Titanic_Logit

IoA

May 11, 2018

Titanic Dataset

```
train <- read.csv("train.csv",header = T, na.strings = "")
test <- read.csv("test.csv", header = T, na.strings = "")
str(train)
```

```
## '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/ 147 levels "A10","A14","A16",..: NA 82 NA 56 NA NA 130 NA NA NA ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
```

```
print("-----------------------------------------------------------------------------------------------------")
```

```
## [1] "-----------------------------------------------------------------------------------------------------"
```

```
str(test)
```

```
## '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 5
##  $ 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/ 76 levels "A11","A18","A21",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
```

Target Variable - Survived

Check for Missing Values

Train

```
colSums(is.na(train))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         177
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0         687           2
```

177 missing values in Age, 687 missing values in Cabin and 2 missing values in Embarked

```
colSums(is.na(test))
```

```
## PassengerId      Pclass        Name         Sex         Age       SibSp
##           0           0           0           0          86           0
##       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           1         327           0
```

86 missing values in Age, 1 missing value in Fare and 327 in cabin

Combine train and test data together. (instead of imputing missing values separately its easy to first combine and then do any operation. You can split back the combined data)

```
#data = rbind(train,test)
```

Error : Because the test dataset doesnot contain survived column. Inorder to merge/combine two data frames the number of colmuns must be same

So for this purpose add a column named survived in our test data.

```
test$Survived <- 1 # Assuming 1
```

Now you can merge the data without any problem

```
data = rbind(train,test)
str(data)
```

```
## 'data.frame':    1309 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : num  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/ 1307 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 58
##  $ 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/ 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/ 186 levels "A10","A14","A16",..: NA 82 NA 56 NA NA 130 NA NA NA ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
```

Change the data types of the columns - Integer to Categorial and vice versa Survived is a categorial column(1 - Yes, 0 - No) therefore it has to be in factor, Similarly Pclass and Sex

```
data$PassengerId <- as.factor(data$PassengerId)
data$Survived <- as.factor(data$Survived)
data$Pclass <- as.factor(data$Pclass)
str(data)
```

```
## 'data.frame':    1309 obs. of  12 variables:
##  $ PassengerId: Factor w/ 1309 levels "1","2","3","4",..: 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       : Factor w/ 1307 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 58
##  $ 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/ 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/ 186 levels "A10","A14","A16",..: NA 82 NA 56 NA NA 130 NA NA NA ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
```

```
colSums(is.na(data))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         263
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           1        1014           2
```

Visualise Missing data

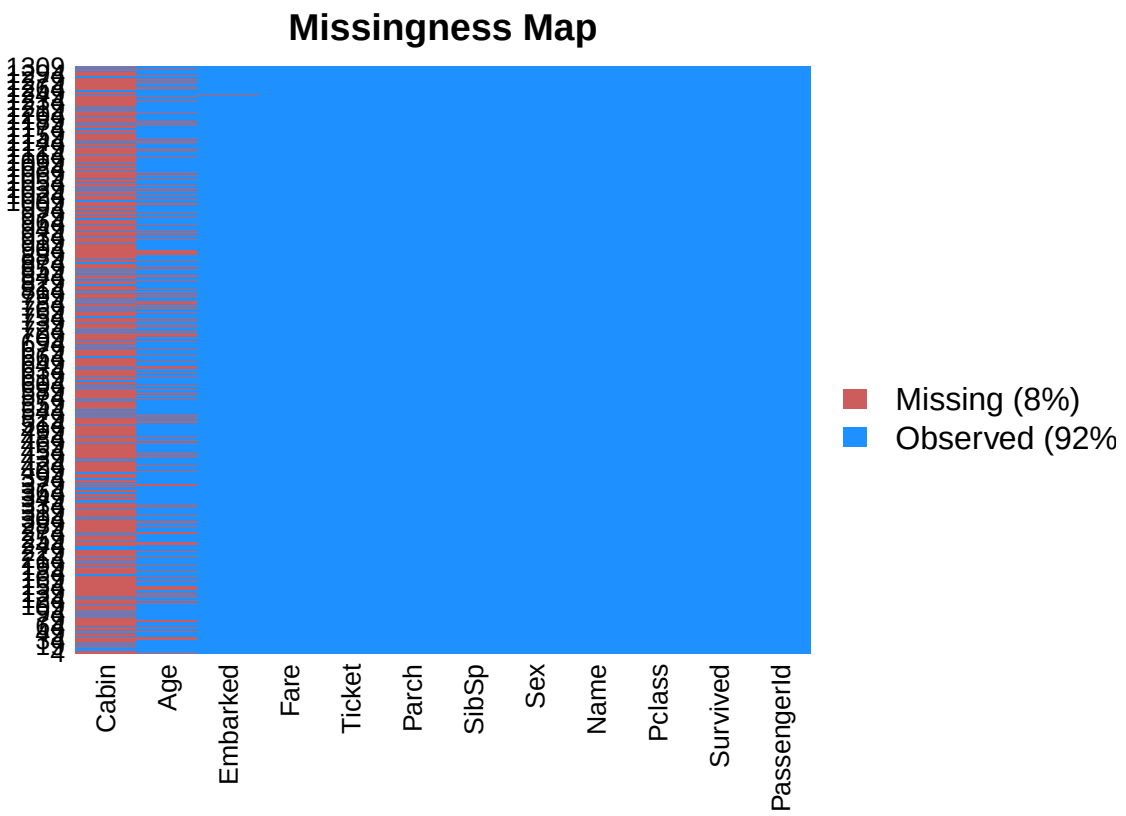
```
library(Amelia)
```

```
## Warning: package 'Amelia' was built under R version 3.4.4
```

```
## Loading required package: Rcpp
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.5, built: 2018-05-07)
## ## Copyright (C) 2005-2018 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
missmap(data)
```



Missing value imputation

We dont know whether age is an important parameter in making our prediction. For now lets replace NA's with mean values of age

```
data$Age[is.na(data$Age)] <- mean(data$Age[!is.na(data$Age)])
colSums(is.na(data))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0           0
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           1        1014           2
```

1 missing values in fare is left

```
which(is.na(data$Fare))
```

```
## [1] 1044
```

Impute fare

```
data_fare = subset(data, Sex == "male" & Embarked == "S" & Pclass == 3 & SibSp == 0 & Parch == 0)
View(data_fare)
```

Inorder to calculate fare price you need to know about the ticket class, gender, the place of boarding, the number of siblings/Spouse and parents/childrens with you.

```
sort(table(data_fare$Fare[!is.na(data_fare$Fare)]))
```

```
##
##  6.2375    6.45  6.4958       7  7.0542  7.1417  7.3125  7.5208   7.575
##       1       1       1       1       1       1       1       1       1
##  7.5792     7.8    7.85   7.875  7.8792  8.1125  8.1583     8.3  8.3625
##       1       1       1       1       1       1       1       1       1
##  8.4333  8.6542  8.7125  9.2167   9.325    9.35  9.4833  9.8458 10.1708
##       1       1       1       1       1       1       1       1       1
##   24.15   6.975  7.4958  7.8875       9    15.1    16.1    7.75   9.225
##       1       2       2       2       2       2       2       3       3
##    14.5  22.525       0   7.125    7.65    7.55 56.4958    7.05  7.7958
##       3       3       4       4       4       6       8       9      10
##     9.5  7.8542   7.925  8.6625    7.25   7.775  7.8958    8.05
##      12      13      13      14      15      18      42      53
```

Now you can see that most used fare price is 8.05 (by 53 passengers)

Now replace the NA in fare column by 8.05

```
data$Fare[is.na(data$Fare)] <- 8.05
```

```
colSums(is.na(data))
```

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

Similary you do for 2 missing values in Embarked column.

```
table(data$Embarked[!is.na(data$Embarked)])
```

```
##
##   C   Q   S
## 270 123 914
```

S - 'Southampton' is way higher than other places of boarding

Lets replace the 2 missing values in Embarked by 'S'

```
data$Embarked[is.na(data$Embarked)] <- 'S'
colSums(is.na(data))
```

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

Cabin has a lot of missing values and therefore lets drop the column.

Separating data backto train and test

```
train <- data[1:891,]
colSums(is.na(train))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0           0
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0         687           0
```

```
test <- data[892:1309,]
colSums(is.na(test))
```

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

```
test$Survived <- NULL
```

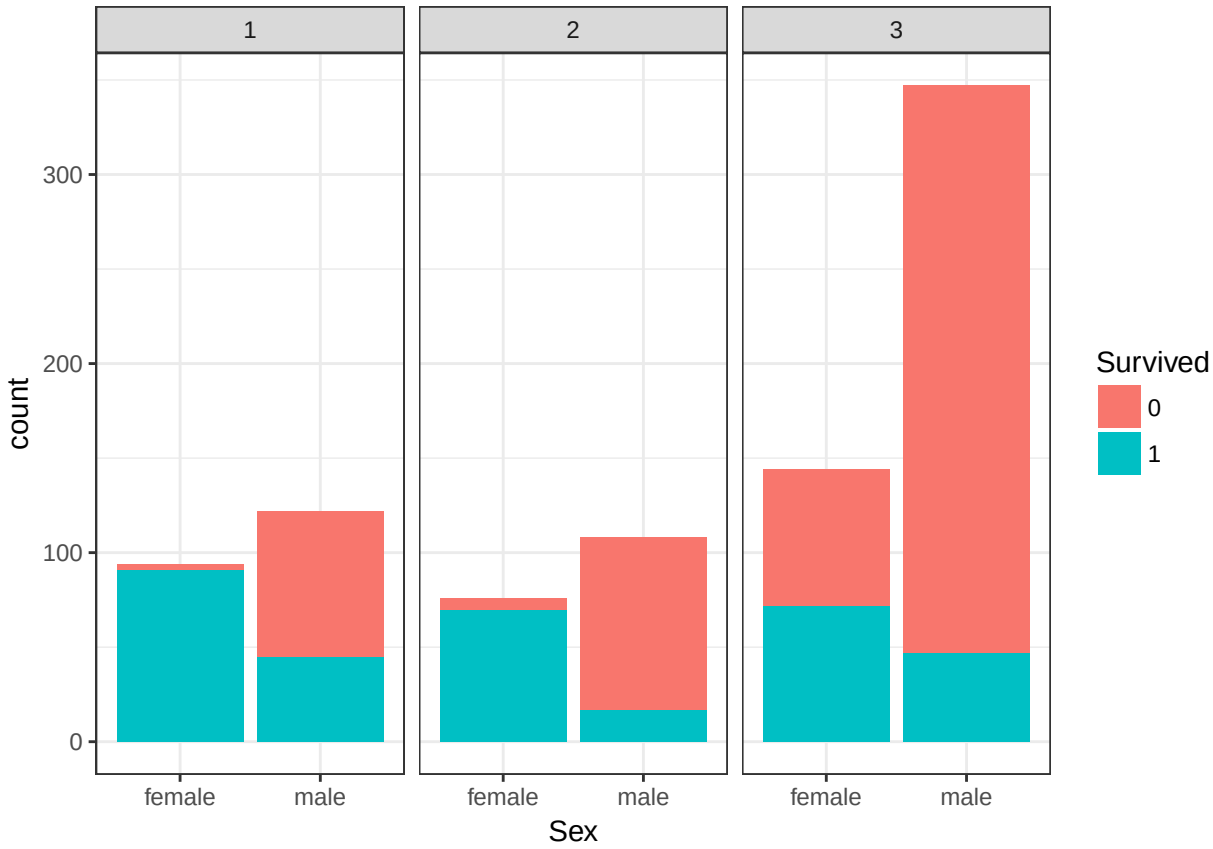
To check the accuracy of a model we need to split train data into sub train and sub test. (because the actual test data doesn't have infomation about Survived)

Data Visualization

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
ggplot(train, aes(x = Sex, fill = Survived )) + facet_wrap(~Pclass) + geom_bar() + theme_bw()
```



Split

```
library(caTools)
set.seed(101)
sample = sample.split(train$Survived, SplitRatio = .80)
sub_train = subset(train, sample == TRUE)
sub_test  = subset(train, sample == FALSE)
```

Logistic Regression Model - Logit

```
model <- glm(Survived~Age+Sex+Embarked+Pclass+Fare, data = sub_train, family = "binomial")
summary(model)
```

```
##
## Call:
## glm(formula = Survived ~ Age + Sex + Embarked + Pclass + Fare,
##     family = "binomial", data = sub_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5757  -0.6368  -0.3901   0.6584   2.4831
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.699684   0.502764   7.359 1.86e-13 ***
## Age         -0.032362   0.008188  -3.952 7.73e-05 ***
## Sexmale     -2.583151   0.210393 -12.278  < 2e-16 ***
## EmbarkedQ   -0.007631   0.412237  -0.019   0.9852
## EmbarkedS   -0.576061   0.259765  -2.218   0.0266 *
## Pclass2     -0.846553   0.328734  -2.575   0.0100 *
## Pclass3     -2.131931   0.323370  -6.593 4.31e-11 ***
## Fare         0.001460   0.002451   0.596   0.5514
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 949.90  on 712  degrees of freedom
## Residual deviance: 637.08  on 705  degrees of freedom
## AIC: 653.08
##
## Number of Fisher Scoring iterations: 5
```

Embarked and Fare doesn't seem to be that significant, Lets remove

```
model <- glm(Survived~Age+Sex+Pclass, data = sub_train, family = "binomial")
summary(model)
```

```
##
## Call:
## glm(formula = Survived ~ Age + Sex + Pclass, family = "binomial",
##     data = sub_train)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6364  -0.6521  -0.4238   0.6483   2.4174
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.510072   0.404544   8.677  < 2e-16 ***
## Age         -0.033177   0.008086  -4.103 4.08e-05 ***
## Sexmale     -2.616834   0.207731 -12.597  < 2e-16 ***
## Pclass2     -1.103989   0.292634  -3.773 0.000162 ***
## Pclass3     -2.266809   0.270445  -8.382  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 949.90  on 712  degrees of freedom
## Residual deviance: 644.34  on 708  degrees of freedom
## AIC: 654.34
##
## Number of Fisher Scoring iterations: 5
```

Prediction

```
prediction <- predict(model, newdata = sub_test, type = "response")
head(prediction)
```

```
##         4        21        22        24        28        29
## 0.9128455 0.2023069 0.2077138 0.4910730 0.5653439 0.5626448
```

It returns probabilites, so set a cut off of 0.5

```
prediction = ifelse(prediction > 0.5,1,0)
head(prediction)
```

```
##  4 21 22 24 28 29
##  1  0  0  0  1  1
```

Confusion Matrix

```
cm <- table(Actual = sub_test$Survived, Predicted = prediction)
cm
```

```
##       Predicted
## Actual  0  1
##      0 95 15
##      1 23 45
```

Accuracy

```
print(sum(diag(cm))/sum(cm))
```

```
## [1] 0.7865169
```

78.6 percent accuracy.

Model with full train data

```
model <- glm(Survived~Age+Sex+Pclass, data = train, family = "binomial")
prediction <- predict(model, newdata = test, type = "response")
prediction = ifelse(prediction > 0.5,1,0)
test$Survived = prediction
```