|  |
| --- |
| EPAM Systems Inc. |
| Survival on the Titanic |
| Statistical Learning Report |
|  |
| **Andrei Kamandzenka** |
| **12.05.2015** |

|  |
| --- |
|  |

## Abstract

This report contains a solution of a test problem by Andrei Kamandzenka - a candidate to fill in a Data Scientist position at EPAM Systems Inc. The goal was to predict survival on the Titanic using 3 different approaches – logistic regression, random forest and support vector machine – in the corresponding Kaggle competition. The resulting predictions from all the three approaches have achieved prediction accuracy of about 80%. This might be considered as an illustration of key importance of data over any learning approach.

## Introduction

The task of the competition is a clear problem of supervised learning for prediction. Thus, the inference from the learning was out of the scope of the present analysis and report. The predictions made by either model were evaluated via submissions to the competition web site in order to estimate prediction accuracy. The submissions scores reflect performance on an unknown half of the test data set. Presumably, this was designed to prevent competitors from using the test set as the validation set, and thus to incentivize emphasis on learning from the training data set only. Unfortunately, it is rather doubtful, that this measure has worked properly, since the unknown half of the test data stays the same from one submission to another. Thus, the intermediate formal standings on the board are to be treated as moderately important.

The present report is composed of four parts: Introduction, Feature Engineering, Statistical Learning, Discussion and Conclusions; followed by a short list of literature.

The survival prediction was performed by 3 different methods – logistic regression, support vector machines and random forests.

### Raw data overview

Raw data is available for download from <https://www.kaggle.com/c/titanic/data> in 2 files: train.csv and test.csv. Each training observation contains 12 entries listed in Table 1. Test observations have all the same entries except for Survived. Figures and conclusions of the present chapter are made with R script in [20141119 - RAW overview.R](Final-1/20141119%20-%20RAW%20overview.R)

Table 1. Raw data entries description

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Name | Meaning | Kind | Values | Note |
|  | PassengerId | Unique passenger identifier | Numeric | Integers starting from 1 | Used for passenger indexing only |
|  | Survived | Survival Status | Binary | 0 = No  1 = Yes | Not supplied for test observations |
|  | Pclass | Passenger Class | Ternary | 1 = 1st  2 = 2nd  3 = 3rd | Pclass is a proxy for socio-economic status (SES): 1st ~ Upper; 2nd ~ Middle; 3rd ~ Lower.  Possibly might be treated as numeric or (ordered?) factor |
|  | Name | Passenger Name | String | Various |  |
|  | Sex | Passenger Sex | Binary | male, female |  |
|  | Age | Passenger Age | Numeric | Various | Age is in Years; Fractional if Age less than One (1). If the Age is Estimated, it is in the form xx.5 |
|  | SibSp | Number of Siblings/Spouses Aboard | Numeric | Non-negativeintegers | With respect to the family relation variables some relations were ignored. The following are the definitions used:  Sibling: Brother, Sister, Stepbrother, or Stepsister of Passenger Aboard Titanic  Spouse: Husband or Wife of Passenger Aboard Titanic (Mistresses and Fiances Ignored)  Parent: Mother or Father of Passenger Aboard Titanic  Child: Son, Daughter, Stepson, or Stepdaughter of Passenger Aboard Titanic  Other family relatives excluded from this study include cousins, nephews/nieces, aunts/uncles, and in-laws. Some children travelled only with a nanny, therefore parch=0 for them. As well, some travelled with very close friends or neighbors in a village, however, the definitions do not support such relations. |
|  | Parch | Number of Parents/Children Aboard | Numeric | Non-negativeintegers |
|  | Ticket | Ticket Number | String | Various | Typically a literal prefix followed by a non-negative integer |
|  | Fare | Passenger Fare | Numeric | Various | A ticket fare rather than a passenger fare |
|  | Cabin | Cabin | String | Various | Very often an empty string |
|  | Embarked | Port of Embarkation | Character | C, Q, S | C = Cherbourg; Q = Queenstown; S = Southampton |

A brief comparison of test and training data is given in Table 2. It shows no big difference in distribution of passengers’ properties between the 2 data sets.

Table 2. Comparison of training and test data

| # | Parameter | Training data set | Test data set |
| --- | --- | --- | --- |
|  | Number of observations | 891 | 418 |
|  | Number of Not Available (i.e. empty) values (non-zero only):   1. Total 2. Age 3. Fare 4. Cabin 5. Embarked | 866  177  0  687  2 | 414  86  1  327  0 |
|  | Share of embarked in   1. Cherbourg 2. Queenstown 3. Southampton | 0.19  0.09  0.72 | 0.24  0.11  0.65 |
|  | Share of passengers by sex   1. Female 2. Male | 0.35  0.65 | 0.36  0.64 |
|  | Share of passengers by class   1. 1st 2. 2nd 3. 3rd | 0.24  0.21  0.55 | 0.26  0.22  0.52 |

Share of passengers by age (among where available) is shown in the histograms below:

|  |  |
| --- | --- |
|  |  |

Figure 1. Distribution of passengers by age

It is noticeable that both data sets have a peak in the same age group of 20-25 years. Although the peak in the test data set is a bit sharper: share of the passengers in their 15-20 and 30-35 years is obviously higher in the training data set. The same is true about little kids of 0-5 years old. Possibly, there were more families with small children in the test data set. The hypothesis is also supported by the SibSp data (also Parch data is less clear about that):

|  |  |
| --- | --- |
|  |  |

Figure 2. Family data comparison

Ticket, Fare and Cabin data appear to be closely related predictors. Firstly, simple analysis of Ticket data shows that some tickets were actually shared by several people: there were 681 unique tickets for 891 passengers from the training dataset and 363 of 418 for the test dataset.

Secondly, all unique tickets had the same fare value (except for the ticket 7534 from the training dataset – there were 2 such tickets with slightly different fares, and except for the ticket 3701 from the test data set where the fare was unknown). Furthermore, fares themselves are less diverse than ticket numbers: 248 unique fares per 891 test observations and 169 per 418 training observations. Finally, fares, although on the average correlate to the passenger class, look very much ­­different for individual tickets regardless of their passenger class:

|  |  |
| --- | --- |
|  |  |

Figure 3. Fares vs. class across data sets

Thirdly, majority of the tickets for which cabins are given in the datasets shared the same cabins – only 14 from the training data set and only 6 from the test data set first class tickets had different cabins specified. Moreover, cabin data is available mostly for the first class passengers’ observations:

|  |  |
| --- | --- |
|  |  |

Figure 4. Cabin data presence

Further analysis of the raw data is infeasible since most of the features might aggregate more information than they seem in the raw data set. Consequently, the next logical step would be to parse features into more “atomic” ones and free them from missing values where possible.

## Feature Engineering

The purpose of the stage is to acquire features effectively utilizable by a learning algorithm. The resulting features have to meet 2 simple criteria: freedom from missing values and representation of atomic properties of their dataset. For the Titanic dataset there are only 2 of 10 original features meeting these criteria – Pclass and Sex. Others either have NAs (Embarked) or contain a mixture of properties (Name, SibSp, Parch, Ticket) or suffer from both deficiencies (Fare, Cabin, Age). Thus, it turns out that some additional features have to be formed both in order to compensate the missing values and to unravel the hidden properties.

In order to create the features some reasoning about them is to be performed, although sometimes it resembles of doing something that can be done rather than something that has to be done. Any preliminary censorship on the features is not justified since the underlying model is not known a priori. It does not imply e.g. decomposition of the original features into separate letters (symbols), but it allows for it even if there is a faintest glimpse of meaning.

Generally feature engineering for the purpose of the analysis was performed from the simple (simply observable) phenomena to the complex ones. In the end the features were pruned in order to make them more statistically meaningful*.*

Due to the availability and boundedness of the complete data set (i.e. there can be no data except for the supplied for the competition) it is wise to combine both train and test data in a single data frame in order to avoid problems with factor variables.

### Simple features

The simplest feature to derive is passenger Family Size given by the sum of Parch and SibSp values plus one (for a passenger him-/herself). It might be descriptive of some aspect of collective behavior of naturally related passengers. Family size allows for a simple split of passengers into those who had family on board (FamilySize>1) and those who had no. This property is stored in the **hasFamily** feature.

Another simple feature is derived from Age. From the raw data overview it is clear that passenger ages were of 3 different “kinds” – known, estimated and unknown (missing). This information extracted to the AgeStatus feature and Ages of the form xx.5 are switched to the form xx, since the age uncertainty is already transferred to the AgeStatus.

The next simple feature is Team Size or Ticket Frequency (**TickFreq**). As it is clear from the caption the feature contains number of passengers holding (travelling by) the same ticket. It might be representative of economic relationships between passengers not captured by their family variables (like Name, Sibsp or Parch).

#### Removing an NA from Fare

At this point the data set is ready to be cleared from a missing Fare value. As long as the person was a lonely 3rd-class passenger embarked in Southampton his Fare is filled in with a median value of similar passengers.

#### Passenger Fare

Since many passengers had been travelling in teams, their personal fares (**PassFare**) can be estimated by dividing ticket fares by ticket frequencies. This feature might possibly be considered as a finer and continuous alternative to passenger class:

|  |  |
| --- | --- |
|  |  |

Figure 5. Fare and Passenger Fare vs. Pclass comparison

On the other hand, it must be noted, that fares per capita of different class passengers still overlap. I.e. there were passengers who paid much less than average passenger of their class. There could be a number of reasons for that – reduced fare prices for children below certain age, different amount of luggage allowance, various on-board services included, ticket reservation fees, typos, etc.

### Less simple features

Further feature engineering is not that straightforward as before. Actually most of the problems are yet unsolved and their solutions might be achieved in multiple ways. In this sections two aggregate features are analyzed – Ticket and Cabin. Both of the features’ entries are strings with complex intrinsic structure.

#### Parsing Ticket

Original Ticket feature is quite rich in information (see the previously defined **TickFreq** and **PassFare**) but is not limited to its frequencies. It turns out that typical Ticket value is combined of a literal prefix and an integer number (or any of the two, usually the latter). It seems useful to split the Ticket variable into to “atomic” features – **ticket prefix** and **ticket number**. Tickets with *LINE* prefix had no number and were assigned zero ticket number.

Ticket prefixes were extracted and considerably reduced in number according to the table below. Such reduction was motivated by presence of obvious/possible typos and a large number of rather rare and statistically insignificant prefixes:

| # | Input | | Output | |
| --- | --- | --- | --- | --- |
| Prefix | Frequency | Prefix | Frequency |
|  | A./5. | 3 | A5 | 29 |
|  | A.5. | 3 |
|  | A/5 | 12 |
|  | A/5. | 10 |
|  | A/S | 1 |
|  | A/4 | 6 | A234 | 13 |
|  | A/4. | 3 |
|  | A4. | 1 |
|  | AQ/4 | 1 |
|  | A. 2. | 1 |
|  | AQ/3. | 1 |
|  | C | 8 | C | 8 |
|  | C.A. | 46 | CA | 69 |
|  | C.A./SOTON | 1 |
|  | CA | 10 |
|  | CA. | 12 |
|  | F.C. | 3 | Fall | 13 |
|  | F.C.C. | 9 |
|  | Fa | 1 |
|  | LINE | 4 | LINE | 4 |
|  | No prefix | 957 | No prefix | 957 |
|  | PC | 92 | PC | 92 |
|  | S.O.C. | 7 | SOC | 8 |
|  | SO/C | 1 |
|  | S.O.P. | 1 | AnyPP | 18 |
|  | S.O./P.P. | 7 |
|  | S.P. | 1 |
|  | S.W./PP | 1 |
|  | SW/PP | 1 |
|  | P/PP | 2 |
|  | PP | 4 |
|  | LP | 1 |
|  | SC | 2 | SCOther | 11 |
|  | SC/A.3 | 1 |
|  | SC/A4 | 1 |
|  | S.C./A.4. | 1 |
|  | SC/AH | 4 |
|  | SC/AH Basle | 1 |
|  | SCO/W | 1 |
|  | SC/Paris | 5 | SCParis | 19 |
|  | SC/PARIS | 11 |
|  | S.C./PARIS | 3 |
|  | SOTON/O.Q. | 16 | SOTON | 49 |
|  | SOTON/O2 | 3 |
|  | SOTON/OQ | 8 |
|  | STON/O 2. | 14 |
|  | STON/O2. | 7 |
|  | STON/OQ. | 1 |
|  | W./C. | 14 | WC | 15 |
|  | W/C | 1 |
|  | WE/P | 2 | WEP | 4 |
|  | W.E.P. | 2 |

This has allowed reducing the number of prefixes from original 52 to only 15. Further reduction is still possible, but it requires some currently not justifiable considerations.

Another property of the ticket numbers is their inhomogeneity. It turns out that, they can be classified into a number of distinct groups as shown in Figure 6. This is encoded in the **TickNumGroup** feature.

|  |
| --- |
|  |

Figure 6. Ticket number grouping

Knowing ticket prefix and number makes possible reconstruction of missing Embarked values. It turned out that ticket 113572 holders likely had embarked in Southampton.

#### Parsing Cabin

Most of the Cabin data is unavailable. On the other hand from the available entries one can extract **number of cabins** per ticket and a cabin **deck**[[1]](#footnote-1). For the unavailable entries number of cabins was set to 0 and deck – to X (not actually present on the ship)[[2]](#footnote-2). Raw distribution of passengers by decks for each passenger class is shown below:

|  |
| --- |
| > table(Combined$Deck[Combined$Pclass == 1])  A B C D E F G T X  22 65 94 40 34 0 0 1 67  > table(Combined$Deck[Combined$Pclass == 2])  A B C D E F G T X  0 0 0 6 4 13 0 0 254  > table(Combined$Deck[Combined$Pclass == 3])  A B C D E F G T X  0 0 0 0 3 8 5 0 693 |

Derivation of the raw deck values is straightforward except for the rare cases when the list of cabins started with a single letter and no number attached to it. This single letter was attributed to as the deck value no matter what the prefixes were in the successive cabin numbers.

Two issues are notable in the raw deck values: single T deck passenger and 67 unknown deck 1st class passengers. For the first case the issue is in the fact that a unique observation is statistically useless (it is impossible neither to learn from nor to make meaningful predictions for it). For the second case the problem is that deck value of X is mostly occupied by the 2nd and the 3rd class passengers. Actually, one can see that none of the 1st class passengers were staying below the Deck E and none of the 2nd and the 3rd class passengers were staying above the Deck D. Since a deck value might be considered as a “geographical” factor of survival chances it can negatively influence prediction accuracy (mostly for the 1st class passengers).

Thus, these 1st class passengers were assigned to the Deck C (as the most popular Deck among 1st class passengers) and the final distribution of passengers over decks is the following:

|  |
| --- |
| > table(Combined$Deck[Combined$Pclass == 1])  A B C D E F G X  22 65 162 40 34 0 0 0  > table(Combined$Deck[Combined$Pclass == 2])  A B C D E F G X  0 0 0 6 4 13 0 254  > table(Combined$Deck[Combined$Pclass == 3])  A B C D E F G X  0 0 0 0 3 8 5 693 |

Another option was to derive the unavailable deck and number of cabins from ticket fares. It would require some additional modelling of fares as a function of other predictors (e.g. Pclass, Age, TickFreq, Embarked and Deck). This option is currently not implemented due to feasibility concerns (cabin data is quite rare for the 2nd and the 3rd class passengers which make chances for successful learning fairly low, while for the 1st class passengers’ distribution over decks had a clear maximum at deck C).

At this stage features Ticket and Cabin become redundant and are removed from the combined data set. All the feature engineering done so far was performed by the R script [20150324 DataPrep.R](Final-1/20150324%20DataPrep.R). Resulting data set and complementary parameters and functions are stored in [CombinedDataPrep.dat](Final-1/CombinedDataPrep.dat)

### Complex features

Up to this point several features remain unattended: Name, Age, SibSp and Parch. Their processing is the main task of the present chapter. These features called complex due to amount of effort required to discover information hidden in them.

#### Name-based features

The original Name values are very rich in data. They hold: Surname, Title, Given Names, Nick Names, Full Maiden Names and Aliases. Illustrative examples of such Name values are shown below:

|  |
| --- |
| > Combined$Name[c(187,557)]  [1] O'Brien, Mrs. Thomas (Johanna "Hannah" Godfrey)  [2] Duff Gordon, Lady. (Lucille Christiana Sutherland) ("Mrs Morgan") |

In the 1st name one can see a typical married woman name containing: husband surname – O’Brien, title – Mrs, husband given name – Thomas, given name – Johanna, nick name – Hannah, maiden surname – Godfrey. In the second name unlike the 1st husband name and nick name are missing, but sort of an alias is present – Mrs Morgan. Names are processed by means of R script in [20150326 Parsing Names Function.R](Final-1/20150326%20Parsing%20Names%20Function.R)

##### Titles and Surnames

At least 2 features are easily extractable from Name values – **Surnames** and **Titles**, as described in (Stephens n.d.). **Surnames** by themselves are hardly related to passengers’ survival, but they allow revealing extended families not identifiable through Parch and SibSp predictors alone. While **Titles** alone are quite significant as they correlate with passenger age, sex and social status which, in turn, might explain passenger behavior. Unfortunately, some of the titles are quite rare and need to be combined together to become statistically meaningful. Original distribution of passengers over titles was the following:

|  |
| --- |
| Capt Col Don Dona Dr  1 4 1 1 8  Jonkheer Lady Major Master Miss  1 1 2 61 260  Mlle Mme Mr Mrs Ms  2 1 757 197 2  Rev Sir the Countess  8 1 1 |

Further, the following mergers were performed:

| # | Input title | Output title |
| --- | --- | --- |
|  | Ms | Miss |
|  | Mlle |
|  | Mme | Mrs |
|  | Capt | Military |
|  | Col |
|  | Major |
|  | Lady | Noble |
|  | Sir |
|  | the Countess |
|  | Jonkheer |
|  | Dona |
|  | Don |

They resulted in the following distribution of passengers over titles:

|  |
| --- |
| Dr Master Military Miss Mr Mrs Noble Rev  8 61 7 264 757 198 6 8 |

##### Other names

Extraction of other names is much less obvious due to variability of their contents – some names are missing, some don’t exist, some contain typos, etc. Firstly, the remaining Name values were checked for presence of parentheses in order to detect and extract maiden names. At this stage missing given names were marked by the string “-empty-”. Then nicknames and aliases denoted by quotes were extracted both from the raw given names and raw maiden names. Finally, a data frame Parsed holding the original Name value, Surname, Title, Given name, Given nick, Maiden name, Maiden nick was formed. Missing values were marked by the string “-empty-” as well.

##### Pruning names

When the names are parsed comes the time to decide which of the names can be meaningful and which – cannot. Firstly, it is useless to retain complete surnames because their last word is practically as unique as the complete surname and much easier to extract from long maiden names. Indeed, there are 875 unique full surnames and 874 unique surnames’ last words. Moreover, even this difference is not essential, since the only affected passenger (#911) had no family on board and was omitted from consideration later on. Further, this approach allows for efficient maiden surname extraction. Later it might simplify for instance discovery of parents and their married children (daughters). Thus, two features are added – **SurnameLW** and **MSurnameLW**.

Secondly, it is hard to expect any meaning from knowing specific nicknames or aliases. On the other hand they are already extracted and simply omitting them seems to be too careless. Thus, a binary flag **hasAnyNick** is added.

Thirdly, given names might be of interest in order to detect married couples (since wives usually officially use husbands’ given names). With given names situation is the opposite of that with surnames. Here the *complete values* are better used in order to simplify spouses matching.

Next, even further pruning of names makes sense. Thus, GivenNames of passengers with titles Master or Miss or with SibSp equal to zero can be ignored. Then – unique GivenNames have to be ignored too. Finally, surnames unique in the joint set of maiden and actual surnames are to be ignored too.

The pruning procedure results in reduction of GivenNames from 954 to 97, of SurnameLW from 874 to 198 and of MSurnameLW from 968 to 206. The corresponding R script is [20150326 Parsing Names Function.R](Final-1/20150326%20Parsing%20Names%20Function.R)

#### Siblings, Spouses, Parents, Children

Motivation for discovery of family relations among passengers is twofold: on one hand it might help to better predict passenger survival, on the other hand it might improve prediction of missing age values. The latter is obvious because natural trams of fertility and social behavior of humans superimpose strict limitations on age differences between children and parents and to some extent between siblings and spouses. Another aspect of motivation is closer to hope than to educated guess.

On the other hand recovery of distinct numbers of siblings and spouses, parents and children is an inverse problem which is likely to be ill-posed. Moreover, among 263 missing Age values only 63 belong to passengers with families aboard, which mean that effect of the reconstruction on age prediction would be rather limited.

In the light of the considerations above the recovery of the family relations (i.e. exact numbers and identities of siblings and spouses, parents and children) looks too expensive (costly) and thus infeasible. All the information available for the reconstruction of the family relations is used while predicting age and survival anyway. Since families are quite off the scope of the present analysis their discovery is rejected.

#### Age

This feature looks much more promising for the prediction of survival. Firstly, due to the famous “women and children first”[[3]](#footnote-3) principle of lifeboat embarkation. Secondly, because of the pattern demonstrated by the available data:

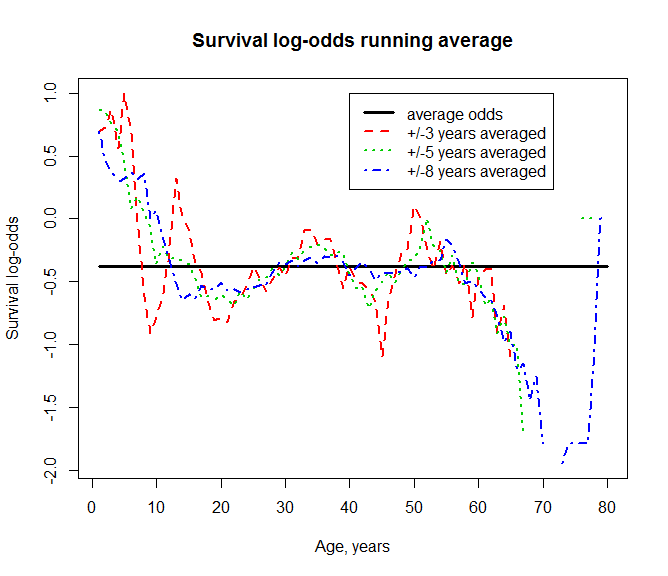


Figure 7. Survival odds vs. Age

The only feasible way to recover missing Age values is to learn them from the available data using some statistical learning method. In this case linear regression was chosen with elasticnet regularization and cross-validation (using glmnet R-package). The corresponding R script is in [20150413 - AgeReconstruction.R](Final-1/20150413%20-%20AgeReconstruction.R)

## Exploratory data analysis

The data set is free from missing values and virtually all meaningful features are extracted and ready to be explored. Previous sections of the report have uncovered some properties of the data but its relation to survival is still obscured (except for the Age – [Figure 7](#_Age)). This section fills in the gap.

The tidy data set contains 1309 observations of 24 fields. One field is used for indexing (PassengerId), one is the response variable (Survived) and the rest 22 variables are the features available for statistical learning. Response is known for the first 891 observations.

Table 3 Survival odds vs. passenger class

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Passenger class | | |
| 1 | 2 | 3 |
| Survival status | 0 | 80 | 97 | 372 |
| 1 | 136 | 87 | 119 |

Table 4 Survival odds vs. passenger sex

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Sex | |
| female | male |
| Survival status | 0 | 81 | 468 |
| 1 | 233 | 109 |

Table 5 Survival odds vs. Age Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Age Status | | |
| Approx. | Exact | Unknown |
| Survival status | 0 | 17 | 407 | 125 |
| 1 | 1 | 289 | 52 |

Table 6 Survival odds vs. SibSp

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Number of Siblings and Spouses | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 8 |
| Survival status | 0 | 398 | 97 | 15 | 12 | 15 | 5 | 7 |
| 1 | 210 | 112 | 13 | 4 | 3 | 0 | 0 |

Table 7 Survival odds vs. Parch

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Number of Parents and Children | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Survival status | 0 | 445 | 53 | 40 | 2 | 4 | 4 | 1 |
| 1 | 233 | 65 | 40 | 3 | 0 | 1 | 0 |

Table 8 Survival odds vs. family size

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Family Size | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 11 |
| Survival status | 0 | 374 | 72 | 43 | 8 | 12 | 19 | 8 | 6 | 7 |
| 1 | 163 | 89 | 59 | 21 | 3 | 3 | 4 | 0 | 0 |

Table 9 Survival odds vs. Title

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Title | | | | | | | |
| Dr | Master | Military | Miss | Mr | Mrs | Noble | Rev |
| Survival status | 0 | 4 | 17 | 3 | 55 | 436 | 26 | 2 | 6 |
| 1 | 3 | 23 | 2 | 130 | 81 | 100 | 3 | 0 |

The tidy data set contains 1309 observations of 24 fields. One field is used for indexing (PassengerId), one is the response variable (Survived) and the rest 22 variables are the features available for statistical learning. Response is known for the first 891 observations.

Table 10 Survival odds vs. presence of a nick name

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Nick name presence | |
| 0 | 1 |
| Survival status | 0 | 534 | 15 |
| 1 | 304 | 308 |

Table 11 Survival odds vs. presence of a family

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Family presence | |
| 0 | 1 |
| Survival status | 0 | 374 | 175 |
| 1 | 163 | 179 |

Table 12 Survival odds vs. ticket frequency

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Ticket frequency | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 11 |
| Survival status | 0 | 351 | 88 | 35 | 12 | 14 | 15 | 19 | 8 | 7 |
| 1 | 130 | 93 | 66 | 32 | 7 | 4 | 5 | 5 | 0 |

Table 13 Survival odds vs. ticket number group

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Ticket number group | | | | |
| 1 | 2 | 3 | 4 | 5 |
| Survival status | 0 | 229 | 35 | 46 | 207 | 32 |
| 1 | 195 | 37 | 42 | 56 | 12 |

|  |  |
| --- | --- |
|  |  |

Figure 8. Survival odds as a function of travel cost

Table 14 Survival odds vs. Deck

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Deck | | | | | | | |
| A | B | C | D | E | F | G | X |
| Survival status | 0 | 8 | 12 | 46 | 8 | 8 | 5 | 2 | 460 |
| 1 | 7 | 35 | 54 | 25 | 24 | 8 | 2 | 187 |

Table 15 Survival odds vs. number of cabins

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Number of cabins | | | | |
| 0 | 1 | 2 | 3 | 4 |
| Survival status | 0 | 481 | 61 | 4 | 3 | 0 |
| 1 | 206 | 123 | 8 | 3 | 2 |

Table 16 Survival odds vs. Embarked

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Embarked | | |
| C | Q | S |
| Survival status | 0 | 75 | 47 | 427 |
| 1 | 93 | 30 | 219 |

According to the tables and plots the new features capture various aspects of passenger records. These aspects are quite sensitive to survival status and create a solid basis for statistical learning. R-script performing the analysis is [20150512 - EDA.R](Final-1/20150512%20-%20EDA.R)

## Statistical Learning

### Logistic Regression

Logistic regression was fit using glmnet R-package with elasticnet regularization and cross-validation. Best implementation is contained in [20150416 - LogRegSolution - 0.79426.R](Final-1/20150416%20-%20LogRegSolution%20-%200.79426.R). Highest score achieved – 79.4%. Best model is a simple linear model of the form:

Any interactions and polynomial Age have not helped to improve performance. Submissions with such performance stand in positions from 485-655 out of 2590 teams.

It has to be noted, that this Kaggle competition allows for some sort of cheating. Thus, doubling penalty weight factor chosen during cross-validation on the training set leads to a noticeable improvement in performance yielding 80.9% accuracy (i.e. improving prediction by 7 people out of 418 with the unknown survival status). In other words, when the score is the goal, it makes sense to use validation approach with validation data being the withheld competition data.

### Random Forest

Random forest solution was developed using randomForest and doParallel R-packages. Best implementation is contained in [20150428 - RandomForestSolutionRF - 0.78947\_2.R](Final-1/20150428%20-%20RandomForestSolutionRF%20-%200.78947_2.R). Highest score achieved – 78.9%. Algorithm steps are:

1. Split the complete training data to train and test subsets
2. Use OOB error rate to optimize number of features used during each tree split
3. Optimize voting scheme to symmetrize error rates
4. Grow the forest using all the training data
5. Make prediction using the optimized voting scheme.

Again, like with logistic regression, some cheating/tuning is possible by using the withheld data as validation set. Another observation is random forest specific – the class predictions are skewed (see Table 3) unlike e.g. logistic regression. This skewness actually has motivated voting scheme optimization in order to get the model less prone to predict passenger death.

Table 17 Random Forest confusion matrix

|  |  |  |  |
| --- | --- | --- | --- |
|  | Confusion matrix: | |  |
|  | 0 | 1 | classification error |
| 0 | 488 | 61 | 0.1111111 |
| 1 | 76 | 266 | 0.2222222 |

### Support Vector Machine

In this section only one implementation (best performing) is covered, although practically all kernels available in e1071 R-package were tried out. The best performing kernel appeared to be radial (see [20150503 - SVM\_radker.R](Final-1/20150503%20-%20SVM_radker.R) for implementation details). The highest score achieved with radial kernel so far is 80.9% like in the logistic regression implementation. doParallel R-package was used too in order to speed-up model selection. The best performing model is given by the following formula:

Possible cheating/tuning opportunities were not attempted with the classifier due to quite a big number of attempts required to perform grid-search of best model/parameters. Nevertheless, SVM has maintained its status of “one of the best “out of the box” classifiers” (ISLR 2013) by achieving the highest score with the least effort compared to the other two methods.

## Discussion and Conclusions

The analysis has shown that any of the three attempted approaches yields similar performance in predicting survival of Titanic passengers. This in turn means that in this case not the approaches but the data governs the success. Since none of the approaches has managed to outperform its rivals then it must has been prevented by the data supplied to the learning algorithms.

Speaking of data two aspects are to be kept in mind: the features themselves and their interactions. Unfortunately, in the analysis covered by the report it has been sort of impossible to find any fruitful interactions except for those concealed by feature engineering process in the tidy data. Indeed, fare is a product of passenger fare and ticket frequency; family size – sum of all family members on board; name-based features were pruned based on values of other predictors. 20% of all age values were actually predicted by virtually all other features. Thus, interactions were actually present and were effectively utilized by the learning algorithms as it can be seen from the model formulae. Then, it must have been something done to and/or with features. Among the unattended opportunities one might think of using fares to identify kids in order to better predict their age; recovery of passenger family roles based on their Sibsp and Parch values; clustering passengers into groups by e.g. their tickets in order to grasp their interconnections outside their families. Possibly something else not listed here could have been done in order to improve prediction accuracy. On the other hand, from the competition leaderboard follows that there is not so many steps to the top. Actually – a lot teams struggle at the same level, about the same number – approximately 1% above and quite rare birds fly around 90%. Honesty of 100% accuracy in the leaderboard is extremely doubtful.

A separate mentioning has to be made about feature engineering. It turned out to be the most challenging and probably the most essential part of the complete analysis. Strange enough is that it seems to be quite weakly covered in the textbooks and lectures used to perform the analysis covered by the report.

Instead of conclusion several widely known statements seem to be very appropriate:

1. There is no free lunch in statistics (James G 2013)
2. Data is the key[[4]](#footnote-4)
3. It’s hard to make predictions, especially about the future[[5]](#footnote-5)

None of the approaches prevail it terms of prediction accuracy yielding about 80%. From the standpoint of ease of application SVM appear to be most convenient.

## Literature

James G, Witten D, Hastie T, Tibshirani R. *An Introduction to Statistical Learning.* New York: Springer Science+Business Media, 2013.

Stephens, Trevor. *Titanic: Getting Started With R - Part 4: Feature Engineering.* n.d. http://trevorstephens.com/post/73461351896/titanic-getting-started-with-r-part-4-feature (accessed Jan 11, 2015).

1. Most of the cabin values also included cabin numbers which seem to be the unnecessary detail since they provide additional information only to the rare tickets with the cabins specified thus splitting them into finer and probably less statistically significant and sensible/detectable groups. [↑](#footnote-ref-1)
2. For the ship design details consider <http://www.encyclopedia-titanica.org/titanic-deckplans/> [↑](#footnote-ref-2)
3. <http://en.wikipedia.org/wiki/Women_and_children_first#See_also> [↑](#footnote-ref-3)
4. These words probably do not belong but surely might have been said by Andrew Ng in his Machine Learning class at Coursera. [↑](#footnote-ref-4)
5. <http://en.wikiquote.org/wiki/Niels_Bohr> (disputable) [↑](#footnote-ref-5)