University of Dhaka

Department of Computer Science and Engineering

CSE-4255: Introducing Data Mining and Warehousing Lab
4th Year 2nd Semester

Session: 2018 -19

Report Topic:

Comparative Performance analysis between Classification Models Decision Tree and Naı¨ve Bayesian Classifier

Submitted by:

Amit Roy, Roll: JH- 40
Md. Tanvir Alam, Roll: SH-61

Submitted to:

Dr. Chowdhury Farhan Ahmed, Professor,
Department of Computer Science and Engineering, University of Dhaka

Abu Ahmed Ferdaus, Associate Professor,
Department of Computer Science and Engineering, University of Dhaka

Date of Submission:

30 September, 2019

Introduction

Classification, a well-known problem in data mining and machine learning used to find important data classes by building models which are called classifier. Those models used to predict or classify data classes. For a better understanding of the data and for various other applications these data classifications are very necessary. For example, a bank manager can learn whether it is “safe” or “risky” to grant loan application by using a classification model learnt on various attributes of the applicants' data. A doctor can decide which treatment a breast cancer patient should receive out of three specific treatment types A, B and C. A super shop owner can decide whether a customer will buy or not a particular product by learning a classification model from different attributes of the customer. The classification models are called classifiers. In this experiment, we have implemented two different types of classification method.

i) Classification using Decision Tree
ii) Naı¨ve Bayesian Classification

We also presented a comparative analysis between them in this report.

Problem Definition

In a classification problem, we will be given a set of tuples with different attributes in each tuple where each tuple has a class label. Each attribute can take a set of values. It can be categorical or numerical. We will split the given data tuple into two sets, a training set and a test set. Classification method has two steps. In the first step, we will build the classification model using the training data. In the second step, we will evaluate the performance measure of our model in the test data. Different methods are used for the cross-validation of training data and test data like K-fold, Bootstrapping, etc. Again to evaluate the performance of a model we can use different measures like accuracy, precision, recall, f-score, etc. Besides the scalability and robustness of a model can be assessed by the running time of a model in datasets of different sizes.

Classification using Decision Tree: This method develops a decision tree on the training data. Each internal node of the tree is a test on a particular attribute and the leaf nodes represent a class. The attribute which best splits a given data partition *D* is chosen based on various measures like Gain, Gain Ratio, Gini Index of the particular attribute. In our implementation, we have used Gain to choose the best splitting attribute. For m data partition and v types of attribute values of an attribute A, Gain(A) is defined as

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i), \qquad Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

$$Gain(A) = Info(D) - Info_A(D).$$

For discrete-valued attributes each branch from an internal node represents a possible value of that attribute and for continuous attributes a node has only two branches less or equal and greater than a particular value of the continuous attribute. We also used pre-pruning in the decision tree to avoid redundancy in the decision tree. For pre-pruning we take pruning threshold from the user and if any branch of the split has less tuples than the minimum pruning threshold percentage than we did not split that branch anymore.

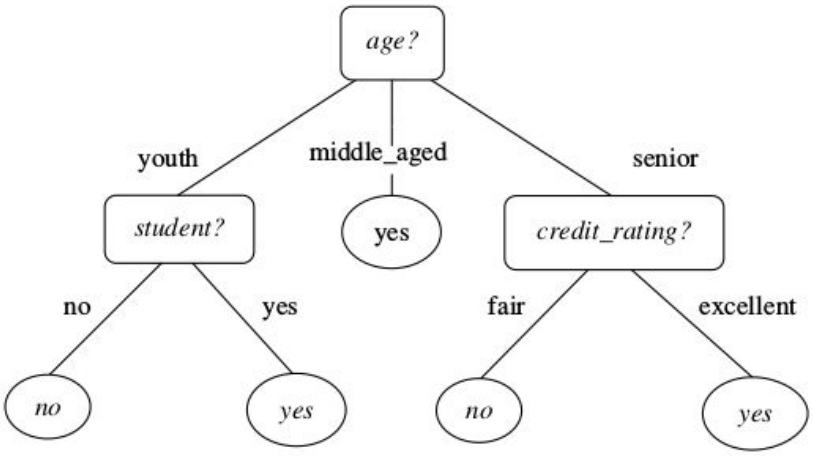

Figure: An example decision tree, internal nodes representing test on an attribute
Leaf node representing the class type for a tuple satisfying tests from root to that leaf

Naive Bayesian Classification:

To classify a data tuple using Naive bayesian classifier we use the **Bayes Theorem** of conditional probability to determine the probability of a tuple being in a particular class. The class which has the maximum probability is predicted. Given a hypothesis H and attribute description X, Bayes Theorem is as follows.

$$P(H|X) = \frac{P(X|H)\ P(H)}{P(X)}$$

Here,

P(H|X) is the posterior probability,

P(H) is the prior probability and

P(X|H) is the posterior probability of X conditioned on H

If we have a data tuple X with n attributes X_1, X_2,...,...,… X_k and there are m classes in the data set named C_1,C_2,…,…, … C_m then for each class we define with the class conditional independence assumption

$$P(X|C_i) = \prod_{k=1}^{n} P(X_k \mid Ci)$$

$$= P(X_1|\ C_i) \text{ x } P(X_2|\ C_i) \text{ x } \ldots\ldots\ldots\ P(X_k|\ C_i)$$

For a particular attribute value A_k,

i) If A_k is categorical then $P(X_k|C_i)$ is tuples of class C_i with attribute value A_k divided by the number of tuples of class C_i in the database D

ii) If A_k is numerical then $P(X_k|\ C_i)$ is defined as

$$P(x_k\ |C_i\ ) = g(x_k, \mu_{Ci}, \sigma_{Ci})$$

$$= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The class with the maximum probability $P(X|C_i)$ is predicted. We also used Laplacian Correction to avoid zero probability of an attribute which has no tuples with a particular class.

Performance Evaluation Measures: In a dataset we need to define a class as the positive class based on the importance or frequency. To define different evaluation measures of the classification model we need to define the following terms.

True Positive, TP: Positive tuples those are correctly classified
True Negative, TN: Negative tuples those are correctly classified
False Positive, FP: Negative tuples those are incorrectly classified as Positive
False Negative, FN: Positive tuples those are incorrectly classified as Negative

Accuracy: Accuracy or recognition rate is defined as the number of positive or negative tuples that are correctly classified divided by the total positive and negative data tuples.

$$accuracy = \frac{TP + TN}{P + N}$$

Recall: Recall is defined as the positive tuples that are correctly classified divided by the total positive tuples in the dataset. Recall is also known as sensitivity or true positive rate.

$$recall = \frac{TP}{P}$$

Precision: Precision is defined as the positive tuples that are correctly classified divided by the total tuples which are classified as positive tuple.

$$precision = \frac{TP}{TP + FP}$$

For class imbalance dataset, precision and recall is a good measure to evaluate the classification model.

F-score: F-score or F_1-score is the harmonic mean of precision and recall.

$$F1-score = \frac{2 \times precision \times recall}{precision + recall}$$

In class imbalanced dataset, F-score or F_1-score consider the effect of both precision and recall. So, it is a good measure to evaluate the classification model.

Experimental Results:

We have implemented decision tree and naive bayesian classifier in python and try to test them in 10 different datasets.

Dataset Description:

Dataset Name	# Tuples	# Attributes
Iris	150	4
Wine	178	13
Breast Cancer	286	9
Breast Cancer Wisconsin (Diagnostic)	699	10
Breast Cancer Wisconsin (Original)	699	10
Breast Cancer Wisconsin (Prognostic)	699	10
Cylinder Bands	540	39
Car	1728	6
Mushroom	8124	21
Wine Quality	1599	11

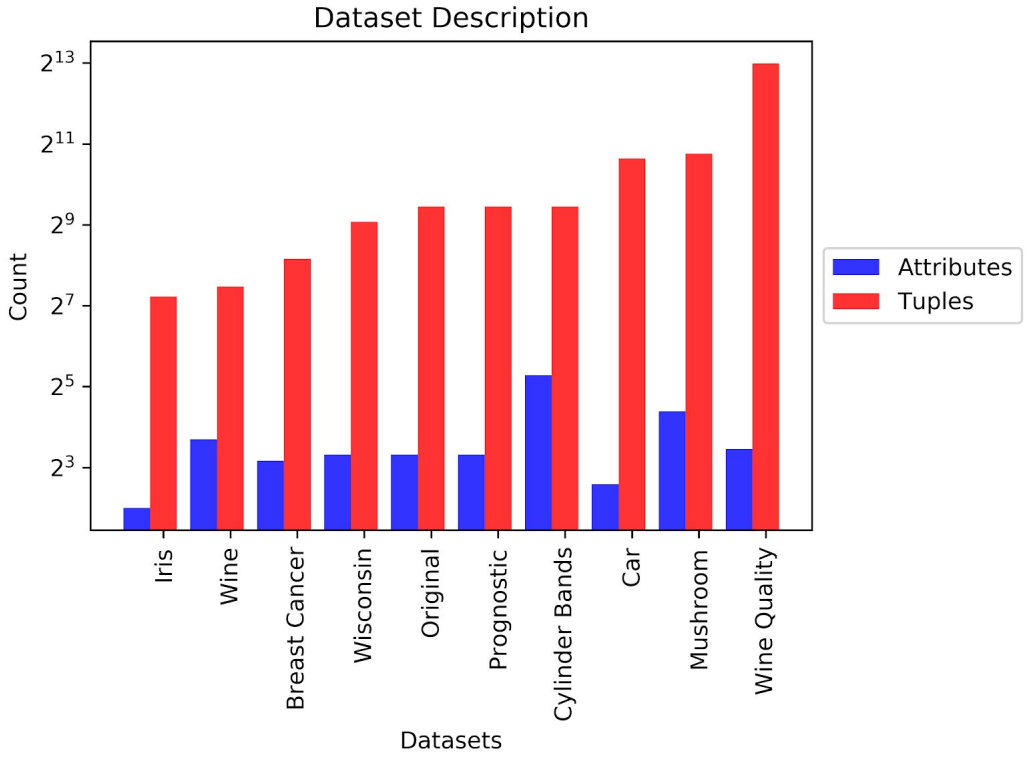

Figure: Dataset Description Attribute and Tuple Count

Decision Tree Experimental Results:

We have used 5-fold cross validation method to divide training and test set.

Dataset Name	Accuracy	Precision	Recall	F-score	Running Time(s)
Iris	94.66	100.00	100.00	100.00	0.195506
Wine	91.015	90.700	87.591	88.902	1.576283
Breast Cancer	66.079	83.120	72.713	77.568	0.266840
Breast Cancer Wisconsin (Diagnostic)	94.991	96.129	96.318	96.223	3.653997
Breast Cancer Wisconsin (Original)	93.278	94.991	94.746	94.868	4.114888
Breast Cancer Wisconsin (Prognostic)	93.991	94.496	96.294	95.386	4.572104
Cylinder Bands	60.185	91.893	60.225	72.692	35.515057
Car	80.957	92.000	91.812	86.125	0.355417
Mushroom	99.323	99.462	99.143	99.302	1.8576
Wine Quality	56.786	69.356	63.750	66.383	201.686458

Naı̈ve Bayes Experimental Results:

We have used 5-fold cross validation method to divide training and test set.

Dataset Name	Accuracy	Precision	Recall	F-score	Running Time(s)
Iris	95.333	100	100	100	0.009613
Wine	96.665	95	100	97.391	0.028092
Breast Cancer	71.693	84.109	77.583	80.623	0.008924
Breast Cancer Wisconsin (Diagnostic)	96.143783	95.205447	98.891164	96.992528	0.048827
Breast Cancer Wisconsin (Original)	95.99	95.195	98.668	96.874	0.044584
Breast Cancer Wisconsin (Prognostic)	96.281516	95.19828	99.10101	97.104764	0.047749
Cylinder Bands	73.0896	43.661	88.189	52.275	0.089919
Car	85.301788	95.785124	91.665833	93.663633	0.042838
Mushroom	95.347063	90.449182	99.886686	94.933085	0.372548
Wine Quality	54.274818	64.898025	66.912366	65.764534	0.297652

Comparative Analysis:

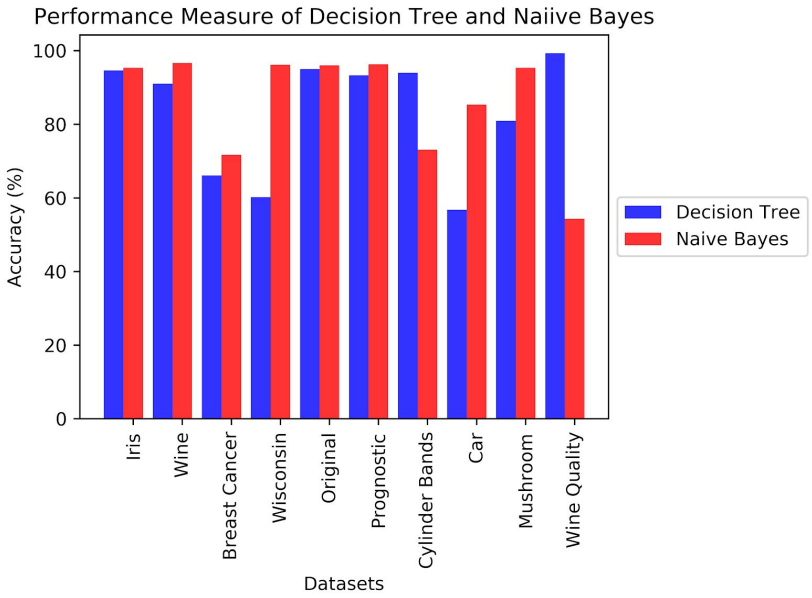

Figure: Accuracy Comparison between Decision Tree and Naı¨ve Bayes

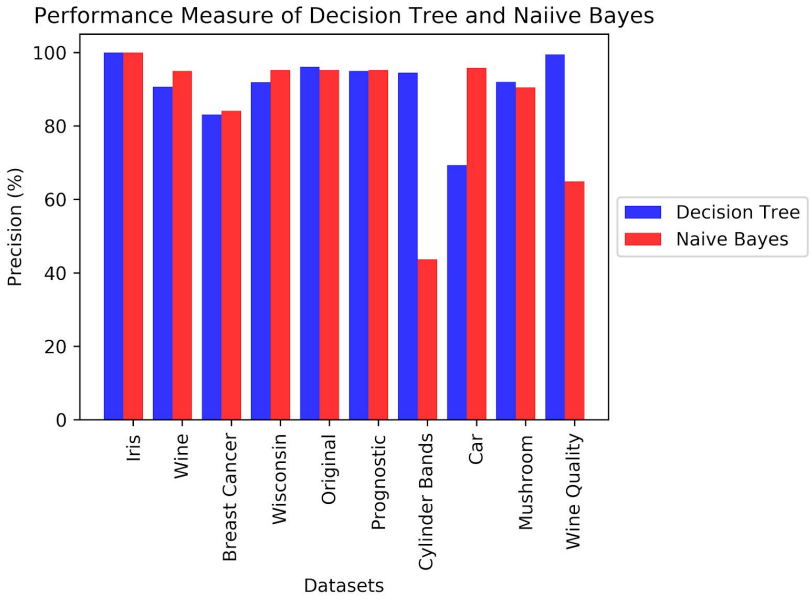

Figure: Precision Comparison between Decision Tree and Naı¨ve Bayes

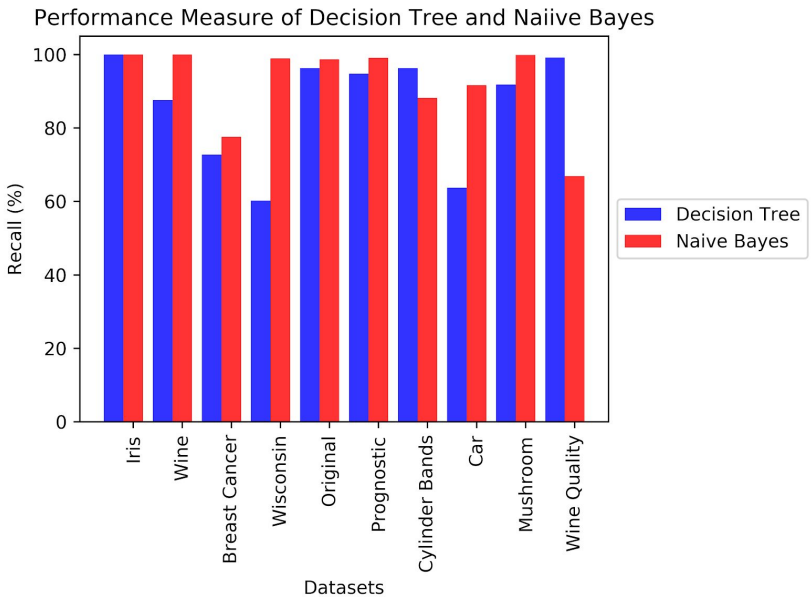

Figure: Recall Comparison between Decision Tree and Naı¨ve Bayes

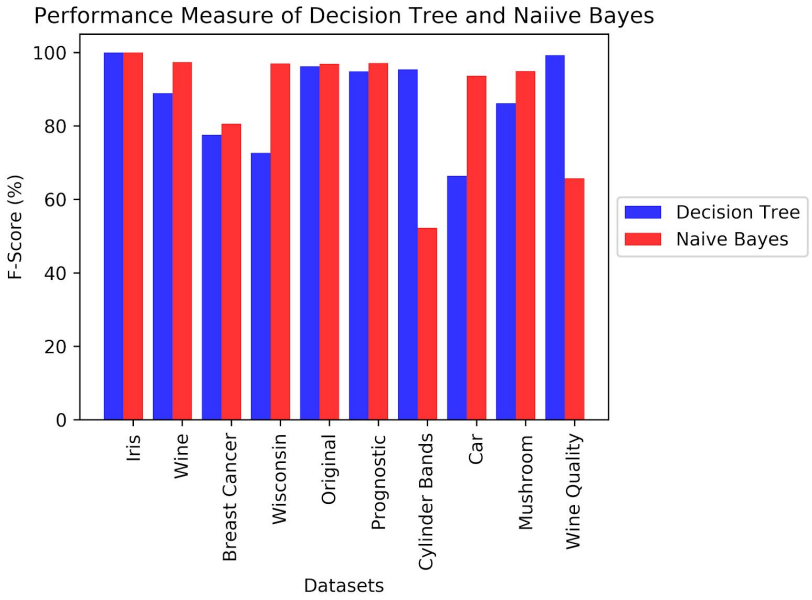

Figure: F-score Comparison between Decision Tree and Naı¨ve Bayes

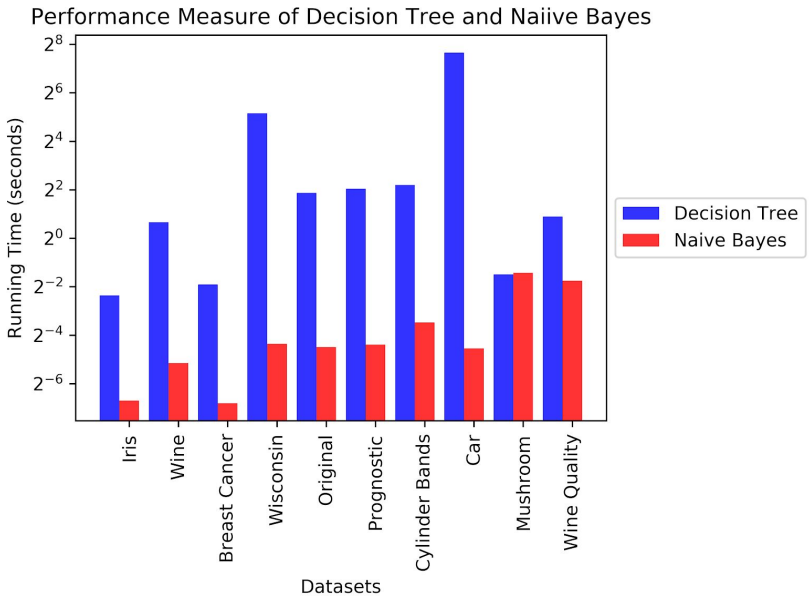

Figure: Running Time Comparison between Decision Tree and Naı¨ve Bayes

Conclusion:

Decision tree classifier and Naive Bayesian classifiers are two very simple and basic classifiers. Naive Bayesian classifier use Bayes rule of conditional probability to classify a tuple and it has better running time performance as we can see from our experimental results. But it shows lower accuracy in some cases because of class conditional independence assumption.