

**CSE 477(1) Project**

***Comparative analysis of different classification algorithms***

Submitted to:

Mohammad Rezwanul Huq, PhD

Assistant Professor

Department of Computer Science and Engineering

Submitted by:

Abeda Sultana Khanam

I.D: 2016-1-60-011

Course code: CSE477

Course title: Data Mining

Section: 1

Date: 19 August 2019

# 1. Introduction

Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. In this paper, we focused on four classification algorithms. We applied these algorithms on a dataset called Chess King Rook. Our target is to analyse which algorithm works well in our dataset. Comparative analysis has been carried out using the performance evaluation measures of accuracy, precision and F-measure.

# 2. Classification Algorithms

**Naïve Bayes Classifier**

The Naive Bayesian classifier is based on Bayes’ theorem. It is a probabilistic algorithm, used in a wide variety of classification tasks.it gives great results, especially in Natural Language Processing. The class with highest probability is considered as the most likely class. It can handle nominal, numeric and binary data. It is a popular method for text categorization.

**Decision Tree Classifier**

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, decision tree algorithm can be used for solving regression and classification problems. The decision tree algorithm tries to solve the problem, by using tree representation. Each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label. In decision trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with record’s attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

**K Nearest Neighbors Classifier**

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection. It can be used for classification and regression both. KNN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification.

**Random Forrest**

Random forests is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

**Support Vector Machine**

SVM is a supervised machine learning algorithm which can be used for classification or regression problems. The objective of the support vector machine algorithm is to find a hyper plane in an N-dimensional space (N — the number of features) that distinctly classifies the data points. Data points falling on either side of the hyper plane can be attributed to different classes. Finding a plane that has the maximum margin, i.e. the maximum distance between data points of classes gives the best result because maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

# 3. Dataset

Our dataset is to predict outcome of chess with two kings and one rook. The dataset consists of 28056 rows and 7 columns. There are 7 categorical attributes and no missing value. The 'result' attribute can be used as the class label. There are eighteen unique class value. It is a multiclass dataset.

Categorical data has been encoded into numeric value using one hot encoder. Class label is converted to numeric data through label encoder. Since there is no missing value, we didn’t keep any function for it.

# 4. Implementation

The classifiers are built using python and using weka tool. In python scikit learn library was used to build the classifier. A few other additional libraries were used and the code was written in jupyter notebook editor.

Screenshots of executing algorithms in Weka Tool

**Naïve Bayes Classifier**

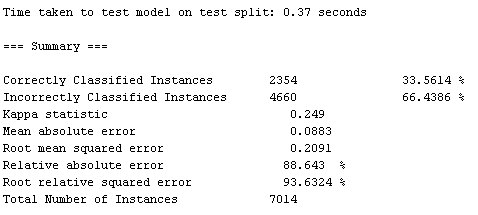
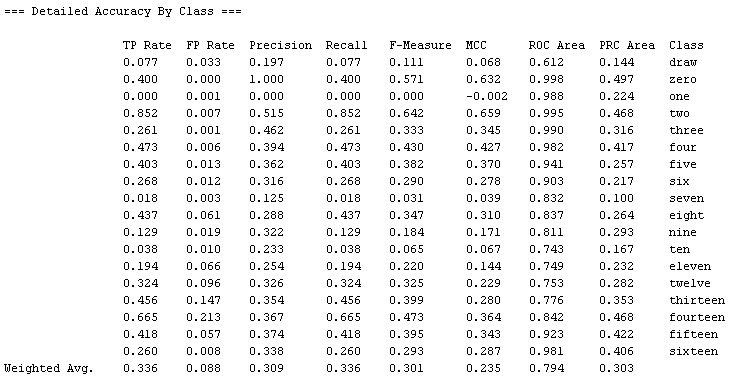


Figure: Accuracy of Naïve Bayes Classifier



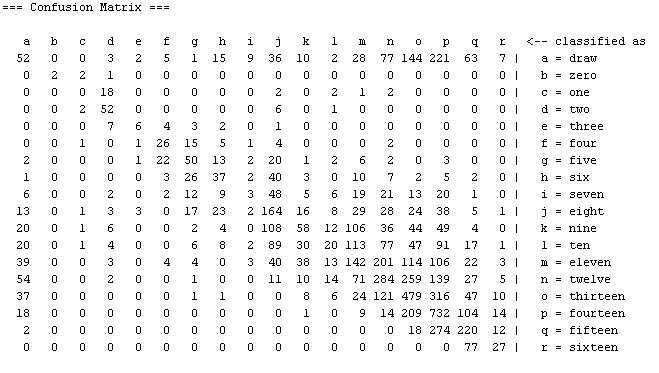
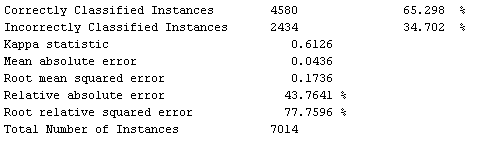
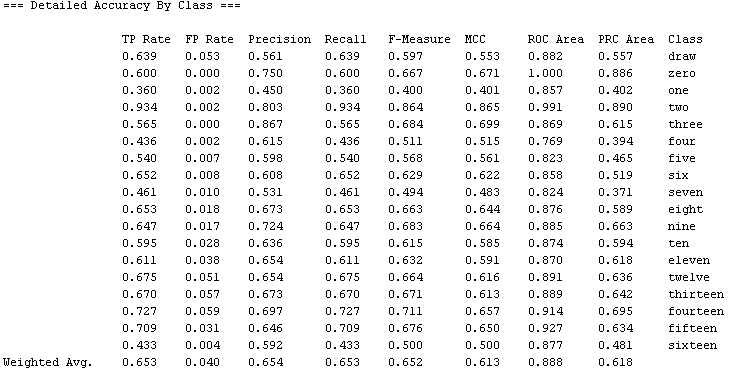


Figure: Naïve Bayes

**Decision Tree Classifier**





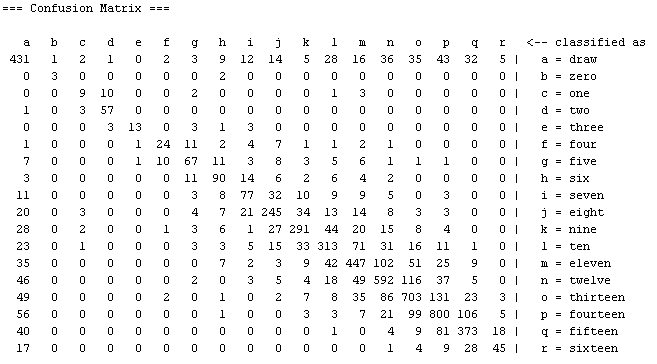
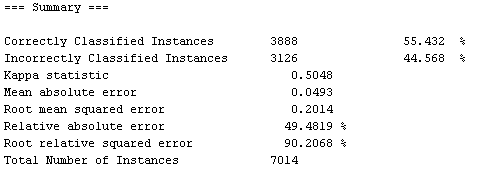
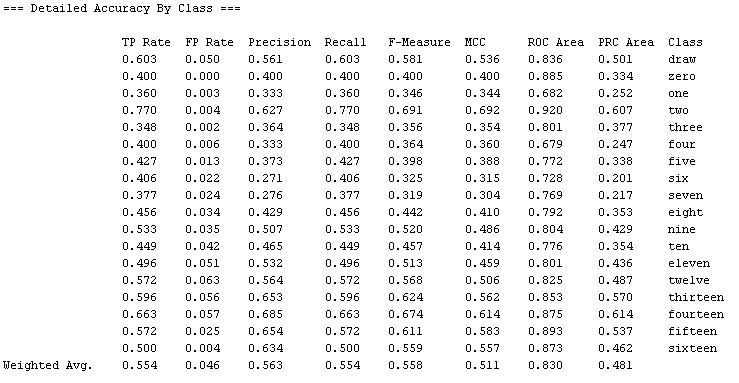


Figure: Decision Tree

**K Nearest Neighbors Classifier**





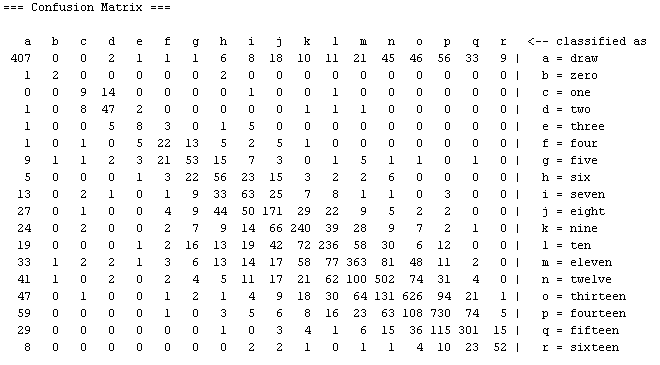
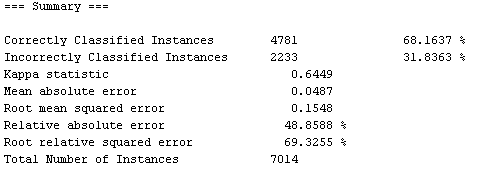
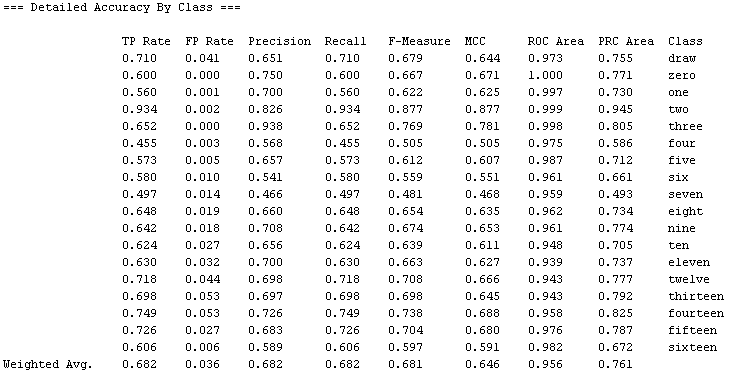


Figure: K Nearest Neighbors

**Random Forrest**





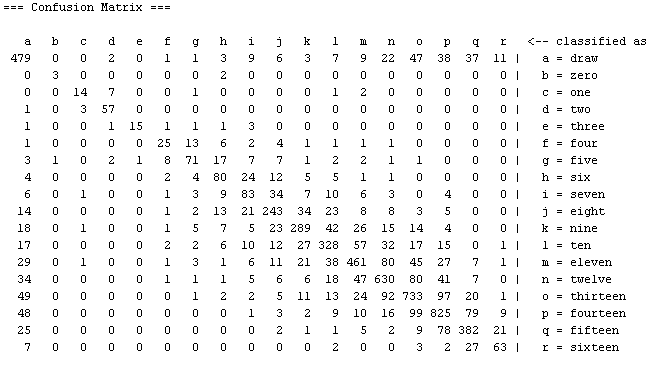
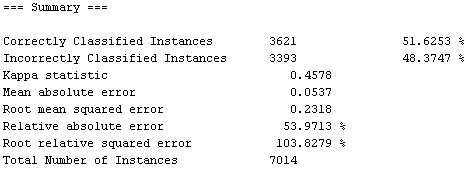
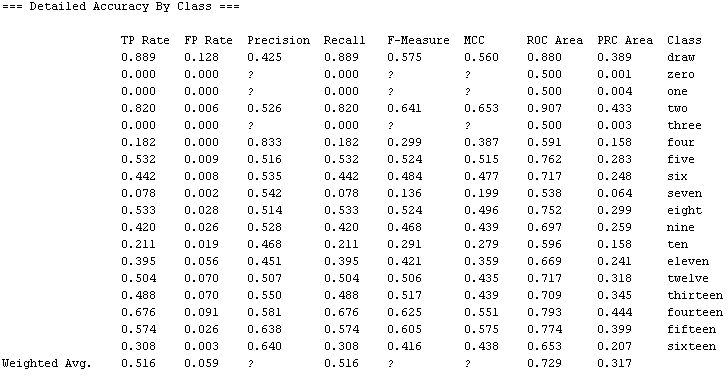


Figure: Random Forest

**Support Vector Machine**





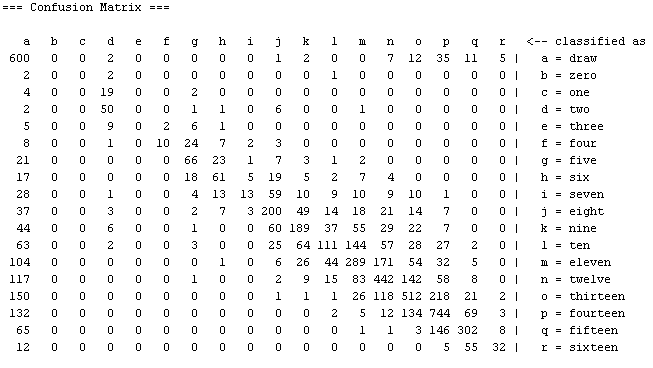


Figure: Support Vector Machine

# 5. Performance Evaluation

| **Algorithm** | **Accuracy** | **TPR/Recall** | **FPR** | **F-measure** | **Precision** |
| --- | --- | --- | --- | --- | --- |
| **Naïve Bayes** | 0.364 | 0.28 | 0.039 | 0.29 | 0.37 |
| **Decision Tree** | 0.785 | 0.78 | 0.013 | 0.79 | 0.81 |
| **KNN** | 0.511 | 0.48 | 0.030 | 0.47 | 0.47 |
| **Random Forest** | 0.672 | 0.68 | 0.020 | 0.66 | 0.66 |
| **SVM** | 0.666 | 0.54 | 0.021 | 0.55 | 0.63 |

Table 1**:** Performance Comparison of the five algorithms based on Accuracy, TPR, FPR, F-measure, and Precision

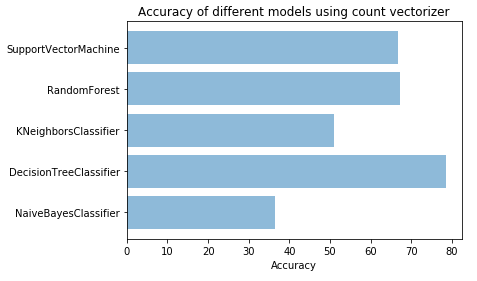


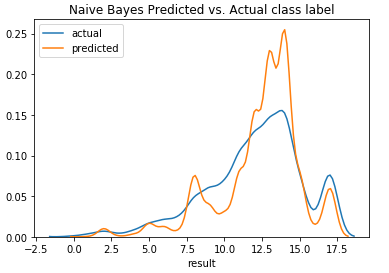
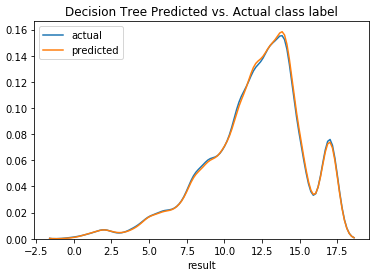
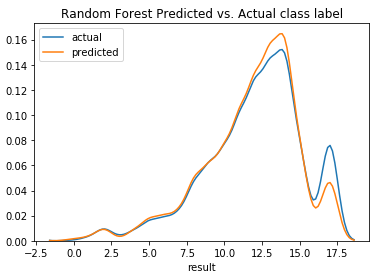
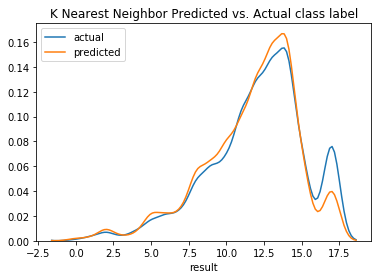
Figure: Accuracy level of different classifier on the given dataset

Figure: Naïve Bayes Predicted vs. Actual class label Figure: Decision Tree Predicted vs. Actual class label

Figure: K Nearest Neighbor Predicted vs. Actual class Figure: Random Forest Predicted vs. Actual class label label

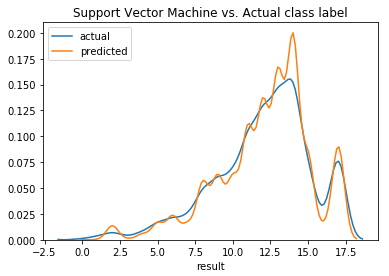


Figure: Support Vector Machine Predicted vs. Actual class

label

# 6. Discussion

Looking at the table.1 we can see Decision Tree Classifier topped all of the criteria we used to compare. So we can conclude that Decision Tree did the best job for the selected dataset. It is the most accurate model of them all with an accuracy rate 78.5%. From the univariate plotting of train class label and test class label, we can see that Decision Tree Classifier have done great job.

Random Forest comes in second with 67.2 percent accuracy rate. Support Vector Machine and Random Forest gives almost identical result in our dataset.

Naïve Bayes has the worst accuracy rate among the five classifier algorithm. Its False Positive Rate is also higher among all of the classifiers.

All the measurements and graph is pointing us to Decision Tree Classifier. It is the best performing algorithm in our dataset.

# 7. Conclusion

In this project we used five different classifier based on five different algorithm in Python and Weka tools. Classifiers attempt to predict with the location of two kings and one rook, whether the game is going to be a draw or result in different position. Since it was a multi class problem (18 class label), different model showed different level of accuracy. Decision Tree model did fairly well on average and topped all of the five criteria.