Georgia Institute of Technology
CS 7641: Machine Learning

Assignment 1 – Supervised Learning

NBA games results and XXX analysis by using five supervised learning algorithms

Name: Tianyu Yang
GTid: 903645962
Date: 2020/9/20

1. Abstract

In this supervised learning project, we will try to design two interesting problems to have a test on five classification algorithms we learned from the course. In this report, I will give the introductions of five algorithms and give the result and output graph of F1 score, training time and predicting time of each algorithm. Last but not least, I will give the comparison of each algorithm and show that which one has the best performance we used on these two datasets.

Codes and related files are uploaded in Github. Please feel free to visit:
https://github.com/simonyang0701/Supervised-Learning.git

2. Introduction

To practice the supervised learning which I studied from the course, I put forward two classification problems to have a test on these algorithms.

The first problem is to use the detailed information of the NBA games to predict the result of a game (win or lose). The reason why I select this problem is that I love NBA and hope that if there is a method to analyze the factors that might affect the result of the game, such as the percentage of shooting balls, or some statistical analysis of a match, such as assists and rebounds. These are all attributes of a match that could use these data to build a model to predict the result.

Another problem is to use LOL games data to analyze which team will win based on some essential information delivered in a game. Like first blood. First tower, first Baron, first Dragon and so on, these will affect the result of a game to a great extent. So that if we know that which team get the first one that I mentioned before, we can predict the result of a game in a high probability. The reason why I am interested in this problem because I am a big fun of LOL games and I hope that if I can build up a model to predict the result of games. Besides, that will also help me know that which is the key factor of a game if I want to improve the possibility of winning a game.

The reason why I use these two datasets is that the outputs of these problems are binary, which is just win or lose. It could be used for supervised learning to use some classification algorithms to run the model, such as Decision Tree, Neural Networks, Boosting Tree, SVM or kNN. Therefore, in this project, I will test on five algorithms based on two datasets and give the analysis of the training result. I will show the training and testing error rates in the Result section. I will give the graph to show the performance of both training and testing data as the comparison. Besides, I will analyze the result of five algorithms and make a comparison based on training time and predicting time. Based on these comparisons, we can conclude that which of these algorithms has the best performance of the datasets.

3. Methodology

3.1 Supervised Learning

Supervised learning is a kind of machine learning method to map an input to an output based on huge input and output pairs.[1] It needs to label the training samples. It requires the learning algorithms to generalize from the training datasets within inductive bias.

3.2 Decision Tree

Decision trees are non-parametric supervised learning method which is used for classification and regression.[2] This algorithm is to make a model to predict the output value of a variable by learning step by step from some simple rules inferred from the data features.

3.3 Neural networks

A neural network is a network or circuit of neurons, or in a modern sense, an artificial neural network, composed of artificial neurons or nodes.[3] Thus, a neural network is either a biological neural network, made up of real biological neurons, or an artificial neural network, for solving artificial intelligence (AI) problems. It is to use the connections of biological neurons to build up a model from the input layers into output layers.

3.4 Boosting

In machine learning, boosting is an ensemble meta-algorithm for primarily reducing bias, and variance in supervised learning, and a family of machine learning algorithms that convert weak learners to strong ones.[4]

3.5 Support Vector Machines

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.[5] The advantage of this algorithm is that it is effective in high dimensional spaces. Besides, it is still effective in cases where number of dimensions is greater than the number of samples. What's more, it uses a subset of training points in the decision function so that it is memory efficient.

3.6 K-nearest neighbors

In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method proposed by Thomas Cover used for classification and regression.[6] In both cases, the input consists of the k closest training examples in the feature space. k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until function evaluation. Since this algorithm relies on distance for classification, normalizing the training data can improve its accuracy dramatically.

4. Result

4.1 Datasets

I divided the data into two sets which testing size is 0.2. To make a comparison, the counts of samples I used are both 965. Here is the dataset analysis of NBA games and LOL games.

NBA games:

	HOME_TEAM_WINS	FG_PCT_diff	FT_PCT_diff	FG3_PCT_diff	AST_diff	REB_diff
count	965.00000	965.000000	965.000000	965.000000	965.000000	965.000000
mean	0.54715	0.011031	0.001882	0.005033	1.174093	0.839378
std	0.49803	0.078655	0.146005	0.131647	6.439211	9.418250
min	0.00000	-0.208000	-0.571000	-0.427000	-22.000000	-39.000000
25%	0.00000	-0.043000	-0.092000	-0.086000	-3.000000	-5.000000
50%	1.00000	0.009000	0.000000	0.003000	1.000000	1.000000
75%	1.00000	0.068000	0.094000	0.098000	5.000000	7.000000
max	1.00000	0.258000	0.571000	0.458000	23.000000	31.000000

LOL games:

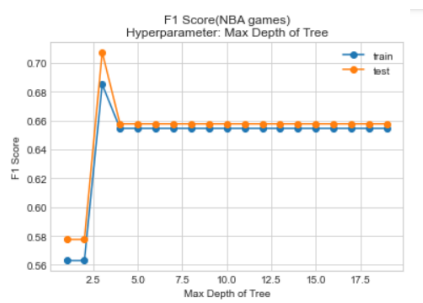
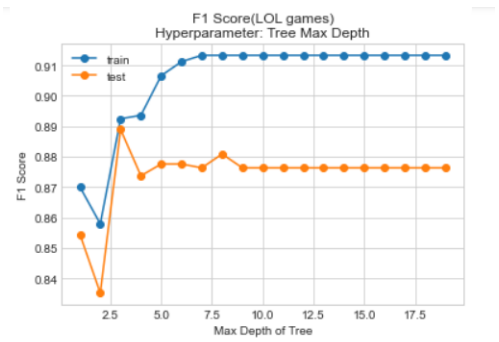
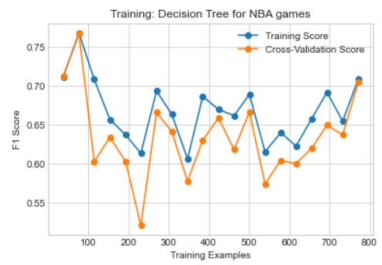
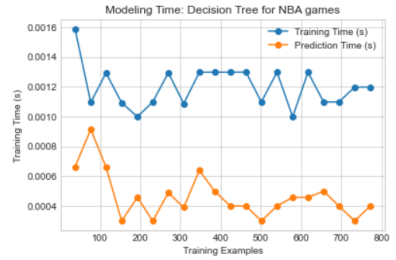
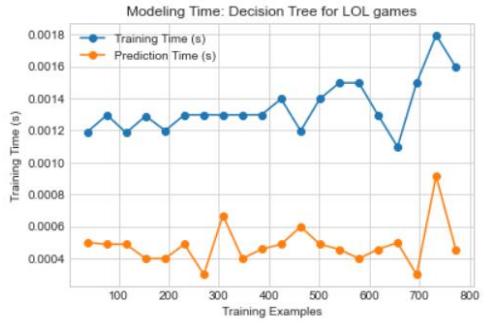
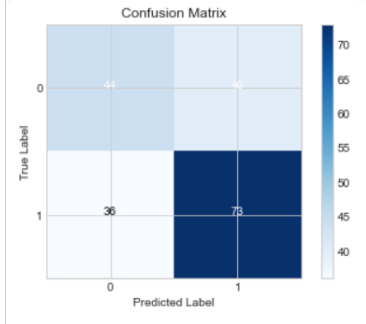
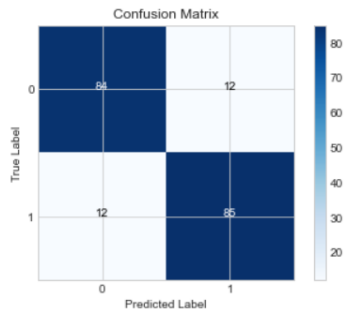
	winner	firstBlood	firstTower	firstInhibitor	firstBaron	firstDragon	firstRiftHerald
count	965.000000	965.000000	965.000000	965.000000	965.000000	965.000000	965.000000
mean	1.491192	1.469430	1.443523	1.304663	0.890155	1.441451	0.757513
std	0.500182	0.511637	0.544848	0.663535	0.832656	0.572486	0.821936
min	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000	0.000000
50%	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
75%	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	1.000000
max	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000

4.2 Decision Tree

In order to get a better result, I adjusted the max depth of the tree into 20 and get the model F1 score graph as pre-training graph.

To get the best parameter for Decision Tree model, I ran the Decision Classifier again and again to get the best max depth and min sample leaf. After that I ran the model again and get the report for F1 Score as best training. To evaluate the result, I use the

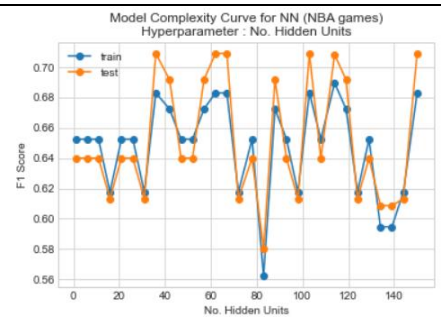
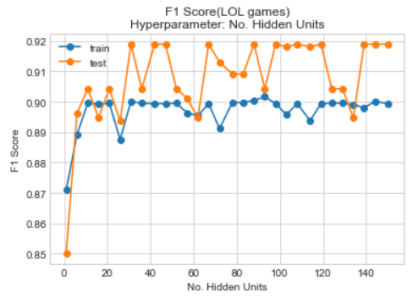
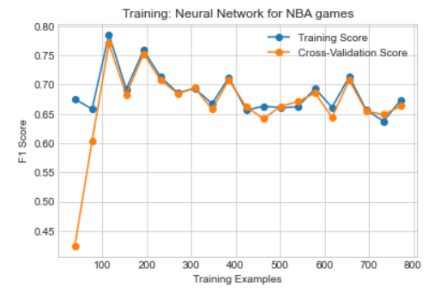
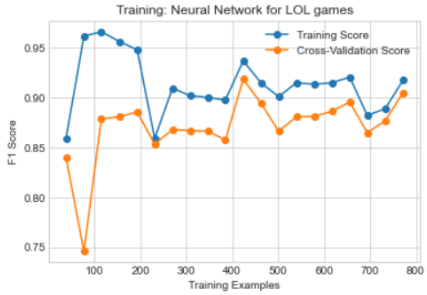
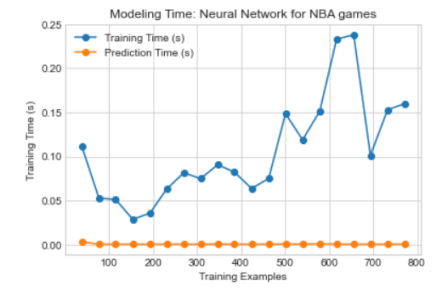
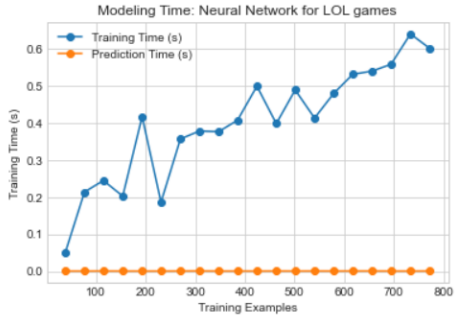
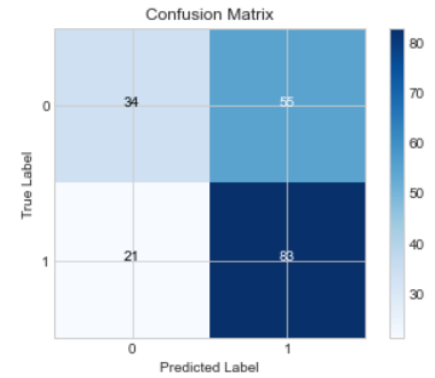
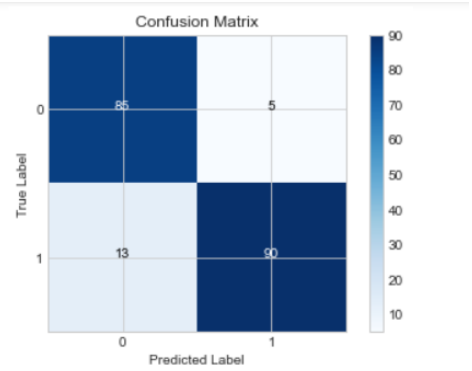
timer in Python to get the fitting time and predicting time for further analysis. The model training time graph and confusion matrix figure are as below:

Datasets	NBA games	LOL games
Pre-training		
Best training		
Training time		
Confusion matrix		

4.3 Neural networks

Similar to Decision tree, I also plot the same figure for Neural Networks as below:

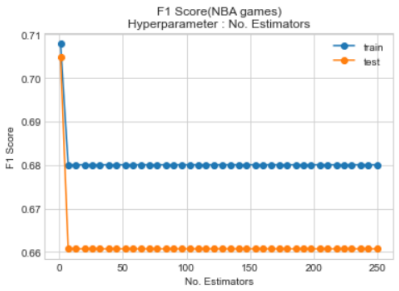
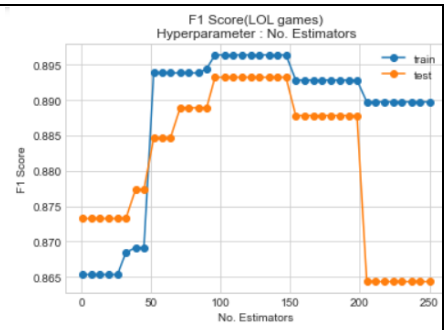
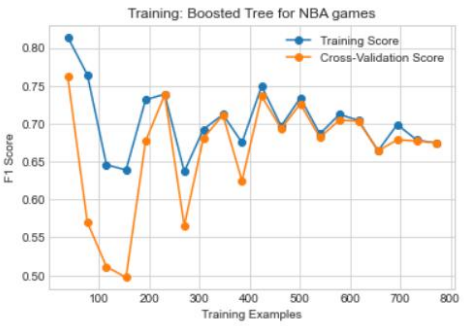
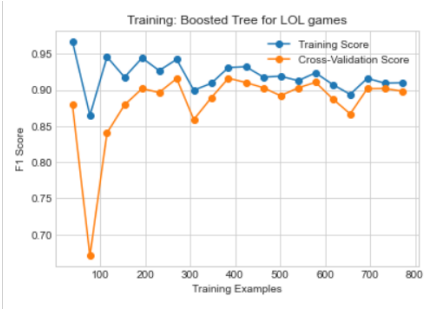
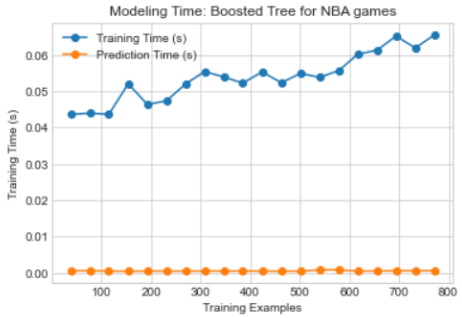
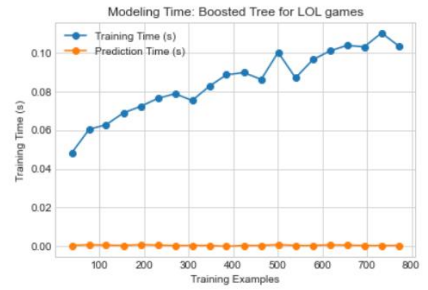
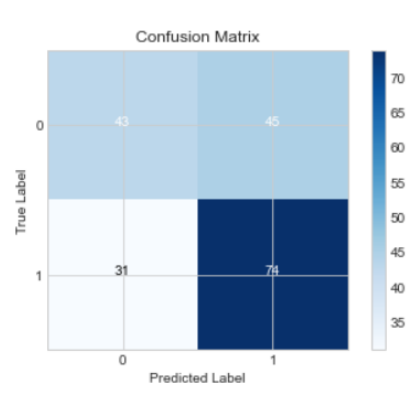
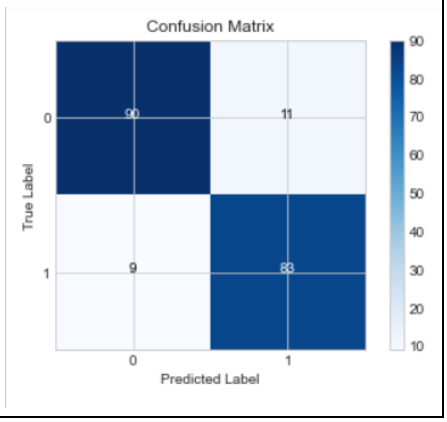
Datasets	NBA games	LOL games

Pre-training		
Best training		
Training time		
Confusion matrix		

4.4 Boosting

Similar to Decision tree, I also plot the same figure for Boosting Tree as below:

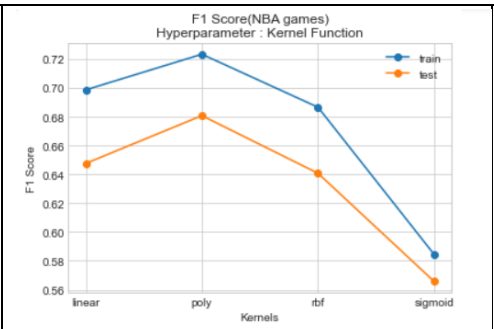
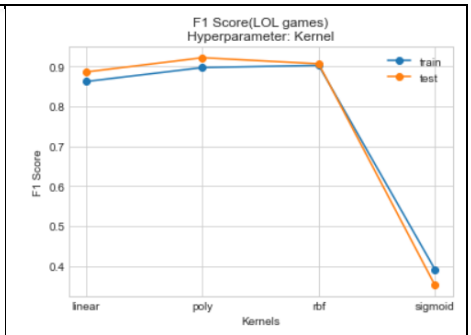
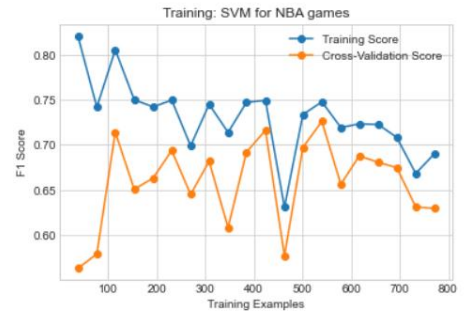
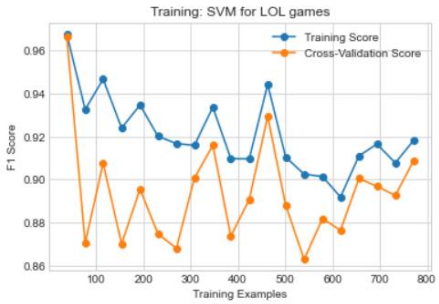
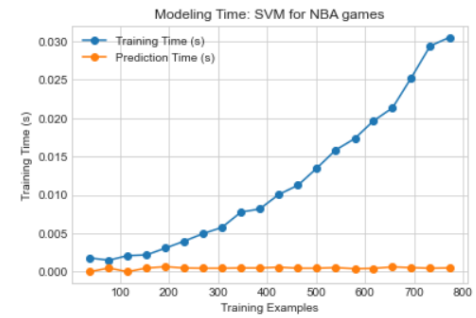
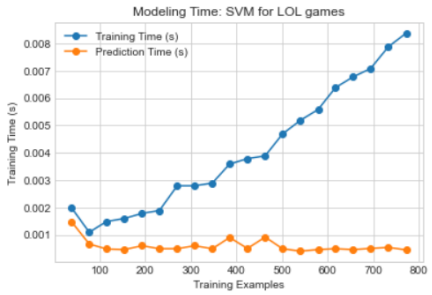
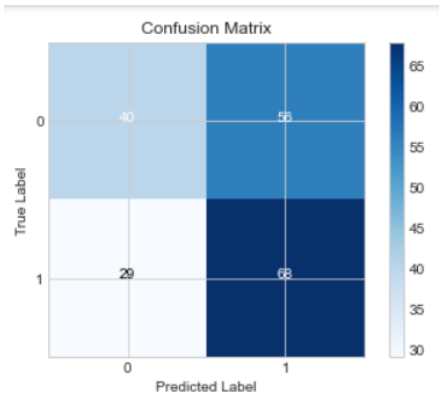
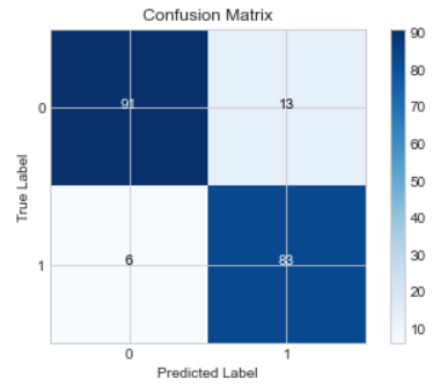
Datasets	NBA games	LOL games

Pre-training		
Best training		
Training time		
Confusion matrix		

4.5 Support Vector Machines

Similar to Decision tree, I also plot the same figure for Support Vector Machine as below:

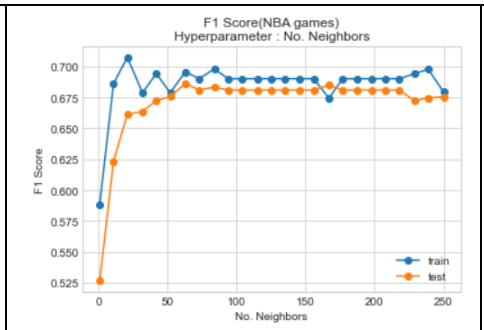
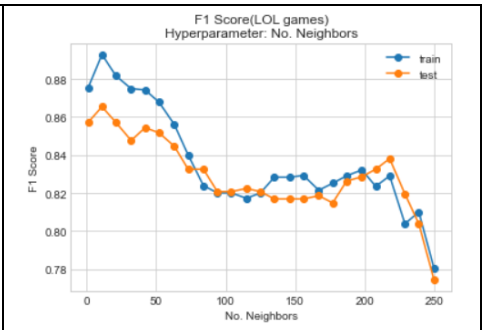
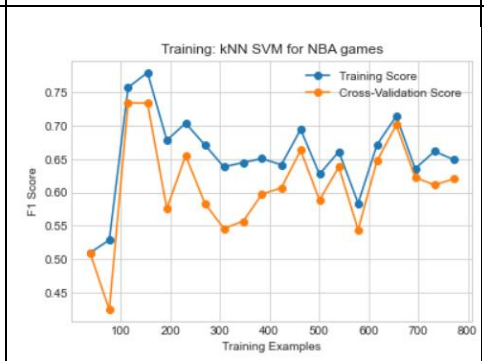
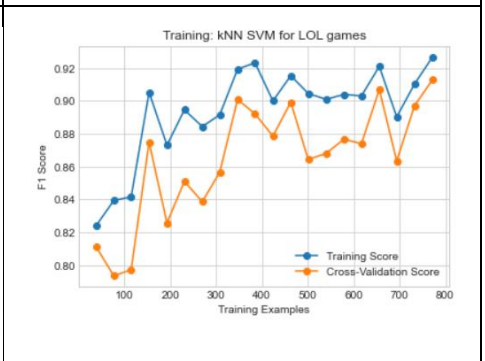
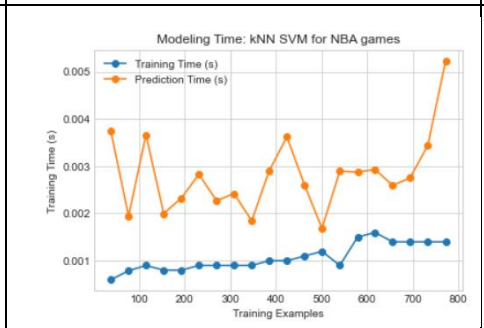
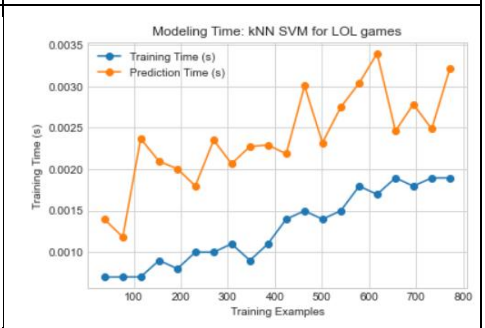
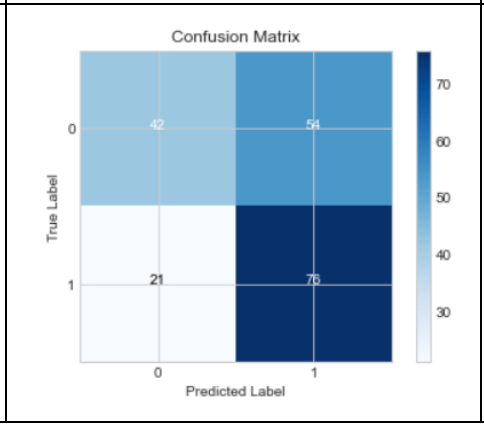
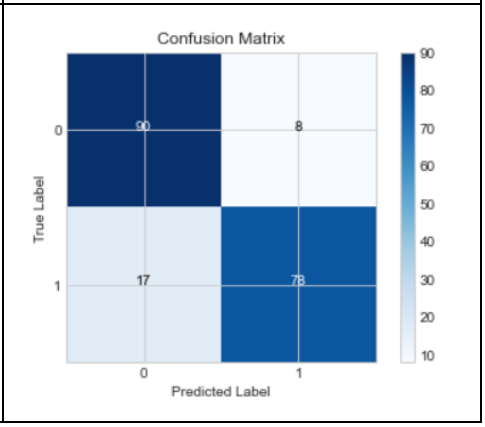
Datasets	NBA games	LOL games

Pre-training		
Best training		
Training time		
Confusion matrix		

4.6 K-nearest neighbors

Similar to Decision tree, I also plot the same figure for k-nearest neighbors as below:

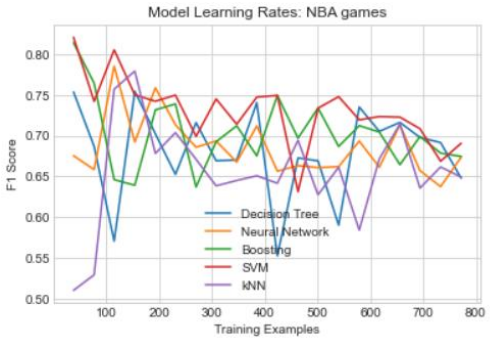
Datasets	NBA games	LOL games

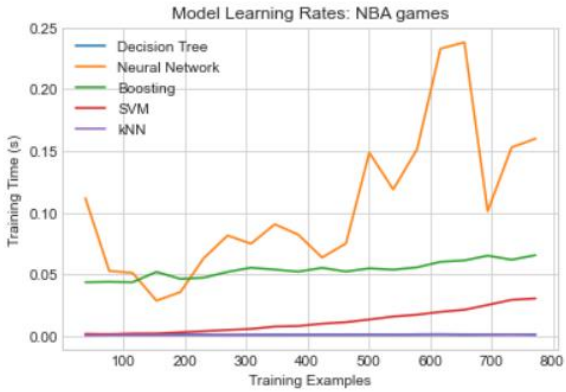
Pre-training		
Best training		
Training time		
Confusion matrix		

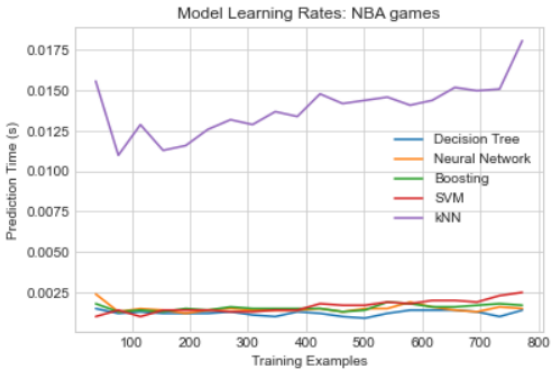
4.7 Comparison of five supervised learning algorithms

After plotting all the graphs for five algorithms, we made a comparison of these five algorithms for F1 Score. Besides, we also calculate the fitting time and predicting time to compare their efficiency.

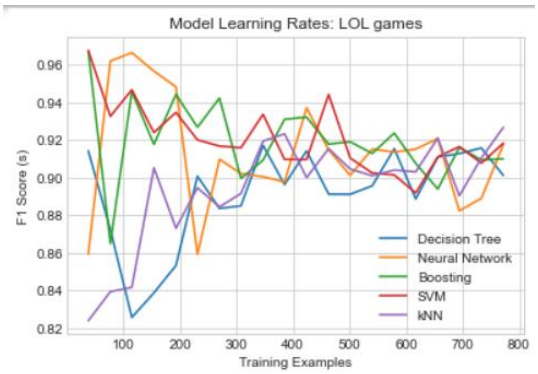
NBA games:

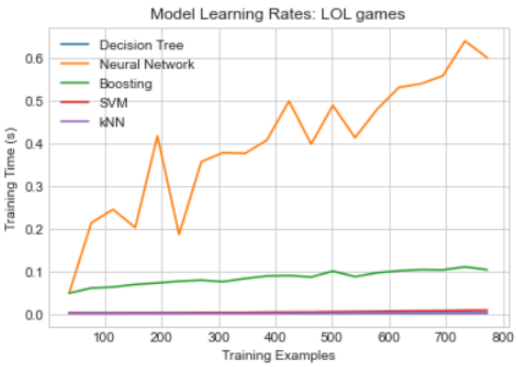


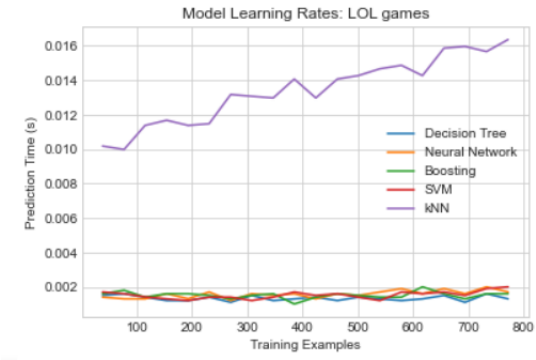




LOL games:







5. Conclusion

From the graph above, we can conclude that for NBA datasets, for all five algorithms, when the sample size is around 750, the F1 scores for all the these are between 0.65-0.70, which means that the results are acceptable. For the model learning time, Neural Networks has the best performance and kNN is the worst. This is because for a small dataset and less attributes, Neural Networks are appropriate for training and kNN is not. As for the predicting time, kNN has the best performance, which means that comparing to other algorithms, kNN is better for prediction.

As for LOL games, we can conclude that for F1 Score, all of these five algorithms are between 0.90 to 0.94, which has a better performance than NBA games. This is because for LOL dataset, the attributes are binary, which means that the data of each column is team 1 or team 2. But in NBA, the attributes are numbers. That's why for the same sample size, LOL games have a better performance. For fitting time and predicting time, this dataset gets the same result as NBA games. Neural Networks has the best training time performance and kNN has the best predicting time performance.

As a result, in this project, we test two different datasets with different features of columns on five popular classification algorithms. We learned a lot from the result and output graphs. In addition, we compare with the algorithm performance so that in the future, when we need to use the supervised learning model, we will choose the best algorithms based on the datasets we use. This is really helpful for the learning and practicing of machine learning.

6. Reference

Datasets comes from Kaggle
(1) NBA games https://www.kaggle.com/nathanlauga/nba-games
(2) LOL games https://www.kaggle.com/datasnaek/league-of-legends?select=games.csv
[1] Stuart J. Russell, Peter Norvig (2010) Artificial Intelligence: A Modern Approach, Third Edition, Prentice Hall ISBN 9780136042594.
[2] https://scikit-learn.org/stable/modules/tree.html
[3] Hopfield, J. J. (1982). "Neural networks and physical systems with emergent collective computational abilities". Proc. Natl. Acad. Sci. U.S.A. 79 (8): 2554–2558. Bibcode:1982PNAS...79.2554H. doi:10.1073/pnas.79.8.2554. PMC 346238. PMID 6953413.
[4] Leo Breiman (1996). "BIAS, VARIANCE, AND ARCING CLASSIFIERS" (PDF). TECHNICAL REPORT. Archived from the original (PDF) on 2015-01-19. Retrieved 19 January 2015. Arcing [Boosting] is more successful than bagging in variance reduction
[5] Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks" (PDF). Machine Learning. 20 (3): 273–297. CiteSeerX 10.1.1.15.9362. doi:10.1007/BF00994018. S2CID 206787478.
[6] Altman, Naomi S. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression" (PDF). The American Statistician. 46 (3): 175–185. doi:10.1080/00031305.1992.10475879. hdl:1813/31637.