Chapter 1: Introduction

Introduction

Classification is a technique that used to extract a classifier that describing important data classes. Classifier are used to predict categorical class labels which are discrete unordered value. Classifier able help us to understand large data in this era. Researcher proposed many classification methods such as machine learning, pattern recognition and statistic. Most of the classification algorithm require a large memory and designed for small dataset. Scalable classification and prediction technique are designed to handling large amounts of disk-resident data. Classification has a lot of application for example it can use to predict the bank loan is safe or risky or the price or house is raised or drop. Classification also can be use in medical diagnosis, manufacturing, fraud detection, target marketing and etc. There is numerous basic classifier such as decision tree classifier, Bayesian classifier, and rule-based classifier. Apart from that, the performance of classifier can evaluate by using performance metrics such as accuracy, sensitivity(recall), specificity and f-score.

Classification involves two steps, which are learning step and classification step. Learning step is the process that construct the classification model by using training set whereas the classification step is the process after constructed classification model that used the model to predict the class labels for given data. Training set include attribute vector and associated with its class labels. The X is represented n-dimensional attribute vector, X = (x1,x2…xn), and each X is assumed associate with a predefined class attribute in database called class label attribute. The class label is discrete-valued and unordered so it is categorical or nominal. The training set (training tuples) are randomly extracted from the training set databse.

Problem Statement

1. Human perspective bias on classifies an object
2. Time consuming to identify possible solution and decision to support business

Objectives

Develop a program that able to train a model and display with performance metrics

1. C, C++, C#, Java (or python discussed verbally in the class)
2. Include two modules, C4.5 algorithm and naïve Bayesian
3. Build classification model based on training set (defined in text file)
4. Evaluated by testing set (defined in text file)
5. Display classification result of each testing data
6. display metrics for evaluating performance of each trained classifier

Scope

User Scope

1. Allow user enter a file name for training set
2. Allow user to enter file name for testing set after training

System Scope

1. Module 1: C4.5 Classification Module
   1. Program will develop a tree model using C4.5
   2. Illustrate each stages of developing decision tree model
   3. Display measure for every iteration in building decision tree
   4. Measure for C4.5
      1. Info(D)
      2. InfoA(D)
      3. Gain(A)
      4. SplitInfoA(D)
      5. GainRatio(A)
   5. Illustrate resulting tree
2. Module 2: Naïve Bayesian
   1. Illustrate the computational process in classifying data tuples
      1. Ci = category, xi attribute value
      2. P(Ci), the prior probability of each of the categories
      3. P (xi | Ci), the posterior probability of each of the attribute values conditioned on each of the categories.
   2. Laplacian correction to avoid any computing probability values of zero
3. Module 3: Evaluation Module
   1. Accuracy
   2. Error Rate
   3. Sensitivity (Recall)
   4. Specificity
   5. Precision
   6. F and F1, Harmonic Mean of Precision and Recall

Chapter 2: Literature Review

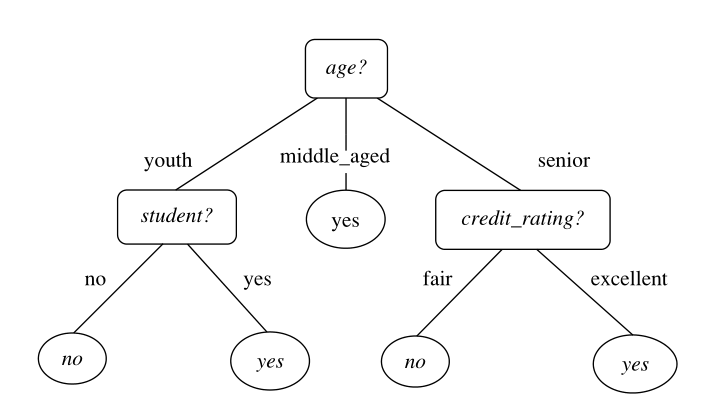
Overview of Machine Learning concept and techniques

According to (Tom Mitchell, 1998), machine learning is a well-posed learning problem which mean a computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E. Machine learning is a new capability for computers such as data mining, self-customizing program, application can’t program by hand for example Natural Language Processing. Machine learning divided into two main category which are supervised learning and unsupervised learning. There is more category from that, such as reinforcement learning, recommender systems and more. Supervised learning divided into two categories. First, regression problem is to predict a valued output based on the input such as house price. Supervised learning does provide a right answer at the end compare to unsupervised learning. Second, classification problem is to classify the object base on the parameter given. Common classification problem is to classify have or don’t have breast cancer based on the tumor size. The classification problem can become more complex based on the situation. Support vector machine (SVM) is introduced to solve when there is two-group of classification problem. (Stecanella, 2017). Unsupervised learning is one of the machine learning algorithms. The goal of unsupervised learning is to draw inferences base on the datasets given without any labeled responses. In other meaning is ask the machine to classify and provide an inference based on the dataset given. The common algorithm included hierarchical clustering, k-Means clustering, Gaussian mixture models, Self-organizing maps, Hidden Markov models. (MathWorks, n.d.).

Literature on Decision Tree and its variants (eg. ID3, C4.5 and CART)

Decision tree induction

Decision tree induction is the learning of decisions tree from class-labeled training tuples. The decision tree is a flowchart-like structure, the most top node in the tree is root node, each node represents the test for an attribute. Each branch represents the outcome of the test and each leaf node contain the class label. The figure below shows the example of the decision tree which used to predict whether a customer at AllElectronics is likely to purchase a computer.

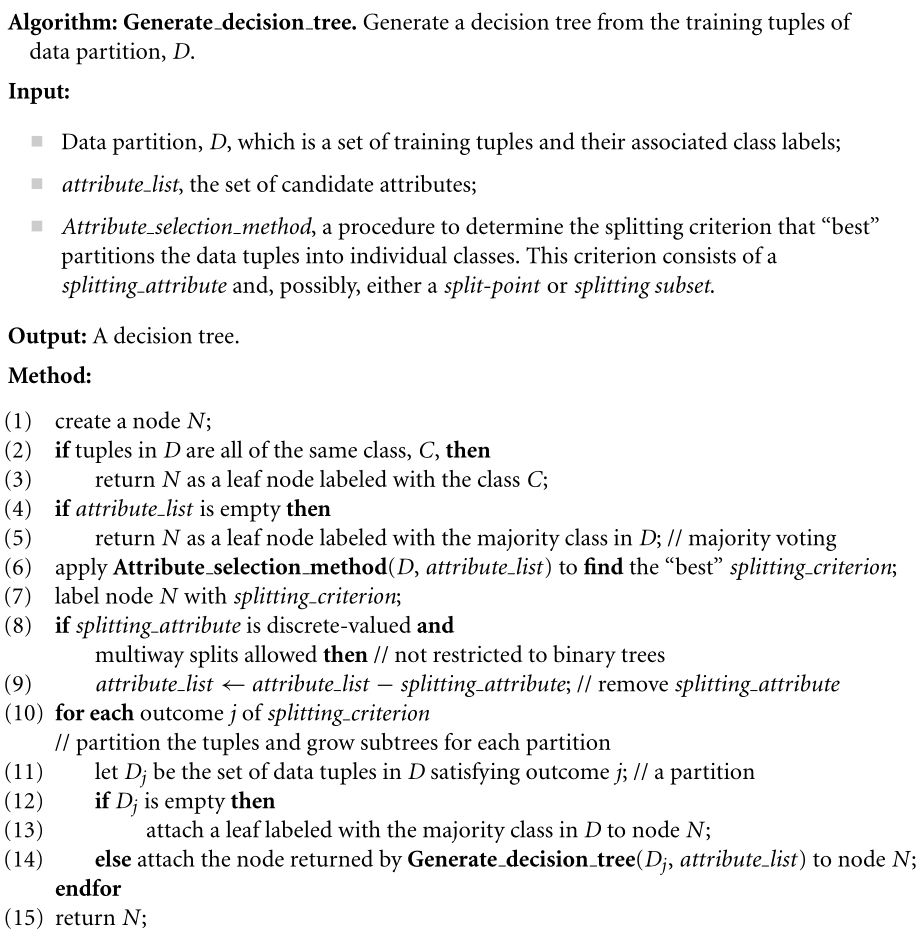


The figure above the rectangles represents the internal node which are the node to test the attribute. The oval represents the leaf node which hold the class label. The decision tree algorithm is variety, some able to produce binary tree but some able to produce non binary tree. During the classification step, the given tuple X will test against the decision tree. The decision tree can easily convert to classification rule. To construct a decision tree does not require any domain knowledge or parameter settings. The decision tree able to handle multi-dimensional data.

In this section will discuss three main type of decision tree which are ID3 (Iterative Dichotomiser), C4.5 and Classification and Regression Tree (CART). ID3 is developed by J.Ross Quinlan during late 1970s and early 1980s. C4.5 is the successor of ID3, it is designed to solve the biased problem on ID3 by introduction gain ratio. CART is a binary decision tree that invented independently with ID3 around the same time. ID3, C4.5 and CART adopt greedy approach such as non-back tracking which means the decision tree is created by using top-down approach and divide-and-conquer manner. Majority algorithm for decision tree induction follow the top-down approach, which mean the training start with a training set and their associated class labels. The training set will divide become a smaller subset recursively during the tree construction process.

A basic decision tree algorithm is summarized at below.

1. It has 3 parameters, which are D (that can refer as a data partition), attribute\_list and Attribute\_selection\_method. The training begins with the complete training set with their associated class labels. The attribute\_list is a list that stored all the attributes that describing the tuples. Attribute\_selection\_method defines a heuristic procedure for choosing the best class discriminators from a collection of tuples. This procedure implements an attribute selection measure such as information gain or Gini index. The attribute selection measure will affect the resulting tree. For example, the Gini index will enforce the resulting tree to be binary. But information gain allowing the algorithm to generate a non-binary tree which will have multi split from a node.
2. The tree will start with a single node, N representing the training tuples in D (step1). Rather than storing the actual tuples at the node, most implementation is storing the pointers to these tuples.
3. The picture below is the basic decision tree algorithm



Attribute selection measure

Attribute selection measure is a heuristic that use to select the best splitting criterion that will use to slice the data partition, D. If we try to slice the data partition (D) into smaller partition, the most ideal scenario is the sliced partition is pure. In another meaning all the tuples will fall into a given partition that belong to the same class. Attribute selection measure also known as the splitting rule because they decided how to split the given data partition.

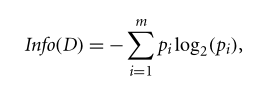
Attribute measure selection will rank each attribute. The attribute that has the highest score will chose as the best splitting attribute for the given tuples. We are restricted to the binary tree if the splitting attribute is continuous-valued.

ID3

Information gain is chosen as the attribute selection measure for ID. In ID3 the attribute that has the highest information gain will chose as the splitting attribute for node N. The chosen attribute will minimize the information requirement to classify the tuples in the resulting partitions. Information gain can minimize the number of tests required to classify an object without label.

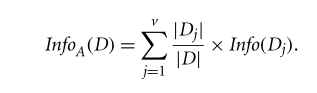
The information gain formula summarized as bellow.

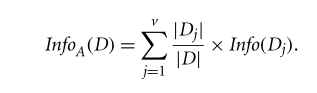
First of all, we start by calculating the information gain for the label.



1. pi is non zero probability for the tuple that belong to class Ci.
2. The log function using base 2 because the information is encoded in bits.
3. Info(D) also known as the entropy of the data partition.

Next, we need to calculate the information gain for the attribute.



1. The  represent the weight of the jth partition.
2. InfoA(D) is the expected information requirement to classify a tuple from D based on the partitioning by A.
3. The smaller the expected information required, the greater the purity of the partitions.

After calculate the info for label and attribute, we need to calculate the Gain of the attribute

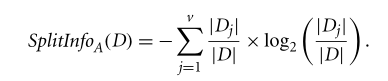


The gain will tell us how much information we will gained if we choose A as the splitting attribute. Knowing the value of A able help us to reduce the information requirement. Since the attribute A has the best gain value, it will be selected as the splitting attribute at node N in ID3. Next, we need to split the given data partition base on the attribute A. So, the information requirement to finish classifying the tuples is the least.

C4.5

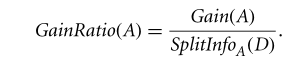
C4.5 is the successor of the ID3. The information gain that used in ID3 will cause biased toward tests with majority outcomes. In another meaning, it is preferring select attributes that hold a larger number of values. user\_id will result a large number of partitions. So, the user\_id will selected as the attribute for node N. It is because each partition divided by using user\_id is pure. When Infouser\_id(D) = 0 the information gained to split by using this attribute is maximal. But it is useless for the classification.

C4.5 is the extension of the ID3. It is designed to overcome the issues mentioned in above.



SplitInfo value shows the potential information will generated by splitting the training dataset into v partitions.

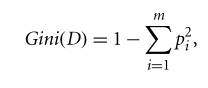
After calculating splitinfo we are able to calculate the gain ratio to overcome biased problem.



The attribute has the highest gain ratio will choose as the splitting attribute. If the split information is 0 will cause mathematical error. A technique called Laplacian correction is used to avoid this issue.

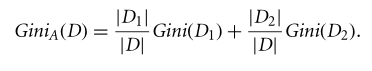
CART

CART is a binary tree classifier that using Gini index as its attribute selection measure. It is independent from ID3 and C4.5. The Gini index is used to measured the impurity of the data partition.



1. pi is the probability for the tuple that in data partition that belong to class Ci

Where the attribute A is discrete-valued which having number of v distinct values {a1, a2…. av}. In CART, all the possible subsets will examine to determine the best binary split for each attribute. For example, the possible subset for temperature {Normal, Cool, High} is {Normal, Cool, High}, {Normal, Cool}, {Normal, High}, {Cool, High}, {Normal}, {Cool}, {High}, {}. But the powerset {Normal, Cool, High} and empty set {} will eliminated. So, there is 2v – 2 number of combinations to form two partitions of the data.



We will calculate weighted sum of the impurity for each resulting data partitions. The subset that having the minimum Gini index will chosen as the splitting subset for discrete value. For attribute that held continues values, a split point must be consider. For example, data partition 1 is for A <= split\_point and data partition 2 is for A > split\_point.



The formula above calculates the impurity for the binary split for the attribute that held continuous or discrete value. The attribute that has the highest impurity value which means it has the lowest Gini index. So, it will be chosen as the splitting attribute.

Literature on Bayes Theorem and Naïve Bayesian classification

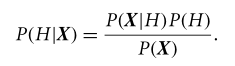
Bayes theorem

Let X = data tuple

Let H = hypothesis = X belongs to class C

Bayes theorem consider the X as the “evidence”.

When we try to determine the probability of X belongs to C P(H|X), the probability of hypothesis H for given observed of data tuple X (evidence). P(H|X) is the posterior probability or posteriori probability.



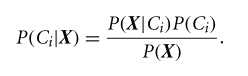
The P(X) is the prior probability of X.

But the (H|X) is the posterior probability of X conditioned on H. and P(H) is independent of X.

Naïve Bayesian Classification



The condition above shows the tuple X belong to the class if and only if the class have the highest posterior probability.



Since the P(X) is constant for classes, so only P(X|Ci) P(Ci) need to be maximized. If we don’t know the class prior probability, it is commonly assumed the classes are equally likely. So we maximize the P(X|Ci) otherwise we maximize the P(X|Ci) P(Ci)

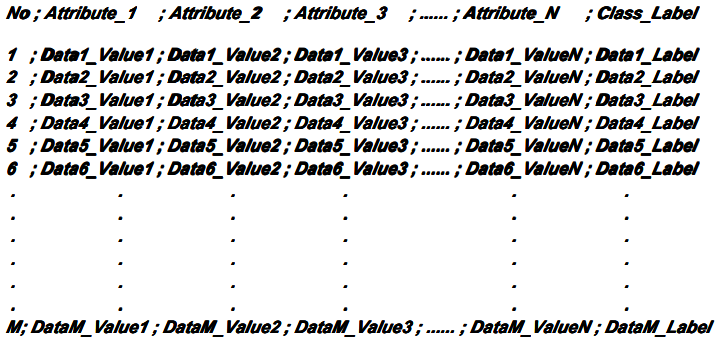
Chapter 3: Methodology

Detailed discussion on the selected methodology in relation to the design and development processes of the program

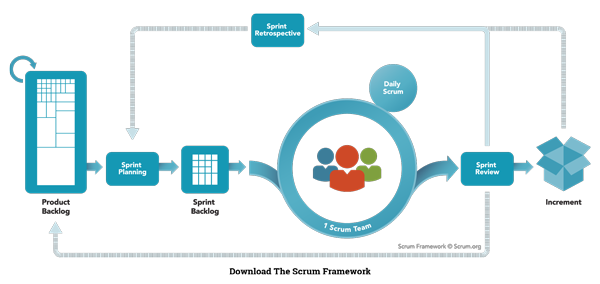
Dataset Design

The picture below showed the file data structure that use to store training set and testing set. The training set is used to create the classification model, and testing set is used to evaluate the performance of the trained model. The format of the file is defined as below:

The first row is the header it defines the list of attribute and class label for each tuple. The first column defines the unique identifier for the record. Each attribute and class label is separated by semicolon(;) and the data tuple separated by a newline.



Methodology



Scrum Model. Adapted from "Nutcache",

retrieved from https://www.nutcache.com/blog/leverage-scrum-to-manage-your-projects/

A scrum model (agile) is implemented in this project. Agile is an iterative approach for software development. The software is developed and delivered to customers in increments. Agile has the flexibility to accommodate frequent changes in the design. Scrum is one of the agile process frameworks which include product owner, scrum master, and development team. Scrum breaks the task into goals that can be completed within the timeboxed iteration, which call sprint. This is a lightweight. Iterative and incremental approach. The sprints should not longer than one month. The development team is self-organized, and responsible convert the backlog into an actual system. Eight members of the development team are required in this project. Product owner representing stakeholders and the voice of the customer. Only 1 product owner is required in this project to maximizing the value delivered by the development team. Scrum Master is responsible for ensuring the Scrum framework is followed and acts as a buffer between the team and any distracting influences. Each team required a scrum master, in this project, there is two development team which are Team A, and Team B. Scrum also included sprint planning, daily scrum, sprint review, sprint retrospective, backlog refinement, cancelling a sprint. These will be implemented in this project. (Fernandes, 2015) (scrum.org, n.d.)

Definition

Scrum events are defined as the following:

**Sprint:**A time-boxed mini project which less than 4 weeks, at the end for each sprint should deliver a releasable product or feature, Sprints include the planning, design, development and testing phase. Each sprint can assign to a synchronized team.

**Sprint planning:** A planning stage before the sprint began. Each backlog is prioritizing and review the requirement (product backlog) and created an order list for a particular sprint. Analyze the feasibility for each requirement and features finish at a particular time.

**Daily Scrum:**A meeting that held every day morning and takes less than 5 minutes. The scrum master will coordinate the team and discuss their daily goal and achievement. The obstacle also will be discussed in the meeting to seek help from another team. An unclear goal can make the team focus on their daily tasks and increase productivity.

**Sprint review:**An informal meeting establishes at the end of the sprint. The increment (product backlog) will be demonstrating to the end-user if any improvement or changes will execute in the next sprint.

**Sprint retrospective:**A formal meeting that gathers all the scrum and reviews the sprint. Each sprint will be review in this stage, which included the factor that makes the sprint or goal fail, way to improve the sprint, and etc. Then continue next sprint.

Scrum artifacts are defined as the following:

**Story:**Describe what users need to solve their problems. It describes the functionality and the features of the system which is also known as user stories. For example, login, pay and update profile.

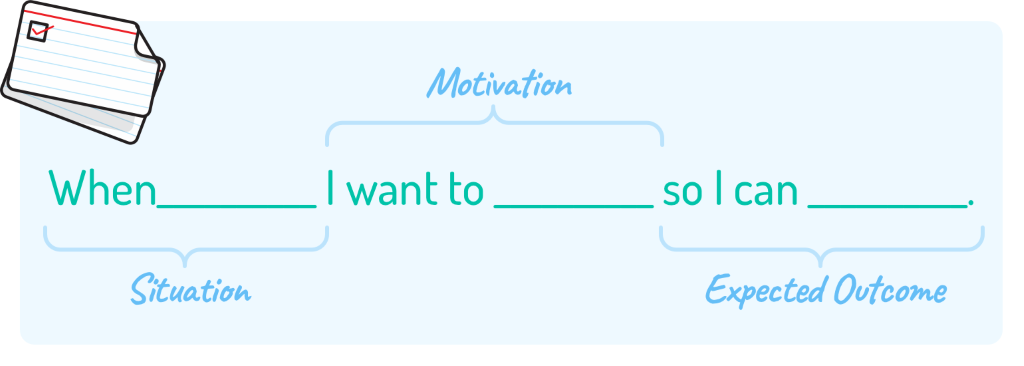


Figure 3.4 User Story. Adapted from "Mountain Goat Software", by Mike Cohn,

retrieved from <http://www.mountaingoatsoftware.com/blog/job-stories-offer-a-viable-alternative-to-user-stori>

es/feed

**Product Backlog**: An ordered list of the requirement for the product or features of the system. Product backlog includes all the requirements of the systems or features such as users can pay via credit card. A product backlog is never complete because product backlog will evolve throughout the entire development process. A product backlog is dynamic and frequently changes to fulfil the requirement of the product and what the product needed to be competitive. Adding detail, estimate, and order to items in the product backlog call product backlog refinement.  (Fernandes, 2015)

**Tasks:**A decomposed of a product backlog. Task refine the product backlog and the requirement of the product or features of the system.

User Story

User story able to help programmer to analyze the user requirement and turn it into module in programming easily.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User ID | As a …. | I need … | So I can… | Priority |
| U001 | User | A program that able to read and input training file | Use to create a classification model based on training set | 5 |
| U002 | User | A program that able to read and input testing file | Classify the object by using classification model | 5 |
| U004 | User | Display performance metrics such as accuracy, error rate, sensitivity, specificity, precision and F-score | Evaluate the performance of classification model | 3 |
| U005 | User | Include C4.5 and Naïve Bayesian classification algorithm | Create classification model | 5 |
| U006 | User | Laplacian correction for Naïve Bayesian algorithm | Avoid any computing probability of values of zero | 4 |
| U007 | User | Display all measure value for c4.5 such as Info(D), InfoA(D), Gain(A), SplitInfoA(D) and GainRatio(A) | Evaluate the features | 3 |
| U008 | User | Display classification result of each testing data | Evaluate the features | 3 |

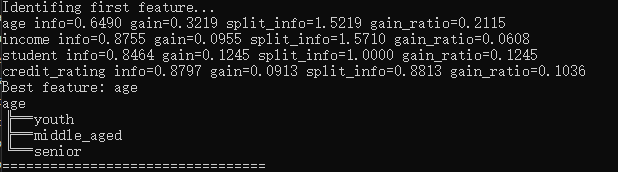
Deliverables

|  |  |  |
| --- | --- | --- |
| **Deliverables** | **Phase** | **Date** |
| Documentation (Chapter 1 to 3) | Analysis Phase (Planning – Define Scope) | 15/1/2021 |
| -Read from csv module  -C4.5 Module | Sprint 1 | 16/1/2021 |
| Performance Metrics Module  Naïve Bayesian Module | Sprint 2 | 17/1/2021 |
| User Interface (View)  Compile to executable file (console application) | Sprint 3 | 18/1/2021 |
| Source Code  Deployment and user manual  Final documentation  Presentation  Presentation video | Sprint 4 | 19/1/2021 |

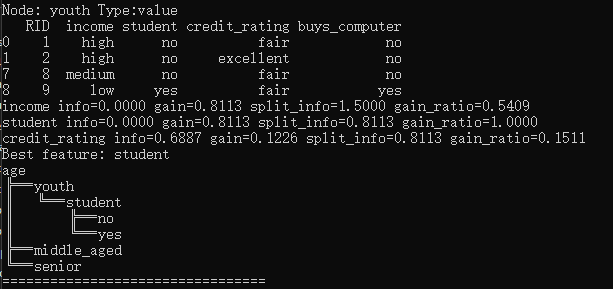
Chapter 4: Implementation

Detailed description and explanation for each of the stages in the classification operations.

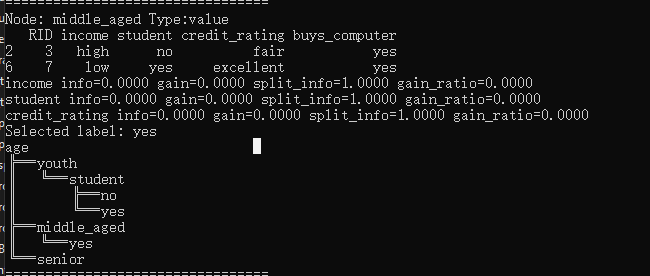
C4.5



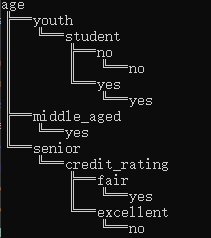
First of all, the software will identify for the first feature to use as the root node. Once it calculates for information gain, gain, split info and gain ratio. It will choose the attribute that have highest gain ratio as the node for the tree. In this case, the best splitting attribute is age. After that, the distinct value for the attribute will added to the node as the child.



After that the data partition will sliced based on the best attribute values. The node will stored the pointer of the sliced data partition and do the step 1 again for this particular node.

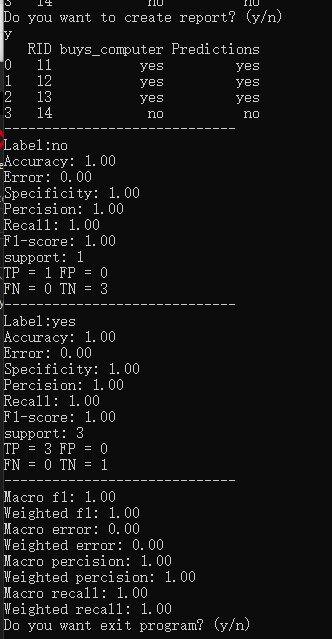


If there is no significance difference and the data partition having the same label, the software will assign the node with the label.

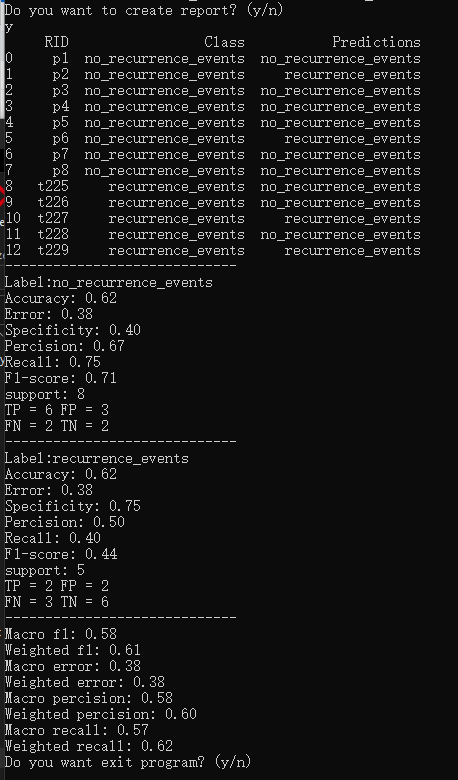


After identifying all the feature and value. A decision tree will be constructed.

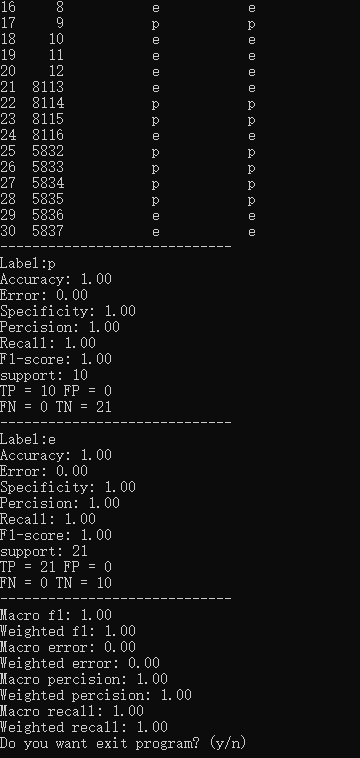
Student test set



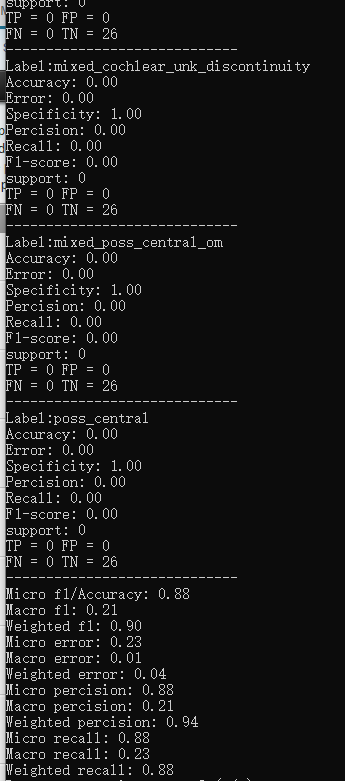
Breast cancer



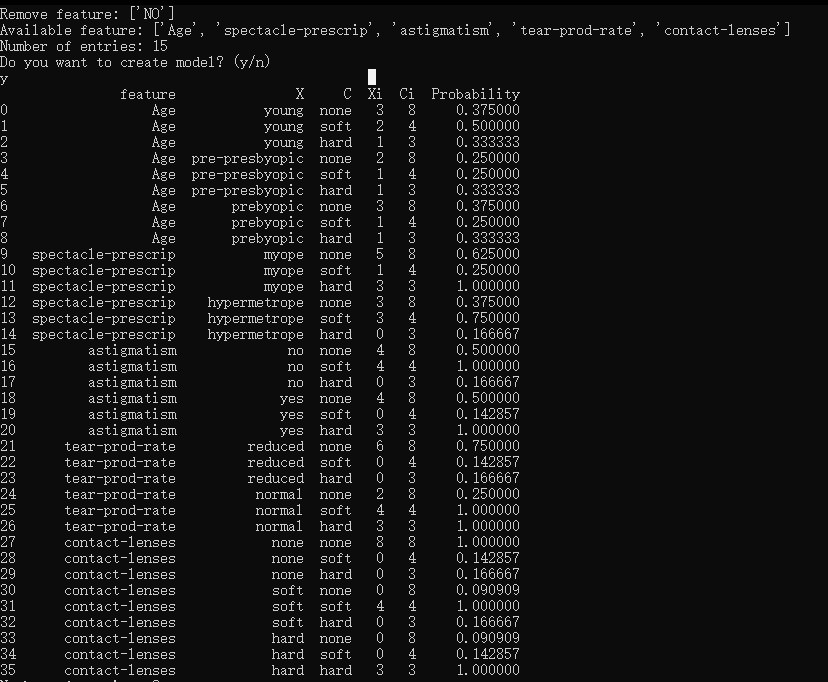
Mushroom



Audiology

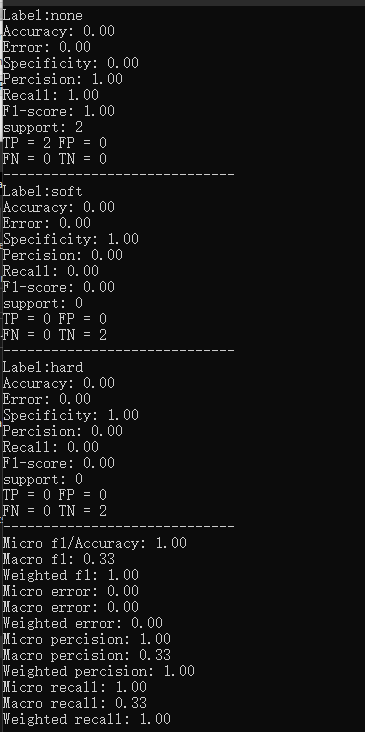


Naïve Bayesian

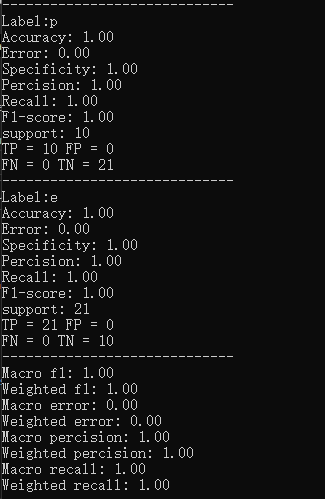


Naïve Bayesian will calculate all the prior probability for every combination of possible given data tuple

The classification result for lense dataset



For mushroom dataset



Chapter 5: Conclusion

In this project, I learn a lot such as the algorithm for decision tree induction, naïve Bayesian algorithm. Apart from that I also learn a lot advanced classification method such as artificial neural network, support vector machine and so on. We also developed C4.5 and naive Bayesian algorithm in this project by using python and compiled to executable file.

Strengths

The strengths of this software using python library such as pandas and numpy which able to process large dataset without considering the memory or array size requirement.

The python provides a more easy and flexible maintenance environment without strictly define the data type.

It is able to store the csv for next iteration.

It is running in console application; it has the flexibility call by another program after few modifications.

Weaknesses

Unable to store the classification model and decision tree.

Cannot export the resulting tree to picture such as jpeg.

Large software size, compiled software required 28MB storage. It can reduce by implementing another programming language or modify the setting for compilation.

Slow training process. The training process performance can increase by implementing parrilla processing.