

**East West University**

A project report on

***“Comparative analysis of different classification algorithms”***

**Course Title:** Data Mining

**Course Code:** CSE477

**Section:** 1

**Submitted To:**

Dr. Mohammad Rezwanul Huq

Assistant Professor  
Dept. of Computer Science & Engineering

**Submitted By:**

Shaykh Siddique

ID: 2016-1-60-053

**Introduction**

We have to predict whether the balloon will be inflated or not with some predictors of balloons. To predict the target class we will use different classification like – Naive Bayes Classifier, Decision Tree Classifier, K Nearest Neighbor Classifier, Random Forest Classifier and Support Vector Machine. After implementing the Python codes for this algorithms, we used Weka data tool to compare the performance and efficiency.

**Classification Algorithms**

In data mining classification is a supervised learning approach to predict/classify new observation from the input dataset. We make some boundaries with conditions to determine the target feature.

**Naive Bayes Classifier:**

Naive Bayes classification is a classification technique which defines the Bayes’ Theorem.

We can find the conditional probability of each predictors from the dataset. And train the machine from that using Bayes’ Theorem. Predictors will be independent.

**Decision Tree Classifier:**

Decision Tree is a machine learning classification technique which splits the population into two or more homogeneous sets based on the most significant attributes making the groups as distinct as possible. The top most decision is the best predictor. We have to build some decision/rules and with the rules the machine trains a decision tree. Every time of prediction the machine will compare the record with that tree.

**K Nearest-Neighbor Classifier:**

K Nearest Neighbor is a takes some points, calculate distance from other points and assign votes for each class points. If the value of K = 1 then, it will be the class of that single nearest neighbor. This algorithm is non-parametric and follows lazy learning approach.

After calculating the similarity it predicts the target class where belongs to.

**Random Forest Classifier:**

Random forest classifier takes the multitude of Decision Tree at the training time and predict the target class from that. Random forest correct for decision trees habit of over fitting to their training set.

**Support Vector Machine (SVM):**

In SVM, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes perfectly. The data points belong to one of two classes, and the goal is to decide which class a new data point will be in.

**Datasets**

We have 100 instances with 5 attributes. All the attributes are nominal variables. We used label encoding technique for encoding features as preprocessing of dataset.

**Implementation**

**Machine Configuration:**

* Processor: Intel® Core™ i5-5200U CPU @2.20GHz.
* Ram: 8GB.
* Windows Environment

**Using Tools:**

* Jupyter Notebook.
* Weka (version-3.8.3).

We split the dataset as 90% for training and 10% for testing. In Naive Bayes we calculate the conditional probability and train the machine.

**Screenshots:**

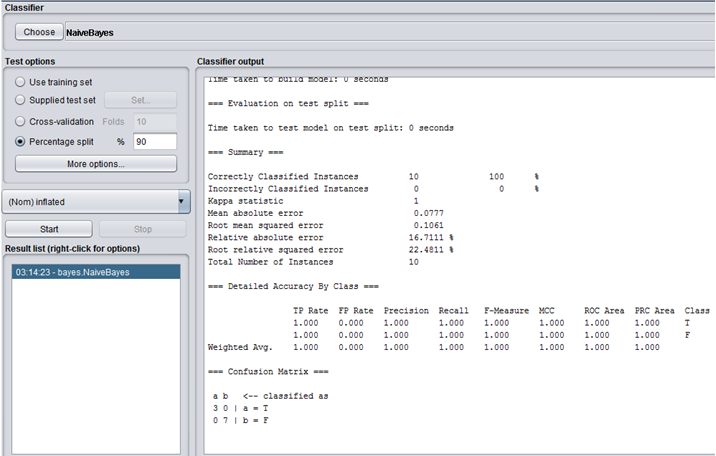


Figure: Weka – Naive Byes Classifier

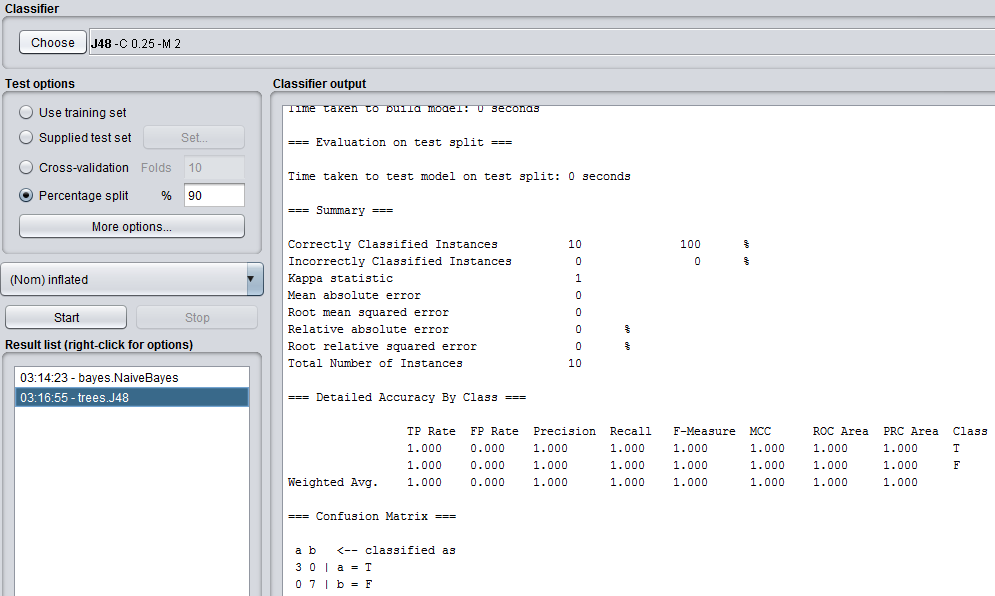


Figure: Weka – Decision Tree Classifier

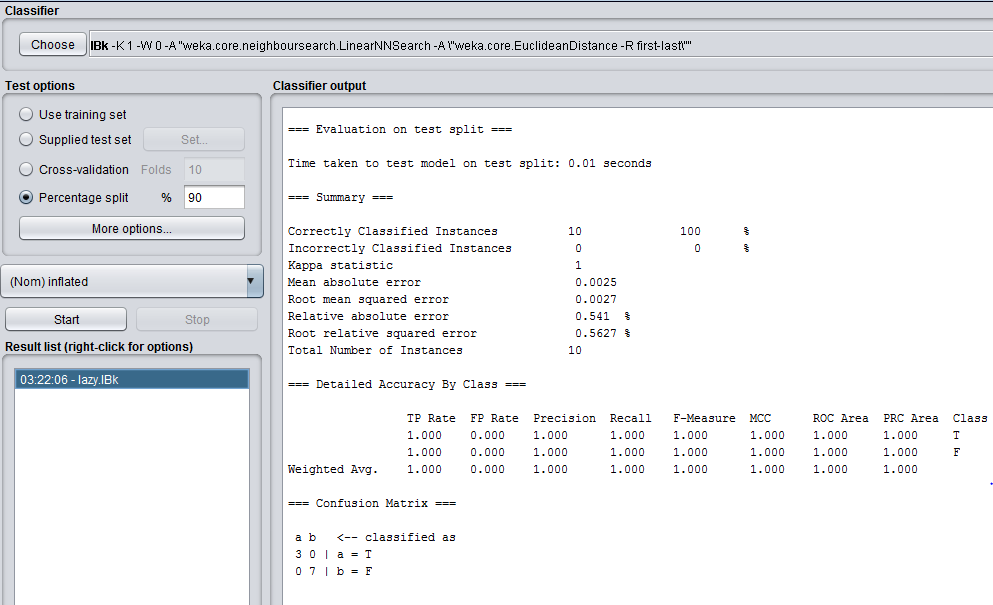


Figure: Weka – K Nearest Neighbor Classifier

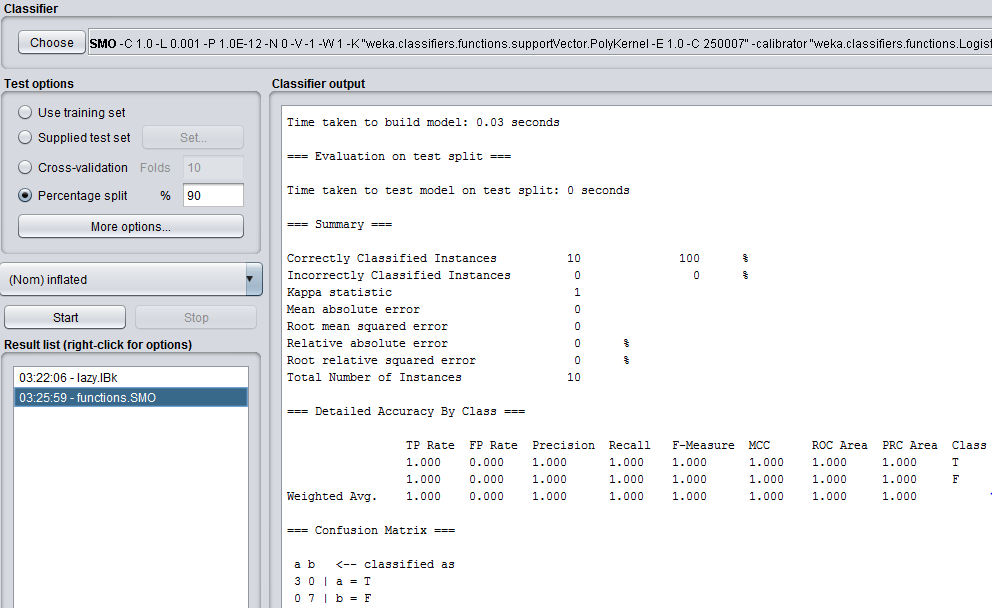


Figure: Weka – Support Vector Machine Classifier

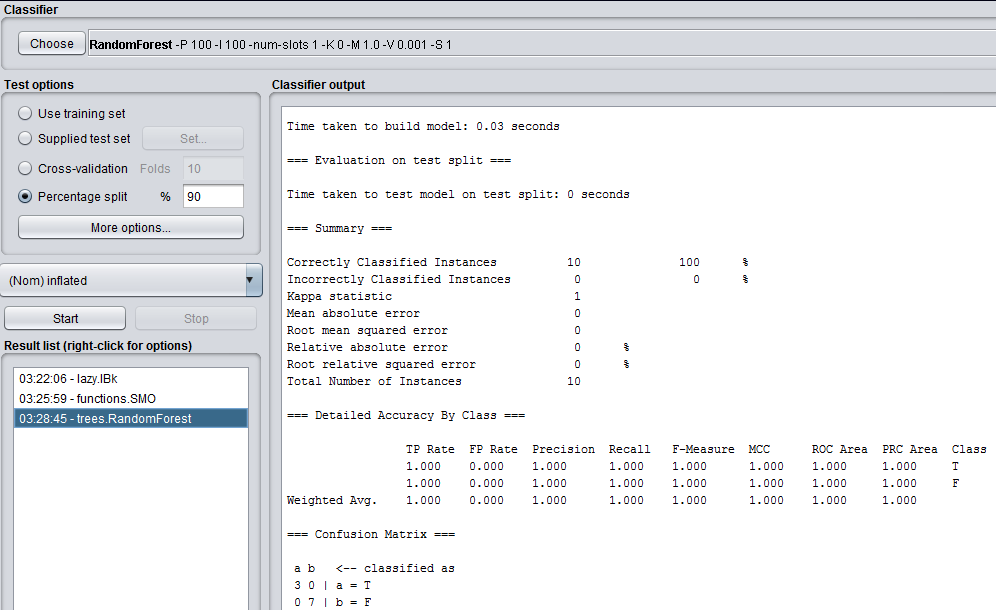


Figure: Weka – Random Forest Classifier

**Permormance Evaluation**

We used the dataset of 100 instances and can make perfect prediction for inflated of balloons.

So the performance table is given below:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name of classifier** | **TP Rate** | **FP Rate** | **Precision** | **Recall** | **F-Measure** | **MCC** | **ROC Area** | **PRC Area** | **Accuracy** |
| Naive Bayes | 1.00 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 100% |
| Decision Tree | 1.00 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 100% |
| K Nearest-Neighbor | 1.00 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 100% |
| Random Forest | 1.00 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 100% |
| Support Vector Machine | 1.00 | 0.00 | .00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 100% |

**Discussion**

With the 100 instance of 5 attribute we can find out the full accuracy for each algorithm. But with the measurement of efficiency and other performance Decision Tree works best.

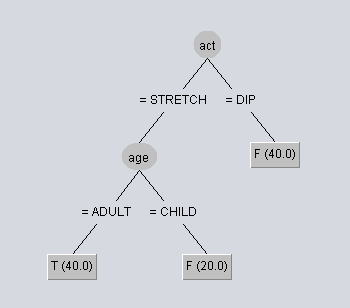


Figure: Weka- Decision Tree

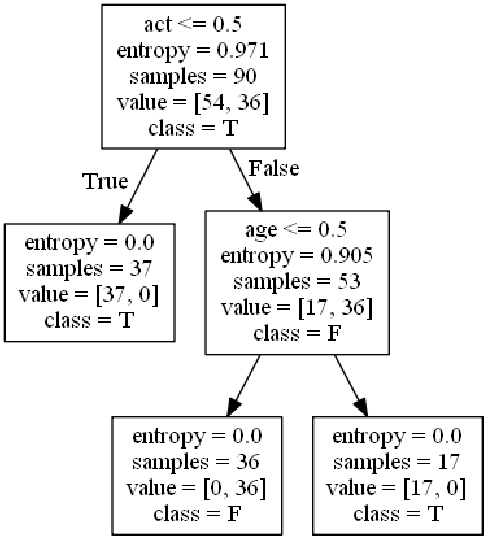


Figure: PythonCode- Decision Tree

We can easily take the act as best splitting criteria and the age will be the second best criteria. So after this two splitting we will find same target class nodes. And when predicting, we can easily find out the predicting class from the tree in a very short time.

**Conclusion**

The classification algorithms works supervised leaning datasets. It is something like prediction. So we cannot say a particular algorithm will be best for all kind of scenarios. But machine can predicts the best matches for each test datasets. But the use of classification increasing now and it becomes the most essential concept in Data Science.