4/24/2016

Darshan Parikh (dkp150130) Dharmam Buch (dsb150030)

THE UNIVERSITY OF TEXAS AT DALLAs

[**Titanic: Machine Learning from Disaster**](https://www.kaggle.com/c/titanic)

CS 6375.501 – Machine Learning.

1. **Introduction:**

The sinking of the RMS Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships.

One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class.

1. **Task Definition:**

2.1 Complete the analysis of what sorts of people were likely to survive. In particular, apply the tools of machine learning to predict which passengers survived the tragedy.

**Dataset Summary:**

#### Attributes:

1. Pclass Passenger Class. (1 = 1st; 2 = 2nd; 3 = 3rd)
2. Name Name
3. Sex Sex
4. Age Age
5. Sibsp Number of Siblings/Spouses Aboard
6. Parch Number of Parents/Children Aboard
7. Ticket Ticket Number
8. Fare Passenger Fare
9. Cabin Cabin
10. Embarked Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

#### Class Attribute:

1. Survival Survived or not ?(0=No, 1=Yes)

**Data Size (Number of Instances):**

1. Training Data 891 instances
2. Testing Data 418 instances

**Training Data Analysis:**

891 observations and 10 variables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Type** | **Levels** | **Missing** | **Mean/Median** |
| Pclass | Numeric |  |  | 2.3/3 |
| Sex | Categorical | 2 (female, male) |  |  |
| Age | Numeric |  | 177 | 29.7/28 |
| Sibsp | Numeric |  |  | 0.5 |
| Parch | Numeric |  |  | 0.4 |
| Ticket | Numeric |  |  |  |
| **Attribute** | **Type** | **Levels** | **Missing** | **Mean** |
| Fare | Numeric |  |  | 32.2/14.5 |
| Cabin | Numeric |  | 687 |  |
| Embarked | Categorical | 3 (C, Q, S) | 2 |  |

**2.2 Algorithm Definition:**

In this project we have used several strong classifiers to build and evaluate the model on the said dataset. We have tried Random Forest, SVM, AdaBoost classifiers in combination of Kmeans-BiCluster clustering to check with performance of various combinations of the mentioned classifiers and clustering.

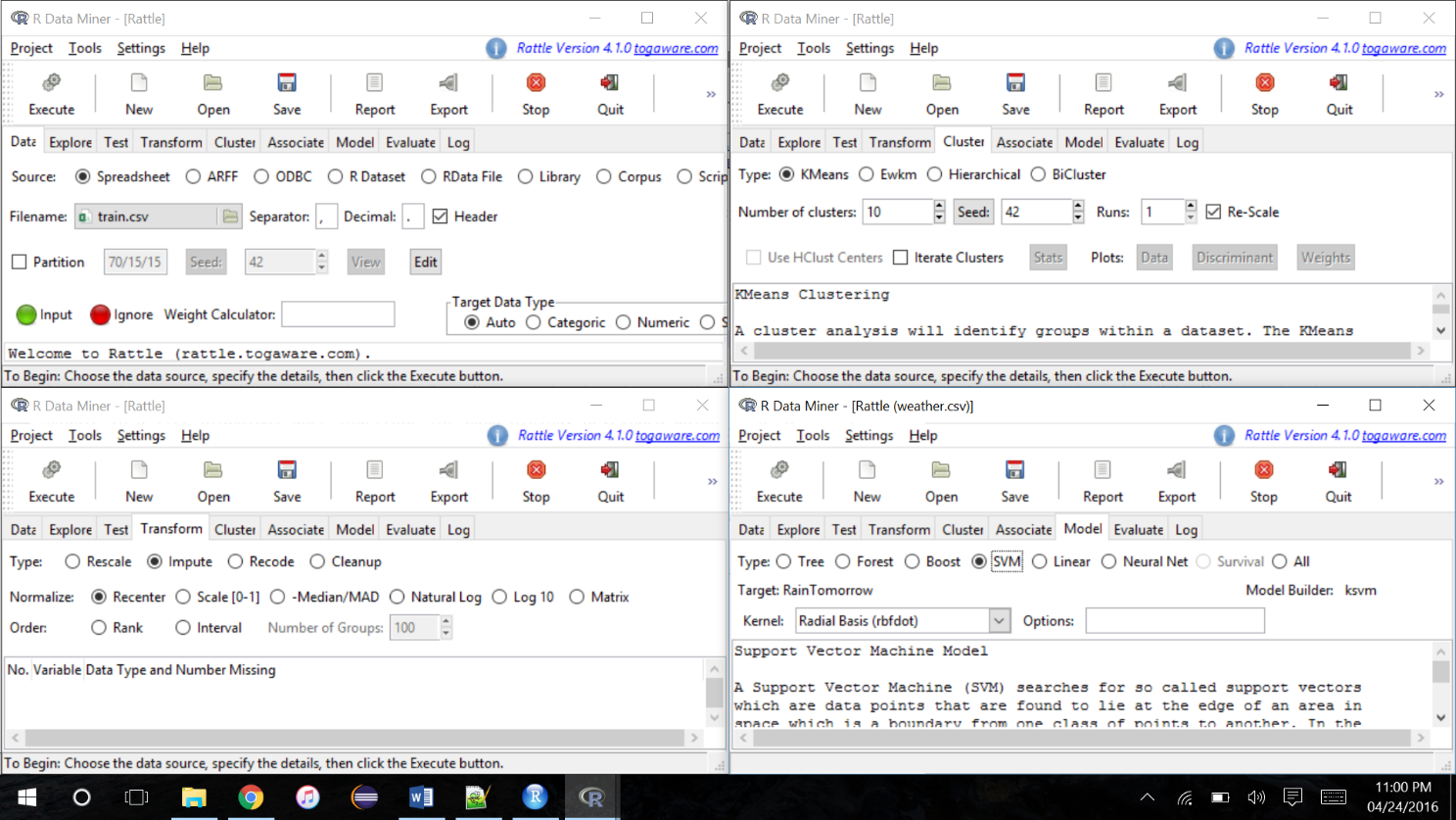
**SVM, also support vector networks** are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

**Random forests** is a notion of the general technique of random decision forests that are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over-fitting to their training set.

**AdaBoost, short for "Adaptive Boosting"**, is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire who won the Gödel Prize in 2003 for their work. It can be used in conjunction with many other types of learning algorithms to improve their performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers. In some problems it can be less susceptible to the over-fitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing (e.g., their error rate is smaller than 0.5 for binary classification), the final model can be proven to converge to a strong learner

1. **Experimental Evaluation**
   1. **Methodology:**

* **Tools Used – R Studio and Rattle.**
* We started building models using above mentioned algorithms at the very initial stages of this project which gave a lesson for giving importance to data imputation. Many a times main focus goes to classifiers, and it should be, but initial stage of this project was really crucial through which gave us an opportunity to understand and learn basics of data imputation.
* As explained above there were many NAs value in the data and they were handled as explained in the following bullets.
* Rattle provides interface based functionality for the data imputation, as explained below.



* The Same objective in R Studio was achieved by writing functional scripts as explained below.
* Age was handled by taking the mean of given ages :-

*age <- train$Age*

*age <- age[!is.na(age)]*

*meanAge <- mean(age)*

*data$Age[is.na(data$Age)] <- meanAge*

* Missing embarked station were filled with median of the given embankment points.

*d <- count(data, 'Embarked')*

*d <- d[ d$freq == max( d$freq ) , ]*

*data$Embarked[is.na(data$Embarked)] <- d$Embarked*

* Missing fare was filled by the median of the fare column.

*f <- count(data, 'Fare')*

*f <- f[ d$freq == max( d$freq ) , ]*

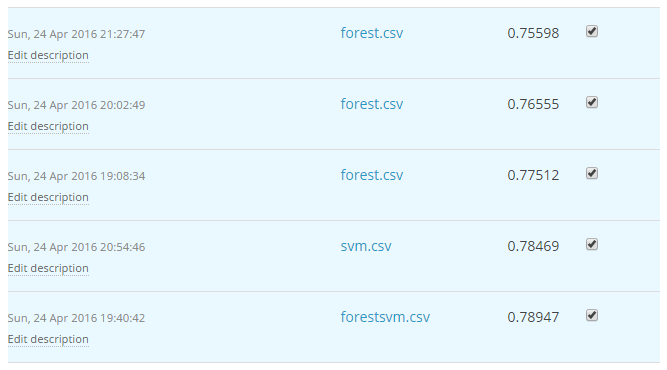
*data$Fare[is.na(data$Fare)] <- d$Fare*

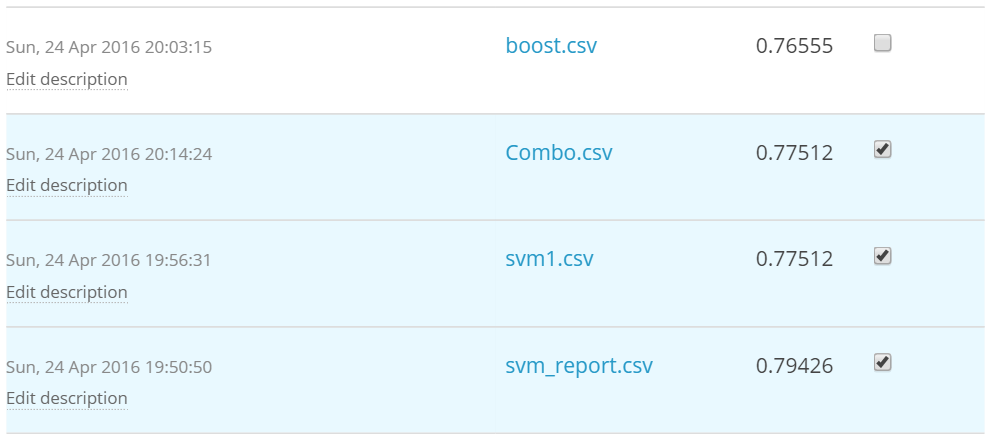
*target <- as.factor(data$Survived)*

* Once the data amputation was done the accuracy increased considerable from 64 to 76 %.
  1. **Results:**

Various classifier results:

|  |  |  |
| --- | --- | --- |
| **Attempt Number** | **Classifier** | **Accuracy (%)** |
| 1 | Random Forest(Without Data Processing) | 64.593 |
| 2 | Random Forest | 77.512 |
| 3 | SVM | 78.947 |
| 4 | AdaBoost | 76.555 |
| 5 | SVM(More tweaking with parameters) | 79.426 |

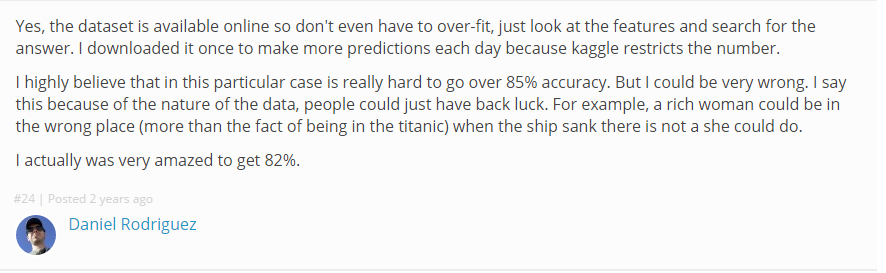




* **We were able to jump from 3500 to 1358 on the leader board among 3928 teams, 4169 Players and 24054 entries.**
* **That is top within top 35% of the leaderboard.**

1. **Related Work :**

* While Some Kagglers used Python and different amputation techniques.
* Some even used different analytical tools like SAS E miner.
* The results of different teams vary in accuracy and while some scores 100 but majority of scores are between 80-85 %, i.e. almost same as our best result.



1. **Future Work:**

* We personally believe that not every solution is perfect and there always exists a scope of improvement. There is a National level Data Mining competition going on hosted by Rang Technologies (<https://rang.shinyapps.io/Competition/>). We have registered for this event and planning to take our project dataset further by participating with this dataset if allowed. This event closes in the end of May’16 and we will be working on this post exam.
* Also, needless to mention that the aim for reaching top 500 on the leader board is always there.

1. **Conclusion:**

* Machine learning is a backbone of data analytics, example like this shows the tremendous scope of ML in real life datasets.
* With even 80% of accuracy also, we could have calculated and do risk mitigation for 1750 odd passengers on such situation.
* On Data Science front, data amputation and analysis of data is the prime importance.
* Selecting an appropriate model is also one of the most important aspect of ML.

**Bibliography:**

* + - 1. Kaggle Forums : <https://www.kaggle.com/c/titanic/forums>
      2. WikiPedia : <https://en.wikipedia.org/wiki/Main_Page>
      3. Inside R Community : <http://www.inside-r.org/>