TITANIC

Machine Learning from Disaster

Author:
Vayne Lover

Supervisor:
Udacity

August 25, 2016

1 Introduction

1.1 Project Overview

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.

1.2 Problem Statement

In this project, i am going to complete the analysis of what sorts of people were likely to survive.In particular, i will apply the tools of machine learning to predict which passengers survived the tragedy.Data Sets are downloaded from kaggle.

1.3 Metrics

In this project,i will choose the F1 score[1] for the result.The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:

$$F1Score = 2 * (precision * recall)/(precision + recall)$$

2 Analysis

2.1 Data Exploration

In this part i will do data enigneering.Above all,i will show the features of the data sets.As you can see ,each passenger has eleven features.The meaning of features can be seen below.

1. survival : Survival (0 = No; 1 = Yes)
2. pclass : Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)
3. name : Name

4. sex : Sex
5. age : Age
6. sibsp : Number of Siblings/Spouses Aboard
7. parch : Number of Parents/Children Aboard
8. ticket : Ticket Number
9. fare : Passenger Fare
10. cabin : Cabin
11. embarked : Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

2.2 Exploratory Visualization

Now i am going to show some related pictures of the data.Above all, i will tell what i have done for data.
I change all string features into numerical features.

1. I let NaN in Age equals to the mean of Age.
2. I let Name equals to the length of Name.
3. I let Embarked equals to 0,1,2 when the value is S,C,Q.
4. I let Sex equals to 0,1 when it is male and female.
5. I Add new features Family equals to SibSp add Parch.
6. I delete PassengerId SibSp Parch,Cabin and Ticket.

It's easy to understand i change features into numerical,and i want to explain why i delete other features.To be honest,I think that the Cabin has a lot of null value which means little information.Besides,Ticket is irregular.Therefore,i decide to delete these features.
This time i will use seaborn[2] to plot figures.
First picture is about age and sex.We can obviously find that female survived more that male.Maybe male choose to let female go that time.That's truely gentleman.

Second picture is about the Embarked.From the picture we can learn that when Embarked is C it has the highest mean of survived.

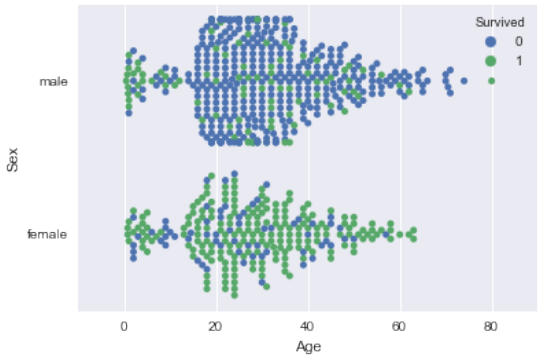

Figure 1: Age-Sex

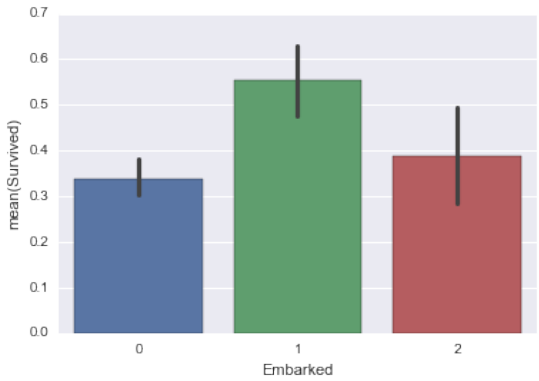

Figure 2: Embarked

Third picture is about Pclass and Fare,i want to explore if people cost more money have higher probablity survived.And i find that when class is 1 it have more probablity to survive cause it close to sky.

Fourth picture is about Family and Age,i want to find that if people have more sisters or parents in the ship have higher probablity to survive.And i find the truth is that people have many relatives do not survive may be it because people can't go themselves.

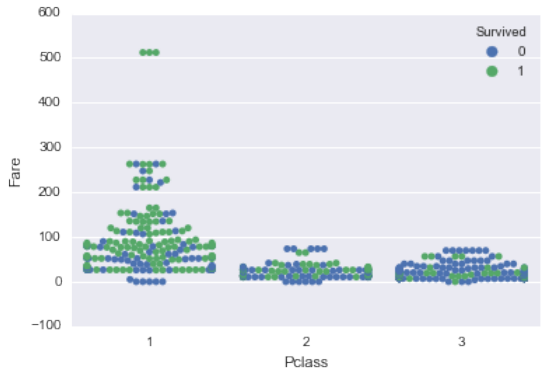

Figure 3: Pclass-Fare

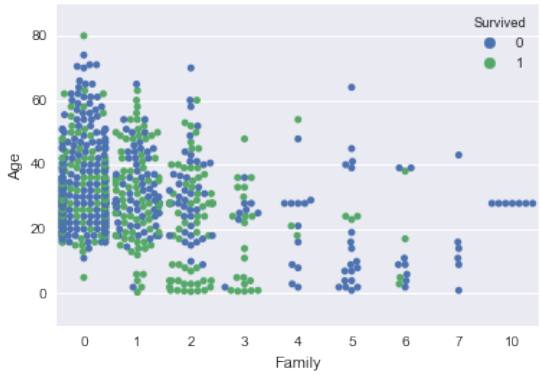

Figure 4: Family-Age

2.3 Algorithms and Techniques

In order to have a good prediction,i will choose a good model first.And now i will test three models to train.Then i will choose best of these to tune.
The models are SVM[3],KNN[4] and RandomForest[5] for my models.

2.4 Benchmark

I will test the f1 score of each model and find the best one.

3 Methodology

3.1 Data Preprocessing

In order to have a good train and test set.i use train_test_split[6] to split data.
And i use three times to test the models,and here are the table.

Models	Size	Time	Train Score	Test Score
SVM1	33%	0.010s	0.944	0.279
SVM2	66%	0.028s	0.931	0.247
SVM3	100%	0.033s	0.927	0.302
KNN1	33%	0.003s	0.678	0.641
KNN2	66%	0.002s	0.687	0.600
KNN3	100%	0.001s	0.708	0.627
Ran1	33%	0.021	0.975	0.741
Ran2	66%	0.028	0.978	0.778
Ran3	100%	0.040	0.976	0.756

Table 1: Comparision Of Models

As we can see in the table,randomforest algorithm have the best test score,so i will choose it for my next model.

3.2 Refinement

Since i have choose the randomforest,now i will tune model for better results.
In order to get the best parameters,i will use grid search[7].
What i am going to change can be seen below:

- n_estimators:10,20,40,80,120,150,180
- criterion:gini,entropy
- max_features:log2,sqrt,None
- max_depth:5,6,7,8,9,10

- min_sample_ssplit:1,2,3
- warm_start:False,True

I will explain the meaning of each parameter.

- n_estimators means the number of trees in the forest.
- criterion means the function to measure the quality of a split. item max_features means the number of features to consider when looking for the best split.
- max_depth means the maximum depth of the tree.
- min_samples_split means the minimum number of samples required to split an internal node.
- warm_start means if reuse the solution of the previous call to fit and add more estimators to the ensemble.

4 Result

After 1269.8s searching,i finally get the optimal model parameters.The value can be seen below.

- 'warm_start': True
- 'oob_score': False
- 'n_jobs': 1,
- 'verbose': 0
- 'max_leaf_nodes': None
- 'bootstrap': True
- 'min_samples_leaf': 1
- 'n_estimators': 180
- 'min_samples_split': 3
- 'min_weight_fraction_leaf': 0.0

- 'criterion': 'entropy',
- 'random_state': None
- 'max_features': 'sqrt'
- 'max_depth': 10
- 'class_weight': None

And i get the final f1 score is 0.7731 which is better than previous test.Therefore i think this model can deal with the problem.

5 Conclusion

You can see the picture below which shows that my tuned model really does better than previous.And it can be known from the higher f1 score.

```
Tuned model has a training F1 score of 0.9272.
Tuned model has a testing F1 score of 0.7731.
Optimize model in 1269.7593 seconds
```

Figure 5: Result

As for my process,i first explore the data,then do feature engineering adding or deleting some features.After that,i firstly choose three model to find the best one.Then i tune the best randomforest model to get best results.Obviously my model works better than ever.
The interesting thing i think is to find the best parameters,firstly i just tune the model by hand.For example,i change the init function to change parameters.And i find it is too hard and too boring.Fortunely,i think of Grid Search and use it to instead my hand.
In order to improve the result,i think we can add more parameters.For example,we can add more detailed parameters like n_estimators:10,20,30,40,50 an so on.Of course it will cost much more time.

6 Reference

1. https://www.kaggle.com/c/titanic
2. http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html
3. http://web.stanford.edu/ mwaskom/software/seaborn/api.html
4. http://scikit-learn.org/stable/modules/svm.html#svm
5. http://scikit-learn.org/stable/modules/neighbors.html#neighbors
6. http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
7. http://scikit-learn.org/stable/modules/cross_validation.html#cross-validation
8. http://scikit-learn.org/stable/modules/generated/sklearn.grid_search.GridSearchCV.html