# Regression Analysis

MANOJ KUMAR - 2048015

20/02/2021

## 1.Install the package “titanic”.

```
# install.packages("titanic")
```

## 2.Load Titanic library to get the dataset

```
# Load Titanic library
library(titanic)

# Load the dataset
data("titanic_train")
data("titanic_test")
```

## 3.Set Survived column for test data to NA.

```
#Note: titanic_test$Survived <- NA

## Setting Survived column for test data to NA
titanic_test$Survived <- NA
```

## 4.Combine the Training and Testing dataset.

```
#Note: complete_data <- rbind(titanic_train, titanic_test)

complete_data <- rbind(titanic_train, titanic_test)
```

## 5.Get the data structure.

```
# Check data structure
str(complete_data)
```

```
## 'data.frame':    1309 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 He
ath (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" ...
```

## 6. Check for any missing values in the data.

```
# Total missing count
colSums(is.na(complete_data))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0         418           0           0           0         263
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           1           0           0
```

## 7.Check for any empty values.

```
colSums(complete_data=='')
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0          NA           0           0           0          NA
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0          NA        1014           2
```

```
# Checking for Empty values
is.null(complete_data)
```

```
## [1] FALSE
```

## 8.Check number of unique values for each column to find out which column we can convert to factors.

```
apply(complete_data,2,function(x) length(unique(x)))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##        1309           3           3        1307           2          99
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           7           8         929         282         187           4
```

```
# sapply() function takes list, vector or data frame as input and gives outpu
t in vector or matrix.
# It is useful for operations on list objects and returns a list object of sa
me length of original set.

sapply(complete_data, function(x) length(unique(x)))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##        1309           3           3        1307           2          99
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           7           8         929         282         187           4
```

## 9.Remove Cabin as it has very high missing values, passengerId, Ticket and Name are not required.

```
# To remove a column from an R data frame

refined_data <- subset (complete_data, select = -c(Cabin, PassengerId, Ticket
, Name))
head(refined_data)
```

```
##   Survived Pclass    Sex Age SibSp Parch    Fare Embarked
## 1        0      3   male  22     1     0  7.2500        S
## 2        1      1 female  38     1     0 71.2833        C
## 3        1      3 female  26     0     0  7.9250        S
## 4        1      1 female  35     1     0 53.1000        S
## 5        0      3   male  35     0     0  8.0500        S
## 6        0      3   male  NA     0     0  8.4583        Q
```

## 10.Convert “Survived”,“Pclass”,“Sex”,“Embarked” to factors

```
refined_data$Survived<-as.factor(refined_data$Survived)
refined_data$Pclass<-as.factor(refined_data$Pclass)
refined_data$Sex<-as.factor(refined_data$Sex)
refined_data$Embarked<-as.factor(refined_data$Embarked)

str(refined_data)
```

```
## 'data.frame':    1309 obs. of  8 variables:
##  $ 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 ...
##  $ 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 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked: Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

```
summary(refined_data)
```

```
##  Survived   Pclass      Sex           Age            SibSp
##  0   :549   1:323   female:466   Min.   : 0.17   Min.   :0.0000
##  1   :342   2:277   male  :843   1st Qu.:21.00   1st Qu.:0.0000
##  NA's:418   3:709                Median :28.00   Median :0.0000
##                                  Mean   :29.88   Mean   :0.4989
##                                  3rd Qu.:39.00   3rd Qu.:1.0000
##                                  Max.   :80.00   Max.   :8.0000
##                                  NA's   :263
##      Parch            Fare          Embarked
##  Min.   :0.000   Min.   :  0.000    :  2
##  1st Qu.:0.000   1st Qu.:  7.896   C:270
##  Median :0.000   Median : 14.454   Q:123
##  Mean   :0.385   Mean   : 33.295   S:914
##  3rd Qu.:0.000   3rd Qu.: 31.275
```

```
##  Max.   :9.000   Max.   :512.329
##                  NA's   :1
```

## 11.Splitting training and test data.

```
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
df<-refined_data%>%
  filter(!is.na(Survived))
```

```
summary(df)
```

```
##  Survived Pclass      Sex           Age            SibSp           Parch
##  0:549    1:216   female:314   Min.   : 0.42   Min.   :0.000   Min.   :0.0
000
##  1:342    2:184   male  :577   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0
000
##           3:491                Median :28.00   Median :0.000   Median :0.0
000
##                                Mean   :29.70   Mean   :0.523   Mean   :0.3
816
##                                3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0
000
##                                Max.   :80.00   Max.   :8.000   Max.   :6.0
000
##                                NA's   :177
##       Fare        Embarked
##  Min.   :  0.00    :  2
##  1st Qu.:  7.91   C:168
##  Median : 14.45   Q: 77
##  Mean   : 32.20   S:644
##  3rd Qu.: 31.00
##  Max.   :512.33
##
```

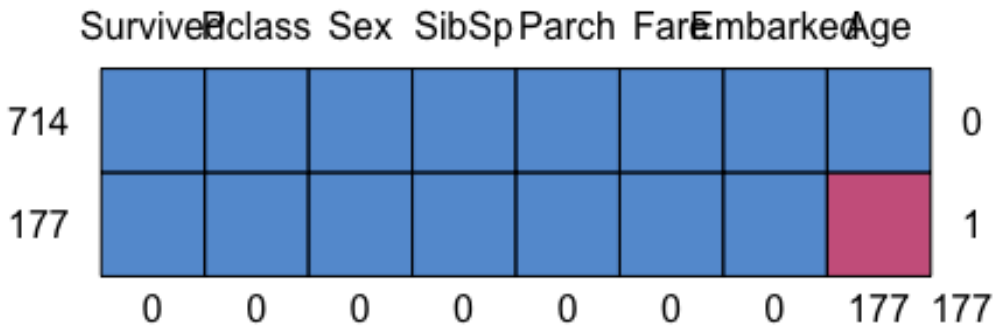
```
library("mice")
```

```
##
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
md.pattern(df)
```




```
##     Survived Pclass Sex SibSp Parch Fare Embarked Age
## 714        1      1   1     1     1    1        1   1   0
## 177        1      1   1     1     1    1        1   0   1
##            0      0   0     0     0    0        0 177 177
```

```
imputed_data<-mice(df,method = 'pmm',seed=50)
```

```
##
##  iter imp variable
##   1   1  Age
##   1   2  Age
##   1   3  Age
##   1   4  Age
##   1   5  Age
##   2   1  Age
##   2   2  Age
##   2   3  Age
##   2   4  Age
##   2   5  Age
##   3   1  Age
##   3   2  Age
##   3   3  Age
##   3   4  Age
##   3   5  Age
##   4   1  Age
##   4   2  Age
##   4   3  Age
##   4   4  Age
##   4   5  Age
##   5   1  Age
##   5   2  Age
##   5   3  Age
##   5   4  Age
##   5   5  Age
```

```
summary(imputed_data)
```

```
## Class: mids
## Number of multiple imputations:  5
## Imputation methods:
## Survived   Pclass      Sex      Age    SibSp    Parch     Fare Embarked
##       ""       ""       ""    "pmm"       ""       ""       ""       ""
## PredictorMatrix:
##          Survived Pclass Sex Age SibSp Parch Fare Embarked
## Survived        0      1   1   1     1     1    1        1
## Pclass          1      0   1   1     1     1    1        1
## Sex             1      1   0   1     1     1    1        1
## Age             1      1   1   0     1     1    1        1
## SibSp           1      1   1   1     0     1    1        1
## Parch           1      1   1   1     1     0    1        1
```

```
imputed_final<-complete(imputed_data)
summary(imputed_final)
```

```
##  Survived Pclass      Sex           Age            SibSp           Parch
##  0:549    1:216   female:314   Min.   : 0.42   Min.   :0.000   Min.   :0.0
000
##  1:342    2:184   male  :577   1st Qu.:20.00   1st Qu.:0.000   1st Qu.:0.0
000
##           3:491                Median :28.00   Median :0.000   Median :0.0
000
##                                Mean   :29.42   Mean   :0.523   Mean   :0.3
816
##                                3rd Qu.:39.00   3rd Qu.:1.000   3rd Qu.:0.0
000
##                                Max.   :80.00   Max.   :8.000   Max.   :6.0
000
##       Fare        Embarked
##  Min.   :  0.00    :  2
##  1st Qu.:  7.91   C:168
##  Median : 14.45   Q: 77
##  Mean   : 32.20   S:644
##  3rd Qu.: 31.00
##  Max.   :512.33
```

```
set.seed(42)

train_pts<-sample(1:nrow(imputed_final),0.75*nrow(imputed_final))

train_dataset <-imputed_final[train_pts,]
test_dataset <-imputed_final[-train_pts,]
```

```
summary(train_dataset)
```

```
##  Survived Pclass      Sex           Age            SibSp
##  0:413    1:161   female:240   Min.   : 0.42   Min.   :0.0000
##  1:255    2:138   male  :428   1st Qu.:20.38   1st Qu.:0.0000
##           3:369                Median :28.00   Median :0.0000
##                                Mean   :29.38   Mean   :0.5344
##                                3rd Qu.:38.00   3rd Qu.:1.0000
##                                Max.   :80.00   Max.   :8.0000
##      Parch            Fare          Embarked
##  Min.   :0.0000   Min.   :  0.000    :  2
##  1st Qu.:0.0000   1st Qu.:  7.896   C:127
##  Median :0.0000   Median : 14.456   Q: 56
##  Mean   :0.3683   Mean   : 32.188   S:483
##  3rd Qu.:0.0000   3rd Qu.: 30.500
##  Max.   :5.0000   Max.   :512.329
```

```
Xtrain = subset(train_dataset,select=-c(Survived))
Ytrain = train_dataset$Survived

library(caret)
```

```
## Loading required package: lattice
## Loading required package: ggplot2
train_final <- upSample (subset(train_dataset,
                                select=-c(Survived)),
                         train_dataset$Survived)
summary(train_final)
```

```
##  Pclass      Sex           Age            SibSp           Parch
##  1:222   female:351   Min.   : 0.42   Min.   :0.0000   Min.   :0.0000
##  2:177   male  :475   1st Qu.:20.00   1st Qu.:0.0000   1st Qu.:0.0000
##  3:427                Median :28.00   Median :0.0000   Median :0.0000
##                       Mean   :29.15   Mean   :0.5254   Mean   :0.3801
##                       3rd Qu.:38.00   3rd Qu.:1.0000   3rd Qu.:0.7500
##                       Max.   :80.00   Max.   :8.0000   Max.   :5.0000
##       Fare          Embarked Class
##  Min.   :  0.000    :  4    0:413
##  1st Qu.:  7.925   C:167    1:413
##  Median : 15.646   Q: 71
##  Mean   : 37.006   S:584
##  3rd Qu.: 32.455
##  Max.   :512.329
```

## 12.Create a model.

```
# The basic syntax for glm() function in logistic regression is -
#           glm(formula, data,family)

# formula is the symbol presenting the relationship between the variables.
# data is the data set giving the values of these variables.
# family is R object to specify the details of the model. It's value is binom
ial for logistic regression.

LogisticModel <- glm(Class ~., train_final, family = binomial(link='logit'))
LogisticModel
```

```
##
## Call:  glm(formula = Class ~ ., family = binomial(link = "logit"), data =
train_final)
##
## Coefficients:
## (Intercept)      Pclass2      Pclass3      Sexmale          Age        Sib
Sp
##   16.696750    -0.969256    -1.996026    -2.720884    -0.048514    -0.4481
74
##       Parch         Fare    EmbarkedC    EmbarkedQ    EmbarkedS
##    0.016519     0.004891   -12.220578   -12.029345   -12.406584
```

```
##
## Degrees of Freedom: 825 Total (i.e. Null);  815 Residual
## Null Deviance:       1145
## Residual Deviance: 736.2     AIC: 758.2
```

## 13.Visualize the model summary.

```
# Model Summary
summary(LogisticModel)

##
## Call:
## glm(formula = Class ~ ., family = binomial(link = "logit"), data = train_f
inal)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.01000  -0.67871  -0.07533   0.60749   2.29966
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  16.696750 427.382580   0.039  0.96884
## Pclass2      -0.969256   0.324794  -2.984  0.00284 **
## Pclass3      -1.996026   0.326350  -6.116 9.58e-10 ***
## Sexmale      -2.720884   0.205386 -13.248  < 2e-16 ***
## Age          -0.048514   0.008085  -6.000 1.97e-09 ***
## SibSp        -0.448174   0.107650  -4.163 3.14e-05 ***
## Parch         0.016519   0.136339   0.121  0.90356
## Fare          0.004891   0.002878   1.699  0.08930 .
## EmbarkedC   -12.220578 427.382386  -0.029  0.97719
## EmbarkedQ   -12.029345 427.382470  -0.028  0.97755
## EmbarkedS   -12.406584 427.382367  -0.029  0.97684
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1145.08  on 825  degrees of freedom
## Residual deviance:  736.23  on 815  degrees of freedom
## AIC: 758.23
##
## Number of Fisher Scoring iterations: 13
```

## 14.Analyse the test of deviance using anova()

```
#Note: anova(model, test="Chisq")

# Using anova() to analyze the table of devaiance

anova(LogisticModel, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Class
##
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                       825    1145.08
## Pclass    2   88.802       823    1056.28 < 2.2e-16 ***
## Sex       1  261.290       822     794.99 < 2.2e-16 ***
## Age       1   30.797       821     764.19 2.864e-08 ***
## SibSp     1   20.705       820     743.48 5.357e-06 ***
## Parch     1    0.137       819     743.35   0.71124
## Fare      1    4.840       818     738.51   0.02781 *
## Embarked  3    2.280       815     736.23   0.51637
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## 15.Compute confusion matrix and ROC curve.

```
## Predicting Test Data

data <- predict( LogisticModel, newdata = test_dataset, type="response")
pred_num <- ifelse(data > 0.5, 1, 0)

pred_data <- factor(pred_num, levels = c(0, 1))
actual_data <- test_dataset$Survived

confusionMatrix( data = actual_data, reference = actual_data)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   0   1
##          0 136   0
##          1   0  87
##
##                Accuracy : 1
##                  95% CI : (0.9836, 1)
##     No Information Rate : 0.6099
##     P-Value [Acc > NIR] : < 2.2e-16
```

```
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.6099
##          Detection Rate : 0.6099
##    Detection Prevalence : 0.6099
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```

```
library(ROCR)
```

```
prediction_obj <- prediction(as.numeric(pred_data), as.numeric(actual_data))
final_set <- performance(prediction_obj, "tpr", "fpr")
plot(final_set, colorize=TRUE)
```