CMP 713 – Data Mining

Spring 2020

CMP 713 - Data mınıng  
the TITANIC DATASET

Erol ÖZKAN

**CONTENT**

[1 INTRODUCTION 2](#_Toc40563419)

[2 DATASET 2](#_Toc40563420)

[3 EXPLORATORY DATA ANALYSIS 2](#_Toc40563421)

[3.1 Load Libraries 2](#_Toc40563422)

[3.2 Read Data 2](#_Toc40563423)

[3.3 Check Data 2](#_Toc40563424)

[3.4 Describe Variables 2](#_Toc40563425)

[3.5 Investigate the Data 3](#_Toc40563426)

[3.6 Explore the Data 3](#_Toc40563427)

[3.6.1 Survived 3](#_Toc40563428)

[3.6.2 Sex 3](#_Toc40563429)

[3.6.3 PClass 4](#_Toc40563430)

[4 FEATURE ENGINEERING 1 5](#_Toc40563431)

[4.1 Title 5](#_Toc40563432)

[4.2 Family Size 5](#_Toc40563433)

[4.3 Cabin 6](#_Toc40563434)

[5 MISSING VALUE IMPUTATION 7](#_Toc40563435)

[5.1 Embarked 7](#_Toc40563436)

[5.2 Fare 7](#_Toc40563437)

[5.3 Age 8](#_Toc40563438)

[6 FEATURE ENGINEERING 2 9](#_Toc40563439)

[6.1 Child 9](#_Toc40563440)

[6.2 Mother 9](#_Toc40563441)

[7 PREDICTION 10](#_Toc40563442)

[7.1 Split Dataset 10](#_Toc40563443)

[7.2 RandomForest Model 10](#_Toc40563444)

[7.2.1 Visualize the Importance 11](#_Toc40563445)

[7.3 SVM Model 11](#_Toc40563446)

[7.4 GBM Model 12](#_Toc40563447)

[8 RESULTS 12](#_Toc40563448)

[9 CONCLUSION 12](#_Toc40563449)

# INTRODUCTION

In this work, the aim is to predict what kind of people are likely to survive in the Titanic dataset. We use data mining methods to predict which passengers survived in the dataset. We do explanatory data analysis and create some illustrative data visualizations to better understand this dataset. Then, we do some feature engineering and missing value imputation to create more meaningful variables. Finally, we build three different models to predict the Survival variable.

The remainder of the document is organized as follows. In Section 2, we explain the Titanic dataset. In Section 3, we explain the EDA. Missing value imputation is explained in Section 5. In Section 4 and Section 6 we show feature engineering steps. Predictive models are explained in the Section 7. Results are shown in Section 8. Finally, we conclude our work in Section 8.

# DATASET

Titanic dataset is used for this project. It contains the survival information of Titanic ship which sank after colliding with an iceberg on April 15, 1912. It killed 1502 out of 2224 including both all the passengers and its crew. One of the reasons this tragedy led to such loss of life was that there were not enough lifeboats for everyone on the ship. This tragedy led to better safety regulations for ships.

# EXPLORATORY DATA ANALYSIS

## Load Libraries

|  |  |
| --- | --- |
| library('ggplot2')  library('ggthemes')  library('scales')  library('dplyr')  library('mice')  library('randomForest')  library('gridExtra')  library('Hmisc') | library('knitr')  library('ggplot2')  library('dplyr')  library('caret')  library('randomForest')  library('gridExtra')  library('ROCR')  library('corrplot') |

## Read Data

train <- read.csv('dataset/train.csv', stringsAsFactors = F, na.strings = c("NA", "")) # Read train dataset.

test <- read.csv('dataset/test.csv', stringsAsFactors = F, na.strings = c("NA", "")) # Read test dataset.

full <- bind\_rows(train, test) # Concatinate train and test dataset

## Check Data

str(full)

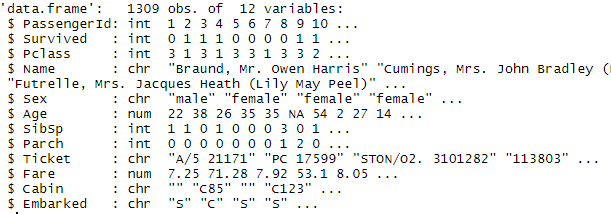


Figure 1. The Output of str(full) command.

There are 1309 observations in total. The training dataset has 891, the test dataset has 418 observations. Note that “Survived” variable is missing from the testing dataset.

## Describe Variables

Description of all the variables is shown in the Table 1.

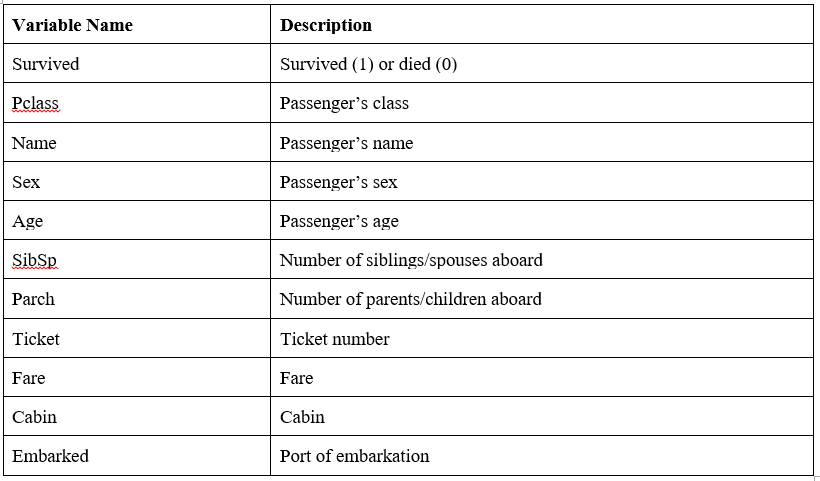


Figure 2. Description of the Dataset

## Investigate the Data

Let’s investigate which variables contain missing values.

sapply(full, function(x) {sum(is.na(x))})

The output of this command is shown in the Figure 3.



Figure 3. The Output of Compleness Function

The 418 observations is missing the Survived column in the test dataset. Cabin is sparsely populated. Age is missing in some observations. In addition, Embarked is missing two values and Fare one value.

## Explore the Data

Let’s convert Sex, Pclass, Survived and Embarked variables into factors.

full$Sex <- as.factor(full$Sex)

full$Survived <- as.factor(full$Survived)

full$Pclass <- as.factor(full$Pclass)

full$Embarked <- as.factor(full$Embarked)

### Survived

Survived variable describes the survival of people survived.

ggplot(full[!is.na(full$Survived),], aes(x = Survived, fill = Survived)) +

geom\_bar(stat='count') +

labs(x = 'How many people died and survived on the Titanic?') +

geom\_label(stat='count',aes(label=..count..), size=7)

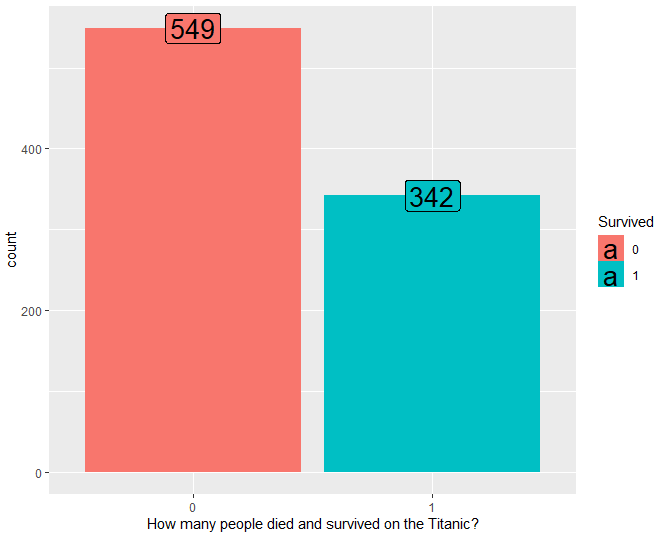


Figure 4. Survival of People

It is clear that 549 people died and 342 people survived in the training data.

### Sex

Let’s visualize the Sex variable.

p1 <- ggplot(full, aes(x = Sex, fill = Sex)) +

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

labs(x = 'All data') +

geom\_label(stat='count', aes(label=..count..)) +

scale\_fill\_manual("legend", values = c("female" = "pink", "male" = "green"))

p2 <- ggplot(full[!is.na(full$Survived),], aes(x = Sex, fill = Survived)) +

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

labs(x = 'Training data only') +

geom\_label(stat='count', aes(label=..count..))

grid.arrange(p1,p2, nrow=1)

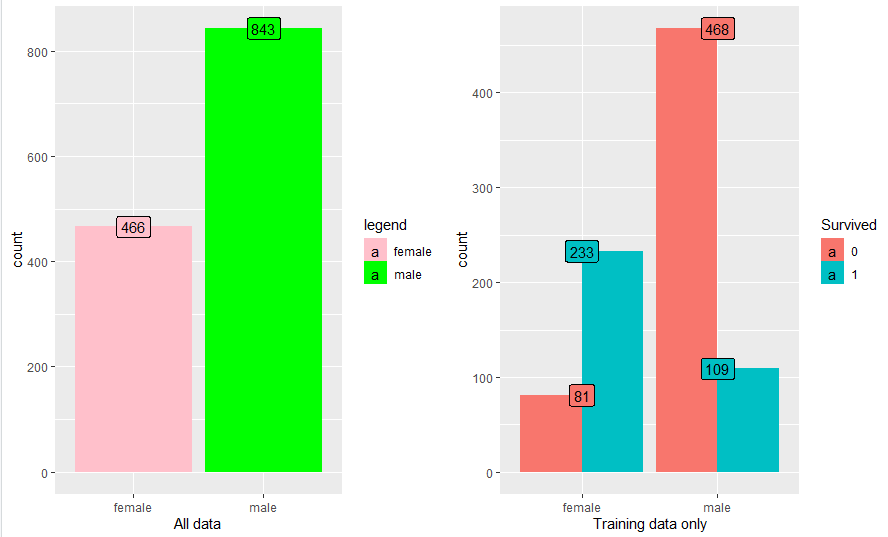


Figure 5. percentage of Male/Female with Survived Information

First plot shows that there are 466 female, 843 male passengers on the training and testing data. We can infer from second plot that 81.1% of the men died, only 25.8% of the women died in the training dataset.

### PClass

Let’s visualize the PClass variable.

p3 <- ggplot(full, aes(x = Pclass, fill = Pclass)) +

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

labs(x = 'Pclass, All data') + geom\_label(stat='count', aes(label=..count..)) +

theme(legend.position="none") + theme\_grey()

p4 <- ggplot(full[!is.na(full$Survived),], aes(x = Pclass, fill = Survived)) +

geom\_bar(stat='count', position='dodge') + labs(x = 'Training data only') +

theme(legend.position="none") + theme\_grey()

p5 <- ggplot(full[!is.na(full$Survived),], aes(x = Pclass, fill = Survived)) +

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

labs(x = 'Training data only', y= "Count") + facet\_grid(.~Sex) +

theme(legend.position="none") + theme\_grey()

p6 <- ggplot(full[!is.na(full$Survived),], aes(x = Pclass, fill = Survived)) +

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

labs(x = 'Training data only', y= "Percent") + facet\_grid(.~Sex) +

theme(legend.position="none") + theme\_grey()

grid.arrange(p3, p4, p5, p6, ncol=2)

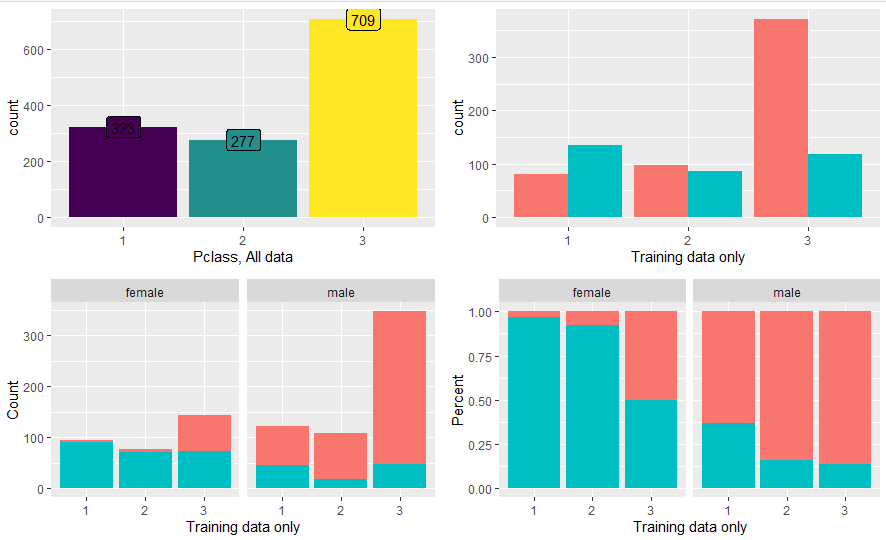


Figure 6. Passenger Class Information

First plot states that most people was in the 3rd class. According to second plot, majority of 1st and 2nd class passengers survived, while majority of 3rd class died. It is noticeable in 3rd and 4th charts that almost all women in 1st and 2nd class survived.

# FEATURE ENGINEERING 1

## Title

We can break the Name variable into additional meaningful variables such as; title, name and surname.

full$Title <- gsub('(.\*, )|(\\..\*)', '', full$Name) # Create Title variable

table(full$Sex, full$Title) # Show Title counts



Figure 7. Title of Passengers

Let’s replace low title counts.

rare\_title <- c('Dona', 'Lady', 'the Countess','Capt', 'Col', 'Don', 'Dr', 'Major', 'Rev', 'Sir', 'Jonkheer')

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

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

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

full$Title[full$Title %in% rare\_title] <- 'Rare Title'

table(full$Sex, full$Title)



Figure 8. Title of Passengers after Replacement

Let’s visualize the Title variable.

ggplot(full[!is.na(full$Survived),], aes(x = Title, fill = Survived)) +

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

labs(x = 'Title') +theme\_grey()

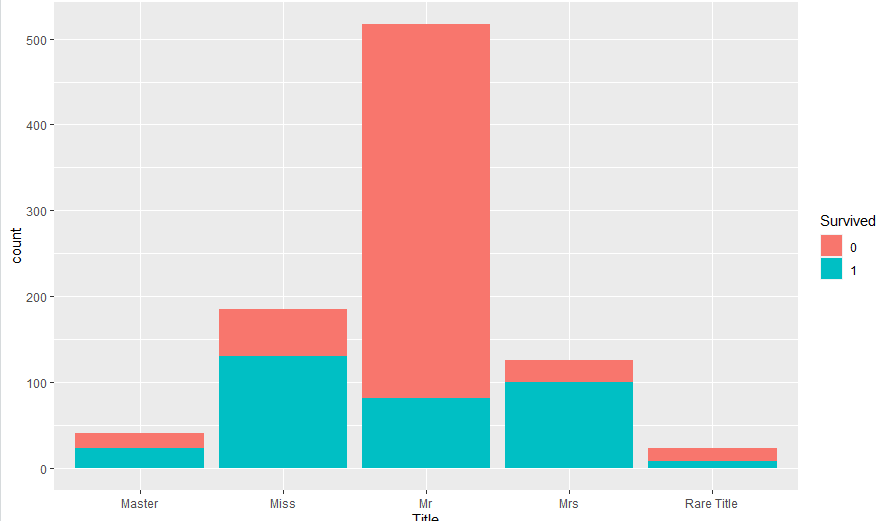


Figure 9. Visualization of Title Variable

It is clear that Title is highly correlated with Survival information.

## Family Size

Family size variable is based on number of siblings/spouse(s) and children/parents. We need to add 1 to include the passenger himself/herself.

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

To better understand the family size variable, let’s use ggplot to visualize the family size and the survival.

ggplot(full[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')

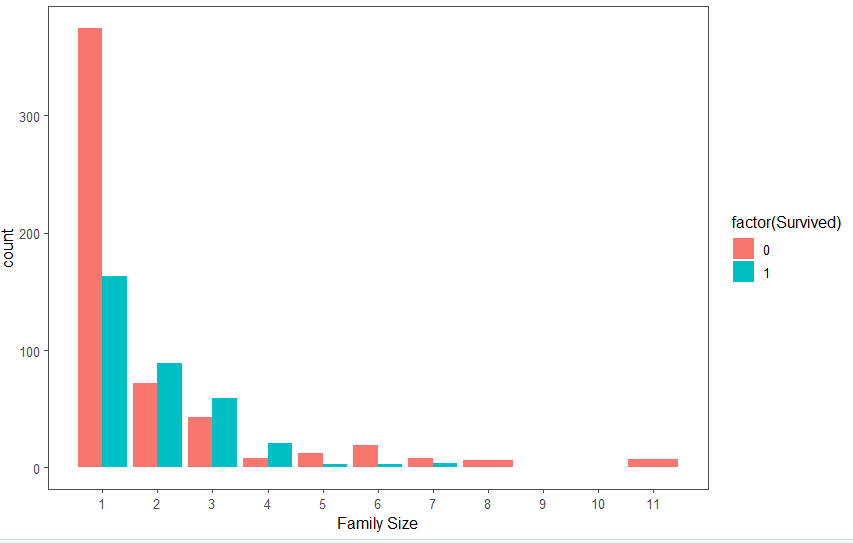


Figure 10. Family Size and Survival

It is clear that there is a survival penalty to single passengers and family sizes above 4. We can further make Family variable a categorical variable.

full$FsizeD[full$Fsize == 1] <- 'singleton'

full$FsizeD[full$Fsize < 5 & full$Fsize > 1] <- 'small'

full$FsizeD[full$Fsize > 4] <- 'large'

mosaicplot(table(full$FsizeD, full$Survived), main='Family Size - Survival', shade=TRUE)

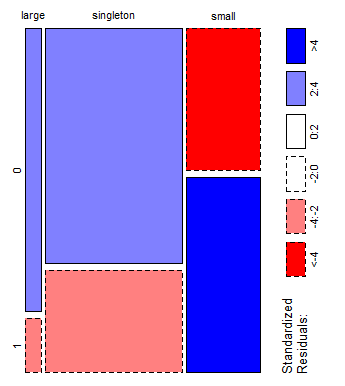


Figure 11. Family Size and Survival

This plot shows that the survival rate of large families is worse than singletons and small families.

## Cabin

Let’s take a look into the Cabin variable.

full$Cabin[1:28]

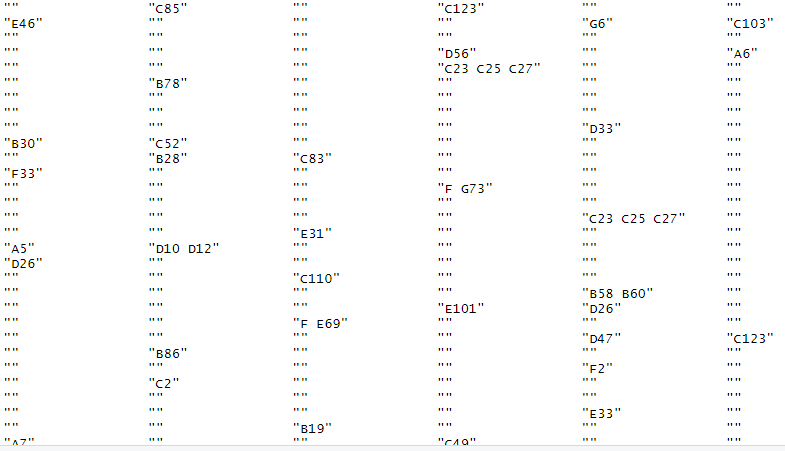


Figure 12. Cabin Variable

As you can see, there are lots of missing values. But, we can also infer Deck variable from Cabin variable.

strsplit(full$Cabin[2], NULL)[[1]]

full$Deck<-factor(sapply(full$Cabin, function(x) strsplit(x, NULL)[[1]][1]))

full$Deck[1:28]

# MISSING VALUE IMPUTATION

There are some missing values in the dataset. Since, we don’t have many observations, we are better not to delete any observations or columns with missing values. Hence, we are trying to replace missing values with a sensible values.

## Embarked

Let’s take a look at Embarked variable.

full[c(62, 830), 'Embarked']

Passengers 62 and 830 are missing Embarkment variable. We can infer their values based on passenger class and fare.

embark\_fare <- full %>%

filter(PassengerId != 62 & PassengerId != 830)

Use ggplot2 to visualize Embarked, Class and median Fare variables.

ggplot(embark\_fare, aes(x = Embarked, y = Fare, fill = factor(Pclass))) +

geom\_boxplot() +

geom\_hline(aes(yintercept=80),

colour='red', linetype='dashed', lwd=2) +

scale\_y\_continuous(labels=dollar\_format())

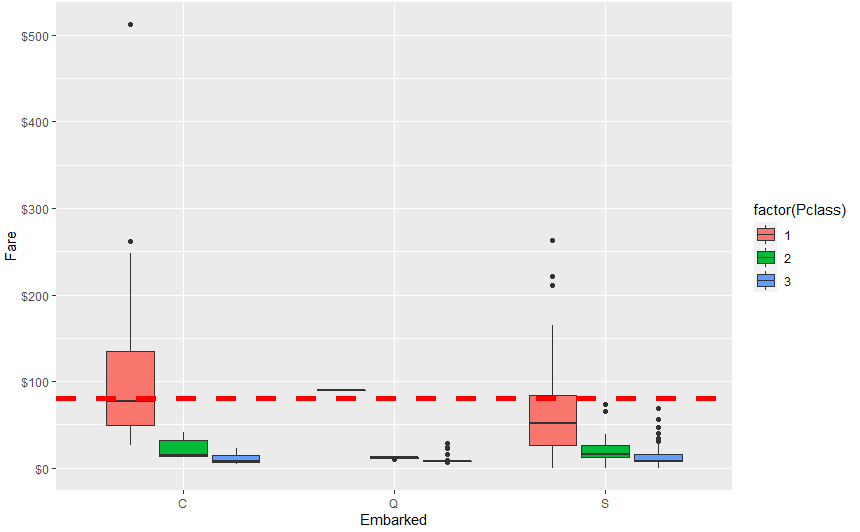


Figure 13. Visualization of *Embarked, Class and median Fare variables.*

The median fare for a first class passenger departing from Charbourg (‘C’) matches with the $80 Fare variable. Hence, we can replace the NA values with ‘C’.

full$Embarked[c(62, 830)] <- 'C'

full[c(62, 830), 'Embarked']

## Fare

Similar to Embarked variable, Fare variable on row 1044 has an NA value.

full[1044, ]

This passenger departed from Southampton (‘S’). Let’s visualize Fares among same class and embarkment.

ggplot(full[full$Pclass == '3' & full$Embarked == 'S', ],

aes(x = Fare)) +

geom\_density(fill = '#99d6ff', alpha=0.4) +

geom\_vline(aes(xintercept=median(Fare, na.rm=T)),

colour='red', linetype='dashed', lwd=1) +

scale\_x\_continuous(labels=dollar\_format())

From this visualization, it seems quite reasonable to replace the NA values with median for their class and embarkment.

full$Fare[1044] <- median(full[full$Pclass == '3' & full$Embarked == 'S', ]$Fare, na.rm = TRUE)

full[1044, ]

## Age

There are many missing Age variable in the dataset. Let’s look into the number of missing Age variables.

sum(is.na(full$Age))

There are 263 missing Age variables. We can use mice package for guessing missing Age variables.

set.seed(129) # Set a random seed

mice\_mod <- mice(full[, !names(full) %in% c('PassengerId','Name','Ticket','Cabin','Family','Surname','Survived')], method='rf')

mice\_output <- complete(mice\_mod) # Save the output

Let’s compare the results with the original distribution.

par(mfrow=c(1,2))

hist(full$Age, freq=F, main='Original Data', ylim=c(0,0.04))

hist(mice\_output$Age, freq=F, main='MICE Output Data', ylim=c(0,0.04))

Distributions looks very similar, Let’s replace the Age variables with the mice model.

full$Age <- mice\_output$Age

sum(is.na(full$Age))

# FEATURE ENGINEERING 2

We can create a couple of new age-dependent variables: Child and Mother.

* A child is someone under 18 years.
* A mother is someone who is female, older than 18, doesn’t have the ‘Miss’ title and has more than 0 children.

## Child

Let’s look at the relationship between age and survival

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

geom\_histogram() +

facet\_grid(.~Sex)

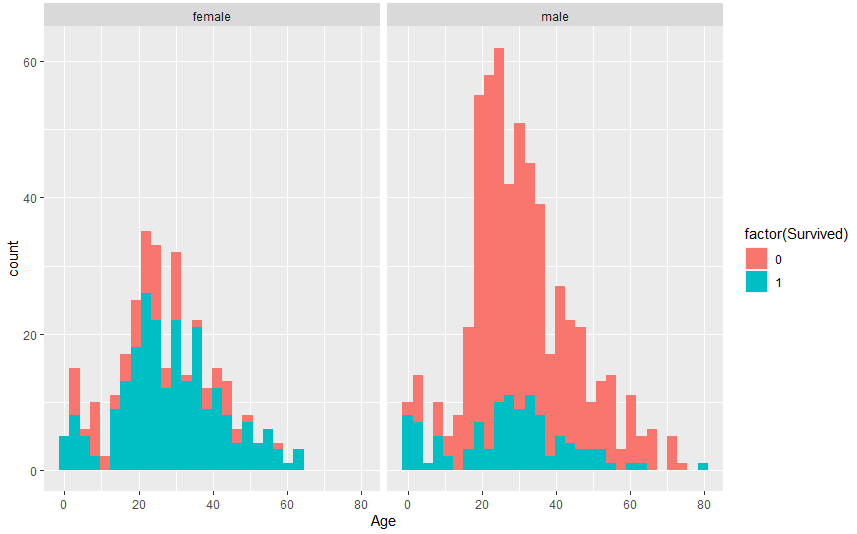


Figure 14. Survival Rate by Age and Sex

Let’s create a Child variable.

full$Child[full$Age < 18] <- 'Child'

full$Child[full$Age >= 18] <- 'Adult'

table(full$Child, full$Survived)

full$Child <- factor(full$Child)



Figure 15. The Output of Child Variable

## Mother

Similarly, let’s create a Mother variable.

full$Mother <- 'Not Mother'

full$Mother[full$Sex == 'female' & full$Parch > 0 & full$Age > 18 & full$Title != 'Miss'] <- 'Mother'

table(full$Mother, full$Survived)

full$Mother <- factor(full$Mother)



Figure 16. The Output of Mother Variable

# PREDICTION

We successfully fill the missing values and created new variables in the Titanic datasets. In this section, we will build some models to predict the Survival variable. Let’s first check the datasets again using the completeness function. Then, factorize some variables.

sapply(full, function(x) {sum(is.na(x))})

full$Sex <- as.factor(full$Sex)

full$Embarked <- as.factor(full$Embarked)

full$Title<- as.factor(full$Title)

full$FsizeD<- as.factor(full$FsizeD)

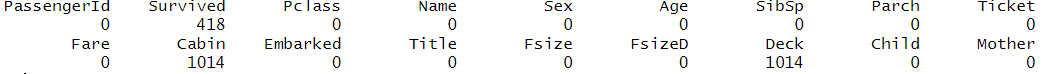


Figure 17. The Output of Complexness Function

As you can see, we have many new variables. Also, we replaced some missing values with their sensible values. Again, we need to convert some variables into factor.

## Split Dataset

Let’s split the data back into a training and test datasets.

train <- full[1:891,]

test <- full[892:1309,]

## RandomForest Model

Let’s build the RandomForest model.

set.seed(2017)

random\_forrest\_model <- randomForest(factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch +

Fare + Embarked + Title +

FsizeD + Child + Mother, data = train)

random\_forrest\_predictions <- predict(random\_forrest\_model, test)

random\_forrest\_solutions <- data.frame(PassengerID = test$PassengerId, Survived = random\_forrest\_predictions)

write.csv(random\_forrest\_solutions, file = 'output/random\_forrest\_solutions.csv', row.names = F)

The final result for RandomForest model is “**0.78947**”. Let’s visualize the error rate.

plot(random\_forrest\_model, ylim=c(0,0.36))

legend('topright', colnames(random\_forrest\_model$err.rate), col=1:3, fill=1:3)

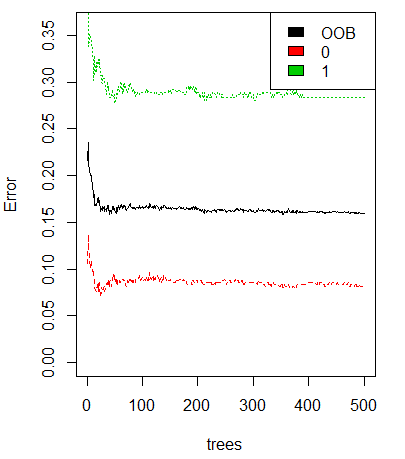


Figure 18. Visualization of the Error Rate.

The black line shows the overall error rate. The red and green lines show the error rate of ‘died’ and ‘survived’.

### Visualize the Importance

Let’s visualize the importance of variables.

random\_forrest\_importance <- varImp(random\_forrest\_model, scale = FALSE)

importance\_scores <- data.frame(Variables = row.names(random\_forrest\_importance), MeanDecreaseGini = random\_forrest\_importance$Overall)

ggplot(importance\_scores, aes(x=reorder(Variables, MeanDecreaseGini), y=MeanDecreaseGini, fill=MeanDecreaseGini)) +

geom\_bar(stat='identity') + coord\_flip() + theme(legend.position="none") + labs(x="") +

ggtitle('Variable Importance Random Forest') + theme(plot.title = element\_text(hjust = 0.5))

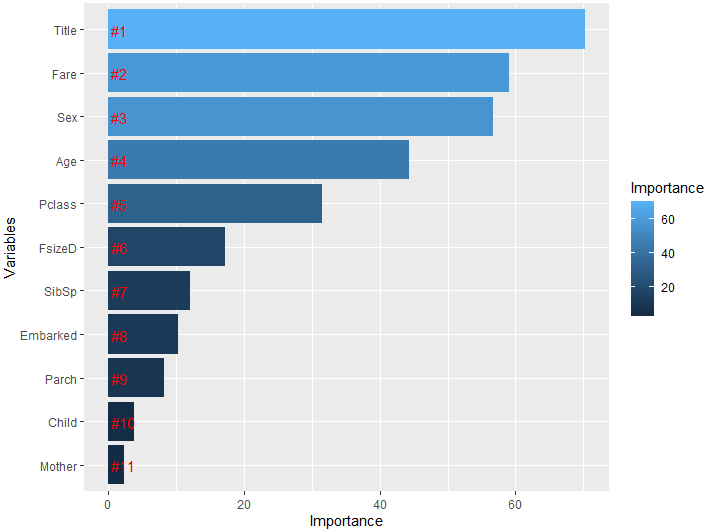


Figure 19. Visualization of Variable Importance

It is clear in the plot that, the Title variable has the highest relative importance.

## SVM Model

Let’s build the SVM model with the following commands.

set.seed(2017)

svm\_model <- train(Survived~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked + Title +

FsizeD + Child + Mother, data=train, method='svmRadial', preProcess= c('center', 'scale'), trControl=trainControl(method="cv", number=5))

svm\_model

svm\_model$results

svm\_predictions <- predict(svm\_model, test)

svm\_solutions <- data.frame(PassengerID = test$PassengerId, Survived = svm\_predictions)

write.csv(svm\_solutions, file = 'output/svm\_solutions.csv', row.names = F)

The output of the svm\_model and svm\_model$results commands are shown below.

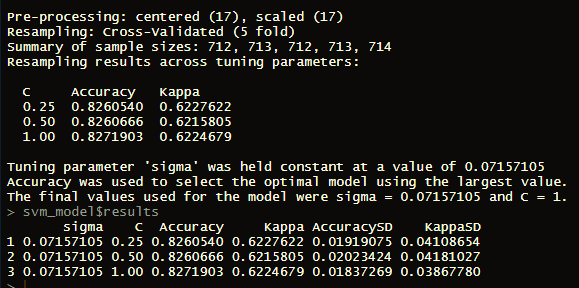


Figure 20. *The output of the* svm\_model *and* svm\_model$results

The final result for SVM model is “**0.80382**”.

## GBM Model

Let’s build the GBM (Gradient Boosting Machine) model with the following commands.

set.seed(2017)

gbm\_model <- train(Survived~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked + Title +

FsizeD + Child + Mother, data=train, method='gbm', preProcess= c('center', 'scale'), trControl=trainControl(method="cv", number=7), verbose=FALSE)

print(gbm\_model)

gbm\_predictions <- predict(gbm\_model, test)

gbm\_solutions <- data.frame(PassengerID = test$PassengerId, Survived = gbm\_predictions)

write.csv(gbm\_solutions, file = 'output/gbm\_solutions.csv', row.names = F)

The output of gbm\_model command is shown below.

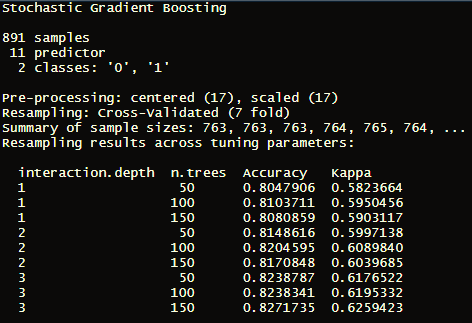


Figure 21. The output of gbm\_model command is shown below.

The final result for GBM model is “**0.76555**”.

# RESULTS

Comparison of all three models is shown in Figure 22.

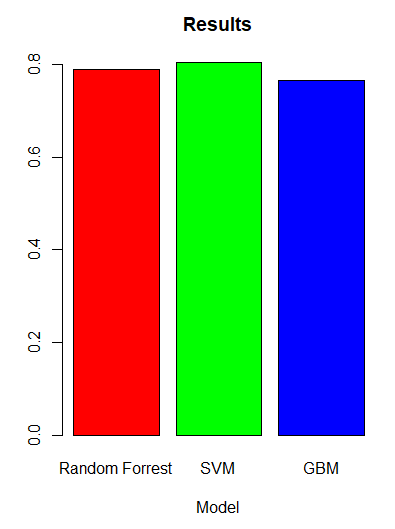


Figure 22. Comparison of the Models

# CONCLUSION

In this work, we tried to predict the survival of people in the Titanic shipwreck. We created some illustrative data visualizations and used data mining methods. We did some feature engineering and missing value imputation. Finally, we created 3 different models to predict the Survival variable.