Machine Learning from Disaster

Titanic Survivor Prediction

Yusi Liu
School of Information and Computing
University of Pittsburgh
yul192@pitt.edu

Yunhao Wu
School of Information and Computing
University of Pittsburgh
yuw122@pitt.edu

ABSTRACT

In this project, we try to complete the analysis of what sorts of people were likely to survive in Titanic by learning the training set using Logistic regression, SVM, K-neighbors, Decision tree, Random forest, Gradient Boosting Decision Tree & xgbGBDT model. And then ensemble the better ones to do prediction on the test set to try Ensemble Learning.

First, we analysis the data and do the feature selection based on the analysis result. We introduced two new features exacted from the original data and apply our findings in the baseline model. The result shows our features improve the model accuracy.

KEYWORDS

Titanic, disaster, machine learning, logistic regression, ensemble 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 [1].

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.

In this project, we are going to complete the analysis of what sorts of people were likely to survive. In particular, we want to apply the tools of machine learning to predict which passengers survived the tragedy.

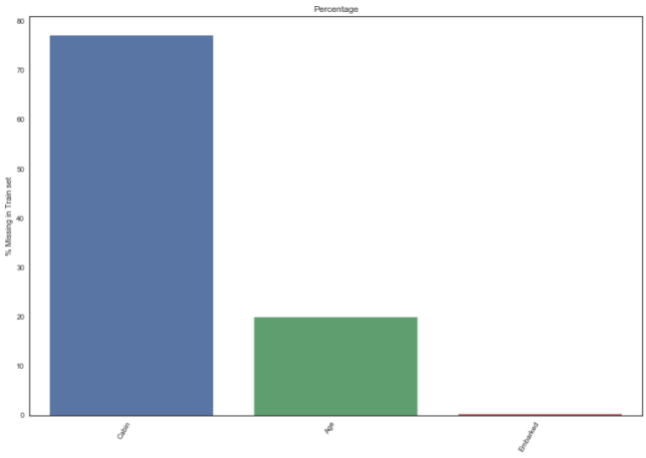
1.1 Data Introduction

This dataset has 11 attributes and 1 target, which includes PassengerId, Pclass, Name, Sex, Age, Sibsp, Parch, Fare, Cabin, Ticket, Embarked and Survived. And the target for classification is Survived.

Index	Variable	Definition	Data type	Key
Target	survival	Survival	int	0 = No, 1 = Yes
1	PassengerId		int	
2	pclass	Ticket class	int	1 = 1st,2 = 2nd, 3 = 3rd
3	Name		str	
4	sex	Sex	str	
5	Age	Age in years	float	
6	sibsp	# of siblings / spouses aboard the Titanic	int	
7	parch	# of parents / children aboard the Titanic	int	
8	Fare	Passenger fare	float	
9	cabin	Cabin number	str	
10	embarked	Port of Embarkation	str	C = Cherbourg, Q = Queenstown, S = Southampton
11	Ticket	Ticket number	str	

Figure 1: Overview of the data

The data has been split into two groups: Train set (train.csv) & Test set (test.csv).



The training dataset has 891 entries, 12 columns, among which there are some data missing in attribute Age (714 entries, 177 missing), Cabin (204 entries, 687 missing) and Embarked (889

Figure 2: Missing percentage in train set

entries, 2 missing).

The test set has 418 entries, 12 columns, also containing missing data in attribute Age (332 entries, 86 missing), Fare (417 entries, 1 missing) and Cabin (91 entries, 327 missing).

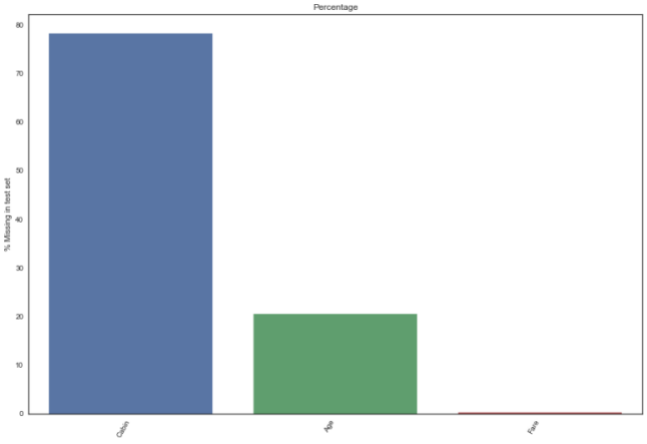


Figure 3: Missing percentage in test set

2 Related Work

Many works have been done before and many models has been adopted. Daniel Manuelpillai[2] used K-Neighbors-Classifier and achieved 65% accuracy on train set. Sagar Jain[3] adopted the sequential model from keras and achieved a 70% accuracy. A single random forest method[4] got a result of over 77%. But these single model methods are clearly not the best choices.

Typically, people are using boosting / bagging algorithm to ensemble several models in order to get a higher accuracy. Some favorable choices for ensemble model are GDBT, random forest and SVM[5]. We are also trying to adopt an ensemble model to improve the accuracy of the prediction.

3 Methodology

3.1 Idea Overview

We first observe all the attributes, trying to figure out their relations to survival (target). Attributes with a strong relation to survival will be kept for later training after potential preprocessing (dealing with missing values, etc.); otherwise they will be abandoned.

Then we use feature engineering techniques to construct new features from existing features to help our training.

We will use basic Logistic Regression model as baseline. And Further, we adopted Logistic regression, SVM[7], K-neighbors, Decision tree, Random forest[6], Gradient Boosting Decision Tree[9] & xgbGBDT[10] model. And then ensemble the better ones to do prediction on the test set to try Ensemble Learning.

3.2 Methodology

3.2.1 Logistic Regression. Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution. Thus, it treats the same set of problems as probit regression using similar techniques, with the latter using a cumulative normal distribution curve instead.

Equivalently, in the latent variable interpretations of these two methods, the first assumes a standard logistic distribution of errors and the second a standard normal distribution of errors. [11]

3.2.2 SVM. In machine learning, support vector machines (SVMs) 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 as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). 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.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data are not labeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The support vector clustering [12] algorithm created by Hava Siegelmann and Vladimir Vapnik, applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data, and is one of the most widely used clustering algorithms in industrial applications

3.2.3 K-neighbors. In pattern recognition, the k-nearest neighbor algorithm (k-NN) is a non-parametric method used for classification and regression. [13] In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression:

In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k

nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors.

NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally, and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.

Both for classification and regression, a useful technique can be to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor.

The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

A peculiarity of the k-NN algorithm is that it is sensitive to the local structure of the data. The algorithm is not to be confused with k-means, another popular machine learning technique.

3.2.4 Decision Tree. Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modelling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

3.2.5 Random Forest. Random forests or random decision forests 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. [14] Random decision forests correct for decision trees' habit of overfitting to their training set.

The first algorithm for random decision forests was created by Tin Kam Ho [14] using the random subspace method, [15] which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, and "Random Forests" is their trademark. The extension combines Breiman's "bagging" idea and random selection of features, introduced first by Ho [14] and later independently by Amit and Geman in order to construct a collection of decision trees with controlled variance

3.2.6 Gradient Boosting Decision Tree. Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

The idea of gradient boosting originated in the observation by Leo Breiman[16] that boosting can be interpreted as an optimization algorithm on a suitable cost function. Explicit regression gradient boosting algorithms were subsequently developed by Jerome H. Friedman [17] simultaneously with the more general functional gradient boosting perspective of Llew Mason, Jonathan Baxter, Peter Bartlett and Marcus Frean. The latter two papers introduced the view of boosting algorithms as iterative functional gradient descent algorithms. That is, algorithms that optimize a cost function over function space by iteratively choosing a function (weak hypothesis) that points in the negative gradient direction. This functional gradient view of boosting has led to the development of boosting algorithms in many areas of machine learning and statistics beyond regression and classification.

3.2.7 XGB-GBDT. XGBoost[18] is an open-source software library which provides the gradient boosting framework for C++, Java, Python, R, and Julia. It works on Linux, Windows, and macOS.From the project description, it aims to provide a "Scalable, Portable and Distributed Gradient Boosting (GBM, GBRT, GBDT) Library". Other than running on a single machine, it also supports the distributed processing frameworks Apache Hadoop, Apache Spark, and Apache Flink. It has gained much popularity and attention recently as it was the algorithm of choice for many winning teams of a number of machine learning competitions.

4 Experimental Results

4.1 Data Overview

4.1.1 Train Set. We describe training set data as follows.

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

Figure 4: Training set description

From this table, we can see that the passengerID is an index, which has little connection to the final survival result. The

average age is younger than what we expected. Most passengers are young adults. And in this accident, most young adults sacrificed their young life to save women and children.

There are some data missing in attribute Age (714 entries, 177 missing), Cabin (204 entries, 687 missing) and Embarked (889 entries, 2 missing). To make the other information clearer, we draw the Correlation matrix.

Figure 5: Correlation matrix of training data

From this matrix, we find something interesting which can be used in the feature selection.

1. The PClass (especially the ticket class) and the Survived are negatively related, which means the more expensive cabin has higher survival rate.
2. The Sex and the Survived are negatively related, which means the women has higher survival rate
3. The Fare and the Survived are positively related, which provides another evidence of finding 1.
4. The PClass and the Fare are negatively related, which means the higher-class cabin has higher value.

Keeping this overview in mind, next we try to explore more about the feature.

4.1.2 Test Set. We describe test set data as follows.

	PassengerId	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	332.000000	418.000000	418.000000	417.000000
mean	1100.500000	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.841838	14.181209	0.896760	0.981429	55.907576
min	892.000000	1.000000	0.170000	0.000000	0.000000	0.000000
max	1309.000000	3.000000	76.000000	8.000000	9.000000	512.329200

Figure 6: Test set description

There are some data missing in attribute Age (332 entries, 86 missing), Fare (417 entries, 1 missing) and Cabin (91 entries, 327 missing).

4.2 Attributes Observation

4.2.1 Age. Due to the missing value in Age, we use (-20) fill in the null values. Then, do age distribution and age survival distribution as follows

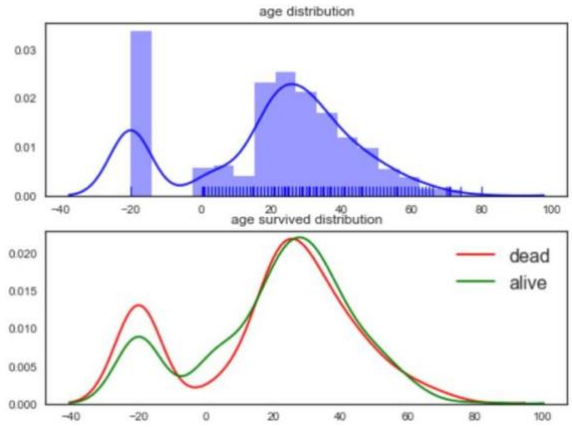


Figure 7: Age distribution and Age-survived distribution

From this figure, we can find that:

1. Regardless of whether they are rescued or not, Age is widely distributed. It is easier for children and middle-aged people to be rescued.
2. Age and survived are not linear. If we use a linear model, this feature may need to be processed discretely and then substituted into the model as a categorical variable.
3. Among the people rescued, the age is less by default.

Then we do Sex-Age distribution to find more.

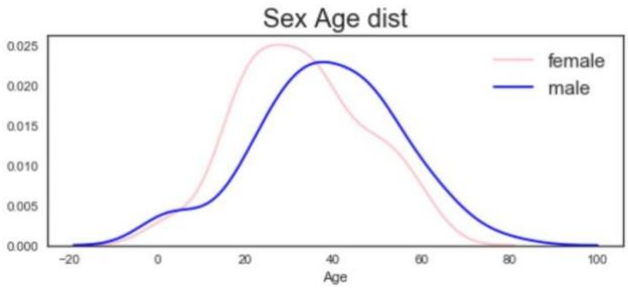


Figure 8: Sex-Age distribution

From the figure we can see that there are more older men and younger women. Boys are more than girls among children.

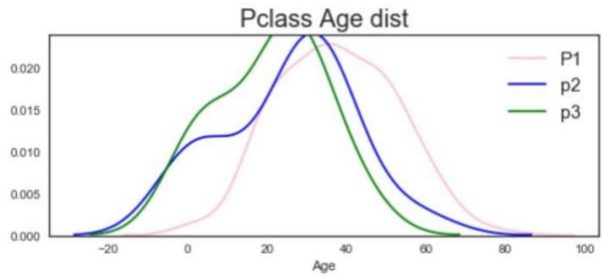


Figure 9: Cabin class and Age distribution

We next analyze the Cabin class and Age. From the figure above, we can see that the higher the Cabin class, the older the age.

4.2.2 Pclass.

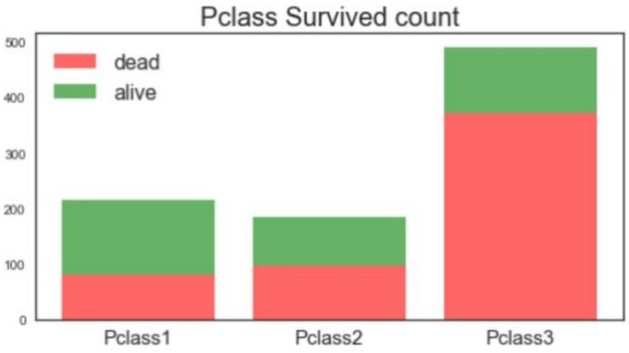


Figure 10: Pclass survived count

Comparison of First Class (Pclass=1), Business Class (Pclass=2), Economy Class (Pclass=3):

1. As expected, the number of economic cabins is far ahead.
2. From the perspective of the percentage of rescued persons, the percentage of rescued first- class cabins is very high, and the proportion of deaths in economic class is quite alarming.

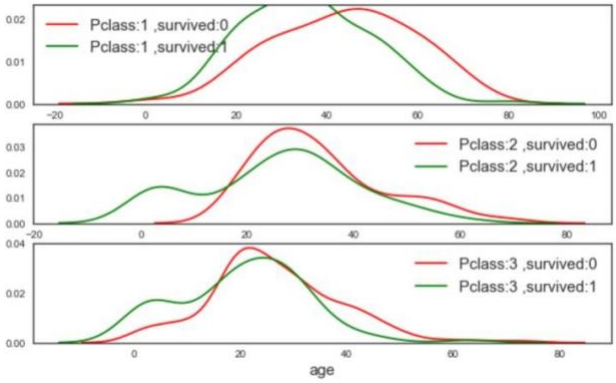


Figure 11: Pclass and Age distribution

Observation results:

1. First-class survivals are younger.
2. Business Class kids are taken care of very well.
3. The child in the economy cabins is also rescued more.

4.2.3 Sex. From the training data, we can see that there are 577 male and 314 females. Nearly 75% (74.2038%) females survived while there are only 18.89% male survived.

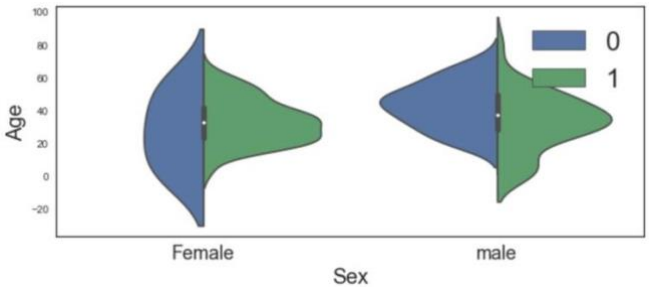


Figure 12: Age-Sex violin map

From this figure we can see that:

1. Among women, the rescued people are concentrated in the middle age;
2. Young male, especially boys, are more likely to be saved.
3. It seems that young and middle-aged male had lower survival rate.

Then we divide the training data by Sex and Pclass, and then count the survival number

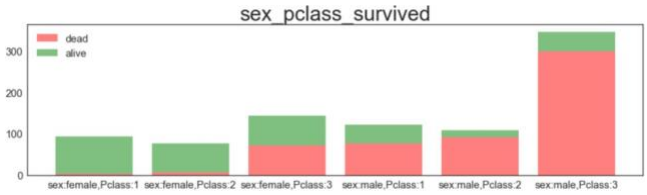


Figure 13: Sex-Pclass-Survived count

From the figure we can see that:

4. In general, women are significantly more likely to be rescued.
5. Under the same gender, the lower the cabin class, the higher the probability of survival.

4.2.4 Fare.

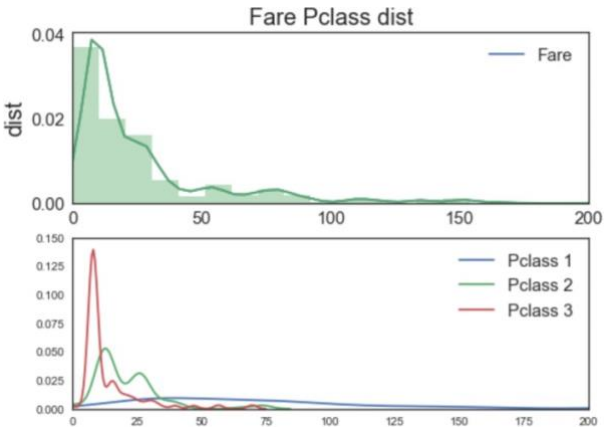


Figure 14: Fare-Pclass distribution

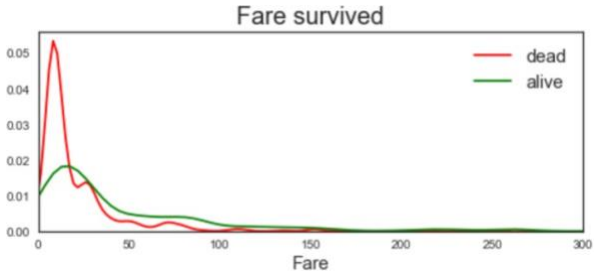


Figure 15: Fare-Survived distribution

From the figures above, we can see that people with higher fares are more likely to survive. This is consistent with the conclusion we got in the Pclass.

4.2.5 Sibsp & Parch

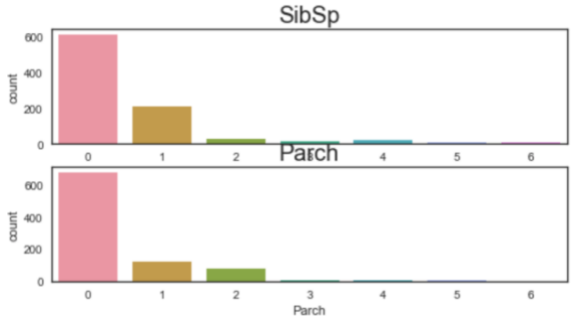


Figure 16: Sibsp count & Parch count

From the figure we can see that most of the passengers have no relatives. There are more people with only one cousin, and more with 1-2 parents/children members.

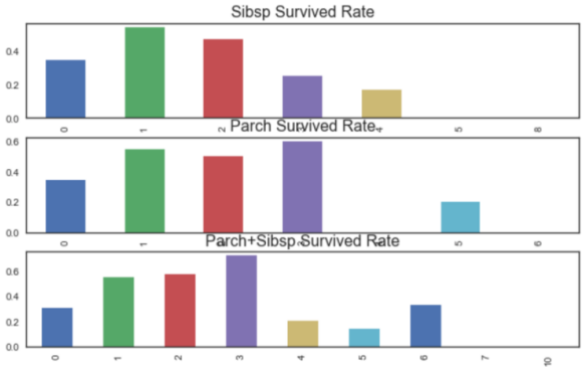


Figure 17: Group statistics on the survival rate

We try to group statistics on the survival rate of different numbers of relatives. From the figure we can see that the survival rate is approximately first high and then low. It is not a simple linear relationship between the number of family members and whether they were saved.

4.2.6 Embarked.

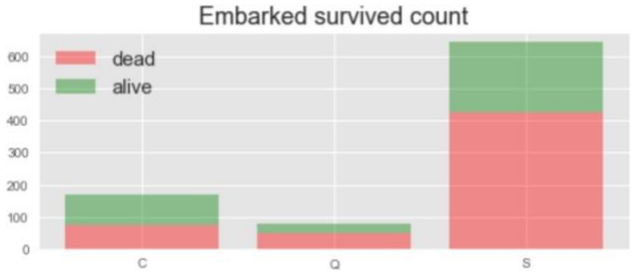


Figure 18: Embarked survived count

From this figure we can see that people who come from port C have a high probability of survived.

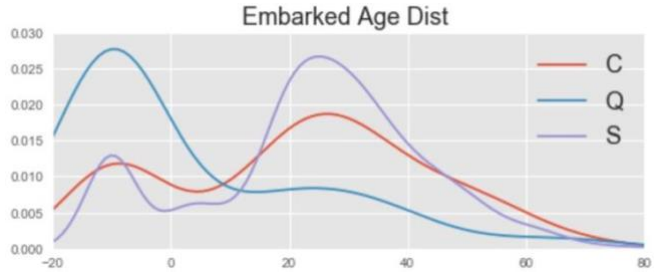


Figure 19: Embarked-Age distribution

We still use -20 to fill in the missing age.

From the above figure we can find:

1. Most of the passengers embarked on port Q have no age information.
2. The age distribution of port C embarked and port S embarked is similar. The difference is that the age

distribution of C's is flatter, and the proportion of children and the elderly is higher.

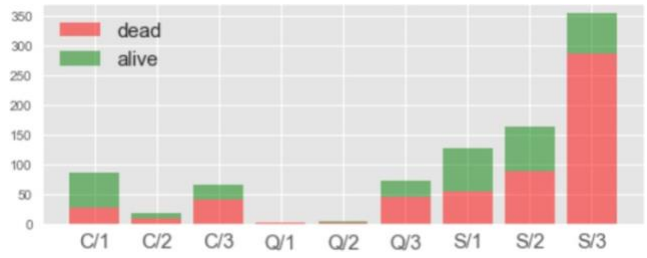


Figure 20: Embarked-Pclass-survived count

We have further categorized statistics on cabin class and port of embarkation. From the above figure, we can see that from the ratio of different positions, it seems that C is more likely to get rescued because the people embarked on port C contained more first-class passengers?

But with further comparison of C/S we found that the within the same cabin class, the probability of C survived is still higher.

4.2.7 Cabin. From the training data, we can see that there are 687 missing values and only 204 valid values. Although over 70% values are missing in this attribute, we found that passengers who have the cabin information also have higher survival probability. Thus, whether or not have cabin information can be consider as a feature for the further learning process.

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
27	28	0	1	Fortune, Mr. Charles Alexander	male	19.0	3	2	19950	263.0000	C23 C25 C27	S
75	76	0	3	Moen, Mr. Sigurd Hansen	male	25.0	0	0	348123	7.6500	F G73	S
88	89	1	1	Fortune, Miss. Mabel Helen	female	23.0	3	2	19950	263.0000	C23 C25 C27	S
97	98	1	1	Greenfield, Mr. William Bertram	male	23.0	0	1	PC 17759	63.3583	D10 D12	C
118	119	0	1	Baxter, Mr. Quigg Edmond	male	24.0	0	1	PC 17558	247.5208	B58 B60	C

Figure 21: Example of finding family members via cabin information

We can see from the above table that some passengers have more than one cabin number. The cabin numbers are consistent with the number of their family members. Therefore, we can complete part of the cabin information by finding the passenger's relatives.

Cabin_Zone	mean	count	Cabin_Zone	mean	count	Cabin_Zone	mean	count
0	0.299854	687	A	0.466667	15	B	0.744681	47
C	0.593220	59	D	0.757576	33	E	0.750000	32
F	0.615385	13	G	0.500000	4	T	0.000000	1

Figure 22: Survival rate in different cabin area

Then we further analyzed the existing cabin information and found that there are differences in surviving rates among passengers with different cabin index letters as shown in Table. And this also can be a feature for the next learning part.

4.2.8 Tickets. From the training data, we found that there are some SAME ticket number. Here's an example.

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Cabin_Zone
257	258	1	1	Cherry, Miss. Gladys	female	30.0	0	0	110152	86.5	B77	S	B
504	505	1	1	Maioni, Miss. Roberta	female	16.0	0	0	110152	86.5	B79	S	B
759	760	1	1	Rothes, the Countess. of (Lucy Noel Martha Dye...	female	33.0	0	0	110152	86.5	B77	S	B

Figure 23: Example of Same ticket number

We can see that these tree girls share the same ticket number and have relative cabin information, but they are not relatives. This may indicate that they are familiar with each other and may also result in same survival probability like family members.

4.2.9 Name. We found that name information includes title and surname, and implicitly contains information such as status and age. This can be used as a feature for learning. It can also be used to supplement default information.

And by grouping the passengers with same name length shown as follows, we find it is interesting that passengers with longer name may have a higher a survival probability. We may use this as a feature for learning, too

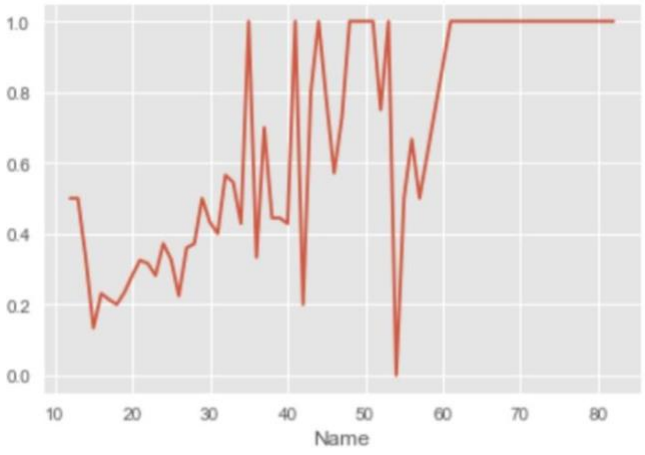


Figure 23: Name length and Survival probability

4 4.3 Feature Engineering

4.3.1 Simple Feature Engineering. We've already know that: In the training set, there are some data missing in attribute Age (714 entries, 177 missing), Cabin (204 entries, 687 missing) and Embarked (889 entries, 2 missing).

In the test set, there are some data missing in attribute Age (332 entries, 86 missing), Fare (417 entries, 1 missing) and Cabin (91 entries, 327 missing).

In the whole data set, age and cabin are missing in both the training set and the test set. Among all these, Cabin information has more missing values. Embarked has only 2 missing values in the training set. Thus, we first deal with the Embarked information.

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Cabin_Zone
61	62	1	1	Icard, Miss. Amelie	female	38	0	0	113572	80	B28	NaN	B
829	830	1	1	Stone, Mrs. George Nelson (Martha Evelyn)	female	62	0	0	113572	80	B28	NaN	B

Figure 24: Searching result of Embarked information missing

We can see that these two passengers are in First-class cabin. Through the training data, we find that most passengers were embarked at port S (S644, C168, Q77) and among the first-class passengers, most of them were embarked at port S (S127, C85, Q2), too. Thus, we decide to fill in the blank with S.

And in the previous analysis, we found that passengers who have the cabin information also have higher survival probability. Here we fill the missing cabin information both in the training and test set with 'Null' and replace the exist cabin information with 'Not Null'.

As we mentioned in the analysis of Name, we need to discretize the age and also handle the default values. In the baseline model, we handle it in a simple way:

1. Group the null value
2. Segmentation by Age and Discrete each segment （Discrete processing of data in a cycle of 5 years, while passengers under 10 years old and over 60 are classified separately）

And we got the new feature 'Age_map' as follows.

Age_map	count	mean	Age_map	count	mean	Age_map	count	mean
10-	62	0.612903	**10-15**	16	0.437500	**15-20**	86	0.395349
20-25	114	0.342105	**25-30**	106	0.358491	**30-35**	95	0.421053
35-40	72	0.458333	**40-45**	48	0.375000	**45-50**	41	0.390244
50-55	32	0.437500	**55-60**	16	0.375000	**60+**	26	0.269231
Null	177	0.293785						

Figure 25: New feature 'Age_map'

There is also a Fare information missing in the test set and we try to use the mean Fare value of passengers who is in the same cabin class and embarked at the same port to fill in the missing data. The distribution of Fare is too wide. Thus, we try to do some scaling to speed up model convergence.

And the last step is using one-hot to encode all the Category variables.

	SibSp	Parch	Fare	Pclass	Sex_female	Sex_male	Cabin_Not Null	Cabin_Null	Embarked_C	Embarked_Q	...	Age_map_20-25	Age_map_25-30	Age_map_30-35
0	1	0	-0.502445	3	0	1	0	1	0	0	...	1	0	0
1	1	0	0.786845	1	1	0	1	0	1	0	...	0	0	0
2	0	0	-0.488854	3	1	0	0	1	0	0	...	0	1	0
3	1	0	0.420730	1	1	0	1	0	0	0	...	0	0	0
4	0	0	-0.486337	3	0	1	0	1	0	0	...	0	0	0

5 rows × 24 columns

4.3.2 Further Feature Engineering

Figure 26: Example of Processed train data

4.3.2.1 Title. Name actually implies a great deal of information, including gender, status, wealth, marital status, and so on. Therefore, we extract the titles in the names. In the previous part we mentioned that the length of the name is positively related to the rate of return. Therefore, we add the length of the name to the feature. And we find that similar titles share similar survival rates.

So, we extracted the title information. Since there are too few people in some titles, we also need to make a mapping.

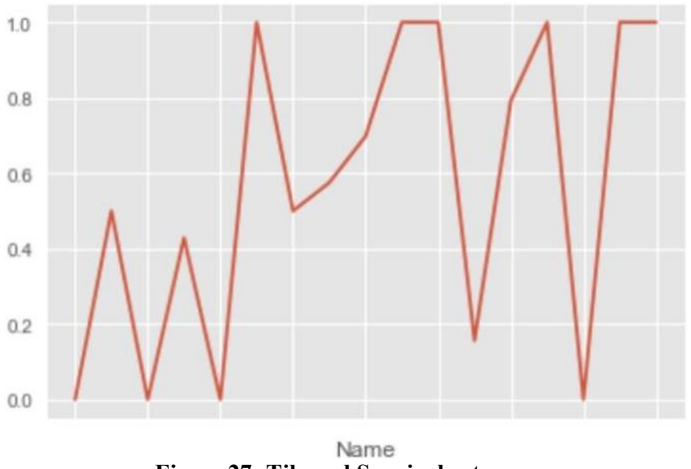


Figure 27: Tile and Survival rate

4.3.2.2 Family. In the Titanic scenario, female deaths and male survival are all small-probability events. The model will easily determine the survival of female passengers and death of male passengers. In order to enhance the ability of the model to identify this type of group, we analyzed the data and found an important feature, family. The survival death pattern under the same family is largely the same. For example: there is a family with a female death, and the family other females have a higher probability of death. Therefore, we mark these special families out.

Then, we will perform the discretization of the Familysize extracted above.

4.3.2.3 Final Data.

	Title_Master	Title_Miss	Title_Mr	Title_Mrs	Name_Len	Dead_female_family	Survive_male_family	IsChild	High_Survival_Ticket	Low_Survival_Ticket	...	Famil
0	0	0	1	0	1	1	1	0	0	1	...	
1	0	0	0	1	4	1	1	0	1	0	...	
2	0	1	0	0	1	1	1	0	0	0	...	
3	0	0	0	1	4	1	1	0	1	0	...	
4	0	0	1	0	2	1	1	0	0	1	...	

5 rows × 24 columns

urvival_Ticket	Low_Survival_Ticket	...	FamilySize_Big	FamilySize_Small	Cabin_isNull	Pclass_1	Pclass_2	Pclass_3	Sex_female	Sex_male	Low_Fare	High_Fare
0	1	...	0	1	0	0	0	1	0	1	1	0
1	0	...	0	1	1	1	0	0	1	0	0	1
0	0	...	0	0	0	0	0	1	1	0	1	0
1	0	...	0	1	1	1	0	0	1	0	0	1
0	1	...	0	0	0	0	0	1	0	1	1	0

Figure 28: Overview of the Final data(a)

Figure 29: Overview of the Final data(b)

4.4 Baseline Model

4.4.1 BASIC LOGISTIC REGRESSION (WITHOUT NAME). In this baseline model, we first try logistic regression (LR) model. We use 5-fold cross validation and L2 regularization. And then we apply GridSearchCV to find the best result. The learning process shown as follow.

```
(0.7994487816366191, 0.029496353096645223)
```

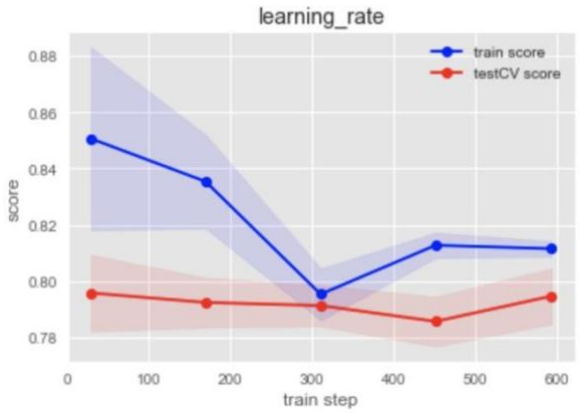


Figure 30: Learning process map of basic LR model

We use the single logistic regression model to learn and predict, the final result of this one is 0.77033

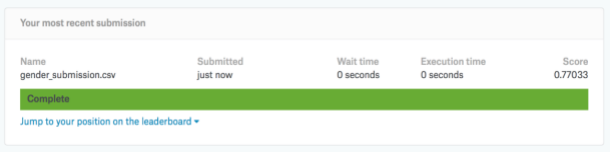


Figure 31: Submission result of baseline model (LR)

4.4.2 Using Name Feature.

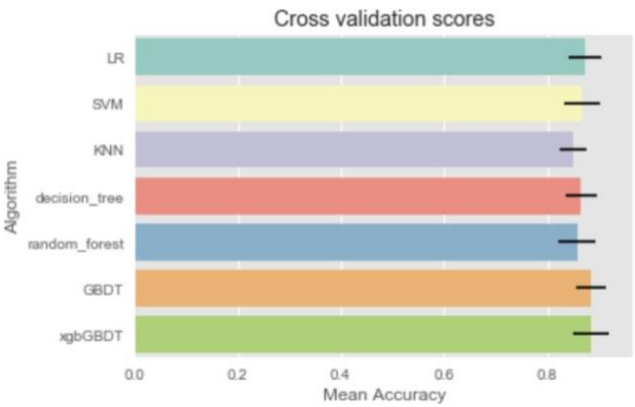


Figure 32 Performance of 7 baseline models

The performance of several algorithms such as logistic regression, support vector machine, nearest neighbor, decision tree, random forest, gbdt, and xgbGBDT are investigated. We have observed that the feature importance of different models is quite different. Thus, we introduce ensemble model.

4.4 Ensemble Model

We used xgbGBDT, Logistic regression, Random forest, SVM, GBDT to form the ensemble model with bagging method. The scores on the train set and test set are 90.8 and 0.80382 (different scoring method, for train set we used score from pandas, while Kaggle uses another score for test set), respectively.

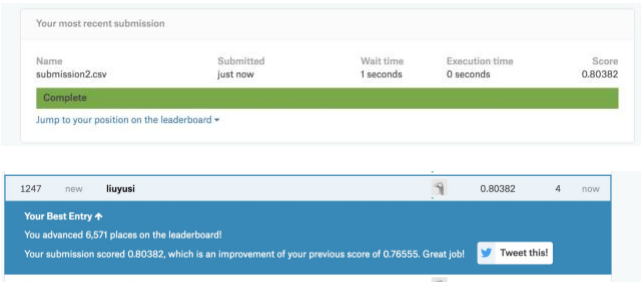


Figure 33: Final rank

5 Analysis of Results

The feature we extracted from the original data makes the correlation between the target value and the input value significantly improved. After introducing new features, which we got in part 4.3, the score of baseline models on the train set can be as good as 86-88, which is much better than the basic logistic regression model in part 4.4.1 (has a score on train set of 84 and on test set of 77).

After adopting the ensemble (bagging) method, the score on train set outperforms all the baseline models (90+). Moreover, the bagging model has a better performance on the testing of Kaggle, resulting in a score of 0.80382.

Thus, we can see that Bagging method has no restrictions on weak classifiers, which is the same as Adaboost. However, the most common ones are also decision trees and neural networks.

The bagging collection strategy is also relatively simple. For the classification problem, a simple voting method is usually used, and one of the categories or categories that obtain the most votes is the final model output. Since the Bagging algorithm samples every time to train the model, the generalization ability is very strong, and it is very useful for reducing the variance of the model. Of course, the degree of fit to the training set will be less, that is, the bias of the model will be larger.

6 Conclusion

The ensemble model using bagging method is indeed better than any machine learning algorithm alone on this task. Our ensemble model got a top 10% rank, which is much higher than our original goal written in the proposal (top 15%).

But we believe that this Titanic survival prediction project still has much room for improvement in variable processing and forecasting methods. We list several further improvement ways:

1. Analyzing the importance of the factors affecting the model can clearly determine which variables have a greater impact on the establishment of the model, and the impact factors on different weights believe that the accuracy of the prediction results can be further improved.
2. Selecting different modeling methods will have an impact on the prediction results.
3. Given attribute Age a more accurate division based on data distribution may reduce errors and improve prediction results.
4. The cabin number corresponds to different locations on the ship and will certainly have an impact on survival. However, due to too many missing data, this variable was not considered in this modeling.
5. This prediction only introduces 2 new features. Through in-depth analysis of different variable data, and then refining new impact features (such as the number of people on the deck, etc.), the prediction results should be further improved.

REFERENCES

[1] https://www.kaggle.com/c/titanic#description
[2] https://www.kaggle.com/daniel83fr/titanic-how-to-start-a-beginners-path/notebook
[3] https://www.kaggle.com/sagarjain2030/titanic-machine-learning-from-disaster
[4] http://trevorstephens.com/kaggle-titanic-tutorial/getting-started-with-r/
[5] https://blog.csdn.net/login_sonata/article/details/54315273
[6] L. Breiman. Random forests. Machine Learning, 45(1): 5–32, 2001. 18.
[7] S.R. Gunn, “Support Vector Machines for Classification and Regression,” technical report, School of Electronics and Computer Science, Univ. of Southampton, Southampton, U.K., 1998, http://www.isis.ecs.soton.ac.uk/resources/svminfo/
[8] Breiman, L. (1996a). Bagging predictors. Machine Learning 26(2), 123–140.
[9] Friedman, J. H. (1999b). Stochastic gradient boosting. Technical report, Dept. Statistics, Stanford Univ.
[10] Chen, T. and He, T. XGBoost: eXtreme Gradient Boosting. https://github.com/dmlc/xgboost. Accessed: 2015-06-05
[11] Rodríguez, G. (2007). Lecture Notes on Generalized Linear Models. pp. Chapter 3, page 45 – via http://data.princeton.edu/wws509/notes/.
[12] Ben-Hur, Asa; Horn, David; Siegelmann, Hava; and Vapnik, Vladimir N.; "Support vector clustering"; (2001); Journal of Machine Learning Research, 2: 125–137
[13] Altman, N. S. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression". The American Statistician. 46 (3): 175–185. doi:10.1080/00031305.1992.10475879.
[14] Ho, Tin Kam (1995). Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282.
[15] Ho TK (1998). "The Random Subspace Method for Constructing Decision Forests"(PDF). IEEE Transactions on Pattern Analysis and Machine Intelligence. 20 (8): 832–844. doi:10.1109/34.709601.
[16] Breiman, L. "Arcing The Edge" (June 1997)
[17] Friedman, J. H. "Greedy Function Approximation: A Gradient Boosting Machine." (February 1999)
[18] ^ Jump up to: a b c Friedman, J. H. "Greedy Function Approximation: A Gradient Boosting Machine." (February 1999)