Titanic Survival Prediction

SAGNIK SAMANTA

CODSOFT TASK1

Use the Titanic dataset to build a model that predicts whether a passenger on the Titanic survived or not. The dataset typically used for this project contains information about individual passengers, such as their age, gender, ticket class, fare, cabin, and whether or not they survived.

PassengerId = PassengerId
Pclass = Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)
Survived= Survival (0 = No; 1 = Yes)
Name= Name of the Passengers
sex = Sex
age= Age
sibsp= Number of Siblings/Spouses Aboard
parch =Number of Parents/Children Aboard
ticket =Ticket Number
fare= Passenger Fare (British pound)
cabin =Cabin
embarked =Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

Importing Dataset

```
titanic=read.csv("C:/Users/shrey/Desktop/Datasets/Titanic.csv",sep =
",",header=TRUE)
```

Dataset Description

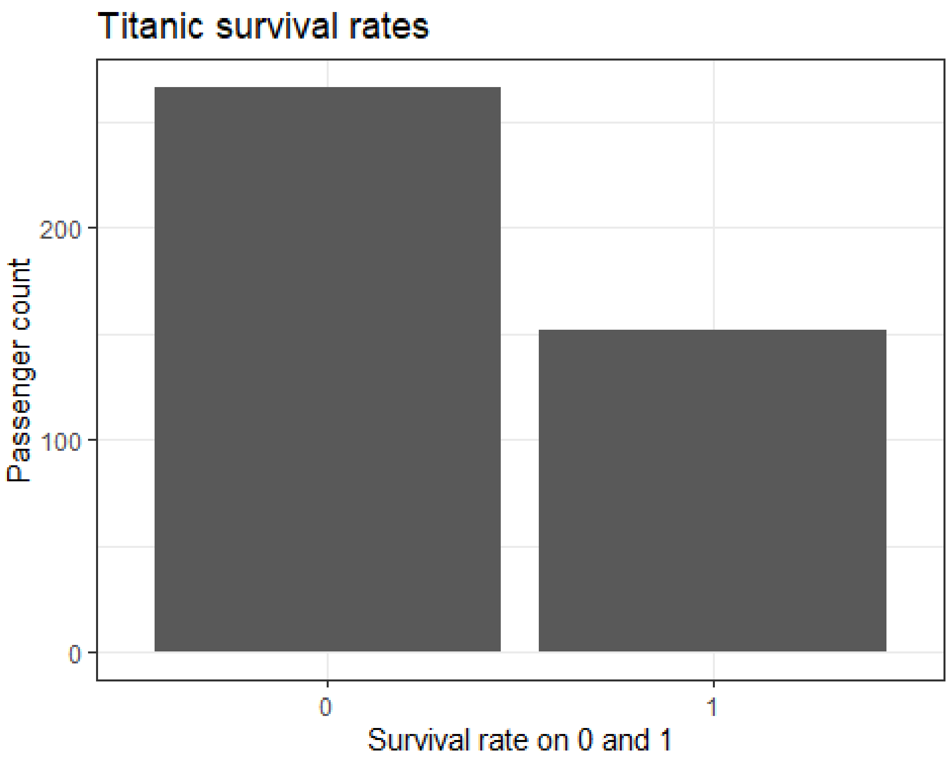
```
str(titanic)
```

```
'data.frame':	418 obs. of  12 variables:
PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
Survived   : int  0 1 0 0 1 0 1 0 1 0 ...
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 ...
```

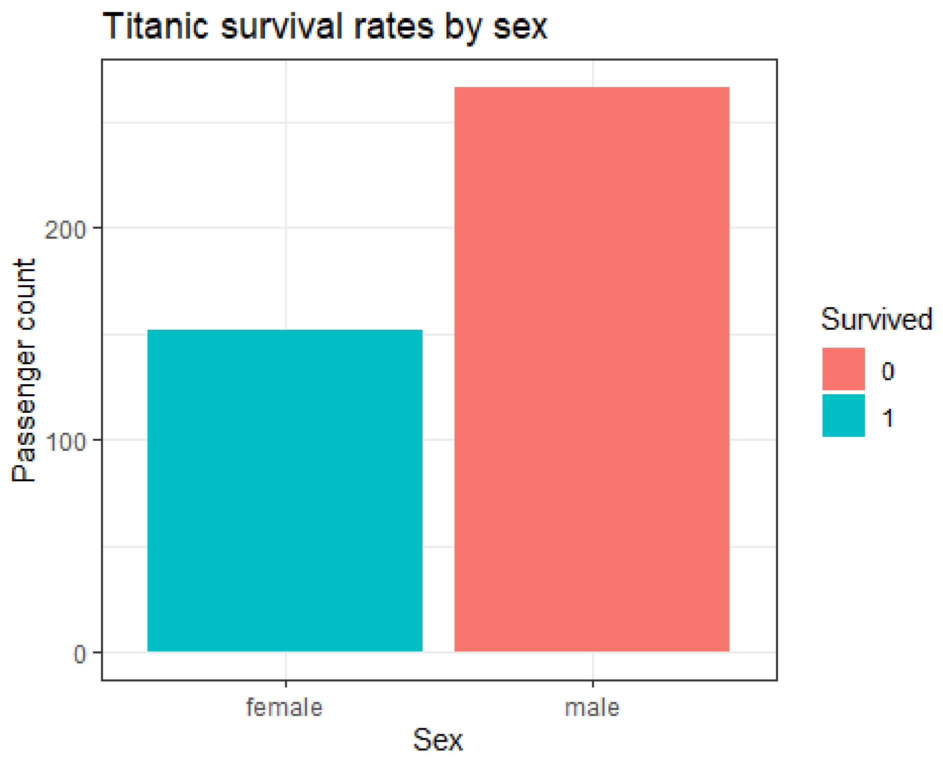
```
Cabin      : chr  "" "" "" "" ...
Embarked   : chr  "Q" "S" "Q" "S" ...

Hence in the titanic dataset there are 418 observations on 12 variables.

Let's check for any missing values in the data
colSums(is.na(titanic))

PassengerId    Survived      Pclass        Name         Sex         Age
          0           0           0           0           0          86
      SibSp       Parch      Ticket        Fare       Cabin    Embarked
          0           0           0           1           0           0

Checking for empty values
colSums(titanic=='')

PassengerId    Survived      Pclass        Name         Sex         Age
          0           0           0           0           0          NA
      SibSp       Parch      Ticket        Fare       Cabin    Embarked
          0           0           0          NA         327           0

Check number of uniques values for each of the column to find out columns
which we can convert to factors
sapply(titanic, function(x) length(unique(x)))

PassengerId    Survived      Pclass        Name         Sex         Age
        418           2           3         418           2          80
      SibSp       Parch      Ticket        Fare       Cabin    Embarked
          7           8         363         170          77           3

Missing values imputation
titanic$Embarked[titanic$Embarked==""]="S"
titanic$Age[is.na(titanic$Age)]=median(titanic$Age,na.rm=T)

Removing Cabin as it has very high missing values, passengerId, Ticket and
Name are not required
library(dplyr)

titanic1=titanic %>% select(-c(Cabin, PassengerId, Ticket, Name))

titanic$Survived=as.factor(titanic$Survived)
titanic$Pclass=as.factor(titanic$Pclass)
titanic$Sex=as.factor(titanic$Sex)
titanic$Embarked=as.factor(titanic$Embarked)
titanic$Cabin=as.factor(titanic$Cabin)

Create dummy variables for categorical variables
install.packages("dummy")
library(dummy)titanic2=dummy(x=titanic)

summary(titanic)
```

```
##   PassengerId     Survived Pclass      Name               Sex
##  Min.   : 892.0   0:266    1:107   Length:418         female:152
##  1st Qu.: 996.2   1:152    2: 93   Class :character   male  :266
##  Median :1100.5            3:218   Mode  :character
##  Mean   :1100.5
##  3rd Qu.:1204.8
##  Max.   :1309.0
##
##       Age             SibSp            Parch           Ticket
##  Min.   : 0.17   Min.   :0.0000   Min.   :0.0000   Length:418
##  1st Qu.:23.00   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median :27.00   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   :29.60   Mean   :0.4474   Mean   :0.3923
##  3rd Qu.:35.75   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :76.00   Max.   :8.0000   Max.   :9.0000
##
##       Fare                    Cabin     Embarked
##  Min.   :  0.000                  :327   C:102
##  1st Qu.:  7.896   B57 B59 B63 B66:  3   Q: 46
##  Median : 14.454   A34            :  2   S:270
##  Mean   : 35.627   B45            :  2
##  3rd Qu.: 31.500   C101           :  2
##  Max.   :512.329   C116           :  2
##  NA's   :1         (Other)        : 80
```

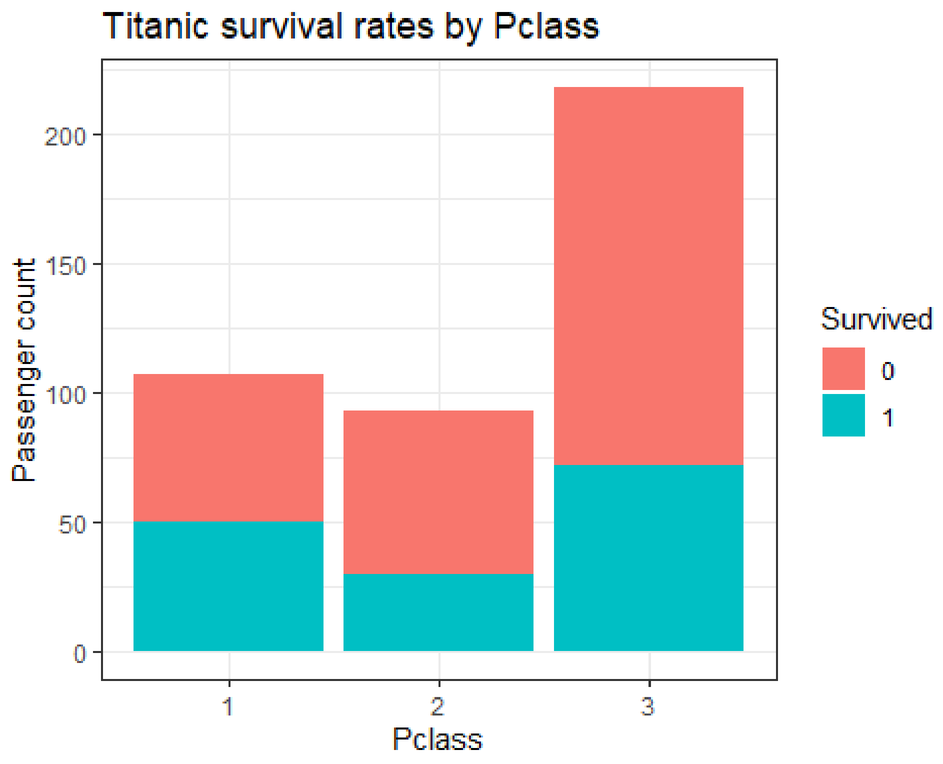
```
Plot how many survived and the percentage of female and male survived
install.packages("ggplot2")
library(ggplot2)

ggplot(titanic, aes(x = Survived)) +
  theme_bw()+
  geom_bar()+
  labs(x = "Survival rate on 0 and 1",
       y = "Passenger count",
       title = "Titanic survival rates")
```



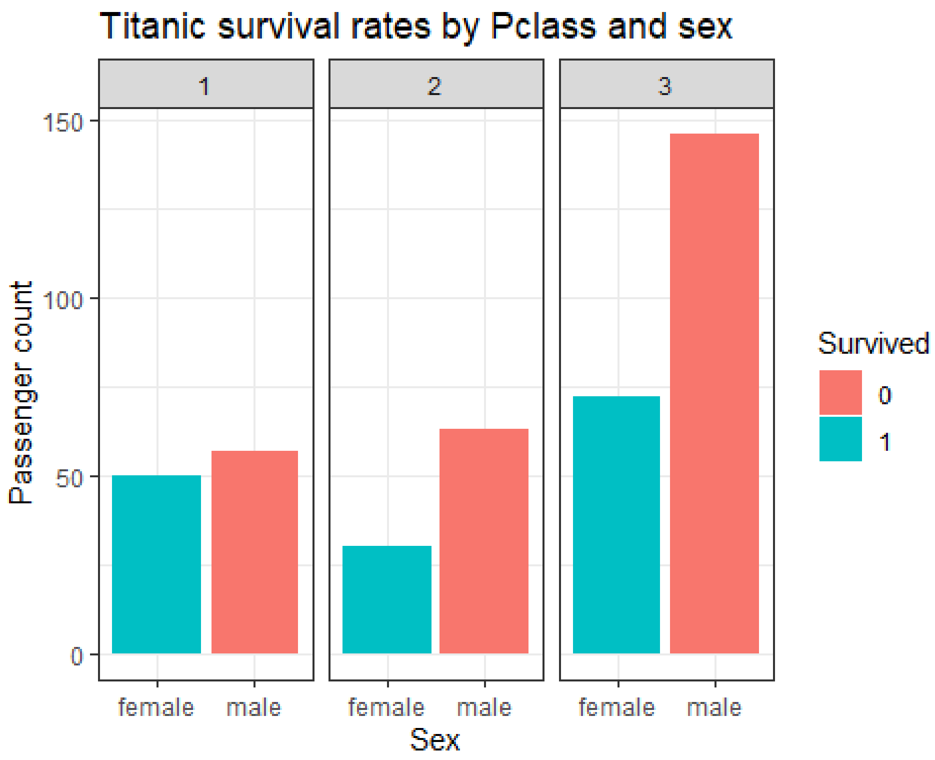
```
prop.table(table(titanic$Survived))


   0         1 
0.6363636 0.3636364 

Survival Rate by Gender
ggplot(titanic, aes(x = Sex, fill = Survived)) +
  theme_bw()+
  geom_bar()+
  labs(y = "Passenger count",
       title = "Titanic survival rates by sex")
```



```
prop.table(table(titanic$Sex))


  female      male
0.3636364 0.6363636

Survival Rate by Gender
ggplot(titanic, aes(x = Pclass,fill=Survived)) +
  theme_bw()+
  geom_bar()+
  labs(y = "Passenger count",
       title = "Titanic survival rates by Pclass")
```



```
prop.table(table(titanic$Pclass))
     1         2         3 
0.2559809 0.2224880 0.5215311



Survival rate by Pclass and gender
ggplot(titanic, aes(x = Sex, fill = Survived)) +
  theme_bw()+
  facet_wrap(~ Pclass)+
  geom_bar()+
  labs(y = "Passenger count",
       title = "Titanic survival rates by Pclass and sex")
```



```
Logistic Regression
install.packages("dplyr")
library(dplyr)

Splitting dataset
Use 80% of dataset as training set and remaining 20% as testing set
sample=sample(c(TRUE, FALSE), nrow(titanic), replace=TRUE, prob=c(0.80,0.20))
train=titanic[sample, ]
test=titanic[!sample, ]
# Training model
logistic_model=glm(Survived ~ Pclass + Sex + Age,data = train,family =
binomial(link = 'logit'))

glm_predict=predict(logistic_model, test, type = 'response')

Survivor=c()
for(i in 1:length(glm_predict)){
  if(glm_predict[i] > 0.9){
    Survivor[i] = "Alive"
  } else {
    Survivor[i] = "Dead"
  }
}



Final_data=cbind(PassengerId=test$PassengerId,Predicted=Survivor)
```

```r
Final_data=as.data.frame(Final_data)
View(Final_data)
```

	PassengerId	Predicted
1	899	Dead
2	903	Dead
3	904	Alive
4	913	Dead
5	928	Alive
6	931	Dead
7	940	Alive
8	951	Alive
9	953	Dead
10	954	Dead
11	955	Alive
12	962	Alive
13	968	Dead
14	980	Alive
15	983	Dead
16	991	Dead
17	993	Dead
18	996	Alive

	PassengerId	Predicted
19	1003	Alive
20	1007	Dead
21	1018	Dead
22	1024	Alive
23	1027	Dead
24	1037	Dead
25	1043	Dead
26	1045	Alive
27	1061	Alive
28	1062	Dead
29	1063	Dead
30	1065	Dead
31	1068	Alive
32	1076	Alive
33	1080	Alive
34	1096	Dead
35	1097	Dead
36	1112	Alive
37	1115	Dead

	PassengerId	Predicted
38	1119	Alive
39	1126	Dead
40	1128	Dead
41	1132	Alive
42	1133	Alive
43	1141	Alive
44	1143	Dead
45	1144	Dead
46	1158	Dead
47	1170	Dead
48	1177	Dead
49	1182	Dead
50	1183	Alive
51	1185	Dead
52	1189	Dead
53	1190	Dead
54	1193	Dead
55	1194	Dead
56	1196	Alive

	PassengerId	Predicted
57	1198	Dead
58	1201	Alive
59	1219	Dead
60	1220	Dead
61	1229	Dead
62	1230	Dead
63	1232	Dead
64	1234	Dead
65	1238	Dead
66	1243	Dead
67	1244	Dead
68	1249	Dead
69	1254	Alive
70	1255	Dead
71	1259	Alive
72	1265	Dead
73	1266	Alive
74	1267	Alive
75	1270	Dead

	PassengerId	Predicted
76	1276	Dead
77	1308	Dead

Showing 1 to 28 of 77 entries, 2 total columns