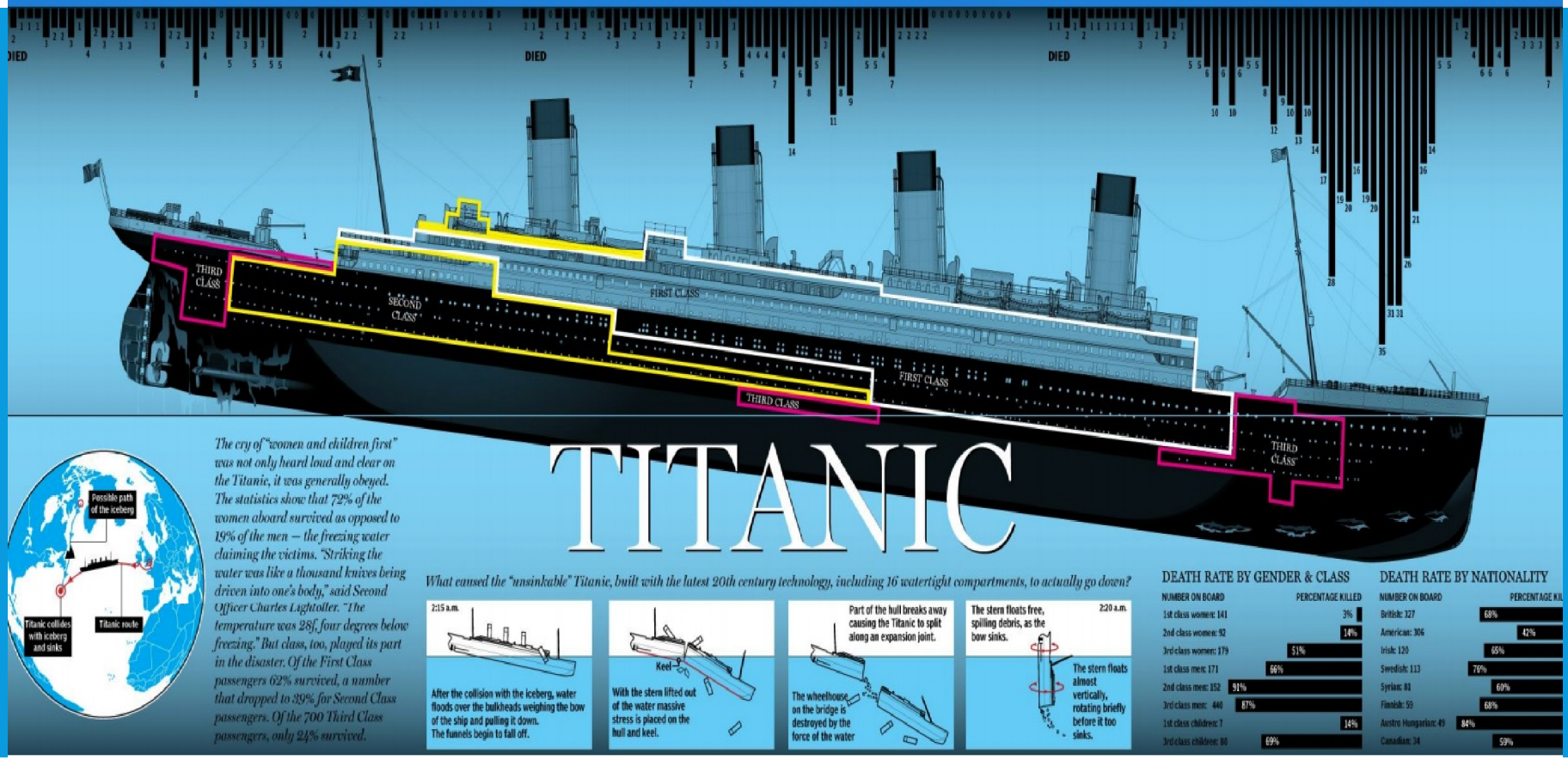
The Titanic: Machine Learning from Disaster

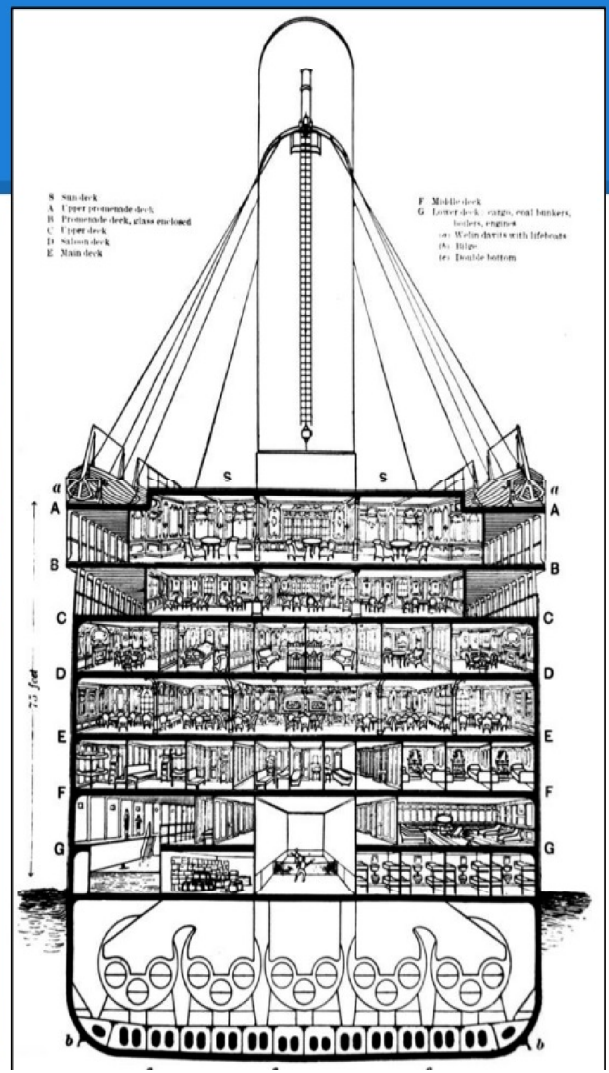
Computational INtelligence and Business Applications Project Two

Gowrishankar Vindhiyavarman | Anand Sampathkumar

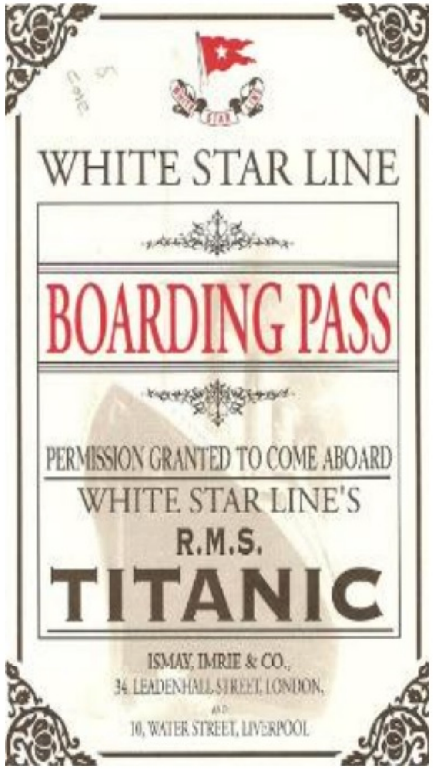
"Any data relating to one's location on the ship could prove helpful to survival predictions..."



April 1912
The Titanic Disaster



AGENDA



- Data Visualization
- Tree-based methods: rpart
- Ensemble Methods: randomForest, cForest
- Support Vector Machine
- Neural Network
- Feature Engineering
- Results

RMS Titanic, April 1912
A priori knowledge from problem domain

What factors contributed to survival?

Gender, Age, Passenger Class, Fare, Family

More likely to perish	More likely to survive
• Males	• Females
• Adults >50	• Children, Adults<50
• 2nd, 3rd class	• 1st Class
• Paid lower fares	• Paid higher fares
• Travelling alone	• Travelling with family
• Immigrants	

Titanic Dataset
Predictor & Target *Variables*

Response
VARIABLE

Survived
(1 = Yes; 0 = No)

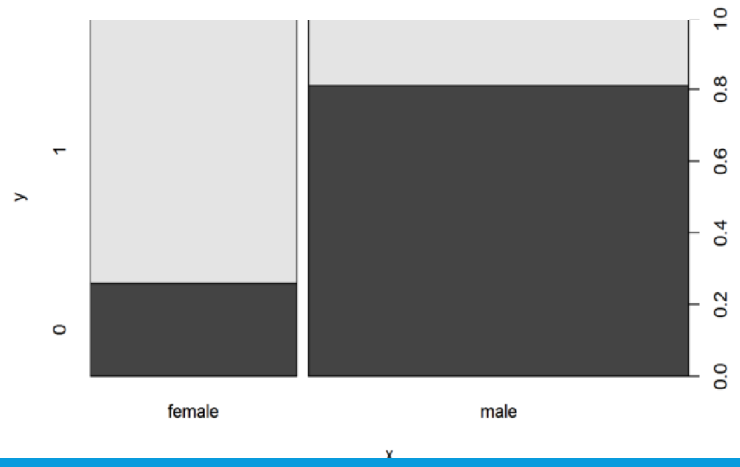
Predictor Variables	*DESCRIPTION*
Pclass	Passenger Class (1=1st; 2=2nd; 3=3rd)
Name	Passenger Name
Sex	Sex ("male", "female")
Age	Age (Numeric fraction e.g., 1.5)
Fare	Passenger Fare
Sibsp	Number of Siblings/Spouses Aboard
Parch	Number of Parents/Children Aboard
Ticket	Ticket Number
Cabin	Cabin
Embarked	Port of Embarkation (C=Cherbourg; Q=Queenstown; S=Southampton)

QUANTITATIVE Variables; the rest are QUALITATIVE.

Data Visualization

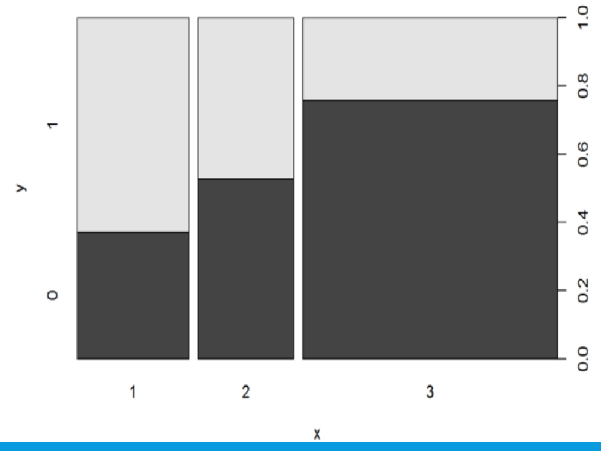
Rate of survival for Males and Females

```
##
##            0   1
##   female  81 233
##   male   468 109
```



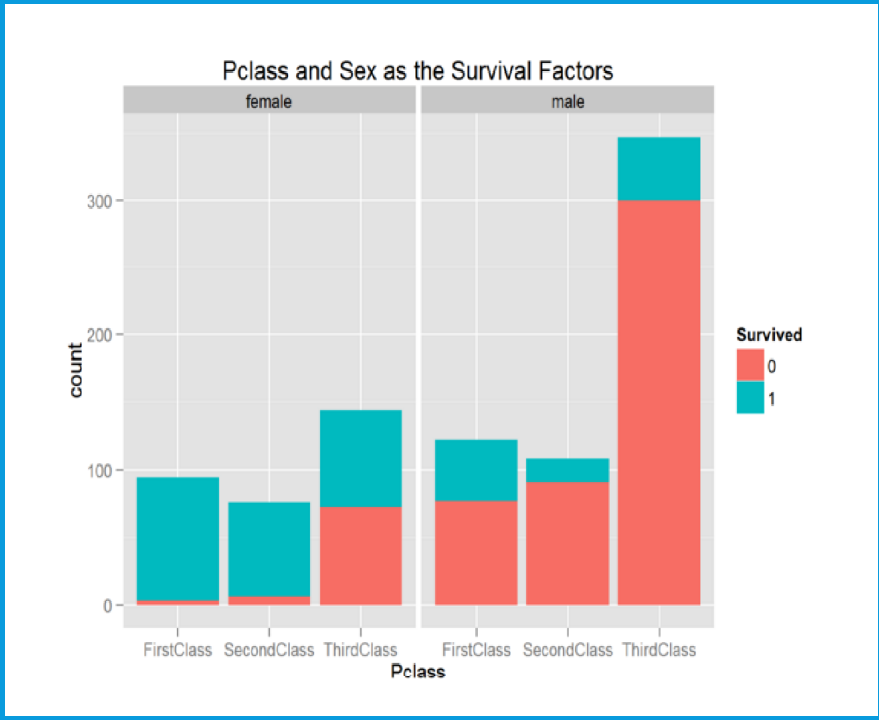
Does the passenger class have an impact on survival?

```
##
##      0   1
##   1  80 136
##   2  97  87
##   3 372 119
```

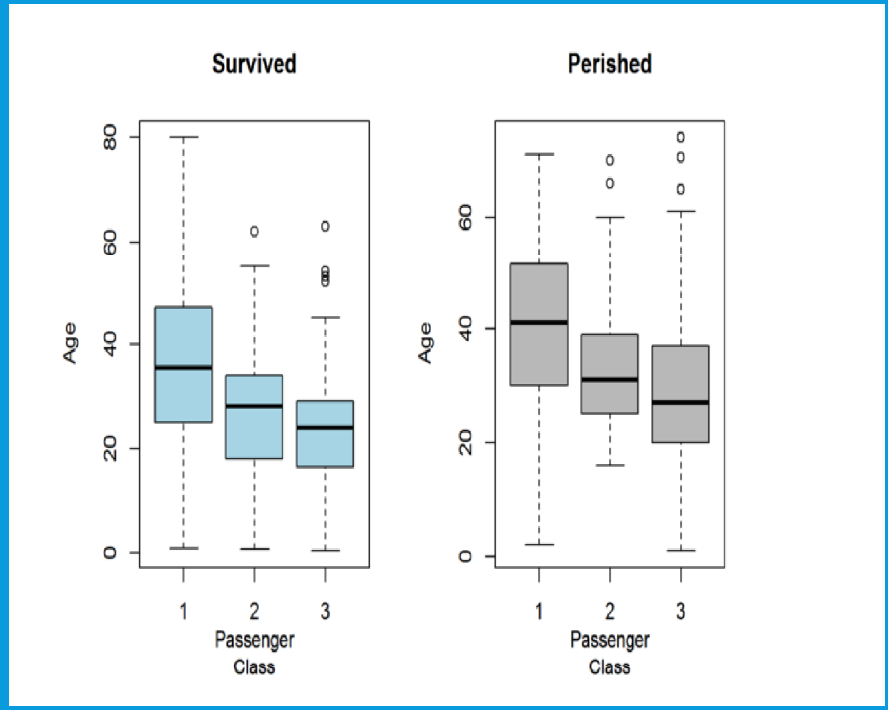


Females have a better survival rate. 75% of the women have survived, whereas only 18% of men survived.

Impact on Survival -Class

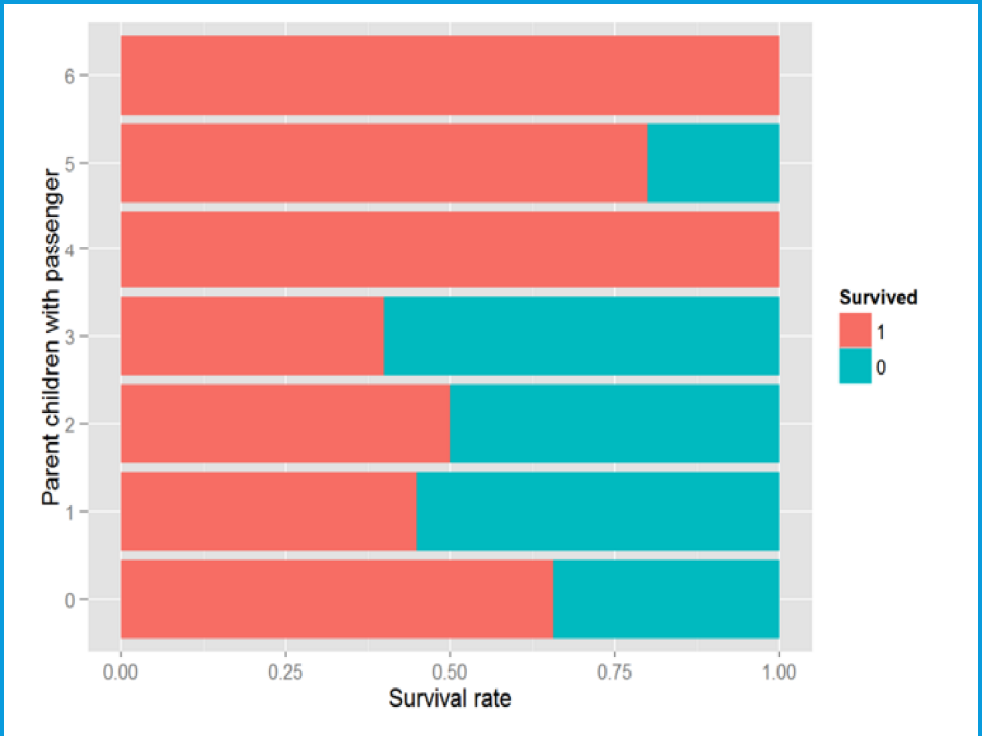


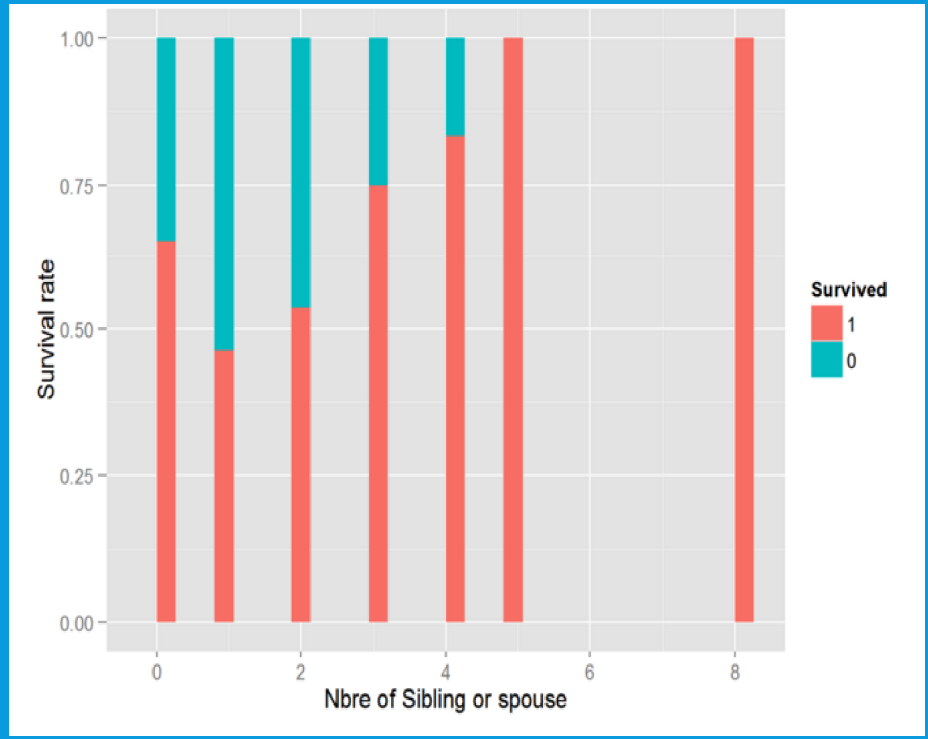
Females in the upper classes have a higher rate of survival



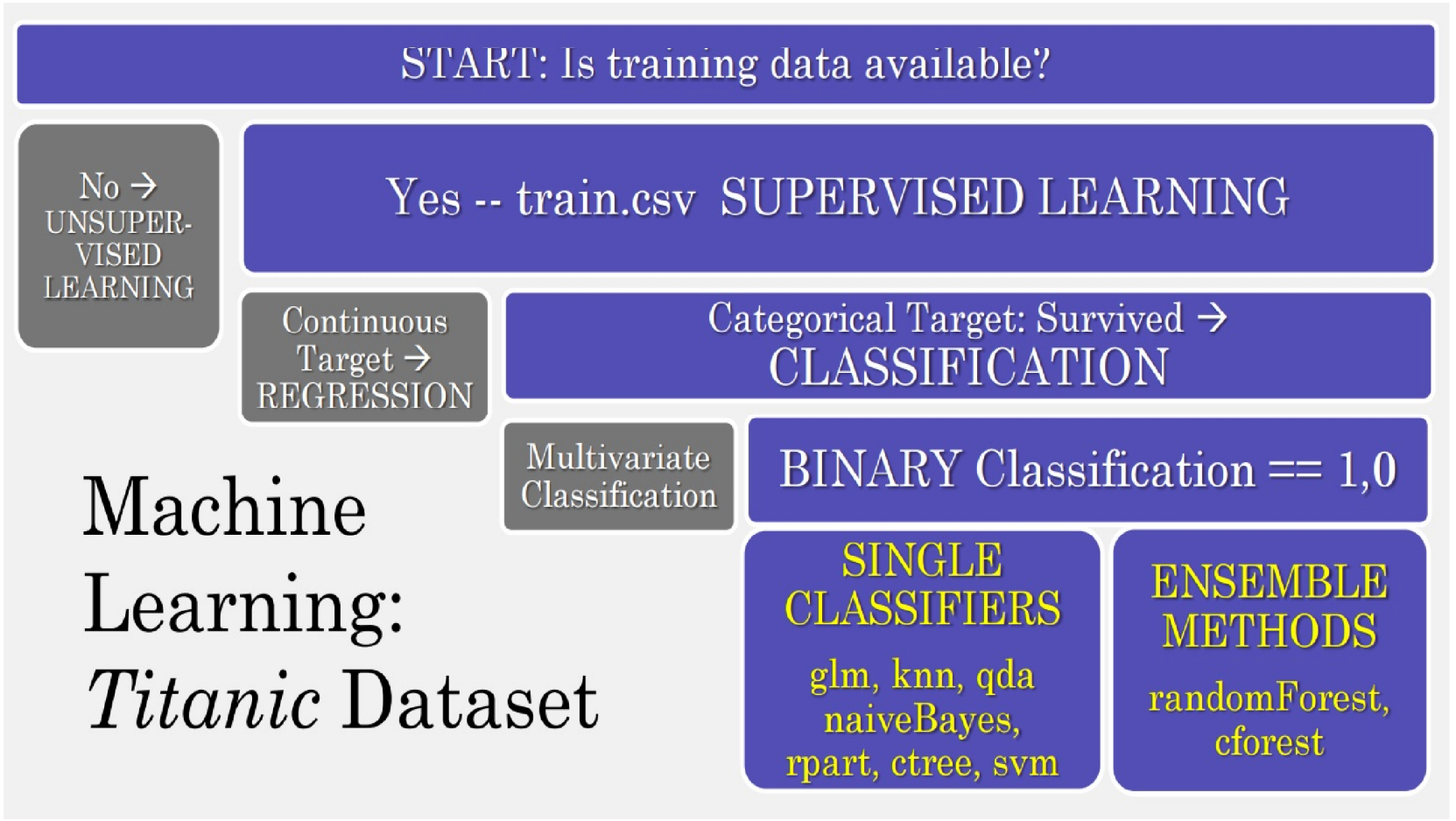
The Mean age of the Survived is less than that of those who perished

Impact on survival class contd..





Survival is Higher for passengers with Parch=3 or SibSp =1


START: Is training data available?
No → UNSUPER-VISED LEARNING
Yes -- train.csv SUPERVISED LEARNING
Continuous Target → REGRESSION
Categorical Target: Survived → CLASSIFICATION
Multivariate Classification
BINARY Classification == 1,0
SINGLE CLASSIFIERS
glm, knn, qda naiveBayes, rpart, ctree, svm
ENSEMBLE METHODS
randomForest, cforest
Machine Learning: Titanic Dataset

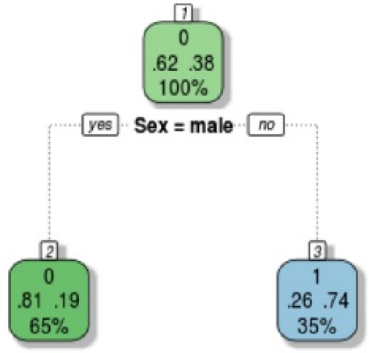
Decision Trees

- A **decision tree** is a simple, but powerful form of multiple variable analysis. It displays a tree-like graph of decisions and their possible consequences.

- **Recursive Partitioning**-> at each step, we identify a question that we use to partition the data.

Advantages:

- **Data-driven:** Makes no prior assumptions; selects significant predictors based on the greatest information gain.
- **Flexible**: No data pre-processing needed! Handles numeric and categorical data.
- **Easy to interpret** and explain to others.



Decision Trees -accuracy

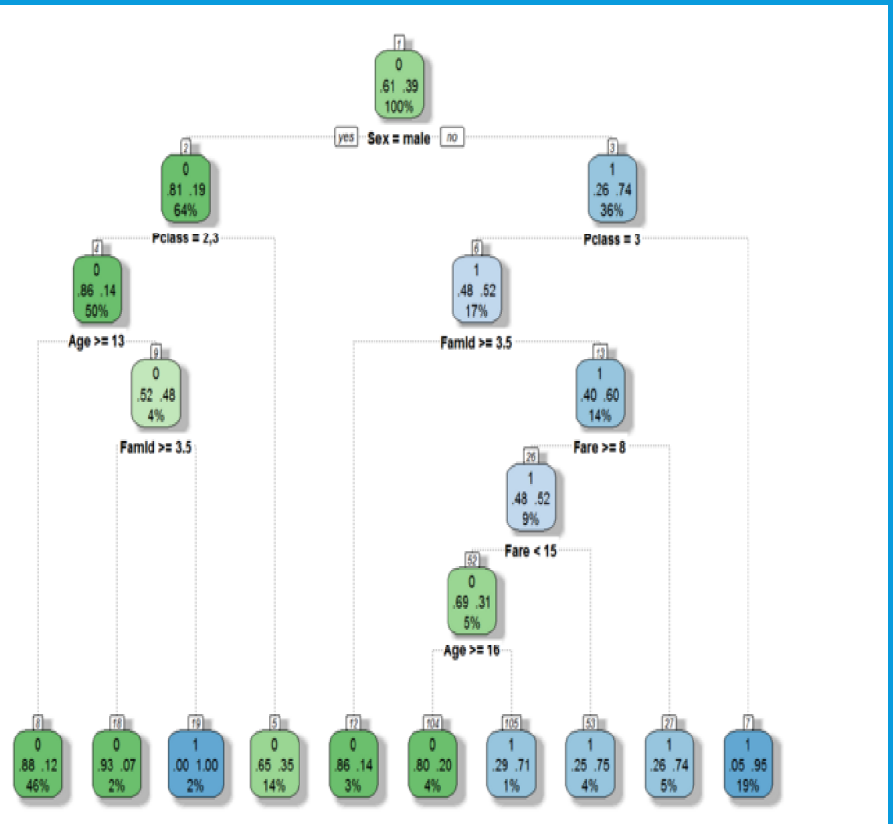
```
library(caret)


#Applying Decision trees
fit <- rpart(Survived ~ Sex + Age + FamId + Pclass + Fare , data = dataTrain, method="class")
fancyRpartPlot(fit)
```

```
dataVal$pred_dt <- predict(fit, dataVal[,-1], type = "class")
conf_Tree <- confusionMatrix(dataVal$pred_dt, dataVal$Survived)
#Confusion Matrix
tab=with(dataVal,table(True=Survived,Predicted=pred_dt))
print(tab)
```

```
##      Predicted
## True   0   1
##    0 110  10
##    1  19  52
```

```
#Accuracy
sum(diag(tab)) / sum(tab)
```

```
## [1] 0.8481675
```

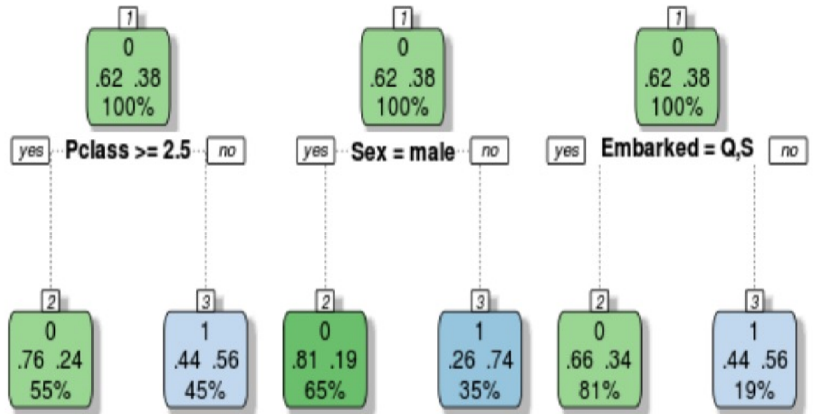


Random Forests

- A group of actors who perform together.
- An example of an **ENSEMBLE METHOD –** combines multiple models to produce one result.
- Unlike single decision trees which can suffer from high variance or high bias, Random Forests use ***random sampling*** and ***averaging*** to find a natural balance between the two extremes.

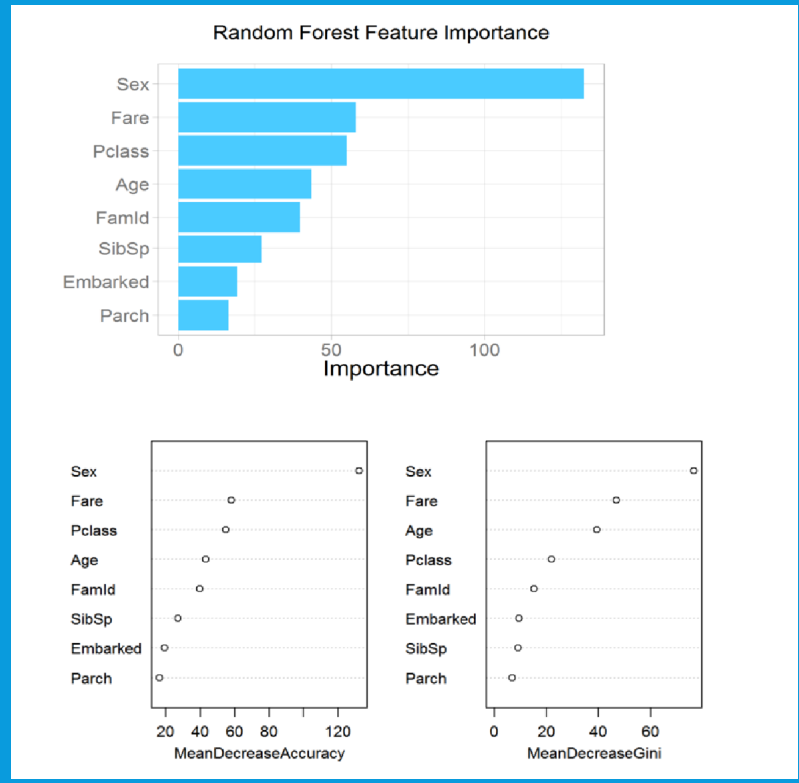
Advantages:

- Easy to use: can be used quite efficiently with default parameters.
- Ideal for people without a deep background in statistics.
- Produces fairly strong predictions with only a small amount of coding.



Random forests-contd..

```
# Random Forests
set.seed(415)
fit2<- randomForest(as.factor(Survived) ~ Pclass + Sex + Age + Fare + Embarked + FamId + Parch + SibSp, data=dataTrain,
                    importance=TRUE, ntree=2000)
varImpPlot(fit2)

imp <- importance(fit2, type=1)
featureImportance <- data.frame(Feature=row.names(imp), Importance=imp[,1])

 ggplot(featureImportance, aes(x=reorder(Feature, Importance), y=Importance)) +
      geom_bar(stat="identity", fill="#53cfff") +
      coord_flip() +
      theme_light(base_size=20) +
      xlab("") +
      ylab("Importance") +
      ggtitle("Random Forest Feature Importance\n") +
      theme(plot.title=element_text(size=18))
dataVal$Pred_rf <- predict(fit2, dataVal)
Conf_RForest <- confusionMatrix(dataVal$Pred_rf, dataVal$Survived)

#C- forest
set.seed(415)
fit <- cforest(as.factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked + FamId, data = dataTrain, control
s=cforest_unbiased(ntree=2000, mtry=3))

Pred_cf <- predict(fit, dataVal, OOB=TRUE, type = "response")
Conf_CForest <- confusionMatrix(Pred_cf, dataVal$Survived)
```



Support vector machine

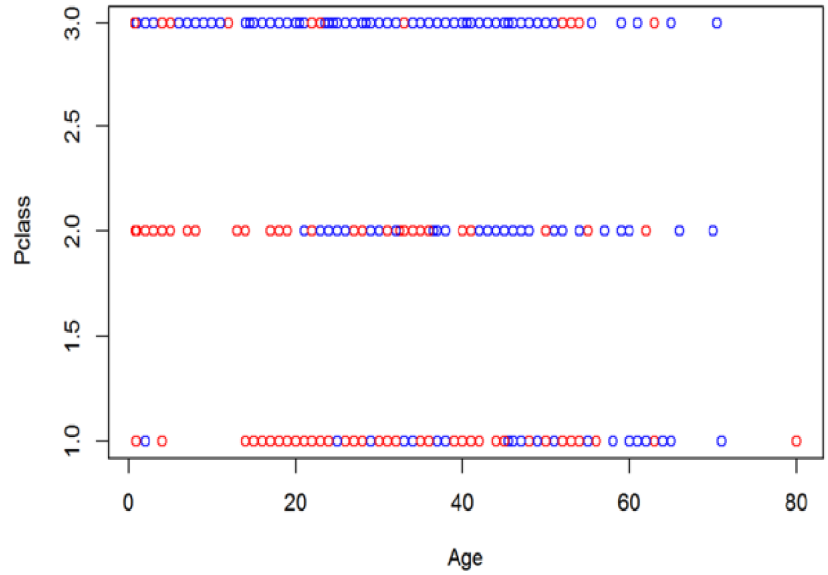
Support Vector Machines (SVMs) are supervised learning methods used for classification and regression tasks that originated from statistical learning theory . As a classification method, SVM is a global classification model that generates non-overlapping partitions and usually employs all attributes . The entity space is partitioned in a single pass, so that flat and linear partitions are generated . SVMs are based on maximum margin linear discriminants, and are similar to probabilistic approaches, but do not consider the dependencies among attributes

Advantages

- SVMs are currently among the best performers for a number of classification tasks ranging from text to genomic data.
- SVMs can be applied to complex data types beyond feature vectors (e.g. graphs, sequences, relational data) by designing kernel functions for such data.
- Produce very accurate classifiers and less overfitting , robust to noise.

Support vector machine -contd..

```
#SVM modelling
plot(dataTrain$Age, dataTrain$Pclass, xlab="Age", ylab="Pclass", col=ifelse(dataTrain$Survived==1, "red", "blue"))

train_svm<-dataTrain[,c("Age","Sex","Pclass","SibSp","Parch","Survived")]
svm.model<-svm(Survived ~ . , data = train_svm, kernel="radial")

dataVal_svm<-dataVal[,c("Age","Sex","Pclass","SibSp","Parch")]
preds<-predict(svm.model, dataVal_svm)


conf_Svm  <- confusionMatrix(preds, dataVal$Survived)
```



Artificial Neural Networks

- Artificial Neural Networks, also known as "Artificial neural nets", "neural nets", or ANN for short, are a computational tool modeled on the interconnection of the neuron in the nervous systems of the human brain and that of other organisms. Biological Neural Nets (BNN) are the naturally occurring equivalent of the ANN. Both BNN and ANN are network systems constructed from atomic components known as "neurons". Artificial neural networks are very different from biological networks, although many of the concepts and characteristics of biological systems are faithfully reproduced in the artificial systems

Advantages

- More like a real nervous system .
- Parallel organization permits solutions to problems where multiple constraints must be satisfied simultaneously.
- Graceful degradation.
- Rules are implicit rather than explicit.

Artificial neural networks contd..
And comparison

```
#Applying Neural network

nnet1 = nnet(Survived ~ Sex + Pclass + Fare + Age + SibSp,
             data=dataTrain, size = 2, rang = 0.1,
             decay = 5e-4, maxit = 200)

prediction$net.result = predict(nnet1, dataVal)

for(i in 1:length(prediction$net.result)){
  if(prediction$net.result[i]>0.6){prediction$net.result[i]<-1}
  else{prediction$net.result[i]<-0}
}
conf_Neuralnet <- confusionMatrix(prediction$net.result, dataVal$Survived)
```

Accuracy

```
## neuralnet 0.8272251
```

```
# Comparison of predictions from each model
compare_df <- data.frame(Accuracy = c(conf_Tree$overall[1],
                                      Conf_RForest$overall[1],
                                      Conf_CForest$overall[1],
                                      conf_Svm$overall[1],
                                      conf_Neuralnet$overall[1]),
                         row.names = c("rpart", "rf", "cforest", "svm", "neuralnet"))
compare_df
```

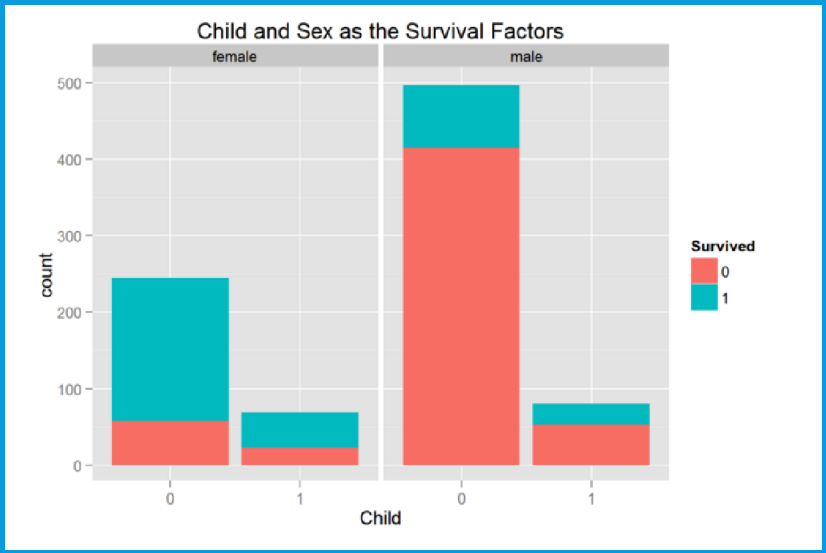
```
# Comparison
```



```
##              Accuracy
## rpart       0.8481675
## rf          0.8534031
## cforest     0.8481675
## svm         0.8481675
## neuralnet 0.8272251
```

Child and sex-survival factor

```
ggplot(train, aes(Child)) +
  geom_bar(aes(fill = Survived )) +
  facet_grid(~Sex) +
  ggtitle("Child and Sex as the Survival Factors")
```



From here, it's also somewhat evident that female children are more likely to survive.

Child or woman -indicator

Creating an indicator for Woman

```
train$Woman <- ifelse(train$Sex=="female",1,0)
```

Creating another indicator for Child or Woman

```
train$CoW <- ifelse(train$Child==1 | train$Woman==1,1,0)
```

NOW COMING TO THE VARIABLES RELATED TO FAMILY.

```
train$FamilySize <- train$SibSp + train$Parch + 1
train$FamilyCat <- cut(train$FamilySize, c(0,1,4,12))
```

What is the family size of the individual? That's what FamilySize gives you. In the next step, I have split the Family Sizes into 3 groups - those who are travelling alone, those whose family size is 2-4 and the third group of individuals whose family size is greater than 4.

To demonstrate how this works, let's tabulate the FamilyCat and survival

```
table(train$FamilyCat, train$Survived)
```

```
##
##           0   1
##   (0,1]  374 165
##   (1,4]  123 169
##   (4,12]  52  10
```

Individuals travelling alone and those with very high family sizes have a bad survival rate.

Now a variable for tracking down the 3rd class passengers alone - the other two passenger classes had a fairly similar survival rate.

```
train$PC3 <- ifelse(train$Pclass=="3",1,0)
```

I have decided to go with the following independent variables:

Pclass # Passenger class - the proxy for socio-economic status FamilyCat # Family Category - Family Size=1, Family Size 2-4, Family Size >4 Sex CoW # Is the passenger a Child or a woman? Title

I'm loading only these required variables into a new data frame:

```
df_train <- train[,c("Survived","FamilyCat","Pclass","Sex","CoW","Title")]
```

The next step is to split the data into a Training Set & a Validation Set

```
dataTrainfe<-df_train[1:700,]
dataValfe<-df_train[701:nrow(df_train),]
```

What the above code means is that the first 700 rows of data are loaded into the data frame object "dataTrainfe" and the remaining rows are loaded into "dataValfe", which is going to be my validation set.

All my variables are categorical, so I'm converting them to factors.

```
dataTrainfe$Survived <- as.factor(dataTrainfe$Survived)
dataTrainfe$Sex <- as.factor(dataTrainfe$Sex)
dataTrainfe$Title <- as.factor(dataTrainfe$Title)
dataTrainfe$Pclass <- as.factor(dataTrainfe$Pclass)
dataTrainfe$CoW <- as.factor(dataTrainfe$CoW)
dataTrainfe$FamilyCat <- as.factor(dataTrainfe$FamilyCat)
```

Feature Engineering - Accuracy

```
# Comparison of predictions from each model
compare_df <- data.frame(Accuracy = c(conf_Treefe$overall[1],
                                      Conf_RForestfe$overall[1],
                                      Conf_CForestfe$overall[1],
                                      conf_Svmfe$overall[1],
                                      conf_Neuralnetfe$overall[1]),
                         row.names = c("rpart", "rf", "cforest", "svm", "neuralnet"))
compare_df
```

```
##                Accuracy
## rpart     0.8586387
## rf        0.8638743
## cforest   0.8586387
## svm       0.8481675
## neuralnet 0.8376963
```

CONCLUSION

Without Feature Engineering

```
##              Accuracy
## rpart     0.8481675
## rf        0.8534031
## cforest   0.8481675
## svm       0.8481675
## neuralnet 0.8272251
```

With Feature Engineering

```
##              Accuracy
## rpart     0.8586387
## rf        0.8638743
## cforest   0.8586387
## svm       0.8481675
## neuralnet 0.8376963
```

THANK YOU