Titanic: Machine Learning from Disaster

Xianlong Zhang, Xinyi Zhang, Xiaofei Huang, Qiqi Cheng

I. Introduction

BASED on provided information of passengers on the Titanic (from Kaggle) [1], this project is aimed at conducting analysis of how these elements are related to being survived in the sinking or not, followed by evaluation with model built.

II. Data Preparation

Three data files named “train.csv”, “test.csv”, and “gender_submission.csv” are given, representing training data, testing data and true results of testing data respectively. In test file, the survived data are removed and stored in the last file for prediction and accuracy calculation. Information of 1309 passengers is provided while 891 of them are training data.

For training data, 12 variables with 891 passengers need to be observed (as shown in Table I). In order to better model and predict, the feature will be classified and performed correlation examination. Missing value may be completed and categorical features may be converted to numeric values.

TABLE I
Definition of Given Features

Feature	Definition
PassengerID	No. of passengers
Survived	0 = not survived, 1 = survived
Pclass	1 = first class, 2 = second class, 3 = third class
Name	passenger name with title
Sex	male or female
Age	estimated age appears as decimal
Sibsp	number of siblings/spouses aboard
Parch	number of parents/children aboard
Ticket	ticket number
Fare	passenger fare
Cabin	cabin number
Embark	embarkation port

In order to get better model and prediction, the features will be analyzed first. Missing values may be completed and categorical features may be converted to numeric values. For 12 features we dropped ‘Name’, ‘Sibsp’, ‘Parch’, ‘Ticket’, ‘Cabin’. Meanwhile, we added ‘Namelength’ to replace ‘Name’, ‘FamilySize’ to replace both ‘Sibsp’ and ‘Parch’. Besides, we added missing values in ‘age’ and “fare” with means, and filled missing values for “Embarked” by using the most frequent embarkation mark, then we mapped ‘Sex’, ‘Embarked’, ‘Pclass’ into numeric values. So we have 7 features right now: ‘Family Size’, ‘NameLength’, ‘Embarked’, ‘Fare’, ‘Age’, ‘Sex’, ‘Pclass’.

Here is the feature correlation as shown in Fig 1:

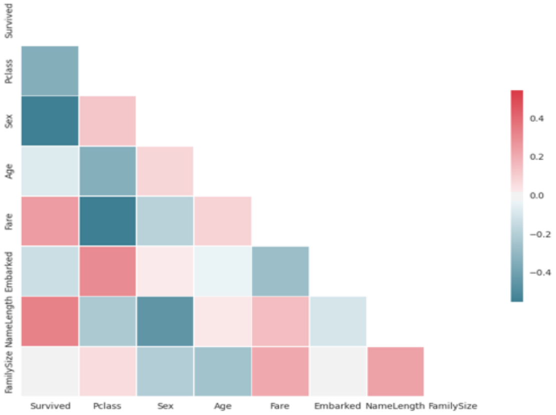
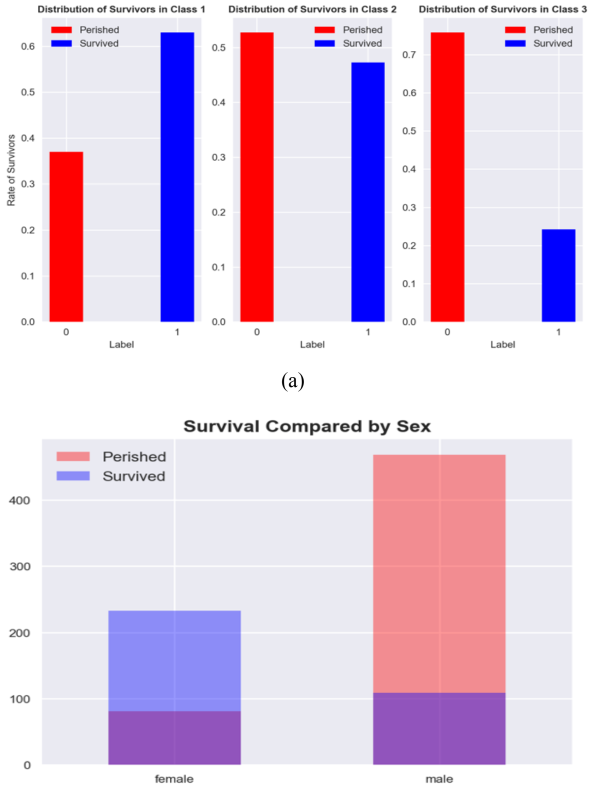


Fig. 1. Correlation graph for features

From the correlation graph, we find there are four features are highly relative to ‘Survived’, which are ‘Pclass’, ‘Sex’, ‘Namelength’ and ‘Fare’, we will take these four as instances to analyze features, as shown in Fig 2 and Fig 3.

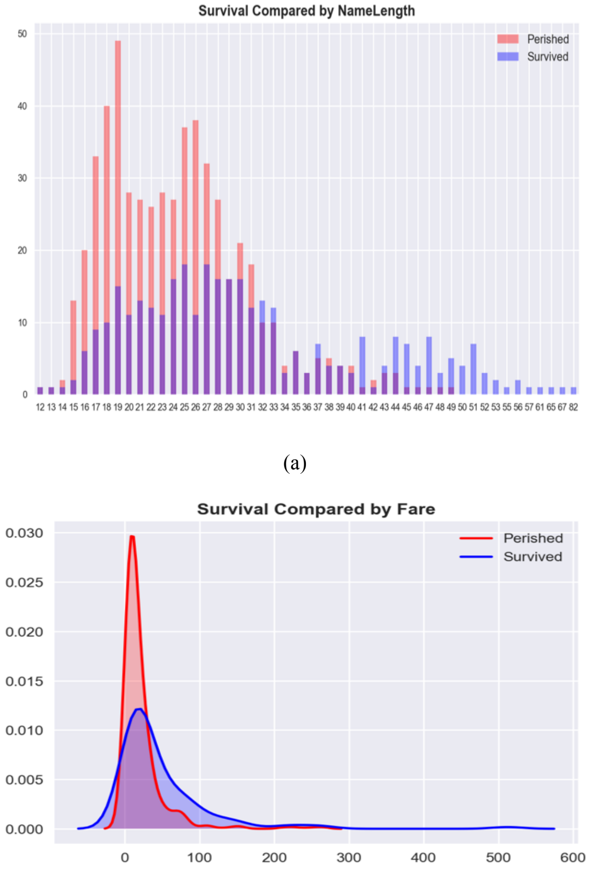


(a)

(b)

Fig 2. Feature analysis for (a)Pclass (b)Sex

In Fig 2(a), we can see that in 1st class has a higher survival rate than the other two classes, and 3rd class has the lowest survival rate with large gap between perished and survived number. In (b), survival rate of female is much higher than perished, which it's completely different when we look into male survival rate.

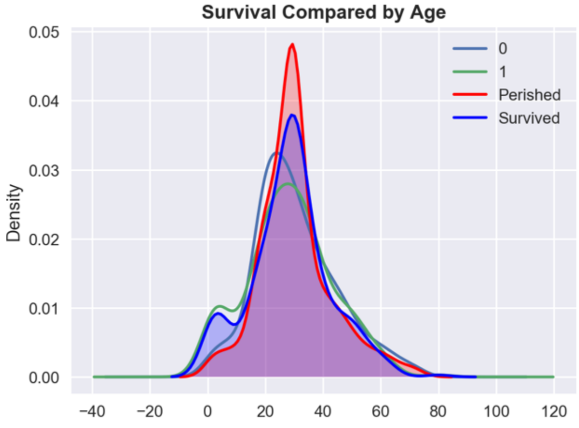


(a)

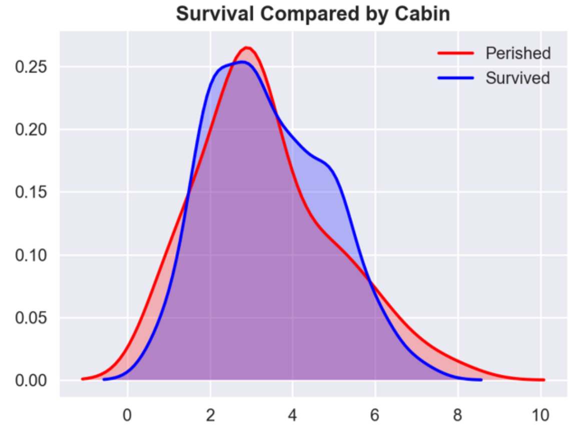
(b)

Fig. 3. Feature analysis of (a)Name length (b)Fare

Also, we can infer from Fig 3(a) that longer name length seems to be more inclined to survive, and (b) tells us that less expensive fares mean higher mortality rate.



(a)



(b)

Fig.4. Feature analysis of (a)Age (b)Cabin

Fig 4 shows influence of "Age" and "Cabin". Actually, according to the "women and children first policy", "age" feature should be a highly-weighed factor to train our model, but from Fig 1 we can see it has few relations with survival rate. Maybe it's because the small ratio of children among passengers. As for "Cabin" feature, the distributions for "Perished" and "Survived" are even, so we dropped "Cabin" feature when we process data.

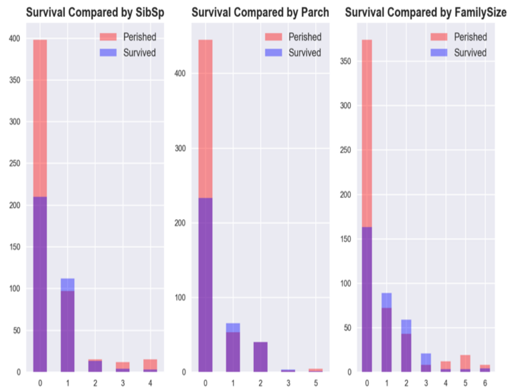


Fig. 5. Feature analysis of family size

In Fig 5, the survival rate is higher when the family size between 1 and 3, which means family member would try to save one between them. But when the family size is greater than 4, the death rate becomes higher, it may be because someone would take much time to find family members and missed the best time to survive.

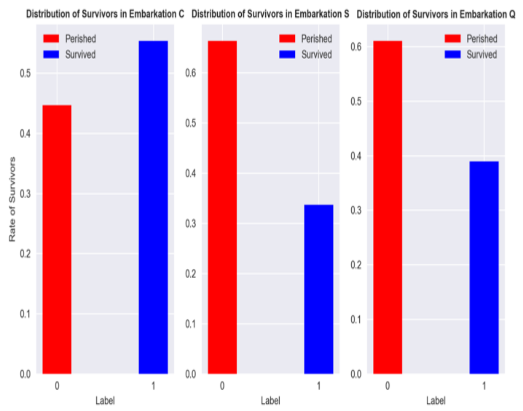
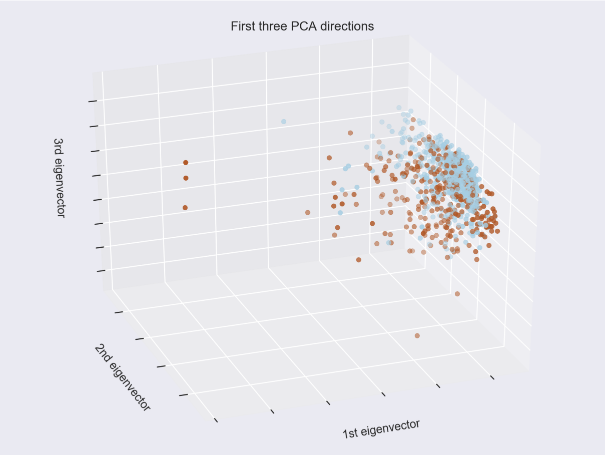


Fig. 6. Feature analysis of embarkation

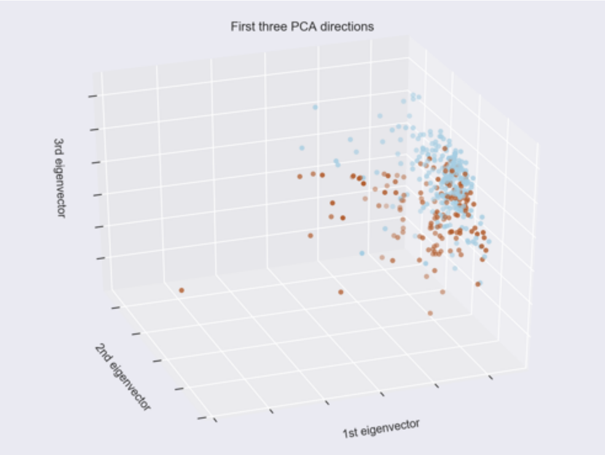
Intuitively, the embarkation shouldn't be a key factor to affect the survival rate, however, Fig 6 presents that people get on aboard from embarkation C have a larger chance to survive, so we kept this feature.

A. *Principal Component Analysis (PCA)*

PCA leads high-dimensional data to a lower-dimensional space. In this project, PCA is used to merge correlated features or cluster from the compact principle components. We decreased our dimension to three in our project. Here is the result figure for train data and test data[2].



(a)



(b)

Fig. 7. First three PCA for (a)train data and(b)test data

III. Methodology

A set of training samples is given while being marked as survived or not survived. In order to achieve supervised learning as well as classification and, the choice of models is narrowed down and three of them are finally be chosen in this project. We compared our prediction result to the test label getting our accuracy.

A. *Support Vector Machine (SVM)*

SVM is eligible to train non-linear data and classify them in two categories with a hyper-plane by mapping the inputs to a sufficiently high dimension. We used linear SVM and Non-linear SVM with RBF kernel, as in [3]:

$$\mathrm{K(x,y)} = e^{\frac{-\left\|x^T y\right\|^2}{2\sigma^2}} \tag{1}$$

The result shows Non-linear SVM performs better than linear one, which is 0.7081 > 0.6387 accuracy.

B. *K-nearest Neighbors Algorithm(k-NN)*

With k-NN classification, an object is assigned to the class among its k nearest neighbors. After performing PCA for feature extraction and dimension reduction, k-NN follows to classify data in reduced-dimension space. The applied distance metric is Euclidean distance:

$$\mathrm{d(i,j)} = \sqrt{\left|x_{i1}-x_{j1}\right|^2+\left|x_{i2}-x_{j2}\right|^2+\cdots+\left|x_{ip}-x_{jp}\right|^2} \tag{2}$$

As shown in Fig 8, we can find out that the best K is 22, and we can get an accuracy of almost 72%.

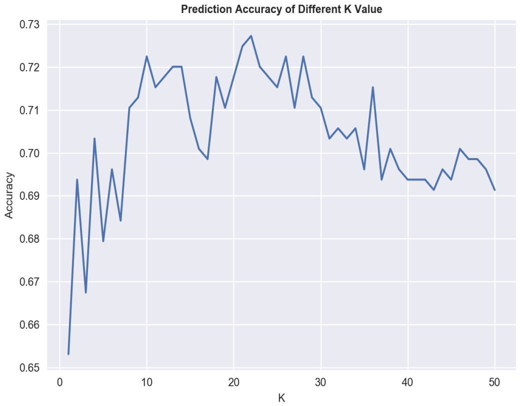


Fig. 8 Prediction accuracy of various K value

C. Linear Discriminant Analysis (LDA)

Linear discriminant analysis (LDA) is a generalization of Fisher's linear discriminant. This method is used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. [4]

We used eigenvalue decomposition which decomposes a square matrix into its eigenvalues and eigenvectors. We used shrinkage to improve estimation of covariance matrix.

In the result, we can see the accuracy is 0.6555.

IV. Conclusion

Results and running time are given in table II. Apparently, our samples are nonlinearly separated, that’s why KNN and nonlinear SVM perform better than linear SVM and LDA.

TABLE II
COMPARISON OF DIFFERENT CLASSIFIER MODELS

Model	Accuracy	Running Time(s)
Linear SVM	0.6387	0.027
Nonlinear SVM	0.7081	0.403
K-NN	0.7273	0.263
LDA	0.6555	0.002

From the results of our project, if a person wants to survive in this disaster, she’d better be a female or child, because women and children always go first. Otherwise if he is a male, he needs to have lots of money or a long name, in that case he’s more likely to be a noble man, and people may let him go first survive before them. But if he is a poor man in middle age, traveling alone in this ship, he will have a high death rate in this disaster.

In this project, there’re some details need to be improved, such as the way to fill missing values. Since we just simply filled missing values by mean or most frequent data, overfitting and bias may be caused. The better way to get missing values is to predict them with the full data set by linear regression or expected maximum, which I believe can be a more reasonable method to train our model.

References

[1] "Titanic: Machine Learning from Disaster Kaggle", Kaggle.com,2017.[Online].Available:https://www.kaggle.com/c/titanic. [Accessed: 21- Apr-2017].

[2] "sklearn.decomposition.PCA — scikit-learn 0.18.1 documentation",Scikit-learn.org,2017.[Online].Available:http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html. [Accessed: 20- Apr- 2017]

[3] J. Cott-Font, "Support Vector Machines", Northeastern University, 2017.

[4] C. Bishop, Pattern recognition and machine learning, 1st ed. New Delhi: Springer, 2006, p. 187.