting into factors

complete$Age\_Cat <- factor(complete$Age\_Cat)

complete$Sex <- factor(complete$Sex)

complete$Embarked <- factor(complete$Embarked)

complete$Title <- factor(complete$Title)

complete$Pclass <- factor(complete$Pclass)

complete$FsizeD <- factor(complete$FsizeD)

Creation of train and test datasets for model building and prediction

# Creating training and test variables from subset

train <- complete\_sub [1:891, ]

test <- complete\_sub [892:1309,]

Building a randomForest model

set.seed (221)

rf1<- randomForest(as.factor(Survived) ~.,data=train,ntrees=200,mtry=5,

importance=TRUE,strata=train$Survived,

sampsize=c(300,300),trace=TRUE,normalize = TRUE,

na.action=na.exclude)

Error Evaluation

#OOB Estimate and confusion matrix

print (rf1)

Call:

randomForest(formula = as.factor(Survived) ~ ., data = train, ntrees = 200, mtry = 6, importance = TRUE, strata = train$Survived, sampsize = c(300, 300), trace = TRUE, normalize = TRUE, na.action = na.exclude)

Type of random forest: classification

Number of trees: 500

No. of variables tried at each split: 6

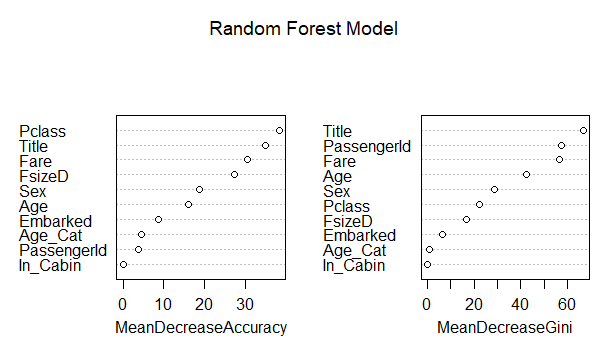
OOB estimate of error rate: 17.06%

Confusion matrix:

0 1 class.error

0 470 79 0.1438980

1 73 269 0.2134503



Predicting the survived passengers in test list:

#Prediction

prediction <- predict (rf1, test)

prediction

# Solution for submission

Survived\_List <- data.frame(PassengerID = test$PassengerId, Survived = prediction)

head (Survived\_List)

# write csv

write.csv(Survived\_List, file = 'Titanic\_Surv\_List.csv', row.names = F)

**Final Outcome:**

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