PassengerId คือ เลขประจำตัวผู้โดยสาร

Pclass คือ ระดับของผู้โดยสาร

1 = ระดับ 1 ระดับสูง

2 = ระดับ 2 ระดับปานกลาง

3 = ระดับ 3 ระดับล่าง

Name คือ ชื่อของผู้โดยสาร

Sex คือ เพศของผู้โดยสาร

Age คือ อายุของผู้โดยสาร

SibSp คือ จำนวนบุคคลที่เดินทางมาด้วย (ญาติพี่น้อง สามี ภรรยา)

Parch คือ จำนวนบุคคลที่เดินทางมาด้วย (พ่อแม่ ลูก)

Ticket คือ หมายเลขตั๋ว

Fare คือ ค่าโดยสาร

Cabin คือ หมายเลขห้องพัก

Embarked คือ สถานที่ขึ้นเรือ

C = Cherbourg

Q = Queenstown

S = Southampton

Survived คือ ผู้ที่รอดชีวิต

0=ไม่รอด

1=รอด

5605104046 นายวิศิษฐ์ เลิศศักดิ์วิมาน

5605104043 นายอานนท์ กันทา

```
#write by 5605104046 นาย วิศิษฐ์ เลิศศักดิ์วิมาน

#and 5605104043 นาย อานนท์ กันทา

# Load the data sets

setwd("D:/Com/ICT_UTCC/DATA MINING/titanic-master/")

train <- read.csv("Data/CSV/train.csv", stringsAsFactors = FALSE)  # 891 obj ชุด

test <- read.csv("Data/CSV/test.csv", stringsAsFactors = FALSE)    # 418 obj ชุด

#Import library

library(rpart)

library(randomForest)

library(party)

library(rattle)

library(rpart.plot)

library(RColorBrewer)

# Clean data sets (1)- (9)     -> (1) ทำการใส่ค่า NA ลงไปในช่องว่างในชุดข้อมูลฝึกสอน

train[train == ""] <- NA

#(2) ทำการใส่ค่า NA ลงไปในช่องว่างของตัวแปรSurvivedของข้อมูล

test$Survived <- NA

#(3) ทำการรวมชุดข้อมูลฝึกสอนและข้อมูลทดสอบไว้ด้วยกันในตัวแปร combi

combi <- rbind(train, test)

#(4) ทำสร้างตัวแปร Title และตัดเอาทำนำหน้าชื่อจากตัวแปร Name ไปสร้างตัวแปร Title

#    และทำการใส่ค่า Mlle แทนลงไปในช่องที่มีคำว่า Mme หรือ Mlle

#    และทำการใส่ค่า Lady แทนลงไปในช่องที่มีคำว่า Dona หรือ Lady หรือ the Countess

#    และทำการใส่ค่า Sir แทนลงไปในช่องที่มีคำว่า Capt หรือ Don หรือ Major หรือ Sir หรือ Jonkheer หรือ Dr

#    และทำการแปลง Title จากข้อมูลแบบvectorให้เป็นข้อมูลแบบfactor โดยฟังก์ชัน factor

combi$Title <- sapply(combi$Name, FUN=function(x) {strsplit(x, split='[,.]')[[1]][2]})

combi$Title <- sub(' ', '', combi$Title)

combi$Title[combi$Title %in% c('Mme', 'Mlle')] <- 'Mlle'

combi$Title[combi$Title %in% c('Dona', 'Lady', 'the Countess')] <- 'Lady'

combi$Title[combi$Title %in% c('Capt', 'Don', 'Major', 'Sir', 'Jonkheer', 'Dr')] <- 'Sir'

combi$Title <- factor(combi$Title)
```

#(5) ทำสร้างตัวแปร FamilySize และทำการใส่ขนาดของครอบครัวลงไปโดย นำ

จำนวนบุคคลที่เดินทางมาด้วย (ญาติพี่น้อง สามี ภรรยา) + จำนวนบุคคลที่เดินทางมาด้วย (พ่อแม่ ลูก) + 1(จำนวนของตัวเอง)

```
combi$FamilySize <- combi$SibSp + combi$Parch + 1
```

#(6) ทำสร้างตัวแปร Surname และทำการใส่สกุลที่ตัดได้จาก ตัวแปรName

```
combi$Surname <- sapply(combi$Name, FUN=function(x) {strsplit(x, split='[,.]')[[1]][1]})
```

#(7) ทำสร้างตัวแปร FamilyID เพื่อแยกประเภทขนาดความใหญ่ของครอบครัว

โดยทำการใส่ Small ลงไปใน FamilyID ที่ FamilySize <= 2

```
combi$FamilyID <- paste(as.character(combi$FamilySize), combi$Surname, sep="")
combi$FamilyID[combi$FamilySize <= 2] <- 'Small'
famIDs <- data.frame(table(combi$FamilyID))
famIDs <- famIDs[famIDs$Freq <= 2,]
combi$FamilyID[combi$FamilyID %in% famIDs$Var1] <- 'Small'
combi$FamilyID <- factor(combi$FamilyID)
```

#(8) ทำค่าลงไปในตัวFare ที่มีIndex = 1044 โดยค่าที่ใส่เป็นค่าเฉลี่ยจาก Fareทั้งหมด

```
combi$Fare[1044] <- median(combi$Fare, na.rm=TRUE)
```

#(9) ทำค่าลงไปในตัว Embarked columลำกับที่ 62และ 830 ด้วยค่า S เนื่องจากค่าSถูกแทนมากที่สุด

```
combi$Embarked[c(62,830)] = "S"
```

#ทำการสร้างตัวแปร Agefit มาเพื่อพยาการณ์หาค่า Ageที่ว่างอยู่โดยมีตัวแปรเป้าหมายคือ Age และมีตัวที่ใช้ในการพยาการณ์ คือ

Pclass , Sex , SibSp , Parch , Fare , Embarked , Title , FamilySize

#โดยใช้ฟังก์ชัน rpart และใช้ method="anova"

```
Agefit <- rpart(Age ~ Pclass + Sex + SibSp + Parch + Fare + Embarked + Title + FamilySize,
data=combi[!is.na(combi$Age),], method="anova")
combi$Age[is.na(combi$Age)] <- predict(Agefit, combi[is.na(combi$Age),])
```

#ทำการแปลงหรือencode (Sex, Embarked ,Survived) จากข้อมูลแบบvectorให้เป็นข้อมูลแบบfactor โดยฟังก์ชัน factor

```
combi$Sex <- factor(combi$Sex)

combi$Embarked <- factor(combi$Embarked)

combi$Survived <- factor(combi$Survived)
```

ทำการแยกชุดข้อมูลฝึกสอนและข้อมูลทดสอบ (combi) ออกจากกัน ไว้ในตัวแปร train test

```
train <- combi[1:891,]

test <- combi[892:1309,]
```

ทำการสร้างตัวแปร fit มาเพื่อพยากรณ์หาผู้รอดชีวิตโดยมีตัวแปรเป้าหมายคือ Survived และมีตัวที่ใช้ในการพยากรณ์ คือ

Pclass , Sex , Age , SibSp , Parch, Fare, Embarked , Title , FamilySize , FamilyID

โดยใช้ฟังก์ชัน cforest กำหนดให้ ntree=5000, mtry=2

```
set.seed(415)

fit <- cforest(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked + Title + FamilySize + FamilyID, data =
train, controls=cforest_unbiased(ntree=5000, mtry=2))
```

สร้างตัวแปร Prediction ขึ้นมา และนำค่า fit ที่ได้จากการพยากรณ์ใส่ลงไปในข้อมูลทดสอบ test

```
Prediction <- predict(fit, test, OOB=TRUE, type = "response")
```

สร้างตัวแปร submit ขึ้นมา ทำการสร้าง frame ข้อมูลโดยใส่

colum แรกชื่อ PassengerId และใส่ค่า PassengerId ของข้อมูลทดสอบลงไป

colum สองชื่อ Survived และใส่ค่า Prediction ที่ได้จากการพยากรณ์ลงไป

```
submit <- data.frame(PassengerId = test$PassengerId, Survived = Prediction)
```

#ทำการนำข้อมูลจากตัวแปร submit เขียนข้อมูลลงไปใน Path “ Data/CSV/ “ ไฟล์ชื่อ CF_27_3.csv

```
write.csv(submit, file = "Data/CSV/CF_27_3.csv", row.names = FALSE)
```