**Titanic Data Challenge**

Table of Contents

[Problem Statement: 2](#_Toc496551456)

[Data Dictionary 2](#_Toc496551457)

[Languages Used: 3](#_Toc496551458)

[Data Exploration: 3](#_Toc496551459)

[Feature Selection and Feature Engineering: 4](#_Toc496551460)

[Removing Features 4](#_Toc496551461)

[Recoding Features: 4](#_Toc496551462)

[Combining Features: 8](#_Toc496551469)

[Synthetic Features: 8](#_Toc496551470)

[Feature Transformations: 8](#_Toc496551471)

[String Indexer: 8](#_Toc496551472)

[Word2Vec: 8](#_Toc496551473)

[Approach: 9](#_Toc496551474)

[Conclusion: 10](#_Toc496551483)

[Acknowledgements: 10](#_Toc496551484)

# Problem Statement:

The sinking of the RMS Titanic is one of the most infamous shipwrecks in history. The objective is to complete the analysis of what sorts of people were likely to survive. In particular, we are asked to apply the tools of machine learning to predict which passengers survived the tragedy.

# Data Dictionary

|  |  |  |
| --- | --- | --- |
| Variable | Definition | Key |
| survival | Survival | 0 = No, 1 = Yes |
| pclass | Ticket class | 1 = 1st, 2 = 2nd, 3 = 3rd |
| sex | Sex |  |
| Age | Age in years |  |
| sibsp | # of siblings / spouses aboard the Titanic |  |
| parch | # of parents / children aboard the Titanic |  |
| ticket | Ticket number |  |
| fare | Passenger fare |  |
| cabin | Cabin number |  |
| embarked | Port of Embarkation | C = Cherbourg, Q = Queenstown, S = Southampton |

# Languages Used:

* Python (actual code written in python using PY Spark API in Jupyter Notebook for Big Data Analysis and also for data exploration on the fly)
* R (for basic data exploration and structure analysis for report)

# Data Exploration:

The basic description of data indicates that there null and missing values. Most Importantly Age and Fare in train and test data

> summary(train$Age)

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

0.42 20.12 28.00 29.70 38.00 80.00 177

The summary statistics of Age column show that mean and median are almost the same. So we fill in the missing values with mean value for both train and test.

> summary(test$Fare)

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

0.000 7.896 14.454 35.627 31.500 512.329 1

The summary statistics of Fare column show that mean and median are different. So we fill in the missing values with median value for both train and test to prevent the effect of outliers on the data.

# Feature Selection and Feature Engineering:

The approach taken consisted of the following:

1)Removing Features

2)Combining Features

3)Recoding features

4)Generating Synthetic features

## Removing Features

We removed irrelevant features like “passengerId” which generally play no role in learning task.

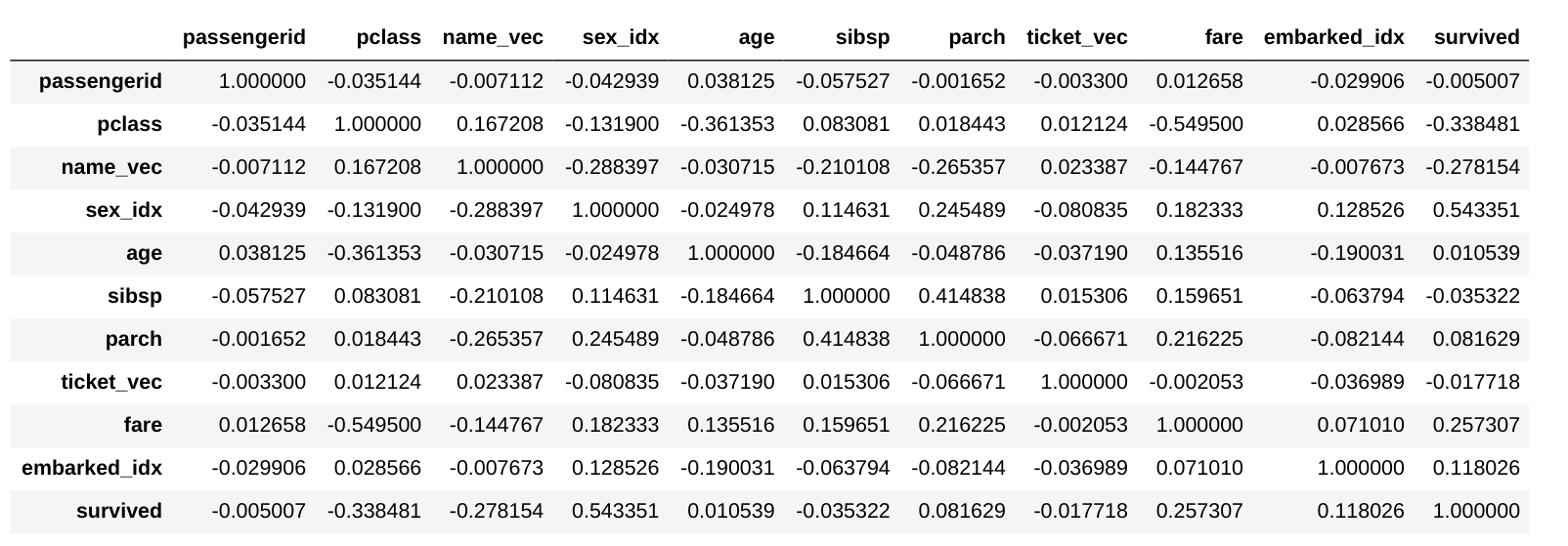
Further features like “cabin” are considered doubtful because of large number of missing values in our data.So in our initial feature selection for training the model we discard it.

We later try to recode cabin feature to extract useful information for it.

## Recoding Features:

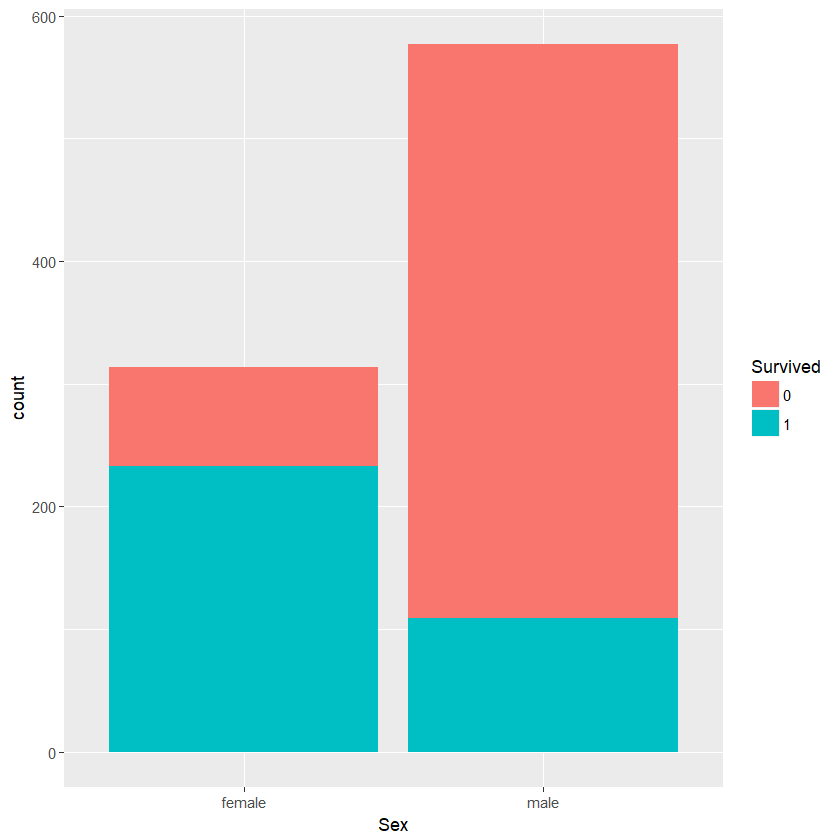
We can recode certain features based on the distribution of the data which can make our classification task much easier.

We take the correlation matrix of train data to give us some more insights on variables. For convenience let us convert all string variables like name and ticket to word2vec of one dimension .



1. The “sex” has the highest correlation with “survived”. To verify the importance we also plotted it. It is already a simple categorical value. So we can deal with it by simply indexing. Same can be said with respect to pclass.

>qplot(Sex,data = train,fill = Survived)



As seen from the plot the the percentage of females who survived is much more than the males. This is an important feature to learn.

1. The next variable is “name\_vec” which shows that name column is significant contributor to explaining variance and can be used to extract information. We could split the huge name into parts. By general intuition on western names this can be split

* Person name (pname variable)
* Title
* Family name (fname variable)

Once the data is extracted using user defined functions and some reg expressions , we can create three columns. We know that by common sense person name is definitely not going to be significant factor in learning a generalized model. The next is “Title”.

Title can provide useful information as Mr and Miss may also mean a kind of gender label. To make sure words of similar semantics are present at same location in dimensional space., Word2Vec is an important feature to use to represent it. This will save us lot of time in figuring out words of same meaning and assigning them a new label. Long story cut short! For example: “captain” and “ Major” mean the same and word2 vec is useful in making sure they are near to each other. Thus we create “title\_vec”.

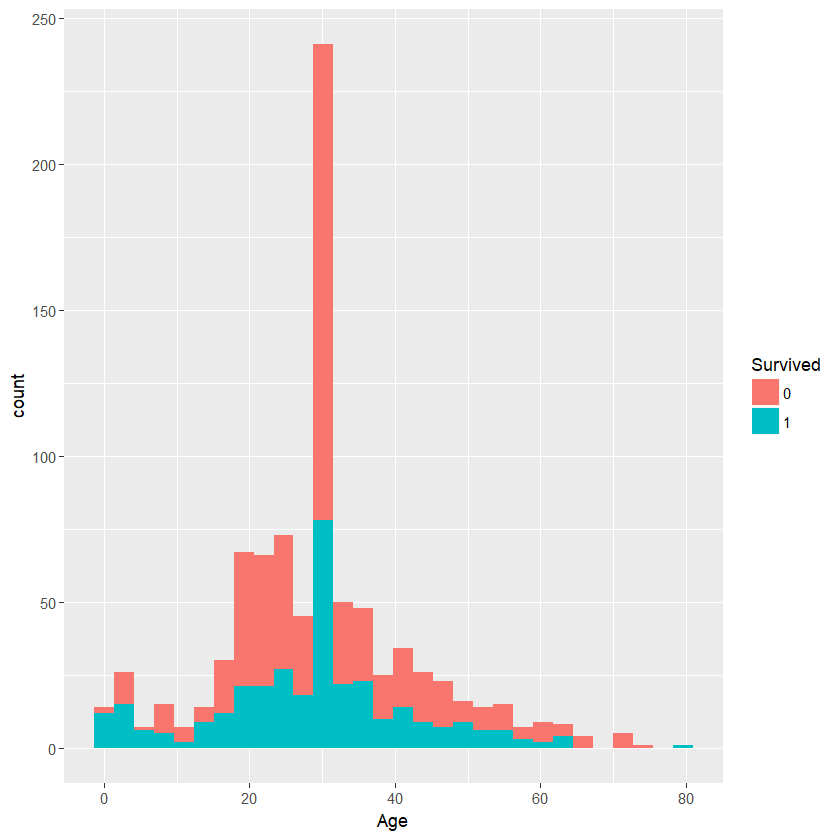
1. The next is “Fare” which has good correlation. To extract further information from it we could make use of the Ticket data. If we group the data by ticket and extract counts we we can see that there are more than one ticket of same kind all given single fare. We could use it to our advantage by extracting the average fare for each person by dividing the fare by count of distinct ticket. This gives actual price paid by each person. Previously some of them had group fares associated with them. Thus a new column “fare\_avg” is created

Then let us focus on insignificant features from correlation matrix:

1. Sibsp and parch are already indexed in data
2. “Age” has low correlation with response variable. We now plot it to see if there is any useful information we can extract.

> ggplot(train, aes(Age,fill =Survived)) + geom\_histogram()

Plot in next page



From the plot we can clearly observe that high percentage of people in (0-7) age group survived. People in middle age group die in greater proportion. And old age people also have good percentage survival rates. Thus it makes sense to divide the age into 3 to 4 groups. Here we divide age into four categories. Creating new feature “age\_category”

* child = 1 adolescent = 2 adult = 3 old =4

1. “cabin” feature has a lot of missing values .To extract useful information from it we divide the cabin info into two parts

* Cabin\_level : indicating the level of the cabin by extracting the characters
* Cabin\_interval: This was complicated as there were multiple room numbers in the same entry . The saving grace was that these numbers were either even or odd. Taking an average or max would have lead to some loss of information. To encode all the information about all the rooms we defined a new interval for even and odd rooms in interval range of 10’s.

For Example: If the room numbers were 23 25 and 27 we defined it as interval 21. Where 2 in ten’s location indicates the room numbers are between 20 and 29. The 1 in one’s location indicates the room numbers are odd. Similary , 22 24 26 are given interval 22 where 2 in ones’ location indicates they are even numbered. 0 value was given to missing cabin room numbers .

## Combining Features:

\*based on inspiration from trevor stephens blog on titanic [1] .

Combined the features “sibs” and “parch” to create a “family\_size” feature. But it did not prove to be useful in the final analysis.

## Synthetic Features:

Generated the following synthetic\_features based on correlation values.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Correlation |  |
| Synthetic1 | sex\_vec\*title\_vec | 0.52 |  |
| Synthetic2 | pclass\*fare\_avg | -0.65 |  |
| synthetic3 | family\_size\*age\_category | 0.366 |  |

# Feature Transformations:

The following are the basic transformations used :

String Indexer:

To index categorical values .

For example: “sex” and “embarked”.

## Word2Vec:

To give numerical values to string and to preserve their semantic similarities.

For Example: “pname”, “title”, “fname” and “sex\_vec”

Note: sex\_vec was created to explore the intrinsic relation between sex and title as a synthetic feature.

# Approach:

Lab\_Notebook\_experiments excel sheet submitted with this document is an important source of Info on the approach taken. We only summarize that here .

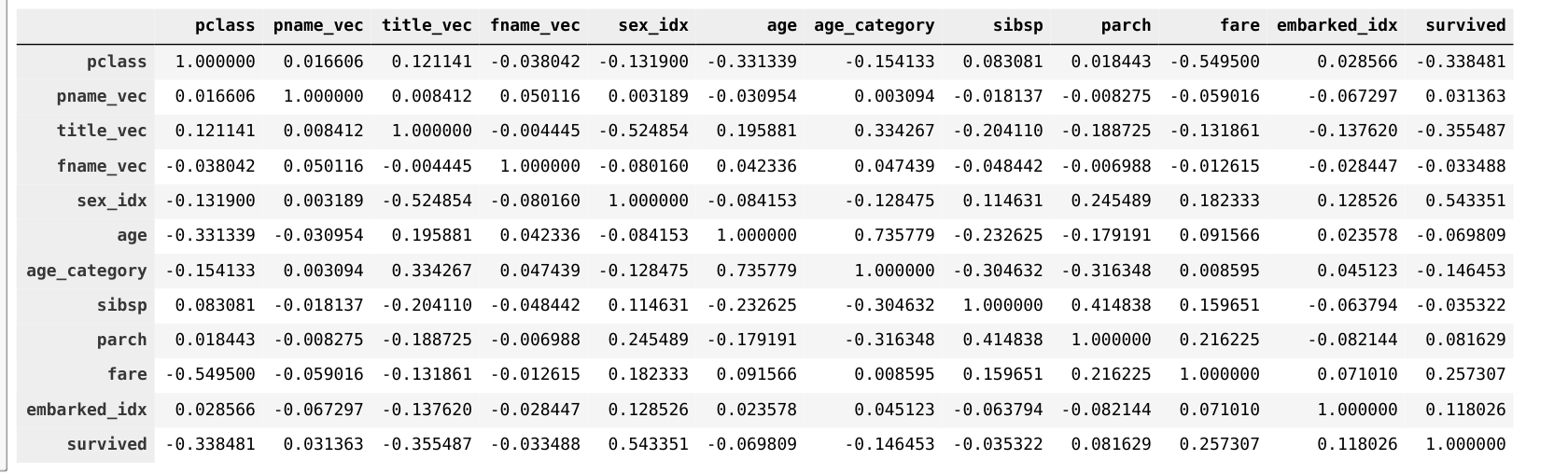
1. Initially did not do any extensive feature engineering. Just plugged in the missing values and transformed non numerical values to numerical values such as sex, embarked and name.Used word to vec with 3 dimension for name . The result was 0.779.
2. Later used only one dimensional vector for word2vec for an increase to 0.79.

Note : In both these cases did not use cabin feature

1. Later introduced the cabin feature which lead to a fall in the score probably because of introduction of noise

All this did not throw any light on the way they are behaving. The solution was Correlation Matrix. The step 1 and 2 in hindsight were useful to generate this Matrix.

1. Later with help of correlation matrix took a step wise approach @ feature selection and engineering as disclosed in Feature selection and Engineering heading.
2. After that generated a new correlation matrix as shown below:



6. From this correlation matrix chose the eight variables which had the highest correlation

with output variable. (pclass, title\_vec, sex\_idx,age\_category,sibs,parch,fare,embarked).

This improved the accuracy to 0.80861.

7. Later continued with the experiments of adding and deleting features but did not

improve accuracy

8. Used an extensive Param Grid builder for Random forest with only above features but it

resulted in lower score of 0.78 as it essentially overfitted the data. The code for builder is

there in jupyter notebook but later only stuck to default random forest with depth of 4.

# Conclusion:

In conclusion, a structured approach of using basic data exploration proved to be the key in improving the accuracy. Plots combined with correlation matrices provided the key to approaching the problem. The features like sex, pclass were an automatic selection based on correlation. The age\_category was the key to improve the accuracy further. This makes ample sense during disaster as sex(women) and age(children) are given priority for evacuation. Some features like fare\_avg even though were intuitive did not provide visible gains. The same is true with the synthetic features..