CSE477(1)-Project Report

Comparative analysis of different classification algorithms

Submitted to:

Dr. Mohammad Rezwanul Huq

Assistant Professor, Dept. of CSE, EWU

Submitted by:

Nazmus Sakib Patwary (2015-2-60-092)

Md.Tanvir Rahman(2015-2-60-027)

Introduction

In modern life, Data have increased rapidly and using those data we can measure the prediction and we also can make a proper decision to execute some works where result may use for business purpose or may use for disease or other sectors. From the large number of dataset, we can find out a patterns and patterns help to take decision. The data analysis occurs by data mining. The Data mining is the process of sorting through large data sets to identify patterns and establish relationships to solve problems through data analysis. Data mining tools allow enterprises to predict future trends.

Here, we had worked with two datasets. For the python implementation, we had used one dataset and for the Weka tools we had used another dataset.

One dataset**[job-experience.csv]** about job experience, here have 26 attributes. Using those attributes, we have to measure weather employee are attrition or not. For this dataset we used Random Forest, Naïve Bayes, Support Vector, Decision tree using python.

Another dataset**[adult.csv]** about the personal information where have age, income, marital status, capital loss and etc. Here have 15 attributes, using those we have to measure weather a person income is greater than or equal to 50k. For this dataset we used Random Forest, Naïve Bayes, Support Vector, Decision tree using weka tools.

Classification Algorithm

We have used four algorithms to measure the result.

1. Decision Tree
2. Naïve Bayes
3. Random forest
4. Support Vector Machine

Decision Tree: A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal. A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes

the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node. This algorithm has the ability to handle binary and multiclass classification problem. We need two different entropies for calculating.one is for total database that means for all the attribute and the equation for that is,

$$\text{Info(F)} = - \sum_{i=1}^{m} pi \log_2(pi)$$

pi is the probability that an arbitrary tuple in F belongs to class Yes or No and another is for each attribute and equation for that is,

$$\text{Info}_{att}\text{(F)} = \sum_{j=1}^{v} \frac{|Fj|}{|F|} \times info(F)$$

After that total gain will be calculated. Here, the formula,

$$\text{Gain(attribute)} = \text{Info(F)} - \text{Info}_{att}\text{(F)}$$

Naïve Bayes: In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

By using the Bayes theorem, posterior probability can be calculated, P(c|x), from P(c), P(x), and P(x|c). The Naive Bayes classification consider the predictor value (x) on a given category (c) is independent which is conditional independence. So, we can formulate the theorem from Bayesian.

Here,

$$P(c|x_i) = p(x_1|c) \text{ x } \dots \text{ x } P(x_n|c) \text{ x } P(c)$$

P(c|xi) is the posterior probability of a given predictor (attribute). P(c) is the prior probability of class. P(xi|c) is the likelihood. P(xi) is the prior probability of predictor.

Random forest: Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. To say it in simple words: Random forest builds multiple

decision trees and merges them together to get a more accurate and stable prediction. One big advantage of random forest is, that it can be used for both classification and regression problems, which form the majority of current machine learning systems.

Random Forest has nearly the same hyper parameters as a decision tree or a bagging classifier. Fortunately, we don't have to combine a decision tree with a bagging classifier and can just easily use the classifier-class of Random Forest. Like I already said, with Random Forest, you can also deal with Regression tasks by using the Random Forest repressor.
Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.
Therefore, in Random Forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. We can even make trees more random, by additionally using random thresholds for each feature rather than searching for the best possible thresholds like a normal decision tree does. This is:

1. Increasing the Predictive Power.
2. Increasing the Models Speed.

Support Vector Machine: "Support Vector Machine" is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, 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 very well.

Accuracy and Statistical measurement

In this projects we had used 4 different algorithms to extract some knowledge. But all the algorithms are not given the same result. So some of algorithms can give better accuracy and some of aren't. For that, we calculated the accuracy of each of the algorithms from the confusion matrix. Confusion matrix is a convenient way of performance measure of any predictive model. We have calculated Accuracy, Precision of the above-mentioned algorithms. Accuracy show that how often the classifier is correct.

		Prediction	
Actual	Total= TP+TN+FP+FN	Positive	Negative
	Positive	TP	FN
	Negative	FP	TN

Table1: Confusion Matrix

The formula of the accuracy calculation,

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

The formula of the Precision,

$$\text{Precision} = \frac{TP}{(TP+FP)}$$

The formula of the True Positive Rate (TRP) or Recall,

$$\text{Recall} = \frac{TP}{(TP+FN)}$$

Where,

'TP' is True Positive

'FP' is False Positive

'TN' is True Negative

'FN' is False Negative

Dataset

Firstly, we had chosen **"Employee Attrition"** dataset for the python implementation. Here have 26 attributes and it's null free. In this dataset, here have 1471 data. This dataset is a company's employee list. The list has briefly show about the employee's job. At the end this data help to predict new comer employee's if they are interest or not in this job.

Secondly, we had chosen **"Adult Census Income"** dataset for the weka tool implementation. Here have 15 attributes and it's null free. In this dataset, here have 32562 data. This dataset is a different types of person information. Using this data, predict whether income exceeds $50K/year based on census data.

Implementation

Python: Using python, we have found some result. In below there have confusion matrix report.

Algorithm	Precision	Recall	f-score	Support	Accuracy
Decision Tree	0.76	0.81	0.78	515	81.0%
Naïve Bayes	0.76	0.52	0.58	515	52%
Random Forest	0.81	0.85	0.80	515	84.4%
Support Vector Machine (SVM)	0.71	0.84	0.77	515	84.3%

Table2: Python Implemented Confusion Matrix Report

Weka Tools: In below, there have all attribute figure.

In the weka tool, we have used 70% data for the training and 30% data for the testing set. So in below we attached different method measures to observe the different performance

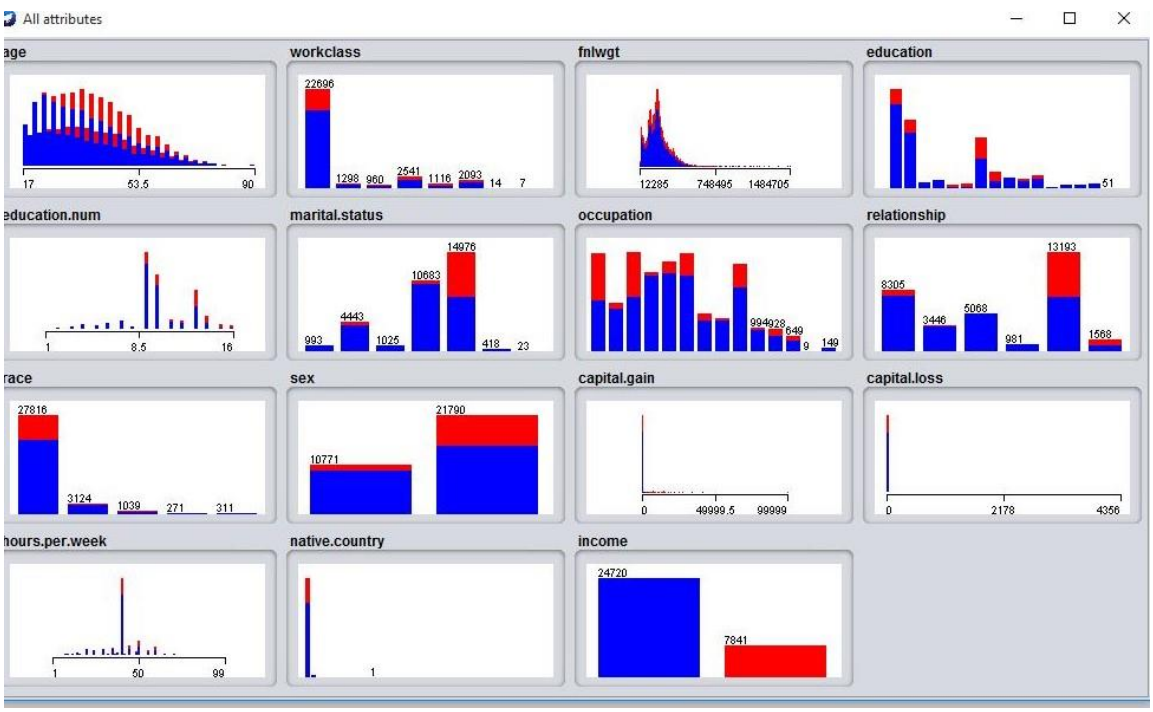

Figure1:- All attributes chart

Decision Tree: Correctly classified instances are 8476, about 86.17% and incorrectly classified instances are 1592, about 16.23% among the total number of 9768 instances. In Decision Tree algorithm the accuracy for that dataset is about 86%

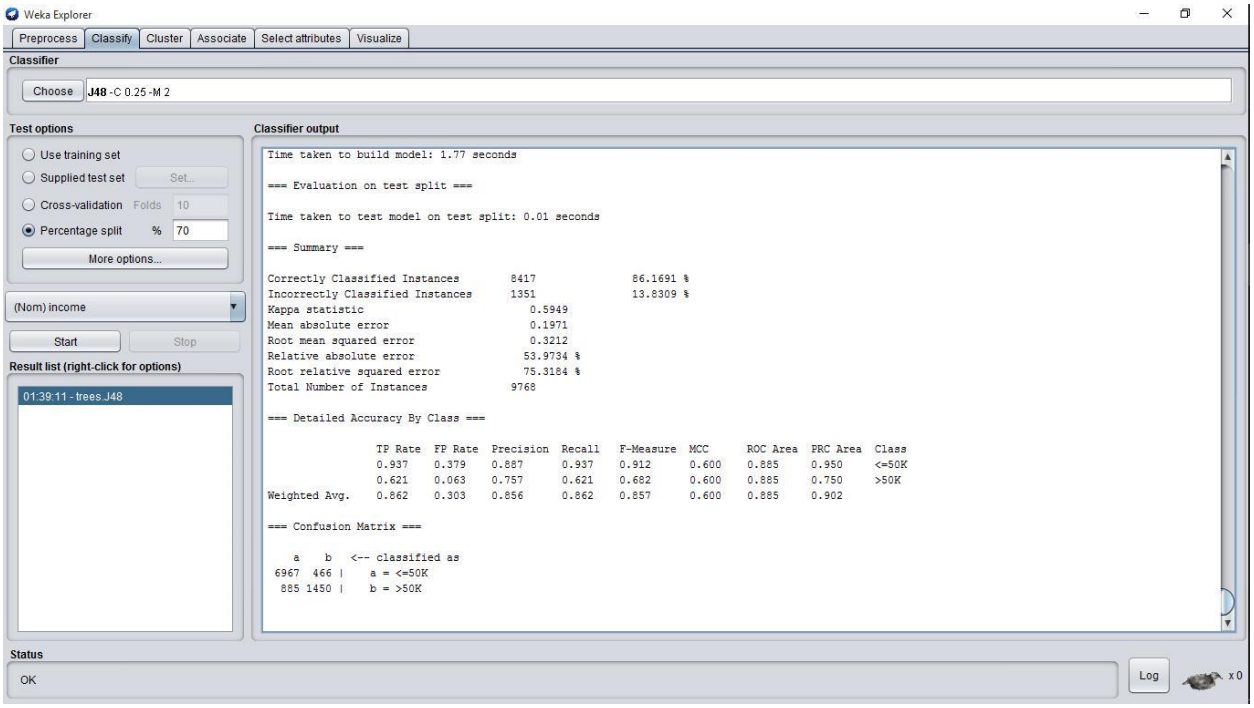

Figure2:- Decision Tree observation

Naïve Bayes: Correctly classified instances are 8176, 83.7% and incorrectly classified instances are 1592, 16.29% among the total number of 9768 instances. In Naïve Bayes algorithm the accuracy for that dataset is about 83%

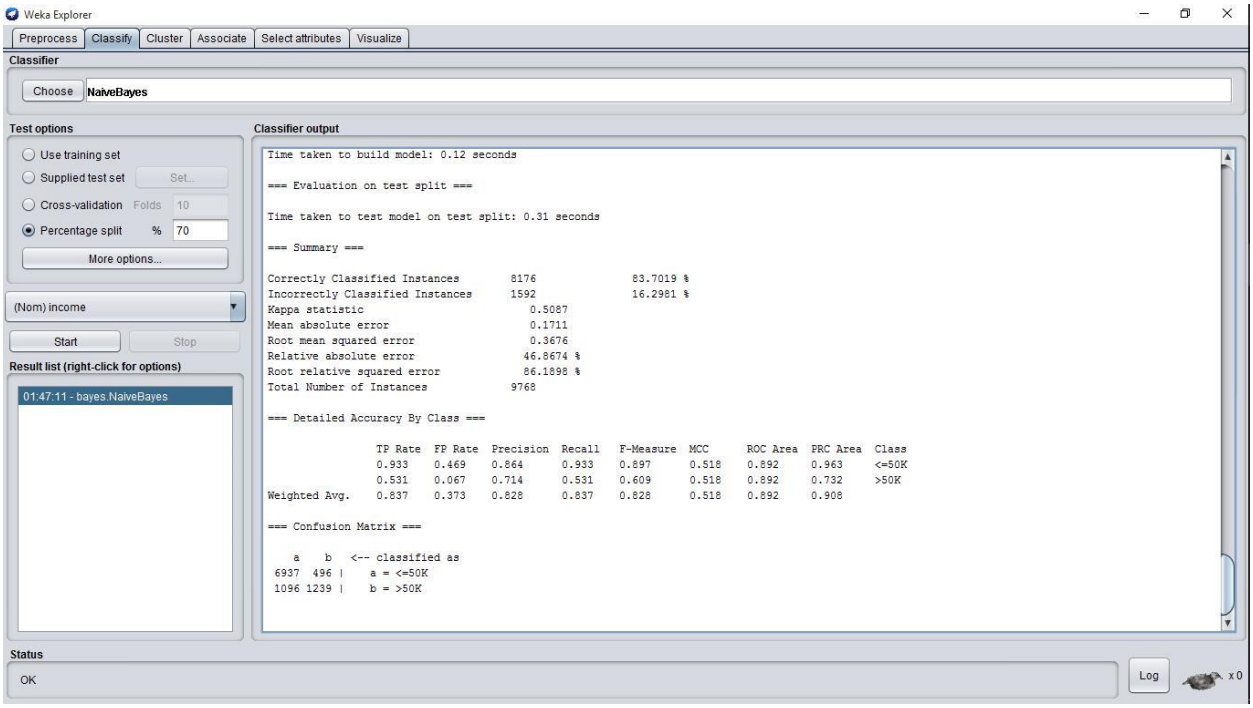

Figure3:- Naïve Bayes observation

Random forest: Correctly classified instances are 8296, 84.94% and incorrectly classified instances are 1472, 15.09% among the total number of 9768 instances. In Naïve Bayes algorithm the accuracy for that dataset is about 84.9%

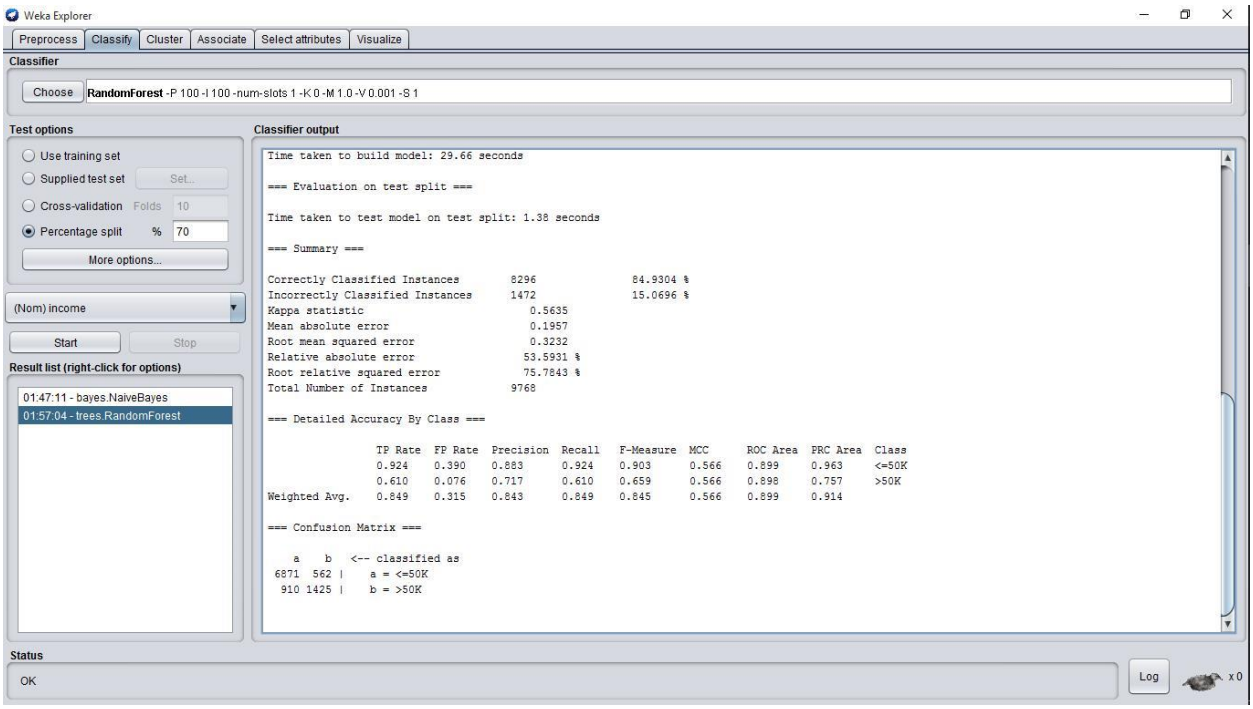

Figure4:- Random forest observation

Support Vector Machine(SVM): For this dataset SVM not gives the result in proper time.

Performance Evolution

In **table-2** we have shown python implemented accuracy and in this section we show about weka tool implemented accuracy.

Algorithm	Precision	Recall	f-score	Support	Accuracy
Decision Tree	0.856	0.862	0.857	9768	86.2%
Naïve Bayes	0.828	0.837	0.828	9768	83.7%
Random Forest	0.843	0.849	0.845	9768	84.9%
Support Vector Machine (SVM)	-	-	-	-	-

Table3: Weka tool Implemented Confusion Matrix Report

Discussion

In table-1 and table-2 we have seen Precision, Recall, f-score, Support, Accuracy. In table-1 shows that, SVM gives the best result. But in table-2, here have more data than table-1 and table-1 has more attribute than table-2. For the increasing of data, SVM is very slower than other algorithms. So in this project SVM gives the best data for less data. Decision tree gives the best result for table-2 where table-2 implies the dense dataset.

In table-1, Naïve Bayes and Random Forest gives the same accuracy but in the table-2 dense dataset Random forest gives the best result than Naïve Bayes.

Table-1 implies that more than 50% people are attrition in job and Table-2 implies that more than 83% people are earn more than or equal to $50K a year.

Conclusion

We learned some essentials about decisions trees, Naïve Bayes random forests. We applied those algorithms to a dataset, and we compared the accuracy of those models. So all of the algorithms are help to take decision or take predict. Using those algorithms, Business man can predict their business and the mining might bring out the maximum profit.