Department of Computer Science and Engineering

University of Dhaka

4^{th} *year* 2^{nd} *Semester*, 2021

CSE4255: Introduction to Data Mining and Warehousing Lab

Assignment: 02

Implementation of Decision Tree and Naïve Bayesian Classification Algorithm.

Submitted by:

Md. Ahasanul Alam, Roll: 10

Submission Date:

22 February 2021

Submitted to:

Dr. Chowdhury Farhan Ahmed.

Professor, Department of Computer Science and Engineering,

University of Dhaka.

1 Introduction

Classification is a two-step process, learning step and prediction step. In the learning step, the model is developed based on given training data. In the prediction step, the model is used to predict the response for given data. Decision Tree is one of the easiest and popular classification algorithms to understand and interpret.In this assignment, we have implemented two classification algorithm. This algorithms are:

- Decision Tree Classificaion Algorithm
- Naïve Bayesian Classification Algorithm

1.1 Decision Tree Algorithm

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data). 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 the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

1.2 Naïve Bayesian Algorithm

Naïve Bayesian Algorithm is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'. Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

2 Implementation Detail

In this assignment, we have implemented the Decision Tree and Naive Bayesian algorithms and experimented on 8 popular real life data-sets to compare the performances. For Decision Tree Classifier, we used **Gain Ration** as attribute selection measure. In the decision tree classifier, if attributes are continuous valued then we determined a best split point for those attributes where the midpoint between each pair of adjacent values is considered as a possible split-point.

While calculating probability in Naive Bayes Classifier, we replaced the zero probabilities with a very small value 10^-6. For continuous valued attributes, we used a normal probability distribution function to calculate the probabilities. If standard deviation of the values of any attribute for a given class is zero then we replaced it with a very small value 10^-6. In the following section, we will analyse the performance of Decision Tree and Naive Bayes Classification Algorithm on 8 different datasets.

3 Comparison Result

The programs are written in Python programming language (version 3.8.5). All the experiments are executed on a laptop with Intel Core i5 2.2GHz CPU and 12GB RAM. Performance were evaluated based on 4 standard measures, such as (i) Accuracy, (ii) Precision, (iii) Recall and (iv) F Measure. For precision, recall and F

Dataset Name	# Attribute	# Class	# Instances	Attr Type	Dataset Area
Iris	4	3	150	Real	Life
Wine	13	3	178	Integer, Real	Physical
Wholesale customers	8	2	440	Integer	Business
ILPD [1]	10	2	579	Integer, Real	Medical
Tic-Tac-Toe Endgame	9	2	958	Categorical	Game
Car Evaluation	6	4	1728	Categorical	N/A
Nursery	8	5	12960	Categorical	Social
Bank Marketing	17	2	45211	Real	Business

Table 1: Caption

measure, we have individually considered all the class labels as true class and calculated the average of the results using python sklearn API [1]. We have used the following 8 datasets to compare the performances of the two algorithms. The characteristics of the datasets are shown in Table 1 We have splitted the data set into train and test set to analyse the performance. The train test split ratio was **80%** train set, **20%** test set throughout the experiment. To eliminate statistical variations, we have executed each algorithms 5 times and taken the average of the outcome. The results are shown in table 1. To avoid complexity, we have removes the row if any of its attribute value is missing. All the datasets are taken from UCI machine Learning Repository [2].

The result of the four measures on the 8 experimental datasets are shown in the following sub section.

3.1 Dataset: Iris

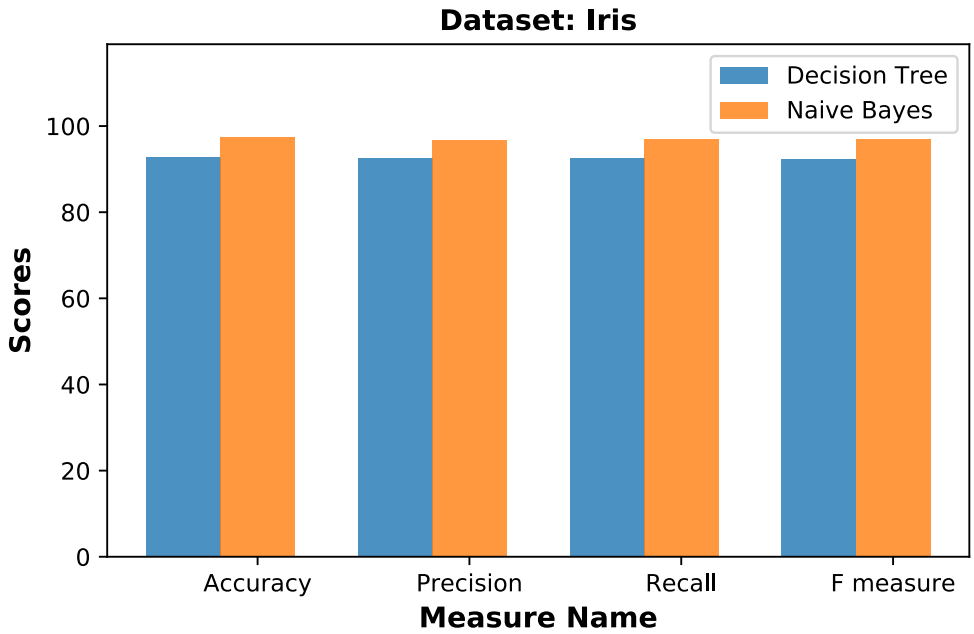

Figure 1: Performance Comparison on Iris dataset

[1] ILPD stands for Indian Liver Patient Dataset

3.2 Dataset: Wine

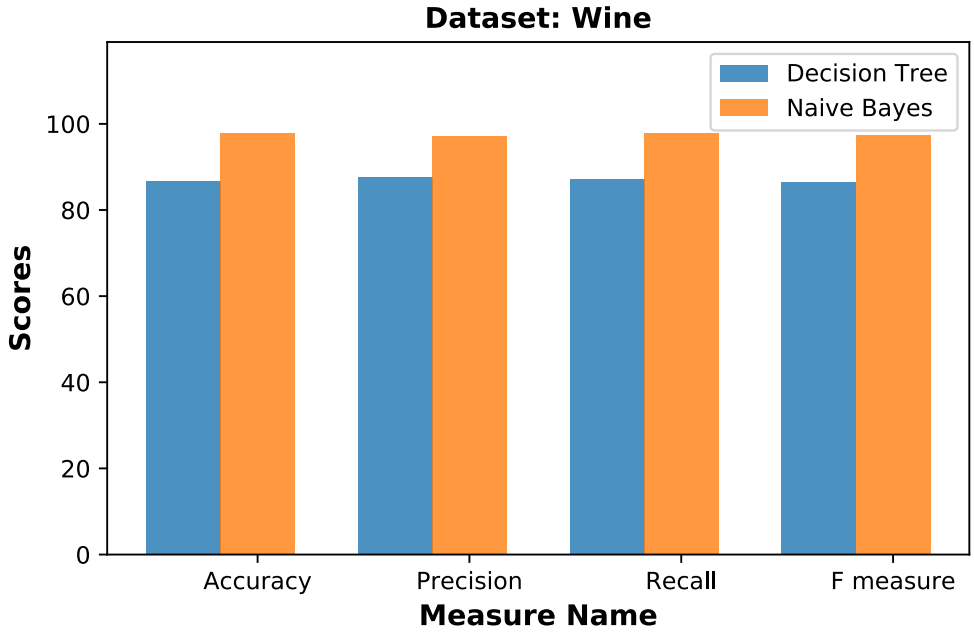

Figure 2: Performance Comparison on Wine dataset

3.3 Dataset: Wholesale Customers

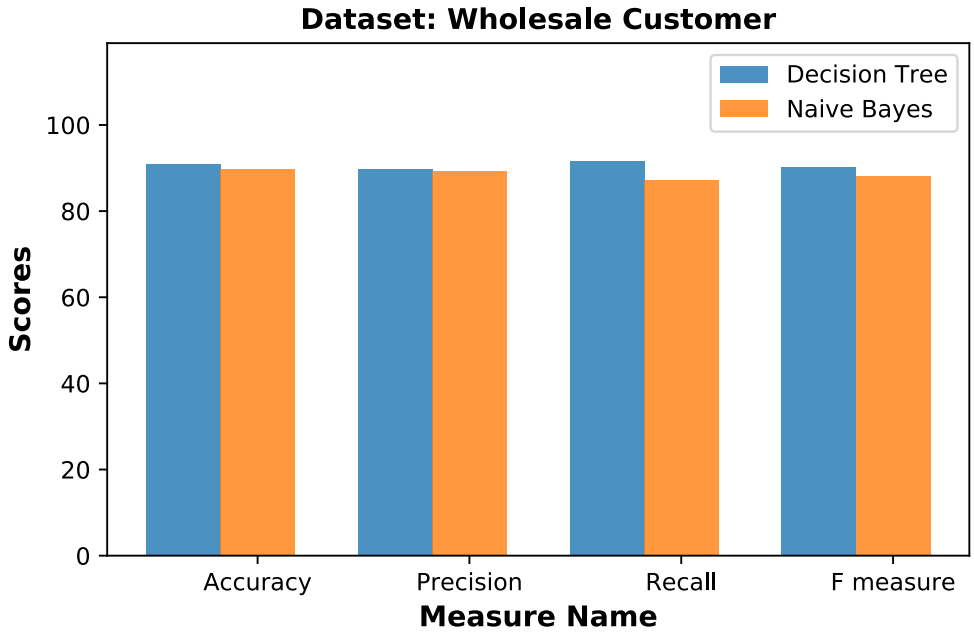

Figure 3: Performance Comparison on Wholesale Customer dataset

3.4 Dataset: ILPD

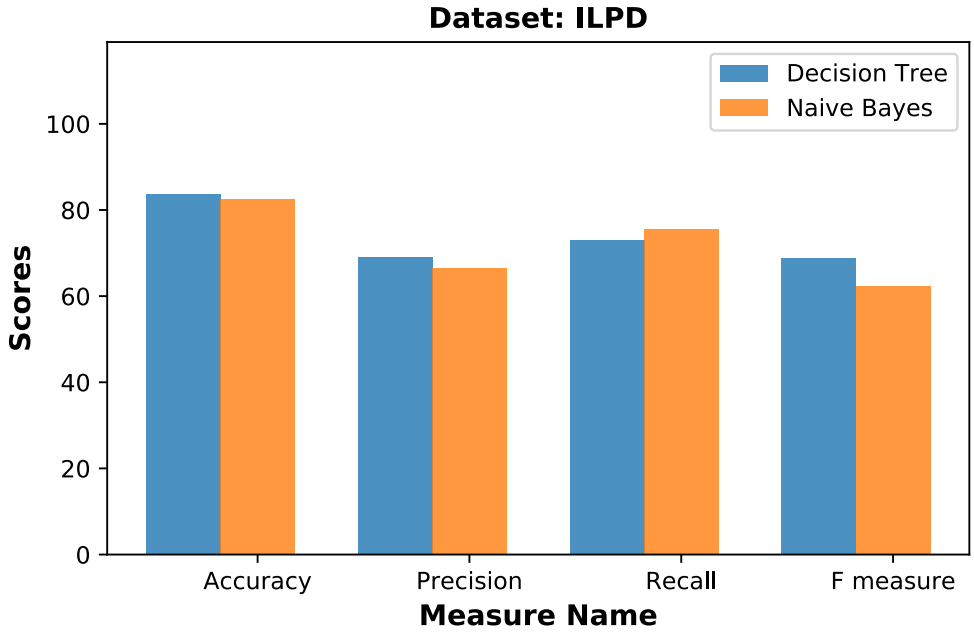

Figure 4: Performance Comparison on ILPD dataset

3.5 Dataset: Tic-Tac-Toe

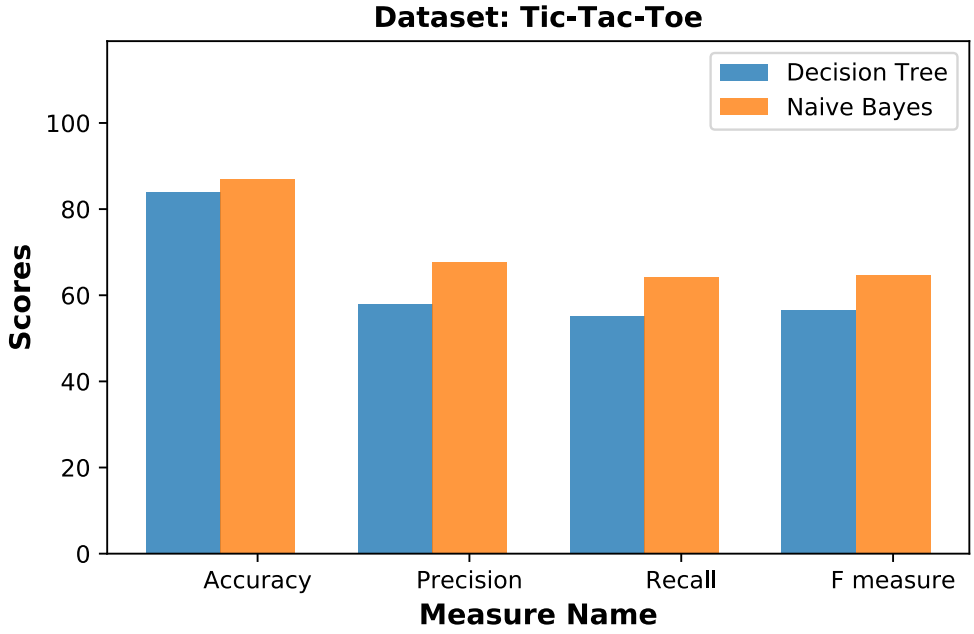

Figure 5: Performance Comparison on Tic-Tac-Toe dataset

3.6 Dataset: car Evaluation

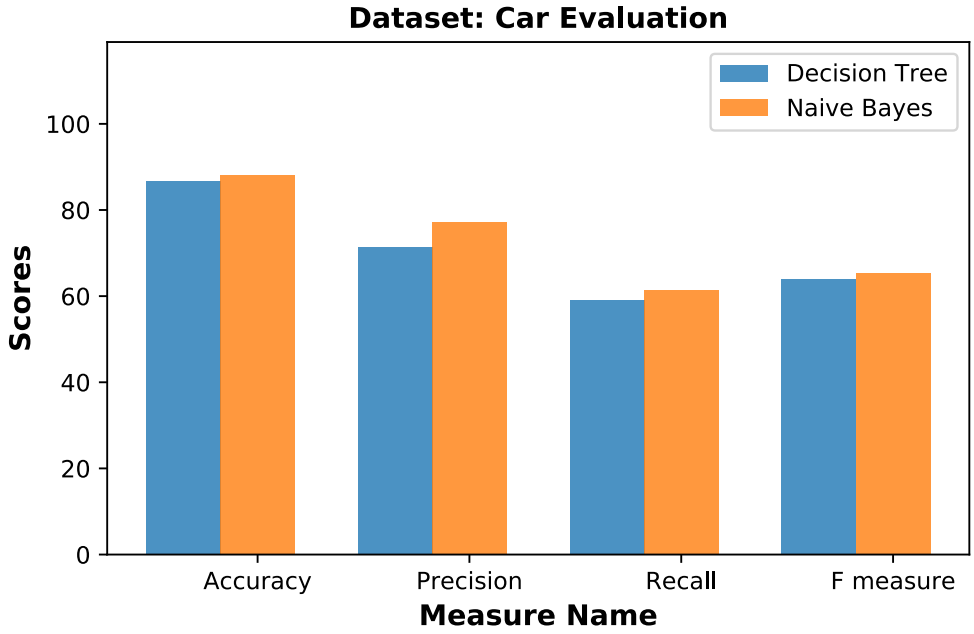

Figure 6: Performance Comparison on Car Evaluation dataset

3.7 Dataset: Nursery

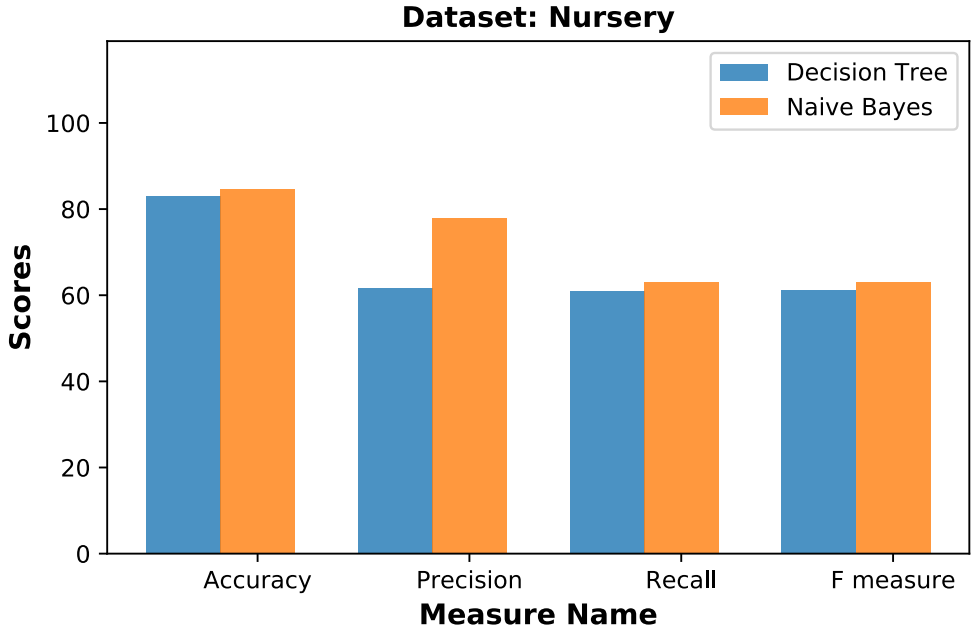

Figure 7: Performance Comparison on Nursery dataset

3.8 Dataset: Bank Marketing

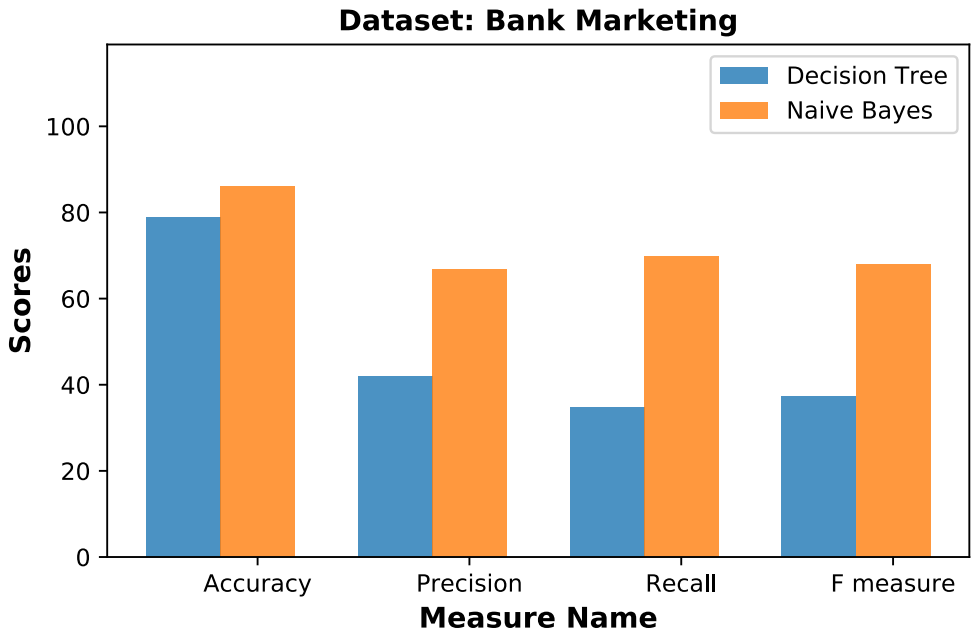

Figure 8: Performance Comparison on Bank Marketing dataset

4 Result Analysis and Discussion

From the Result Comparison section, we can see that both Decision Tree and Naive Bayes classifier perform almost similar to all the data set. Between this two classifier Naive Bayes performs slightly better in most of the datasets. This is because of the characteristics of these two classifier. While Decision tree considers each tuple in the process, Naive Bayes calculates a Bayesian probability of the tuple with respect to the whole training dataset. By doing so, anomaly or outlier data has less impact on Naive Bayes classifier compared to Decision tree Classifier. Where, decision tree tends to overfit the training data. As a result, Naive Bayes performs better than Decision Tree Classifier.

For Tic-Tac-Toe, Car Evaluation and Nursery dataset where all the attributes are categorical, Naive Bayes performs significantly better in terms of precision, recall and F measure. This is because as Decision tree is more prone to outlier, categorical data can direct the decision tree to wrong path more easily than real/continuous data. For continuous data we have binary splitted the data in two part using best split method. Hence an anomaly data has less impact than categorical data. Some problem is present for large datasets (Nursery, Bank Marketing) too. Here the number of samples are higher than other datasets. Higher number of samples tends to have higher anomaly data which downgrades the performance of Decision tree classifier compared to Naive Bayes.

5 Conclusion

Both Decision Tree and Naive Bayes are one of the most simple and classic classification algorithms. Naive Bayes works fast and doesn't need to pre-calculate any information. But It consider that all the features are independent. This may not true in real live data. Thus degrades the performance. While on the other hand, Decision tree is simple and easy to understand. Though it requires pre computation to buid the decision

tree, but once the tree is built, it can predict faster with having any heavy calculation. But as it consider all the samples explicitly, decision tree is more prone to outlier. However, many improvement has been made for both the algorithms which can accelerate the performance of the classifier significantly.

References

[1] API Reference. *Scikit Learn*. URL: `https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics`. (accessed: 22.02.2021).

[2] UCI. *UCI Machine Learning Repository*. URL: `https://archive.ics.uci.edu/ml/index.php`. (accessed: 22.02.2021).