ID: ywang3564
Dataset used in the following analysis: Titanic from Kaggle.

The first dataset I am choosing is Titanic problems from Kaggle. Here is a brief introduction about the sinking of Titanic.. Personally, I watched the Titanic movie at a very young age, which left an unforgettable memory for me. This is also the first dataset when I joined the Kaggle competition. Practically, I am very familiar with the dataset and will learn how to improve the analysis with other's previous experience.

The second classification problem was also derived from this dataset. The Embarked information was shown for each person. Does the Embarked location have some correlation with other factor. For example, all are first class embarked in a certain place. Based on other information, the goal is to predict the embarked location for each person. From my exploration from the below methods, it's actually predictable. It means certain groups of people board the Titanic boat at a certain place.

Description of dataset:
I split the original training dataset as two-part: 1) train 2) test. The total number for training and the testing dataset is 703 and 188, roughly 4:1. Besides, the original data have 12 columns, with the column of "Survived" is the outcome.

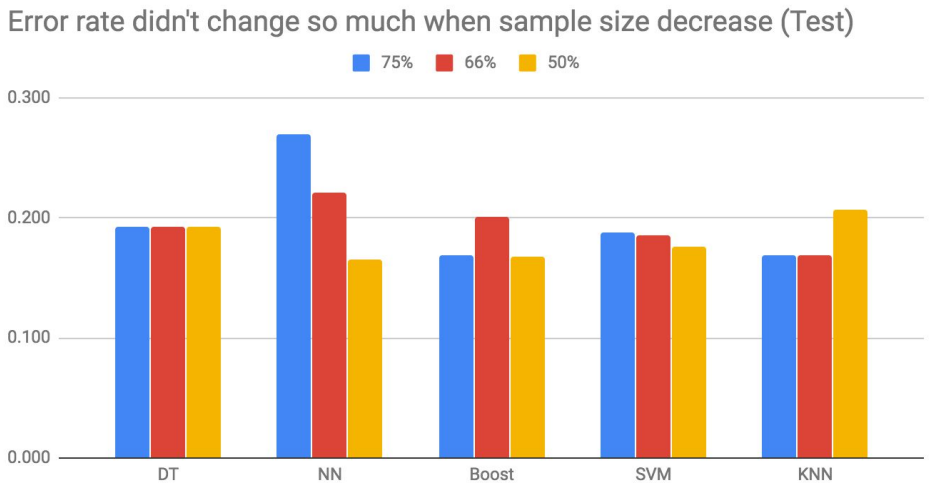
Description of the methods:
I am going to use R package for all analysis. Here are all the packages I am using:

1. Decision Tree: Rpart;
2. Neural Network (NN): neuralnet
3. Boosting: gbm
4. Support Vector Machine(SVM): e1071
5. k-Nearest Neighbors (KNN): class

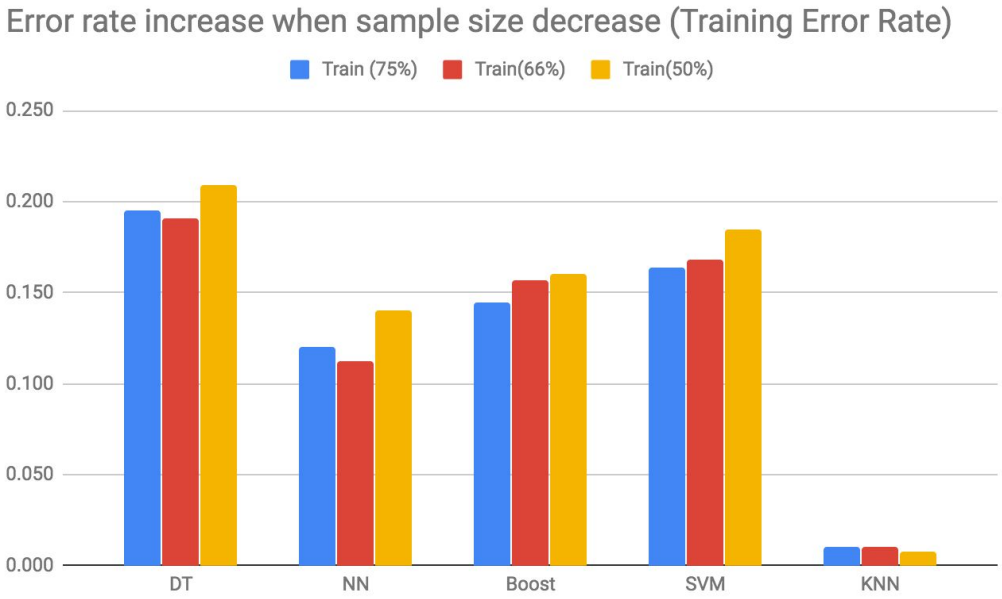
Data Preparation:
Following procedures were applied for train and test data.

- Remove those with NA value;
- Remove unused columns like Passenger Name or ID;
- Covert PClass (Cabin class) to three binary factors: First, Second, Third;
- Covert Sex to two binary factors: Male, Female;
- Covert Embarked to three binary factors: EmbarkedC, EmbarkedQ, EmbarkedS.
- Re-scale Age and Fare with (x-min)/(max-min), and the final results will range from 0 to 1.

Below is the summary for error rate of test data for different methods. Boost and KNN are top performance methods among all. Except for NN, the final result does not change so much after the sample size change. These suggest a robust method for all these four.



For the train data, when the sample size decreases, the error rate increases. This is true, because the sample size decreased, the model can not match training data very well. However, the generalization of the model still works, because the error rate in test data didn't change (see above).



Conclusion: DT, Boost, SVM and KNN all work very well on my dataset since the problem itself is really a binary problem: classify the people into two categories. Especially boosting and KNN works the best. Compared with the decision tree, boosting with the decision tree can improve the power of prediction. KNN also works very well for its best performance on clustering, which is the essence of my problem. The neural network seems like a little over-fitting because its error rate for train is low. But with decreasing the sample size, the performance of NN has a quick improvement. If we use only 50% of sample, NN is the best performer. The train error rate of KNN is an outlier, when you choose K=1. Decision tree gave me a reasonable way to understand my data, please see the following sections for more details.

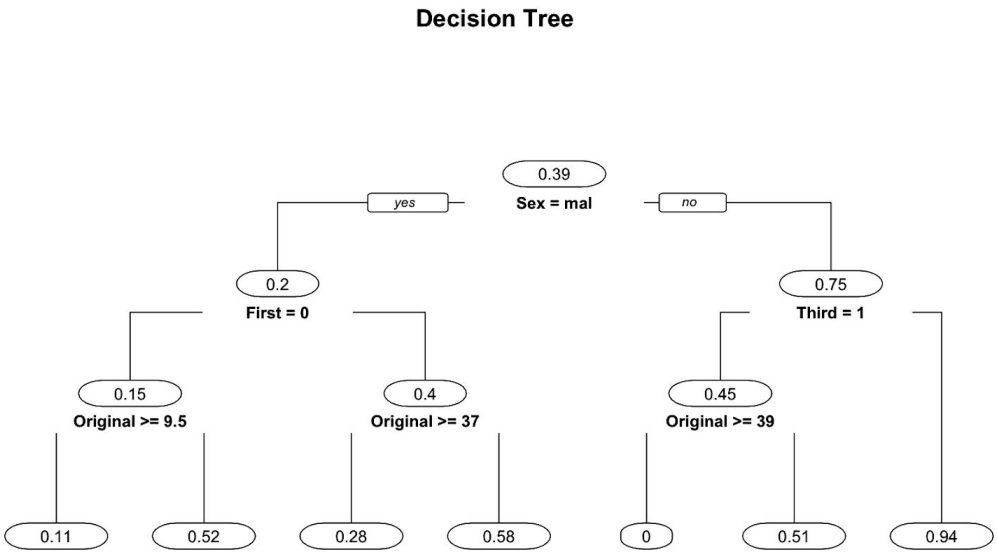
For the second classification, it was very surprising to find that the boarding location is actually predictable. The best result came from KNN, which highly agrees with the first classification. KNN is designed based on similarity. All group boarding at a certain place must have a certain similarity.

1. Decision Tree with pruning.

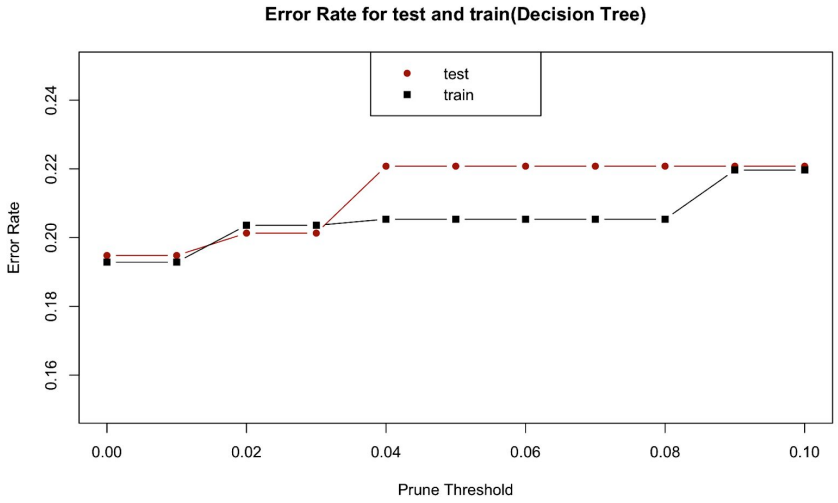
Decision Tree is a tree-like model to make a decision and give every possibility at its leafs.

All datasets are converted to binary data and re-scaled with the methods of (x-min)/(max-min). I have tried different combinations of factors used in the decision tree. Because of the paragraph is limit, I will directly go to the final best combination: **Survived ~ Sex + Original_Age + First + Second + Third.** Below is the decision tree before any prune.

From the tree, **gender** is the **first** important factor for survival. Male has a chance of 20% to be saved, while the saving chance for the female is 75%. If those females are not from third class (Third = 1, FALSE), then the probability will increase to **95%**. For females from the third class, they only have a chance of 45% to be saved. Same things happened for the male. If they are from the first class, they have a chance of 40% to be saved, while the chance for Non-first class is only 15%.



Additionally, I also pruned the tree with different criteria. The tree without any pruning seems like the best tree and the **minimum error for train and test data is 0.193 and 0.195**, respectively.



For the second classification problems, after applying the model of “Embarked ~ Sex + Age + Pclass”, I got the error rate for testing is: 0.216. All results indicate that they are all from EmbarkedQ.

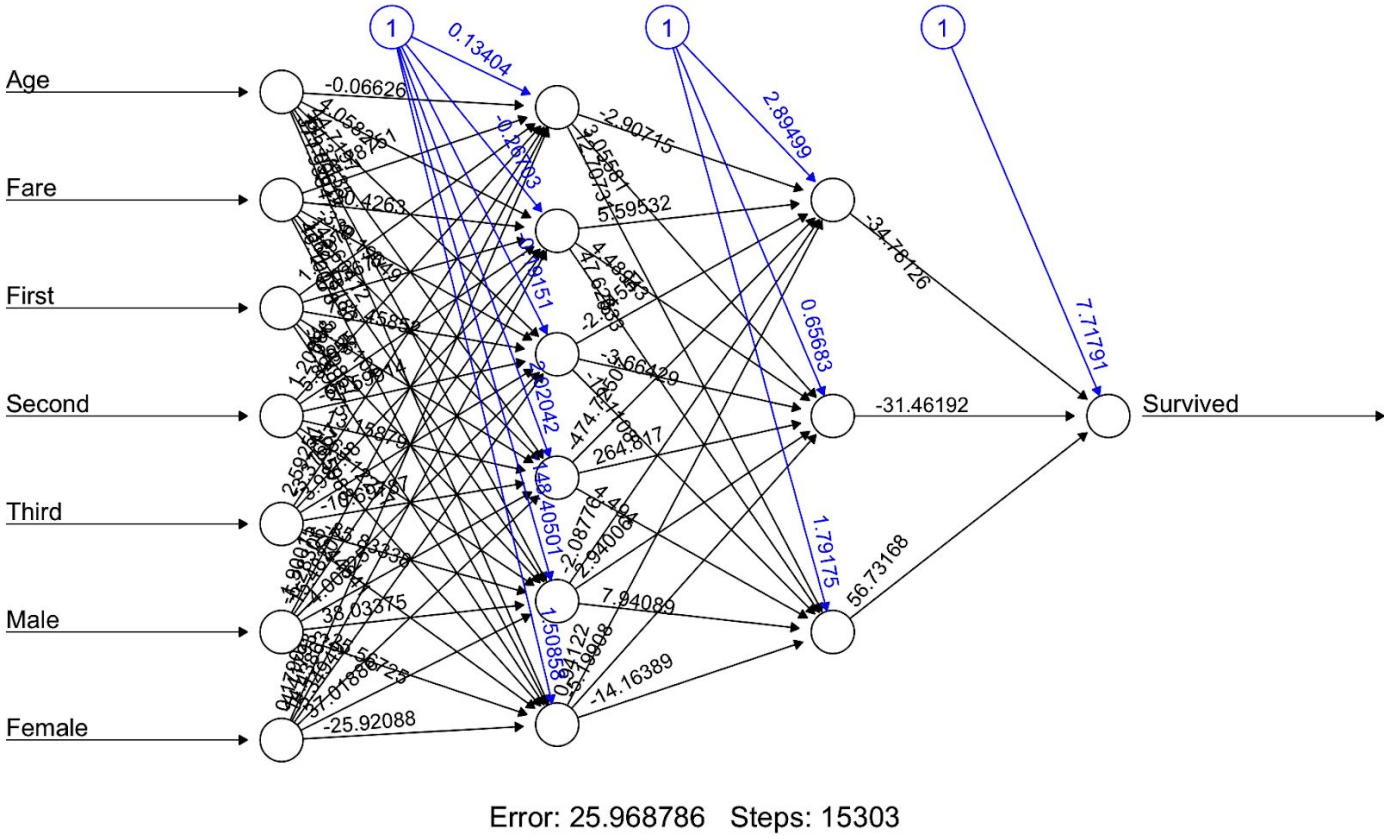
Reference: R in action (V2)

2. Neural Network

Neural Network can both work as a machine learning algorithm or as a glue of other machine learning techniques. Here used NN as a machine learning algorithm and compare its performance with others. Same with the previous analysis, all data are converted and binary as additional factors.

At first, I chose all factors as input and draw the Neural Network. Based on the weight on different factor, I kept those factors with high weight and use this formula: Survived ~ Age + Fare + First + Second + Third + Male + Female.

For computational consideration, I only try two layers. For the neuro numbers, **I started from 3 to 6 and the error rate for test decreased from 0.27 to 0.22, while the error rate for the train is always 0.12.** It seems like a over fitting for my dataset.



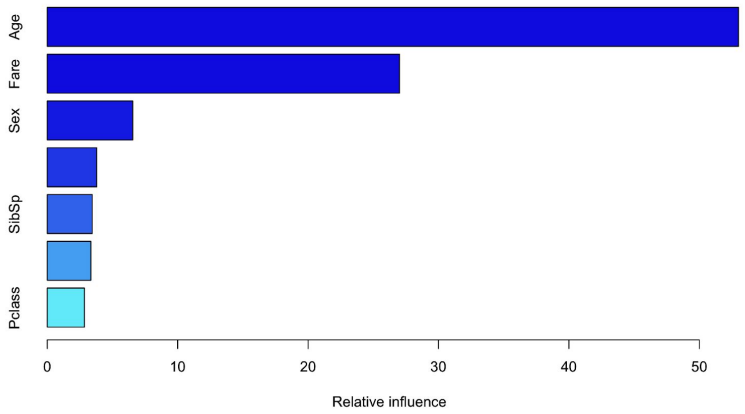
For the second classification problem, the error rate for test is: 0.24, while using two hidden layers with 5 and 3 neurons.

Reference: https://www.r-bloggers.com/fitting-a-neural-network-in-r-neuralnet-package/

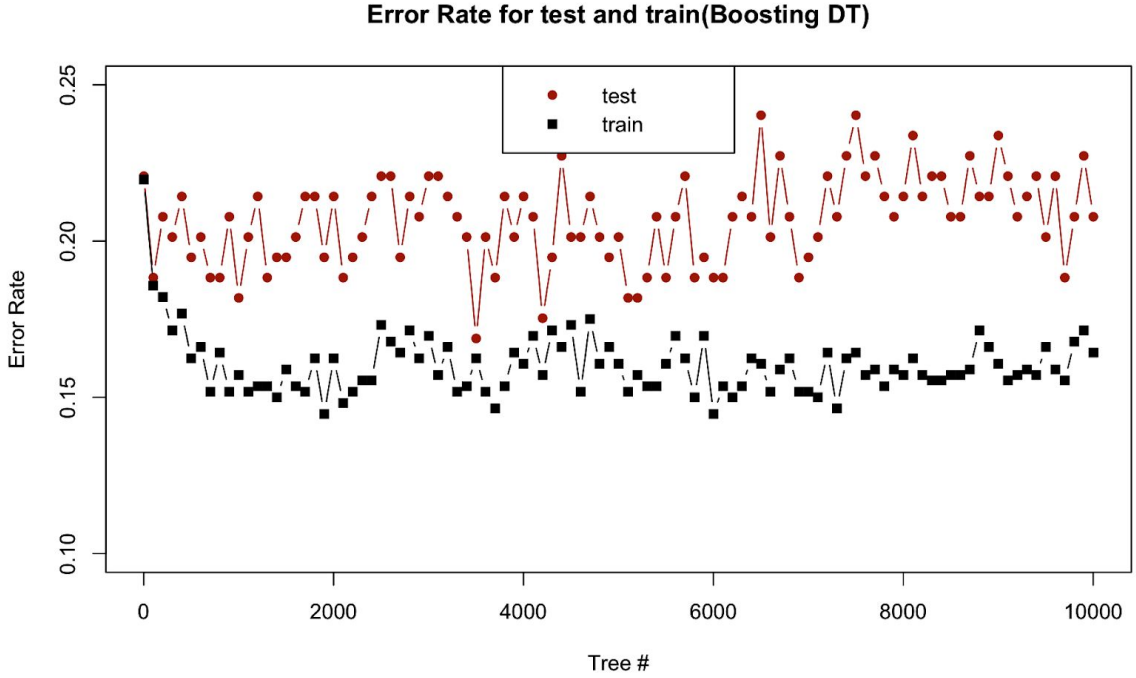
3. Boosting (Gradient boosting with decision tree)

Gradient boosting first builds learner to predict the output, and calculate the loss. Then, it will build a second learner to predict the loss after the first step. This step will continue until a certain threshold. Both train and test data are prepared as described in the first page.

I first evaluated the impact of different factors, and the figure below showed that **Age, Sex, and Fare** were the top three important factors. However, after several test, this combination has the lowest error rate: **Sex + Original_Age + First + Second + Third, which is the same with the previous decision tree.**



I tried different tree number for boosting to impreve, but didn't purne the tree (no effect in DT section). The result changed from time to time, so the lowest value here may be different with what you get. And this is the lowest error after several test: **minimum error rate for test: 0.169, tree number: 3501. The minimum error rate for train: 0.145, tree number: 1901.** Both train and test result are close to the lowest value after a certain round but show a perturbation trend.



For the second classification problem, the error rate for test is about 0.205, when # of trees is 1901.
Reference: https://www.r-bloggers.com/gradient-boosting-in-r/

4. SVM

SVM covert the classification to a convex problem to find the maximum margin. I will use R package ksvm in the assignment requirement (for sufficently loose definition of implement including “downloading”). SVM required binary dataset and all input should be scaled at first. Data quality is crucial for SVM method.

To start, I tried a very complex model Survived ~ Age + SibSp + Parch + Fare + First + Second + Third + Male + Female + EmbarkedC + EmbarkedQ + EmbarkedS, and the error rate for test was 0.188.

Interestingly, after removing the EmbarkedC, EmbarkedQ, EmbarkedS, the result didn’t change. For SVM, these three factors don’t have weight at the beginning. The training error rate is 0.164, slightly lower than that of test.

I also tried different kernels, and the error rate for test are:

- Linear: 0.221.
- Polynomial: 0.188
- Radial: 0.188
- Sigmoid: 0.39

The two best kernel are polynomial and Radial (gaussian), and it makes sense because the input data are following a kind of gaussian and polynomial distribution. For example: age follows a distribution of gaussian, and gender follows polynomial. I don’t have any figure here.

For the second classification problem, SVM also works well:
First, you can predict the embarked location just by the custom information. The error rate for testing data is about 0.24 for polynomial kernel, the error rate for radical is 0.22.

Reference: R in action.

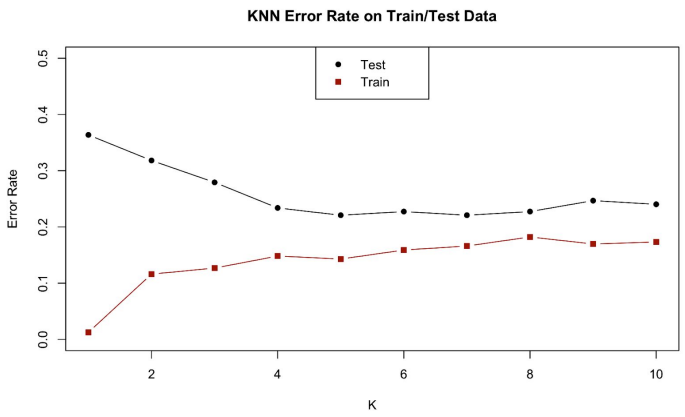
5. *k*-Nearest Neighbors (KNN)

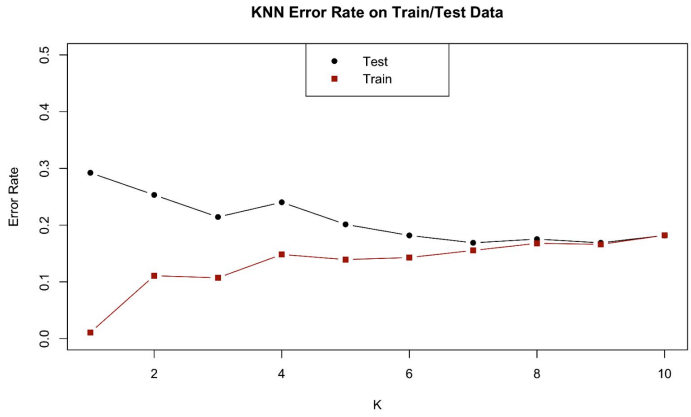
KNN is a non-parametric classification technique based on similarity measure (eg. distance).

First, all dataset are converted to binary data and re-scaled with the methods of (x-min)/(max-min). *Eclucide* distance is preferred here for the distance measure.

To start, I tried all the informative factor of training data and compared the performance of different K. For the testing data, the minimum error rate decrease and keep stable when K>=5. For the training data, the minimum error is close to zero when K = 1. This makes sense because KNN is actually clustering the point with itself and predict the output, therefore, the error rate is very low.

To get better performance, I gradually removed the factors in the training data and found the best result. The minimum error was produced by this combination: Survival ~ Age + Fare + First(cabin) + Second(cabin) + Third(cabin) + Male + Female. **The best K for testing data is: 7 or 9, with the minimum error rate of 0.169. The best K for training data is 1, and the error is 0.0107.** Factors like Sibsp, Parch (siblings or parents number), and Embarked areas may have some contribution to the training data, but may bring a little over-fitting on the training.





Factor used in the left figure: "Age","SibSp","Parch","Fare","First","Second","Third","Male","Female","EmbarkedC","EmbarkedQ","EmbarkedS"
Factor used in the right figure: "Age","Fare","First","Second","Third","Male","Female"

For the second classification problems, I got the error rate for test of 0.182 when K = 5.

Reference: http://rpubs.com/Nitika/kNN_Iris