SPM4450: Fundamentals of Data Analytics - Final Assignment Report

Performance of various Data Analytics Techniques on Kaggle's Problem Set 'Titanic: Machine Learning from Disaster'

Authors:

Nidhi Singh
4242246
n.singh-2@student.tudelft.nl
MSc. Computer Science

K. Chaitanya Akundi
k.c.akundi@student.tudelft.nl
MSc. Computer Science

List of Figures

Chapter 1

Titanic Data Set

1.1 Problem Description

For our final assignment, we have taken up a challenge from Kaggle 'Predict survival on the Titanic'. The dataset includes details of people who travelled on RMS Titanic which sank in 1912 killing 1502 out of 2224 passengers. The aim of the Kaggle challenge is to complete the analysis of what sorts of people were likely to survive. In order to do so, we will apply different predictive models to the dataset and will finally evaluate their performance against each other. Kaggle also supports Leaderboards which evaluate the submitted results, but since this evaluation is based on only 50% of the test data, it makes sense to do performance evaluation of all the models.

Since we are given both training and test data set, this problem's predictive models will fall under the umbrella of Supervised Learning Algorithms. Also we have to decide whether a passenger survived or not, this makes it a classic Classification problem.

1.2 Data Exploration

Before diving deep into prediction making on test data, we will explore the dataset. We are given two sets of data, training (data containing attributes and known outcomes [survived or perished] for a subset of the passengers) and test (data containing attributes without outcomes for a subset of passengers).The given training data set has 891 observations of following 12 variables:

- PassengerId - Unique generated Id for each passenger
- Survived - Survival(0 = No; 1 = Yes)
- Pclass - Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)
- Name - Name of the person
- sex - Sex
- Age - Age
- Sibsp - Number of Siblings/Spouses Aboard

- Parch - Number of Parents/Children Aboard
- Ticket - Ticket Number
- Fare - Passenger Fare
- Cabin - Cabin in the ship
- Embarked - Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

Let us start by looking at the type of these variables

```
str(train_csv)

## '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 5
##  $ 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",..: 525 596 662 50 473 276 86 39
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
##  $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

Here Factor refers to categorical data, since all the names are unique, we have 891 levels equal to number of observations.

```
prop.table(table(train_csv$Survived))

##
##      0      1
## 0.6162 0.3838
```

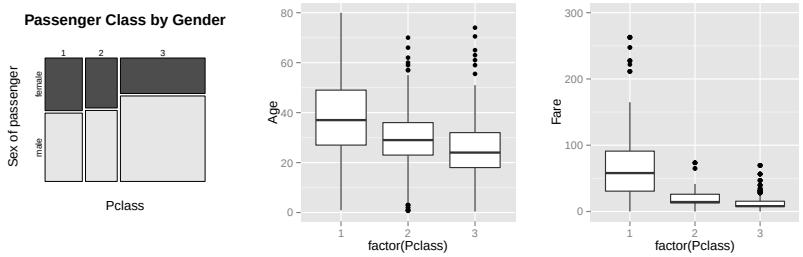
This shows that 61.6% of the passengers perished and only 38.3% survived. Running the same code for Sex, we find 35.2% females and 64.7% in the training data set.

```
summary(train_csv$Age)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.42   20.10   28.00   29.70   38.00   80.00     177
```

Summary results on Age shows that this variable is missing for 177 passengers and the minimum age is 0.42 or 5 months and maximum is 80, while 90% of the passengers were below 50.

```
prop.table(table(train_csv$Pclass))

##
##      1      2      3
## 0.2424 0.2065 0.5511
```

More than 55% passengers were travelling in third class. It will be worthwhile to see the age and sex of people in each class.



(a) Passenger Class by Gender. (b) Passenger Class by Age. (c) Passenger Class by Fare.

Figure 1.1: Passenger class by Sex, Age and Fare

We see in Figure1.1 that third class has mostly males, since third class cabins were at the bottom of the ship this might be one of the reasons that most of the males could not survive. Also passengers in third class were younger with median below 25.With just one outlier above $500 for first class ticket fare, fare is below $100.

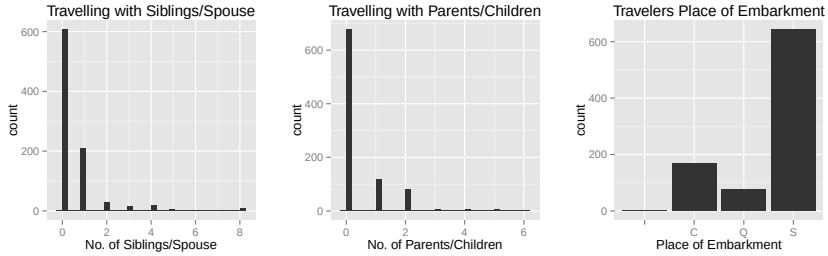


Figure 1.2: Frequency of Sibsp, Parch and Embarkment

We now look at other variables to see if they can have some influence on predictions. From Figure 1.2 we can see that most passengers travelled alone and started their journey from Southampton.

Other varibales 'Ticket' and 'Cabin' do not tell much as they have unique values, and are un-related to other variables.

1.2.1 Survived variable with other variables

Till now we looked at the variables and their values and frequencies and tried to get an initial understanding of the data. Since we have to predict the 'Survived'

variable for the test set, in this section we will look at the relation between 'Survived' variable and other variables.

As we can see from Figure 1.3 age and fare doesnt seem to give much information about the survived variable, moreover most of the passengers were from Southampton so Place of Embarkment doesn't seem to play much role too.

But from Figure 1.4 we can find some interesting facts, people in 1st class outnumbered the people from 3rd class in survival rate.So there was a clear preference for elite poeple. From the second plot in Figure1.4 we can see another preference was for females, we would like to believe that there was preference for children but this is not yet evident from our data. The last plot in Figure 1.4 shows that surely there was a clear bias for females in 1st and 2nd class compared to males. This is a clear indicator that 'Sex' variable is hihgly important for our analysis with maybe 'Pclass' coming next.

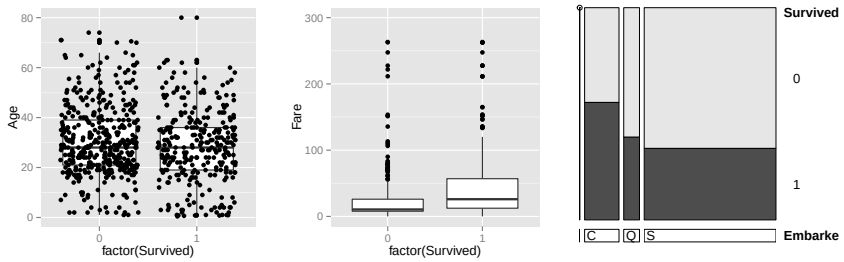


Figure 1.3: Passengers Survived by Age, Fare and Place of Embarkment

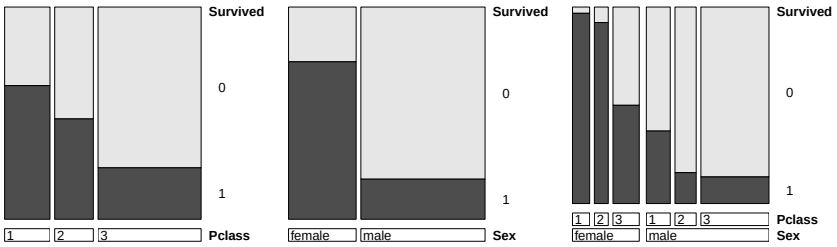


Figure 1.4: Passengers survived by Class and Gender

1.3 Feature Engineering

There are few variables which have 'NA' and missing values, before we run our prediction models we need to get rid of these

Chapter 2

Prediction Models

Since the 'Predict survival on the Titanic' challenge is a classification problem, we will start with Linear Classifiers and then go further with methods like Decision trees and Ensembles of classifiers. In the following sections we will explore each model in detail and will also report its evaluation on Kaggle.

2.1 Logistic Regression

Logistic Regression is a classical Classification algorithm. R's *glm* function is used to fit generalized linear models, With *family* variable set to "binomial", glm( ) produces a logistic regression.

We will run our first regression model with basic features provided within the dataset and can look at the results by calling summary on this model.

```
logit.m1 <- glm(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked,
                data = train.batch,
                family="binomial")
summary(logit.m1)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
##     Fare + Embarked, family = "binomial", data = train.batch)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.639  -0.617  -0.446   0.666   2.424
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.80e+01   6.07e+02    0.03    0.976
## Pclass      -1.10e+00   1.61e-01   -6.81  1.0e-11 ***
## Sexmale     -2.65e+00   2.23e-01  -11.93  < 2e-16 ***
## Age         -4.47e-02   8.88e-03   -5.04  4.7e-07 ***
## SibSp       -2.55e-01   1.20e-01   -2.13    0.033 *
```

	Actual Value	
Prediction	0	1
0	115	17
1	7	39

Table 2.1: Confusion Matrix - Initial Decision Tree

```
## Parch       -7.07e-02   1.30e-01   -0.54    0.588
## Fare         3.04e-05   2.83e-03    0.01    0.991
## EmbarkedC   -1.24e+01   6.07e+02   -0.02    0.984
## EmbarkedQ   -1.26e+01   6.07e+02   -0.02    0.983
## EmbarkedS   -1.29e+01   6.07e+02   -0.02    0.983
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 957.90  on 712  degrees of freedom
## Residual deviance: 646.84  on 703  degrees of freedom
## AIC: 666.8
##
## Number of Fisher Scoring iterations: 13
```

2.2 Decision Trees

Decision tree algorithms work by repeatedly splitting the dataset into subsets based on a particular attribute value. This process is recursively carried out until further splitting does not add any value to the predictions. This is a greedy algorithm, which means that decisions with the highest immediate value are given preference.

We applied recursive partitioning on the Titanic dataset using the Decision Tree algorithm from R's *rpart* package . For this, the dependent variables used were Pclass, Sex, Age, Fare, Embarked, Title and FamilyID2. The training dataset containing 891 records was further partitioned with 80% of it taken as the training set, and the remaining 20% as the test set, to evaluate the performance of the model. So there were 418 records in the training partition and 178 records in the test partition. The algorithm is invoked as follows.

```
dt.1 <- rpart(Survived ~ Pclass + Sex + Age + Fare + Embarked +
    Title + FamilyID2, data = train.batch, method = "class",
    control = rpart.control(minsplit = 2, cp = 0.001))
```

2.2.1

In the second run, we added the engineered features Title and FamilyID2 to the estimation, while removing Age, SibSp and Parch which are already composed

Statistic	Value
Accuracy	0.8652
Accuracy with unseen data(Kaggle)	0.78469
95% CI	(0.8061, 0.9117)
No Information Rate	0.6854
P-Value [Acc >NIR]	2.329e-08
Kappa	0.6715
Mcnemar's Test P-Value	0.06619
Sensitivity	0.9426
Specificity	0.6964

Table 2.2: Evaluation - Initial Decision Tree

	Actual Value	
Prediction	0	1
0	117	18
1	5	38

Table 2.3: Confusion Matrix - Engineered Features

in Title and FamilyID2.

```
dt.2 <- rpart(Survived ~ Pclass + Fare + Embarked + Title + FamilyID2,
    data = train.batch, method = "class")
```

This resulted in marginal gains in accuracy and sensitivity of the model, thus giving better results on unseen data.

2.2.2

In the third run, we added control parameters which specified *minsplit* as 2 and *cp* as 0. This allowed unbounded growth for the tree, and resulted in a complex structure with large a number of branches.

Statistic	Value
Accuracy	0.8708
Accuracy with unseen data(Kaggle)	0.79426
95% CI	(0.8124, 0.9163)
No Information Rate	0.6854
P-Value [Acc >NIR]	7.677e-09
Kappa	0.6803
Mcnemar's Test P-Value	0.01234
Sensitivity	0.9590
Specificity	0.6786

Table 2.4: Evaluation - Engineered Features

	Actual Value	
Prediction	0	1
0	122	2
1	0	54

Table 2.5: Confusion Matrix - Unbounded Growth Tree

Statistic	Value
Accuracy	0.9888
Accuracy with unseen data(Kaggle)	0.74163
95% CI	(0.96, 0.9986)
No Information Rate	0.6854
P-Value [Acc >NIR]	<2e-16
Kappa	0.9737
Mcnemar's Test P-Value	0.4795
Sensitivity	1.0000
Specificity	0.9643

Table 2.6: Evaluation - Unbounded Growth Tree

```
fit <- rpart(Survived ~ Pclass + Age + Fare + Embarked + Title +
    FamilyID2, data = train.batch, method = "class", control = rpart.control(minsplit = 2,
    cp = 0))
```

The results were very accurate on the training data. However, the model performed poorly with the test data. From this, we could conclude that the model is overfitting to a high degree.

2.3 Support Vector Machines

Support Vector Machine (SVM) algorithms perform classification by representing the given data as points in space, with a clear plane of separation between the different classes to which the points belong. SVMs use the 'kernel trick' to work with high-dimensional data without ever having to map the points in such spaces. In R, the `e1071` and `kernlab` packages offer methods for creating SVM models. We used the `kernlab` package to generate our SVM models.

2.3.1

Initially, the algorithm was run in linear classification mode, using the features that were already present in the given dataset. The results are as follows.

	Actual Value	
Prediction	0	1
0	119	17
1	3	39

Table 2.7: Confusion Matrix - Linear SVM

Statistic	Value
Accuracy	0.8876
Accuracy with unseen data(Kaggle)	0.62679
95% CI	(0.8318, 0.93)
No Information Rate	0.6854
P-Value [Acc >NIR]	2.044e-10
Kappa	0.7206
Mcnemar's Test P-Value	0.00365
Sensitivity	0.9754
Specificity	0.6964

Table 2.8: Evaluation - Linear SVM