CLASSIFICATIONS ON WINE INFORMATICS

By

Hai Le

A thesis presented to the Department of Computer Science
and the Graduate School of University of Central Arkansas in partial
fulfillment of the requirements for the degree of

Master of Science
in
Computer Science

Conway, Arkansas
May 2015

TO THE OFFICE OF GRADUATE STUDIES:

The members of the Committee approve the thesis of Hai Le presented on April 17, 2015.

Dr. Bernard Chen,
Committee Chairperson

Dr. Mark Smith

Dr. Shengli Sheng

Dr. Yu Sun

PERMISSION

Title Classifications on Wine Informatics

Department Computer Science

Degree Master of Science

In presenting this thesis/dissertation in partial fulfillment of the requirements for graduate degree from the University of Central Arkansas, I agree that the Library of this University shall make it freely available for inspections. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my thesis/dissertation work, or, in the professor's absence, by the Chair of the Department or the Dean of the Graduate School. It is understood that due recognition shall be given to me and to the University of Central Arkansas in any scholarly use which may be made of any material in my thesis/dissertation.

Hai Le

April 17, 2015

ABSTRACT

Nowadays, because of the rapid development of the Internet and search engines, more people are seeking for knowledge through computers. As a result, more data is being transformed from paper to digital type. Therefore, a new study area called data science is also gaining more focus. The ultimate goal of it is from raw data, the researchers apply techniques to extract meaningful knowledge from it, and provide a data product to the users.

This paper seeks to classify wines based on sensory data derived from Wine Spectator magazine wine reviews. In the new data science field of Wine Informatics, our research serves to support the validity of classification based upon organoleptic properties versus physiochemical analysis as a creditable source for wine classification. Our research included using four classification algorithms, Naïve Bayes, Decision Tree, K-nearest neighbors, and Support Vector Machine. Our data set included a 1000 wine data set with 500 scored as 90+ and 500 scored as 90-. The data set was normalized using Computational wine wheel for preprocessing. We used the 5-fold cross validation to validate the performance of our algorithms with results of 85.7% accuracy prediction achieved using the Naïve Bayes algorithm with k = 2. Even though it is lower than two of the Support Vector Machine methods, it is still very high and can be considered a great achievement.

Table of Contents

List of Tables

List of Figures

List of Formulas

Chapter 1 Introduction

Nowadays, with the improvement of Internet speed and bandwidth, almost any e-commerce application is a data-driven application. The question facing every company today, every start up, every non-profit, every project that wants to attract a community, is how to use data effectively – not just their own data, but all the data that is available and relevant. But merely using data is not really what "data science" means. Using data effectively requires something different from statistics, where actuaries in business suits perform arcane but fairly well-defined kinds of analysis. What differentiates data science from statistics is that data science is a holistic approach. People are increasingly finding data in the wild, and data scientists are involved with gathering data, massaging it into a tractable form, making it tell its story, and presenting that story to others. A data application acquires its value from the data itself, and creates more data as a result. It is not just an application with data; it's a data product. Data science enables the creation of data product. (Loukides)

Data mining is often set in the broader context of knowledge discovery in databases, or KDD. This term originated in the artificial intelligence (AI) research field. The KDD process involves several stages: selecting the target data, preprocessing the data, transforming them if necessary, performing data mining to extract patterns and relationships, and then interpreting and assessing the discovered structures. The process of seeking relationships within a data set – of seeking accurate, convenient, and useful summary representations of some aspect of the data - involves a number of steps. First, determining the nature and structure of the representation to be used. Second, deciding how to quantify and compare how well different representations fit the data. Third, choosing an algorithmic process to optimize the score function. Finally, deciding what principles of data management are required to implement the algorithms efficiently. Data mining is an interdisciplinary exercise. Statistics, database technology, machine learning, pattern recognition, artificial intelligence, and visualization, all play a role. (Hand, Mannila and Smyth)

The earliest evidence of wine making was found in China in 7000 BCE based on fermented honey, rice, and fruit. Since then, with the development of society, and the rise in standard of living, the qualities and varieties of wines are increasing year by year. According to OIV (International Organization of Wine and Vine) (International Organisation of Vine and Wine) estimates, 2011 global production (not taking into account must and grape-juice) is around 2,558 million hectoliters, 700,000 more than in 2010 (Foods & Wines from Spain). OIV also estimates that 2011 global wine consumption is at about 2, 419 million hl (1 hl = 100,000 ml), which is an increase from the previous year of 1.7 million hl (Foods & Wines from Spain). In accordance with this information, it is obvious that wine is one of the most widely consumed beverages in the world and has commercial value as well as social importance. Therefore, the evaluation of the quality of wine plays a very important role for both manufacture and sale (Sun, Li-Xian and Danzer). An established approach to investigate which aspects have significant effects on willingness to pay for food products is to focus on objective characteristics (such as price, brand, and appearance), consumer demographics (such as age, income and education level), and frequency of consumption. One of the most popular and important approaches is chemical analysis for the winery. In this practical evaluation, a wine will be analyzed based on many aspects. For example: Brix: hygrometry or refractometer measures total soluble solids. pH: measure free H^+ ions. Titratable acidity (TA): concentration of all available hydrogen ions (both free and bound to dissociated acids), and many more factors. (Oberholster). Each of the measurements evaluates the wine quality solely based on what wine was made from. For example: Brix value gives estimation of amount of sugar to evaluate: indication of fruit ripeness, potential ethanol production in wine, and follow process of fermentation (Oberholster). When all the aspects are tested, the tester will give a quality estimation of the wine solely based on what it is made from. As a result, sensory properties such as taste, aroma, texture, and flavor are typically not included. However, sensory qualities are often the major factors that affect consumers' perception of a product; therefore, it is necessary to include them in accessing a consumer's preference (Yang). It is also the way this study will focus on applying data mining to evaluating wine.

The second method is of sensory qualifications which is what a professional wine reviewer perceives via organoleptic properties – these being the aspects as experienced by the senses of taste, sight, and smell (Villiers, Alberts and Tredoux). When experts taste a wine, they will try to distinguish as many attributes of the wine as possible. After that, they write a very detailed review about how they judged the wine. As a result, the review would include a distinct set of possible attributes. Even though these qualifications are subjective since a wine reviewer may word their experiences differently from another, most of the time the wine is given a very similar grades among experts because training and experience are required to separate the subjective opinion from the objective evaluation. As the result, the real test for professional wine reviewers is that even if they do not all describe the same sensory attributes in a wine, they similarly and consistently grade the wine into the same classifications. Since analyzing for wine chemical compounds usually contains many technical values such as: Brix, TLC, or FOSS, it is difficult for many people to understand exactly their meaning. Sensory Data, on the other hand, comes from the wine reviewer's ability to derive the tastes of a wine and put to words what our perceptions should be. As a result, it is much easier for manufacturers, sellers, and customers to understand what s inside a wine and why its quality deserved the experts' grades.

Wine expert reviews were stored in human language format, but not many researchers have extracted that infomation into something more useful. For example, *Wine Spectator* consists of wine reviews which are derived from their publication (15 issues a year), in which there are between 400 and 1000 wine reviews in each issue (Spectator.). Another example is *eRobertParker* with over 225,000 researchable professional notes about wine exploration. There are different tastes (different kinds of berries , apples, bananas, and so on), sweetness and bodies in these data of wine. Checking each review might be useful for a few specific wines, but the sheer scale of reading all the reviews to extract useful knowledge makes it impossible. For that reason, Data Science and Data Mining are used because we need methods that computers can handle to scan through all the reviews and extract important key terms called attributes, then process those attributes to archive useful knowledge. As a result, data mining is the perfect field to run wine informatics experiments.

The previous works that inspire us to continue is from Wine Informatics: Applying Data Mining on Wine Sensory Review. (Chen, Rhodes and Crawford). In their work, they implemented methods to scan through each review and extract the important key terms. Among all expert reviews, their data is derived from wine reviews from the *Wine Spectator* magazine using sensory data. They used the *Wine Spectator* data source primarily for its impact on the wine culture due to its extensive wine reviews, ratings and general consistency not to logomachy in wine review. They chose *Wine Spectator* for three reasons: First, it has consistent reviews from prestigious experts. Second, they use blind testing, and finally, expert's reviews are straight to the point. As a result, there is no confusion for the team to extract key terms from each review. The research team applies wine savory as the domain knowledge, and extract useful knowledge based on the wine expert's review. Continuing their work, we apply four other data mining classification techniques called: Decision tree, Naïve Bayes, and K-nearest neighbor, and Support Vector Machine to the same dataset. Our goal is to further analyze the dataset to extract even more useful knowledge.

Chapter 2: Data Preprocessing

2.1: Wine Spectator

As mentioned above, the previous work that inspired me to work on this thesis is from Wine Informatics: Applying Data Mining on Wine Sensory Review (Chen, Rhodes and Crawford). In our work, we implemented the methods to scan through each review and extract the important key terms. Among all expert reviews such as *Wine Advocate*, *Decanters Magazine*, or *eRobertParker*, our data is derived from wine reviews from the *Wine Spectator* magazine using sensory data. We used the *Wine Spectator* for three reasons: First, each year, its editors choose more than 15,000 wines to review with detailed tasting notes, rating, and drink recommendations. As a result, its data source is primarily for its impact on the wine culture. Second, they use a blind-tasted method to ensure that their tasters remain impartial and unbiased, with all wines presented on a level playing field (Shanken and Matthews) and each editor generally covers the same wine regions from year to year. These "beats" remain constant, allowing each lead taster to develop expertise in the region's wines. Other tasters may sit in on blind tastings in order to help confirm impressions; however, the lead taster always has the final say on the wine's rating and description. Finally, it has consistent reviews from prestigious experts, and the reviews are straight and to the point in blind tastings. They set stringent standards for themselves and rely on the proven ability and experience of their editors as tasters and critics and follow the guidelines in order to maintain the integrity of their tastings. The following is an example of sensory attributes contributed by the wine reviewer in a wine review sample:

Kosta Browne Pinot Noir Sonoma Coast 2009 Ripe and deeply flavored, concentrated and well-structured, this full-bodied red offers a complex mix of black cherry, wild berry and raspberry fruit that's pure and persistent, ending with a pebbly note and firm tannins. Drink now through 2018. 5,818 cases made. (Spectator.)

Wine Spectator tasters review wines on the following 100-point scale:

Grade Range	**95-100**	**90-94**	**85-89**	**80-84**	**75-79**	**50-74**
Classifications	Classic	Outstanding	Very Good	Good	Mediocre	Not recommended

Table 1. 100-point scale of Wine Spectator

Like many other wine reviews, *Wine Spectator* provides a wine score for each wine review. Ratings reflect how highly *Wine Spectator* expert regards each wine relative to other wines. Before March 2008, *Wine Spectator* used a single score to reflect the quality of the wine, but after that it switched to point range because experts believed this will better reflect the subtle difference between wines and give readers better information for their buying decisions. There are six different range scores and classification, and table 1 represents all of them. In brief, [100-95] is considered classic: a great wine that is strongly recommended. [90-94] is outstanding: a wine of superior character and style. [85-89] is very good: a wine with special qualities. [80-84] is good: a solid, well-made wine. [75-79] is mediocre, a drinkable wine that may have minor flaws, and [50-74] is not recommended to drink.

2.2. Natural Language Processing.

Each year *Wine Spectator* publishes thousands of wine reviews, so it is impossible for the team to read and process all the reviews manually. As a result, the team needs to develop a Natural Language Processing (NLP) technique to extract key terms from each review automatically. The process of building computer programs that understand natural language involves three major problems: (1) the thought process, (2) the representation, and (3) meaning of the linguistic input and world knowledge (Chowdhury). For example, if using the team's NLP technique processes seen in the review on section 2.1, all the terms that are in bold will be extracted and considered characteristics of the wine.

Kosta Browne Pinot Noir Sonoma Coast 2009

Ripe *and* ***deeply flavored****,* ***concentrated*** *and* ***well-structured****, this* ***full-bodied*** *red offers a complex mix of* ***black cherry****,* ***wild berry*** *and* ***raspberry*** *fruit that's* ***pure*** *and* ***persistent****, ending with a* ***pebbly*** *note and* ***firm tannins****. Drink now through 2018. 5,818 cases made.* (Spectator.)

In this example, (1) the thought process happens after wine experts taste the wine and consider how they want to express the feeling in words. (2) The representation and meaning of the linguistic input is how the experts arranged their words to make a good quality comment, and (3) the world knowledge is the wine knowledge that experts have gained through judgment experiences (look, smell, and taste). That is also the reason why in the previous work, the team included Lorri Hambuchen, a wine expert. During the extraction information process, the team found that several of the attributes could be categorized into a single attribute. Such as FRESH-CUT APPLE, RIPE APPLE, and APPLE could be categorized into a single category APPLE; but GREEN APPLE stands out enough to make its own unique attribute. It is very easy for people who are not familiar with wines to get confused. For that reason, the team 's wine expert helps to distinguish all the similar terms, to make sure that the team does not mess up characteristics of the wines.

The Natural Language Processing technique that we use is called Information Extraction. This technique refers to the automatic extraction of structured information such as entities, relationships between entities, and attributes describing entities from unstructured source. In previous research, we mainly focused on speech tagger, which assigns each word to a grammatical category coming from a fixed set, and its most important step is preprocessing libraries for wine reviews. In other words, we need to build a dictionary of key terms. Wine reviews are unstructured text because they do not form into a list or database, so they are just plain texts that can be treated as a set of key terms concentrated together (Sarawagi). The key terms dictionary is built based on each word coming from the reviews, and the set of key terms includes the conventional part of text such as noun, verb, adjective, adverb, conjunct, and pronoun (Sarawagi). In our research, since *Wine Spectator's* reviews mostly contain flavor and feeling expression words, we focused on noun and adjective key terms. At the beginning, we used the Wine Aroma Wheel, made by professor Ann C. Noble, as an initial start. The wheel has very general terms located in the center (e.g. fruity or spicy), going to the more specific terms in the outer tier (such as strawberry or clove). These key terms are not the only words that can be used to describe wines, but represent ones that are most often encountered. (Noble). Figure 1 is a sample of a Wine Aroma Wheel

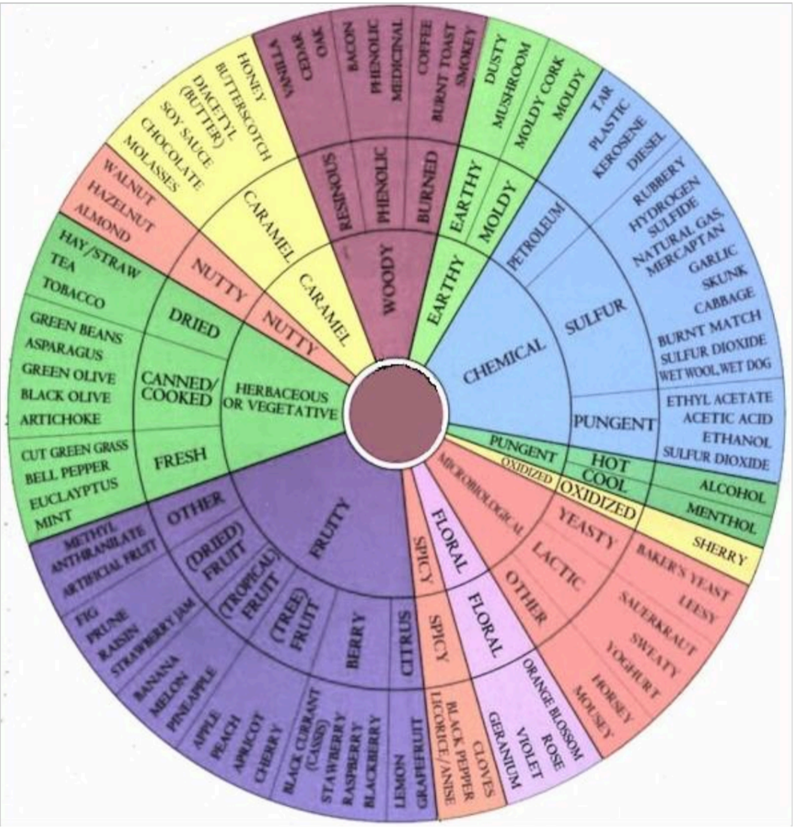


Figure 1. Ann C. Noble's Wine Aroma Wheel

While the Wine Aroma Wheel is a great start, we found it lacking to capture adjective key terms such as tannins, acidity, body, structure, or finish. We also wanted to capture attributes that give the feeling of expression such as: BEAUTIFUL, or SMOOTH. These descriptions, while not actual tasting notes, still add a subtle amount of character to a wine that we wished to capture. As the result, we developed our own wine wheel called the Computational Wine Wheel.

2.3 Computational Wine Wheel

The purpose of the Computational Wine Wheel is not only to capture all flavors but also feeling expressions described in adjective in experts' reviews. In our opinion, those key terms play important roles in our research as well. For example, if APPLE flavor appear in both [90-94] and [70-75] wines, those adjective words such as WELL-STRUCTURES, BEAUTIFUL, or AGE WELL might show the difference between them. Our Computational Wine Wheel is compiled from the list of “Top 100 Wines of 2011” (*Spectator.*), and the goal is capture all attributes demonstrated by representative wines of the year. After analyzing all one hundred wine reviews and adding all necessary subcategories, the Computational Wine Wheel came out with a total of 547 distinct attributes. Since many experts write the reviews, we realize that many attributes just represent the same thing. Such as PEPPER and PEPPERY both represent PEPPER. Because of this, the Computational Wine Wheel introduces Normalized Attributes to capture all similar attributes. As mentioned in section 2.2, the team has Lori Hambuchan, a wine expert, to help us distinguish all the key terms, which attributes are considered the same and which attributes are important enough to be unique. After Normalized Attributes process, the normalized attributes were cut down from 547 to 376, and formed our Computational Wine Wheel. Below is a small example of the Categorical Summary of Wine Attributes that we used to normalize our data attribu*tes* (eRobertParker).

```
CATEGORY_NAME|SUBCATEGORY_NAME|SPECIFIC_NAME|NORMALIZED_NAME|WEIGHT
FRUITY|TREE FRUIT|APPLE|APPLE|3
FRUITY|TREE FRUIT|RIPE APPLE|APPLE|3
OVERALL|TANNINS|TANNIC BASE|TANNINS_LOW|2
OVERALL|TANNINS|TANNINS|TANNINS_LOW|2
OVERALL|FLAVOR/DESCRIPTORS|ACCENTS|ACCENTS|1
OVERALL|TANNINS|FINELY WOVEN TANNINS|TANNINS_MEDIUM|2
OVERALL|TANNINS|FIRM TANNINS|TANNINS_HIGH|2
OVERALL|TANNINS|FLESHY TANNINS|TANNINS_HIGH|2
OVERALL|TANNINS|GRACEFUL TANNINS|TANNINS_MEDIUM|2
```

```
EARTHY|EARTHY|SLATE|MINERAL|3
EARTHY|EARTHY|WET EARTH|MINERAL|3
EARTHY|MOLDY|MOLDY|MOLDY|3
```

(B. Chen)

Each characteristic has five different descriptions. First, the category name is the most general term. For example, any characteristic that involves fruit will be listed under fruity category. Second is the subcategory name that is the more detailed version of the category name. Following the example, the fruit category will have sub categories such as: berry, tree fruit, tropical fruit, dried fruit, etc. Third is the specific name; this is the original name that was extracted from expert's reviews. Fourth is the normalized name. This is where the team of wine experts will tell exactly what *Wine Spectator's* expert means in their reviews. For example: FLESHLY TANNINS, FIRM TANNINS, or GRAINY TANNINS are considered TANNING HIGH, but GRACEFUL TANNINS, FINE TANNINS, or STURDY TANNINS are considered TANNING MEDIUM. Finally, the weight describes how important one attribute is compared to others, ranging from 1 to 3. For example, if a wine has both APPLE and AGE WELL in an expert's review, then in some circumstances, the APPLE attribute will be considered more important than the AGE WELL attribute.

In order to explain how we use the wheel, here is an example that uses a simplified version of our Computational Wine Wheel.

CATEGORY	SUBCATEGORY	SPECIFIC	NORMALIZED	WEIGHT
FRUITY	TROPICAL FRUIT	PLUM	PLUM	3
FRUITY	TROPICAL FRUIT	BANANA	BANANA	3
HERBS/VEGETABLES	CANNED/COOKED	BLACK OLIVE	BLACK OLIVE	3
OVERALL	FINISH	BALANCED FINISH	EXCELLENT FINISH	2
OVERALL	FINISH	BEAUTIFUL FINISH	EXCELLENT FINISH	2
OVERALL	FINISH	FINISH EXPANDS	LONG FINISH	2

Table 2. Simplified computational wine wheel

The simplified Computational Wine Wheel in table 2 has six specific attributes and five normalized attributes. Because PLUM, BANANA, and BLACK OLIVE are unique enough, the key terms are kept as original words after normalization. With category OVERALL and subcategory FINISH, we follow Lori Hambuchan's suggestions, so that BALANCED FINISH and BEAUTIFUL FINISH are normalized to EXCELLENT FINISH, while FINISH EXPANDS is considered LONG FINISH. Here is the process of how we apply the wheel on the following wine:

CAYUSE Syrah Walla Walla Valley En Cerise Vineyard, 2009:

Rich, supple and opulent, this is generous with its blackberry, purple plum, black olive, tobacco and dusky spice flavors, remaining complex and harmonious through the long, balanced finish. Drink now through 2019. (Spectator.)

First, we use the specific key terms (the 3rd column in table 2) to scan through the whole review starting with the longest number of combinations in table 2. Since the longest key terms in the table are 2, we start with BLACK OLIVE, BALANCED FINISH, BEAUTIFUL FINISH, and FINISH EXPANDS. Each time we hit a match key term in the review, the wine will have a positive attribute in the corresponding NORMALIZED attribute. Next, we remove the match word from the review. The processing is repeated until all two-combination words are checked. Finally, we scan the review again with single word specific attributes, and below is *CAYUSE Syrah Walla Walla Valley En Cerise Vineyard* wine after the process. Using the simplified Computational Wine Wheel, the bold key terms are detected and represented as value 1.

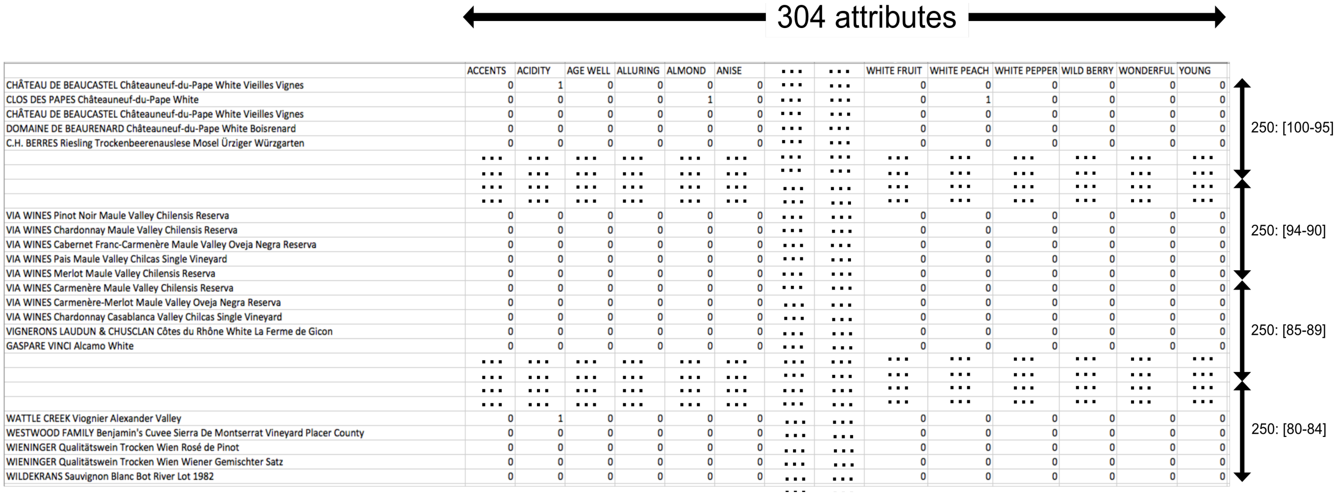
CAYUSE Syrah Walla Walla Valley En Cerise Vineyard, 2009: Rich, supple and opulent, this is generous with its blackberry, purple ***plum****,* ***black olive****, tobacco and dusky spice flavors, remaining complex and harmonious through the long,* ***balanced finish****. Drink now through 2019.*

Wine name	PLUM	BANANA	BLACK OLIVE	EXCELLENT FINISH	LONG FINISH
Syrah Walla Walla Valley En Cerise Vineyard	1	0	1	1	0

Please note that even though there are more important key words in the wine such as: blackberry, tobacco, or dusky spice, they are not mentioned in this example. The reason is we use the simplified version of our Computational Wine Wheel. The complete wheel with 374 NORMALIZED attributes will capture all of them.

2.4 Data Set

The data set we used in this thesis is based on 1000 wine reviews that we retrieved from *Wine Spectator*, and we processed these wines through our computational wine wheel. Out of 374 normalized attributes, 304 of those are appeared in our 1000 wines, and figure 2 is a representation of 1000 wines.



304 attributes
250: [100-95]
250: [94-90]
250: [85-89]
250: [80-84]

	ACCENTS	ACIDITY	AGE WELL	ALLURING	ALMOND	ANISE	...	...	WHITE FRUIT	WHITE PEACH	WHITE PEPPER	WILD BERRY	WONDERFUL	YOUNG
CHÂTEAU DE BEAUCASTEL Châteauneuf-du-Pape White Vieilles Vignes	0	1	0	0	0	0	...	...	0	0	0	0	0	0
CLOS DES PAPES Châteauneuf-du-Pape White	0	0	0	0	1	0	...	...	0	1	0	0	0	0
CHÂTEAU DE BEAUCASTEL Châteauneuf-du-Pape White Vieilles Vignes	0	0	0	0	0	0	...	...	0	0	0	0	0	0
DOMAINE DE BEAURENARD Châteauneuf-du-Pape White Boisrenard	0	0	0	0	0	0	...	...	0	0	0	0	0	0
C.H. BERRES Riesling Trockenbeerenauslese Mosel Ürziger Würzgarten	0	0	0	0	0	0	...	...	0	0	0	0	0	0
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
VIA WINES Pinot Noir Maule Valley Chilensis Reserva	0	0	0	0	0	0	...	...	0	0	0	0	0	0
VIA WINES Chardonnay Maule Valley Chilensis Reserva	0	0	0	0	0	0	...	...	0	0	0	0	0	0
VIA WINES Cabernet Franc-Carmenère Maule Valley Oveja Negra Reserva	0	0	0	0	0	0	...	...	0	0	0	0	0	0
VIA WINES Pais Maule Valley Chilcas Single Vineyard	0	0	0	0	0	0	...	...	0	0	0	0	0	0
VIA WINES Merlot Maule Valley Chilensis Reserva	0	0	0	0	0	0	...	...	0	0	0	0	0	0
VIA WINES Carmenère Maule Valley Chilensis Reserva	0	0	0	0	0	0	...	...	0	0	0	0	0	0
VIA WINES Carmenère-Merlot Maule Valley Oveja Negra Reserva	0	0	0	0	0	0	...	...	0	0	0	0	0	0
VIA WINES Chardonnay Casablanca Valley Chilcas Single Vineyard	0	0	0	0	0	0	...	...	0	0	0	0	0	0
VIGNERONS LAUDUN & CHUSCLAN Côtes du Rhône White La Ferme de Gicon	0	0	0	0	0	0	...	...	0	0	0	0	0	0
GASPARE VINCI Alcamo White	0	0	0	0	0	0	...	...	0	0	0	0	0	0
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
WATTLE CREEK Viognier Alexander Valley	0	1	0	0	0	0	...	...	0	0	0	0	0	0
WESTWOOD FAMILY Benjamin's Cuvee Sierra De Montserrat Vineyard Placer County	0	0	0	0	0	0	...	...	0	0	0	0	0	0
WIENINGER Qualitätswein Trocken Wien Rosé de Pinot	0	0	0	0	0	0	...	...	0	0	0	0	0	0
WIENINGER Qualitätswein Trocken Wien Wiener Gemischter Satz	0	0	0	0	0	0	...	...	0	0	0	0	0	0
WILDEKRANS Sauvignon Blanc Bot River Lot 1982	0	0	0	0	0	0	...	...	0	0	0	0	0	0

Figure 2. A representation of 1000 wines.

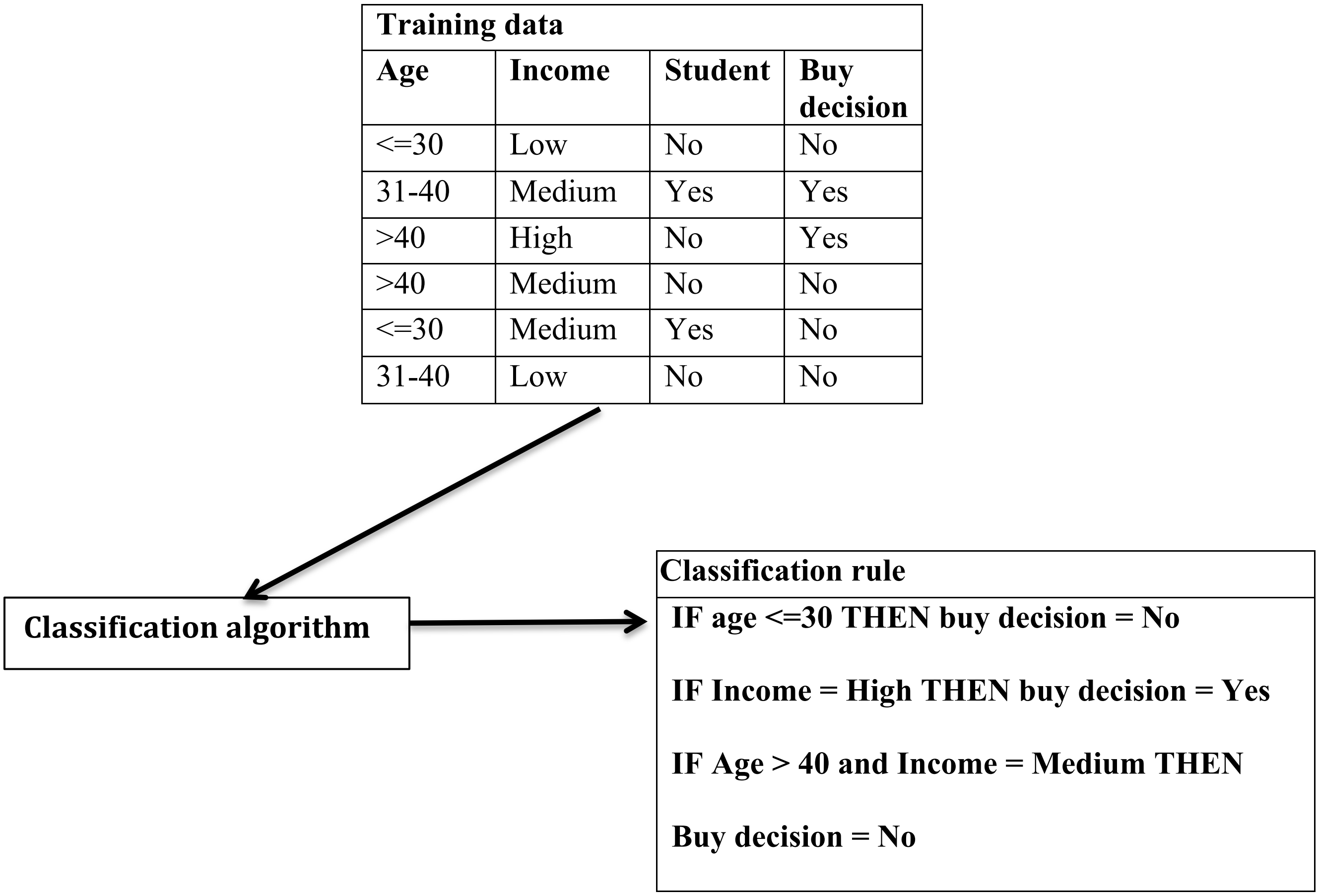
304 columns represent the entire normalized attributes that the computation wine wheel captures, while the rows represent 1000 wines. Among these wines, 250 were in the [80-84] scores category, 250 were in the [85-89] scores category, 250 were in the [90-94] scores category and 250 were in the [95-100] scores category. All of the wines follow the same process as *CAYUSE Syrah Walla Walla Valley En Cerise Vineyard, 2009* in section 2.3; which means if a wine review for an individual wine contained an attribute, a 1 was listed in the column for the attribute indicating ‘positive’, otherwise a 0 was listed for ‘negative’. Because the thesis will mainly focus on classification algorithms, we need to assign labels for each wine. Therefore, we further divided all 1000 wines into two subcategories: 500 wines in [90-100] scores are ‘90+’ category, and 500 wines between [80-89] score are ‘90- category.

Chapter 3: Classifications

3.1 What is classification?

Classification is one of the most common learning models in data mining, and it is a form of data analysis that extracts models describing important data classes. Such models, called classifiers, predict categorical class labels (Han, Kamber and Pei). Researchers have proposed many classification methods in machine learning, pattern recognition, and statistics, and one of their goals is to apply those methods to real life applications. For example, the manager of an electronics store can use information of a customer such as: age, job, or income and predict whether he/she will buy a computer or not. An owner of a supermarket needs to analyze all the transactions to decide if a new product will sell well or not. In each of these examples, the data analysis task is classification, and its model or classifier is constructed to predict class labels, such as "buy computer" or "not buy computer" for the electronics store, and "sell" or "not sell" for the supermarket.

Data classification is a two-step process: learning and classification. First, a classification model is constructed by using algorithms in the learning step (Han, Kamber and Pei). A dataset called training data will be used to build the model, and it is provided based on the decision-making records in the history. Each record represents a profile, and it must have a label. For example, the label of a customer of the electronics shop will either "buy computer" or "not buy computer". Without the labels, classification algorithms would not have enough information to build their own decision-making models. For example, the learning step of the electronics store above is shown in figure 3.



Training data

Age	Income	Student	Buy decision
<=30	Low	No	No
31-40	Medium	Yes	Yes
>40	High	No	Yes
>40	Medium	No	No
<=30	Medium	Yes	No
31-40	Low	No	No

Classification rule

IF age <=30 THEN buy decision = No

IF Income = High THEN buy decision = Yes

IF Age > 40 and Income = Medium THEN

Buy decision = No

Figure 3. The data classification process

As the figure shows, each tube (row) in the training data stores customer information and his/her decision in the store's past, and "buy decision" is the class label. A classification algorithm will be implemented to process all the information in the training set. After that, a set of classification rules is built for the second step, which is a classification step where the model is used to predict class label for given data. Figure 4 shows how classification step processes follow the electronics store example

Classification rule

IF age <=30 THEN buy decision = No
IF Income = High THEN buy decision = Yes
IF Age > 40 and Income = Medium THEN
Buy decision = No

New data				
	Age	**Income**	**Student**	**Buy decision**
Customer A	<30	Low	Yes	?
Customer B	>40	High	No	?
Customer C	31-40	Medium	Yes	?
Customer D	<30	Low	Yes	?
Customer E	>40	High	No	?

Figure 4. The process of classification step

A real life example using classification is: The manager wants to run a promotion by providing coupons to potential customers. Since the cost to design, print out, and distribute coupons is not free, the manager does not want to provide them to all customers. Therefore, he will use the classifier to choose which customers have the highest chance of purchasing a new computer. Based on the rules and data provided in figure 4, customers who have high incomes are promising, so the manager will provide coupons to only customers B and E.

Because all that classifiers do is prediction, there will be a chance they may predict incorrectly. To evaluate the effectiveness of the classification models, we use prediction accuracy, which is the percentage of test sets that are correctly classified by the classifier (Han, Kamber and Pei). For example, if the model predicts that customer B will buy a computer, and in the end, he does buy one, then the model predicts correctly. Otherwise, the model predicts incorrectly. After the model predicts all the customers in the test set, the percentage of accuracy will be generated. If the dataset has ten customers and the model predicts the decision correctly eight out of ten, we say that the accuracy of the model is 80%. The accuracy value is very important because the higher the accuracy, the better quality the model. Depending on the training data, different models will provide a different quality of the predictions.

3.2 Classification applications

One of the main purposes of the classification models is how we can apply them to the real life fields. Below are the descriptions of four different applications in medical, business, biology, and security, and all of them use classifiers to predict the labels. Many classifiers have been developed in order to provide options for researchers; therefore in these applications, researchers not only use one model, but also many more in order to compare them and find which one is the best for their datasets.

3.2.1 Medical Image Classification

In their research, Antonie, Za and Coman (Antonie, Za and Coman) tried to use computers to assist doctors in predicting breast cancer in women. They claimed that mammography is considered the most reliable method in early detection of breast cancer. Since the digital mammograms are among the most difficult medical images to be read because of their low contrast and differences in the types of tissues, the accuracy rate tends to be low when physicians read them. That is why the computer aided diagnosis systems are necessary to assist the medical staff to achieve high efficiency and effectiveness. The methods proposed in their paper classify the digital mammograms into two different labels: normal and abnormal. The normal ones are those characterizing a healthy patient, and the abnormal ones are characterizing potential cancer cases. First, they collected and pre-processed 322 images. Second, they used neural networks (Hagan, Demuth and Beale) and association rules as classification algorithms (Hipp, Jochen and Güntzer)to build the model. Basically, they compared each pixel between the test image and the classification rule to predict the accuracy. In the experimental results, neutral networks achieved the average accuracy of 81.24%, while the association rule performed worse with an accuracy of 69.11%. In conclusion, they claimed "the computer-aided methods they presented could assist medical staff and improve the accuracy of detection". (Antonie, Za and Coman)

3.2.2 Credit scoring decision

In the research of Huang, Chen, and Wang (Huang, Chen and Wang), the team built a model to classify credit score decisions using support vector machines. They cited the motivation for their work was "the credit card industry has been growing rapidly recently, and the credit-scoring manager often evaluates the consumer's credit with intuitive experience. However, with the support of the credit classification models, the manager can accurately evaluate the applicant's credit score" (Huang, Chen and Wang). Credit scoring models were developed to categorize applicants as either accepted or rejected, and the team wanted to increase the prediction accuracy. First, they noticed that the credit department of the bank collects huge numbers of consumers' credit data, so the data became a valuable source for training datasets. They decided to use Australian (690 instances) and German (1000 instances) credit data sets from the UCI Repository of machine Learning Databases. Next, they applied three different strategies of support vector machines (Furey) namely "SVM + Grid," "SVM +Gird + F-score," and "SVM +GA," to built three different models. Next, the team applied each model to the test dataset and generated the accuracy, and the accuracy is represented in table 3.

Dataset Strategies	**Accuracy**	
	Australian	**German**
SVM + Grid search	85.51%	76.00%
SVM + Grid search + F-score	84.20%	77.50%
SVM + GA	86.90%	77.92%

(Huang, Chen and Wang).

Table 3. The results of two datasets and three strategies

Overall, all three models performed very well. "SVM + GA" strategy generated the highest accuracy. Notice that the accuracy of the Australian dataset is higher than the German dataset, and the reason is because the German dataset is more unbalanced. While the ratio between accepted and rejected of the Australian dataset is 307/383, the ratio of the German dataset is 700/300. In conclusion, the team claimed that compared to traditional statistical techniques, "the artificial intelligence techniques (such as SVM, GP, BPN or decision tree) do not require the knowledge of the underlying relationships between input and output variables" (Huang, Chen and Wang). Based on the accuracy, these techniques are worthwhile as an extra method to strengthen the final decision of a credit-scoring applicant.

3.2.3 Apply classification to Protein Interaction Prediction

"Protein–protein interactions (PPI) form the physical basis for formation of complexes and pathways that carry out different biological processes, and play a key role in many biological systems. High-throughput methods could directly detect the set of interacting proteins in yeast, but the results were often incomplete and exhibit high inaccuracy rates" (Qi, Bar-Joseph and Klein-Seetharama). The team proposed other methods using supervised learning to integrate direct and indirect biological data sources for the protein interaction prediction task. In this example, the label of the classifier was either "positive" or "negative". Positive meant there were interactions between proteins, and negative meant no interactions. First, they used three gold standard training datasets including: Database of Interacting Proteins (DIP), the Munich Information Center for Protein Sequences (MIPS), and the Kyoto Encyclopedia of Genes and Genomes (KEGG). The authors claimed that these three databases provide the gold standard for inferring pathway networks of protein interactions. Next, they applied 6 different classification methods: Support vector machines (Furey), Naïve Bayes (McCallum, Andrew and Nigam), Logistic regression (Menard), Decision tree (Safavian), Random forest, and Random forest-based k-Nearest Neighbor (Liaw, Andy and Wiener) in order to predict the test datasets. They varied the specific prediction tasks by predicting (a) direct (physical) protein–protein interactions using the DIP dataset, (b) protein co-complex relationships using the MIPS dataset, and (c) protein co-pathway relationship using the KEGG-pathway dataset. They achieved an accuracy of 68% for MIPS dataset, and lower than 40% for DIP and KEGG datasets. In the conclusion, they showed that the co-pathway relationship was the easiest one to predict. The reason was the MIPS dataset was built to favor co-complex relationships prediction. The team believed that if they had more training data, the accuracy of other two tasks would increase as well.

3.2.4 Classification for Building Intrusion Detection models

"As network-based computer systems play increasingly vital roles in modern society, they have become the target of intrusions by our enemies and criminals. In addition to intrusion prevention techniques, such as user authentication (e.g. using passwords or biometrics), avoiding programming errors, and information protection (e.g., encryption), intrusion detection is often used as another wall to protect computer systems. Many Intrusion Detection Systems (IDS) only handle one particular audit data source and their updates are expensive and slow" (Lee, Stolfo and Mok). Therefore, in their research (Lee, Stolfo and Mok) proposed several methods using data mining techniques to detect computer intrusions. They chose association rule (Hipp, Jochen and Güntzer) to classify each audit record into either normal or a particular kind of intrusions. They developed classification rules by combining the rules of existing models with new rules that were trained on new data. They participated in the DARPA Intrusion Detection Evaluation Program, and DARPA provided them a standard set of extensively gathered audit data. The training data included four main categories of attacks: denial-of-service (DOS), unauthorized access from a remote machine (R2L), unauthorized access to local super user privileges by a local unprivileged (URL), and PROBING. Beside the association rule, they also used frequent episodes to represent the sequential audit record patterns. With that, the model could understand the nature of many attacks, and increase its detection accuracy. Table 4 shows their experimental results.

Category	Accuracy %
DOS **PROBING** **U2R** **R2L**	79.7 97.0 75.0 60.0
Overall	80.2

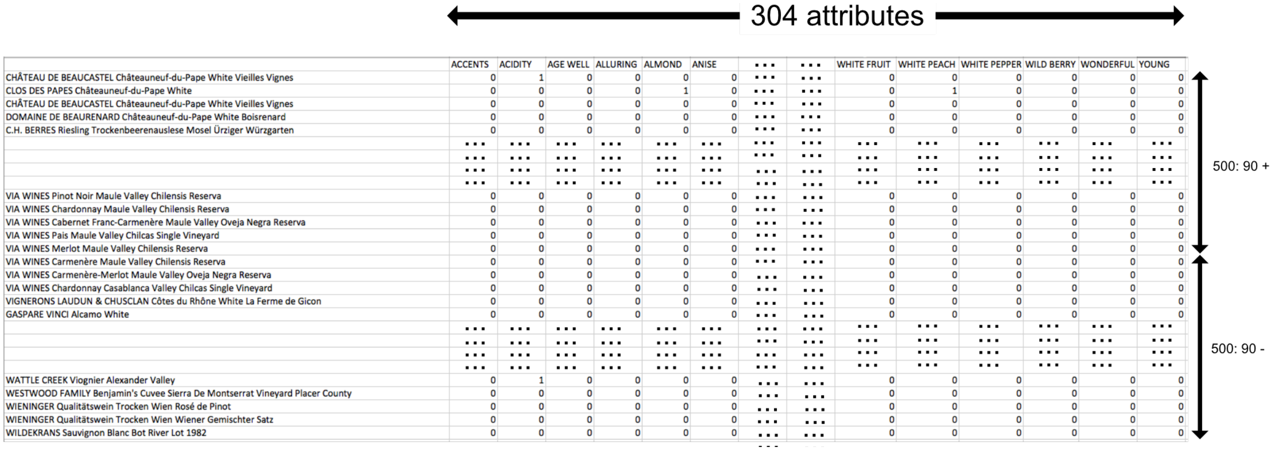
(Lee, Stolfo and Mok)

Table 4. Detection Rates of the model

Their classification models predicted very well in three types of attack with the accuracy over 70%. They claimed that R2L is always the most difficult type to detect, so 60% was acceptable, but not good enough in a mission critical environment. In conclusion, their experiments showed that "the frequent patterns mined from audit data can be used as reliable user anomaly detection models, and as guidelines for selecting temporal statistical features to build effective classification models" (Lee, Stolfo and Mok)**.**

3.3 Wine Informatics and classifications

All of the examples above prove that classification models are worth consideration as valuable methods to strengthen the final decision in many real life situations. The researchers not only have work on those four fields, but also many more such as: sports (Lavrač), music (Pachet, Gert Westermann and Laigre.), or climate change (Steinbach). One of the fields which has a long history, but not many people use data mining techniques to evaluate is wine informatics. Expert wine reviews were stored in human language format, and just a few researchers have extracted that infomation into something more useful. With thousands of reviews being released over the years, they are valuable sources to apply data mining methods. Due to the fast speed, low cost, and consistency of computers, we want to use them to develop several techniques, and research useful information in wine reviews. Because the previous work from Wine Informatics: Applying Data Mining on Wine Sensory Review (Chen, Rhodes and Crawford) provides a very good training data (figures 2.4.1) which clearly classifies all 1000 wines into 4 different categories, we continue using that training set to build more classification models. All the examples in section 3.2 use classifiers to differentiate between two labels because it is very challenging to predict more than two decisions. Time complexity increases a lot, and the accuracy usually is very low. Therefore, instead of trying to categorize four labels: [80-84], [85-89], [90-94], [95-100], we divided 1000 wines into two categories: “90+” and “90-” as figure 5 shows:

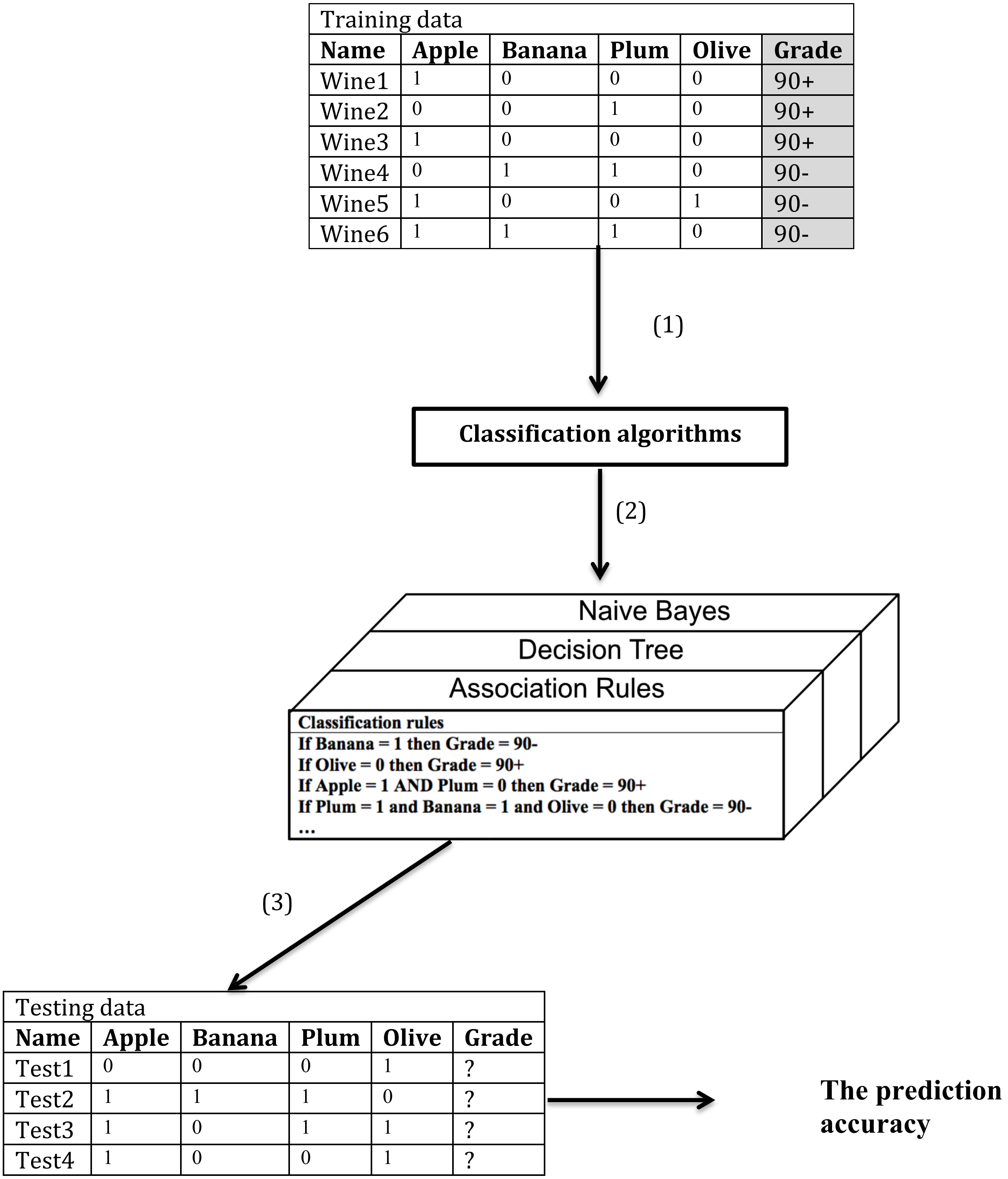


	ACCENTS	ACIDITY	AGE WELL	ALLURING	ALMOND	ANISE	...	...	WHITE FRUIT	WHITE PEACH	WHITE PEPPER	WILD BERRY	WONDERFUL	YOUNG
CHÂTEAU DE BEAUCASTEL Châteauneuf-du-Pape White Vieilles Vignes	0	1	0	0	0	0	...	...	0	0	0	0	0	0
CLOS DES PAPES Châteauneuf-du-Pape White	0	0	0	0	1	0	...	...	0	1	0	0	0	0
CHÂTEAU DE BEAUCASTEL Châteauneuf-du-Pape White Vieilles Vignes	0	0	0	0	0	0	...	...	0	0	0	0	0	0
DOMAINE DE BEAURENARD Châteauneuf-du-Pape White Boisrenard	0	0	0	0	0	0	...	...	0	0	0	0	0	0
C.H. BERRES Riesling Trockenbeerenauslese Mosel Ürziger Würzgarten	0	0	0	0	0	0	...	...	0	0	0	0	0	0
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
VIA WINES Pinot Noir Maule Valley Chilensis Reserva	0	0	0	0	0	0	...	...	0	0	0	0	0	0
VIA WINES Chardonnay Maule Valley Chilensis Reserva	0	0	0	0	0	0	...	...	0	0	0	0	0	0
VIA WINES Cabernet Franc-Carmenère Maule Valley Oveja Negra Reserva	0	0	0	0	0	0	...	...	0	0	0	0	0	0
VIA WINES Pais Maule Valley Chilcas Single Vineyard	0	0	0	0	0	0	...	...	0	0	0	0	0	0
VIA WINES Merlot Maule Valley Chilensis Reserva	0	0	0	0	0	0	...	...	0	0	0	0	0	0
VIA WINES Carmenère Maule Valley Chilensis Reserva	0	0	0	0	0	0	...	...	0	0	0	0	0	0
VIA WINES Carmenère-Merlot Maule Valley Oveja Negra Reserva	0	0	0	0	0	0	...	...	0	0	0	0	0	0
VIA WINES Chardonnay Casablanca Valley Chilcas Single Vineyard	0	0	0	0	0	0	...	...	0	0	0	0	0	0
VIGNERONS LAUDUN & CHUSCLAN Côtes du Rhône White La Ferme de Gicon	0	0	0	0	0	0	...	...	0	0	0	0	0	0
GASPARE VINCI Alcamo White	0	0	0	0	0	0	...	...	0	0	0	0	0	0
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
WATTLE CREEK Viognier Alexander Valley	0	1	0	0	0	0	...	...	0	0	0	0	0	0
WESTWOOD FAMILY Benjamin's Cuvee Sierra De Montserrat Vineyard Placer County	0	0	0	0	0	0	...	...	0	0	0	0	0	0
WIENINGER Qualitätswein Trocken Wien Rosé de Pinot	0	0	0	0	0	0	...	...	0	0	0	0	0	0
WIENINGER Qualitätswein Trocken Wien Wiener Gemischter Satz	0	0	0	0	0	0	...	...	0	0	0	0	0	0
WILDEKRANS Sauvignon Blanc Bot River Lot 1982	0	0	0	0	0	0	...	...	0	0	0	0	0	0

Figure 5. A presentation of 1000 wines with two labels

All 304 attributes and 1000 wines order are kept the same as the previous work dataset. The only difference is now the first 500 wines are classified as above 90 (90+), and the last 500 wines are classified as below 90 (90-).

In order to explain how we apply classification methods to the Wine dataset, an example with simplified dataset is provided in figure 6.



Training data

Name	Apple	Banana	Plum	Olive	Grade
Wine1	1	0	0	0	90+
Wine2	0	0	1	0	90+
Wine3	1	0	0	0	90+
Wine4	0	1	1	0	90-
Wine5	1	0	0	1	90-
Wine6	1	1	1	0	90-

Classification rules

If Banana = 1 then Grade = 90-
If Olive = 0 then Grade = 90+
If Apple = 1 AND Plum = 0 then Grade = 90+
If Plum = 1 and Banana = 1 and Olive = 0 then Grade = 90-
…

Testing data

Name	Apple	Banana	Plum	Olive	Grade
Test1	0	0	0	1	?
Test2	1	1	1	0	?
Test3	1	0	1	1	?
Test4	1	0	0	1	?

Figures 6. The classification process on simplified wine dataset

The wine classification process includes 3 steps: (1) After using Information Extraction to extract all the key terms from wine reviews, a training data set is formed with grade labels of 90+ or 90-, and we analyze it with different classification methods including: Association rule, decision tree, and Naïve Bayes. (2) After that, a set of classification rules is generated for each model. In figure 3.3.2, the rules appeared in the figure are a set of association rules, which "generating frequent item-sets in other to reveal the underlying patterns in wine profiles" (Chen, Rhodes and Crawford). For example, one of the association rules is: **If Banana = 1 then Grade = 90-.** It means if a new wine we want to test has flavor banana, the model will predict the wine's grade is below 90. (3) A test dataset will be tested by the classification rules to generate the accuracy. Because there is no such method suitable for all the datasets, we try to apply some of the most popular classification methods including: Decision tree, Naïve Bayes, and k-nearest neighbor. Based on the accuracy, we will not only know which method suits best for wine dataset, but also extract even more useful knowledge such as: how wine dataset interacts with each classification model, or how weight attributes affect the final results.

3.4 n – fold cross validation

In order to evaluate how good a model is when applying it on a training dataset, we will need a test set to generate the prediction accuracy. In a real life situation, finding a good testing set is usually a difficult task and very time consuming. Back to the example of the electronics store in section 3.1, the manager needs to put some investigation and time to gather a potential customer list, then he has to wait until customers make decisions to know if the model predicts well or not. In a wine dataset, it is almost impossible for us to find a new wine without any reviews to analyze its attributes, predict its label, and wait for the expert's grade to compare. Therefore, instead of making our own testing data, we use n- fold cross validation.

Cross validation is "a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction and one wants to estimate how accurately a predictive model will perform in practice" (Kohavi). In a wine dataset, 1000 wines will be divided into two different subsets: training subset and testing subset. In order to avoid over-fitting, which is when some rows in the training set also are used in the testing set, we apply k-fold cross validation techniques. Basically, 1000 wines are partitioned into k equal parts. k – 1 parts will be used as the training set and the left our part will be used as the testing set. To avoid any bias, each part is used as the testing set. As a result, a classification model needs to run k times to fulfill the requirement, and the overall result is the average of k results after the model generates them. Suppose we use 5 fold cross validation on a 1000 wine dataset; Figure 7 shows exactly how it is processed:

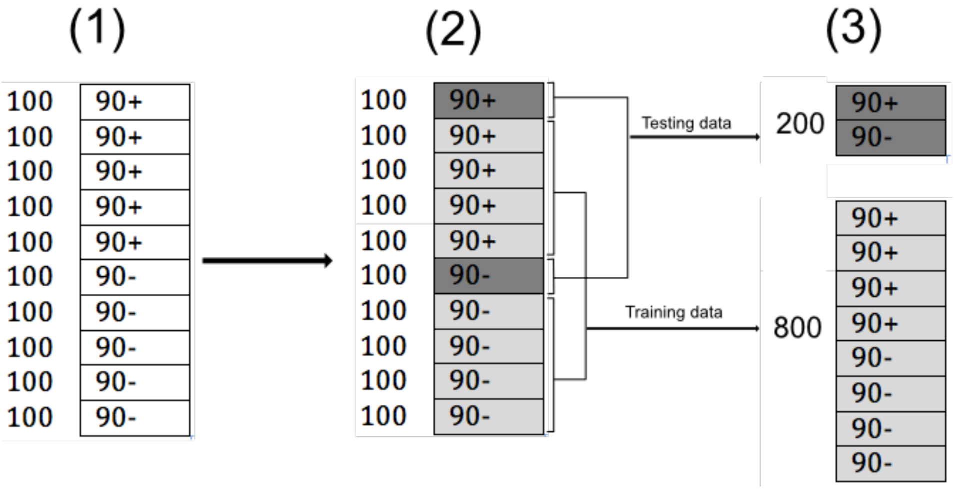


Figure 7. 5 fold cross validation on 1000 wines

First, 1000 wines are divided into ten equal partitions. The order of the wines are maintained, so the first 5 partitions are 500 "90+" wines, and the last 5 partitions are 500 "90-" wines. Second, we take the first 100 wines of both labels and form the testing dataset of 200 instances. The remaining 800 wines are formed into the training dataset. Please notice that after the division, the number of instances of two labels is still balanced. Next, we apply classifiers to these 800 wines, build models, and generate accuracy prediction. The process is repeated until all partitions are formed into testing data. Figure 8 shows the how 5 fold cross validation chooses its testing data each time.

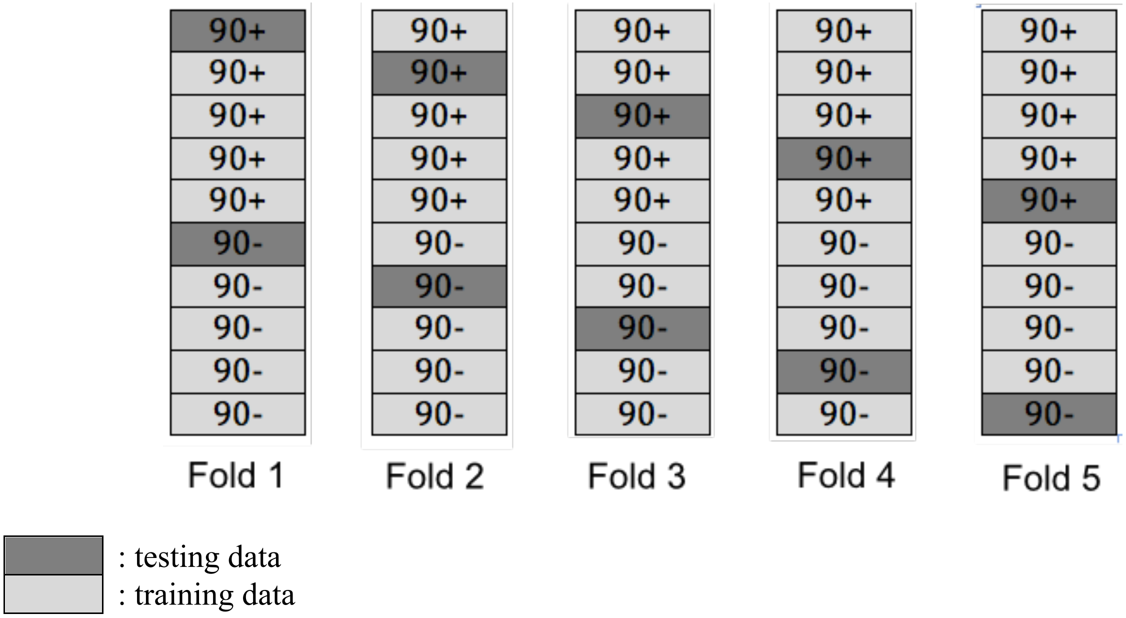


Figure 8. 5 fold cross validation process

Each partition takes a turn to become the testing data, since we use 5 fold cross validation, the process is repeated five times. Since each testing data set will have 200 instances, the accuracy formula will be: $accuracy = \frac{n}{200}$ with n is the number of correct predictions. For example, if the classification model predicts 165 instances correctly, the accuracy is $\frac{165}{200} = 82.5\%$. After that, we will have five different accuracy prediction results, and the average of these five are the final result. As a result, even though we do not have the real testing dataset, we are still able to describe how good the classification models are when applying them to the wine dataset.

Chapter 4: Decision Tree

4.1 Definition

Decision Tree induction is the learning of decision trees from class-labeled training tuples. The tree consists of nodes that form a rooted tree, meaning it is a directed tree with a node called "root" that has no incoming edges. All other nodes called internal nodes have exactly one incoming edge that denotes a test on an attribute, but splits or branches to represent an outcome into two edges according to the input variable. Each leaf node holds a class label or an attribute. The Decision Tree algorithm is a tree that is constructed in a top-down recursive divide and conquer manner. In the beginning, all attributes are listed at the root. To determine which attribute is to become the root, we used a statistical measure called information gain. The attribute with the highest information gain is the root of the tree.

Let p_i be the probability that an arbitrary tuple in D belongs to class Ci, estimated by |Ci,D| / |D|. Expected information (entropy) needed to classify a tuple in D is:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i) \qquad (1)$$

p_j = probability of the label being positive or negative

m = number of attributes

Information needed (after using A to split D into c partitions) to classify D is computed by using information gain formula:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j) \qquad (2)$$

v = attribute entropy

D_j = how many 1's or 0's are present in the attribute

D = total number of data set

I = how many 1's and 0's are present in the 90+ or 90- classification

Information gained by branching on attribute A is:

$$Gain(A) = Info(D) - Info_A(D) \quad (3)$$

Gain(A) = the attribute with the highest gain in the root.

4.2 Decision tree and wine example

After calculation, the attribute that has the highest gain (A) becomes the root of the tree. The process is repeated until "the subset at a node has all the same value of the target variable, or when splitting no longer adds value to the predictions." (Quinlan) In order to better explain how decision work on wine dataset, a simplified example is provided below:

	CHERRY	APPLE	PURE	BERRY	Grade
Wine 1	1	1	0	1	90+
Wine 2	0	0	1	1	90+
Wine 3	1	0	0	1	90+
Wine 4	0	1	1	1	90-
Wine 5	1	0	0	0	90-
Wine 6	0	1	0	1	90-

Table 5 Example dataset to apply Decision Tree

Dataset on table 5 has 6 wines and 4 attributes includes: CHERRY, APPLE, PURE, BERRY. Among 6 wines, the first 3 are graded 90+, and the last 3 are graded 90-. First, Info(D) in formula (1) is computed

$$Info(D) = I(3,3) = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1$$

Next, Info(D) of each attribute is calculated based on formula (2):

$$Info_{CHERRY}(D)=\frac{3}{6}I(1,2)+\frac{3}{6}I(2,1)=\frac{3}{6}(-\frac{1}{3}\log_2\frac{1}{3}-\frac{2}{3}\log_2\frac{2}{3})+\frac{3}{6}(-\frac{2}{3}\log_2\frac{2}{3}-\frac{1}{3}\log_2\frac{1}{3})=0.918$$

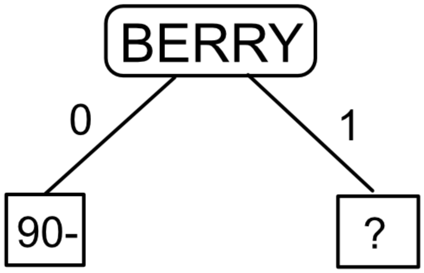
$Info_{CHERRY}(D)$ has two parts. First, $\frac{3}{6}I(1,2)$ means "CHERRY = 0" has 3 out of 6 samples, with one of them belong to "90+" label, and two of them belonging to "90-" label. $\frac{3}{6}I(2,1)$ means "CHERRY = 1" has 3 out of 6 samples, with two of them belongs to "90+" label, and one of them belongs to "90-" label. The process is repeated for APPLE, PURE, BERRY:

$$Info_{APPLE}(D)=\frac{3}{6}I(2,1)+\frac{3}{6}I(1,2)=0.918$$
$$Info_{PURE}(D)=\frac{2}{4}I(2,2)+\frac{1}{2}I(1,1)=1$$
$$Info_{BERRY}(D)=\frac{1}{6}I(0,1)+\frac{5}{6}I(3,2)=0.809$$

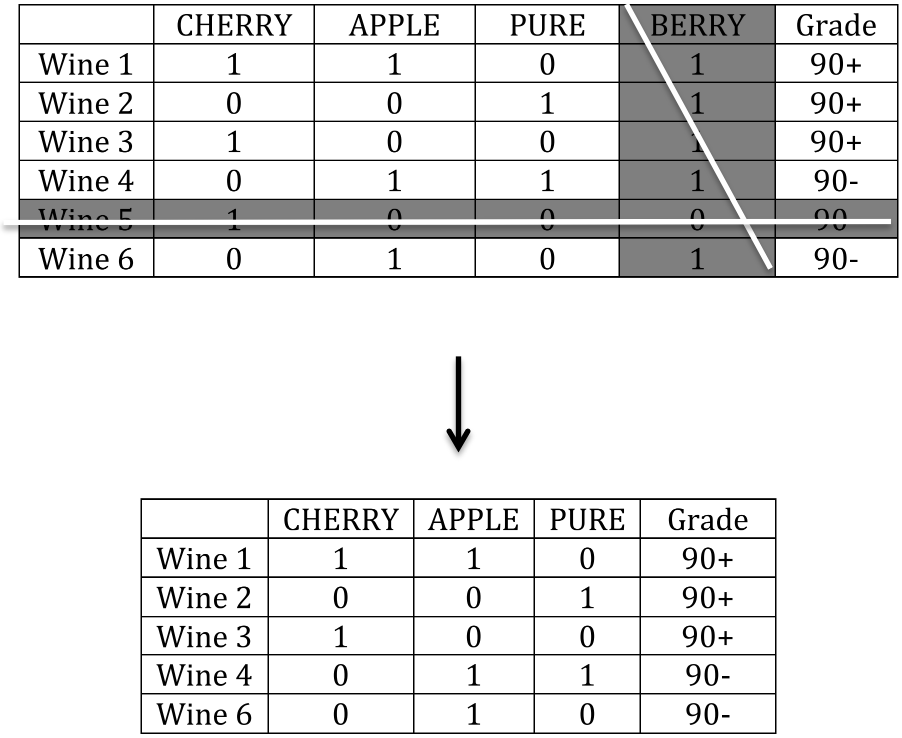
Finally, formula (3) is applied to find which attribute has the highest value.

$$Gain_{CHERRY}(A)=Info(D)-Info_{CHERRY}(D)=1-0.918=0.092$$
$$Gain_{APPLE}(A)=Info(D)-Info_{APPLE}(D)=1-0.918=0.092$$
$$Gain_{PURE}(A)=Info(D)-Info_{PURE}(D)=1-1=0$$
$$Gain_{BERRY}(A)=Info(D)-Info_{BERRY}(D)=1-0.809=0.191$$

Since BERRY gets the highest gain, it becomes the root of the tree. Please notice that there is only one wine that has BERRY = 0, and it belongs to label "90-" (Wine 5), so the branch BERRY = 0 is pure, and no need to expand it anymore.



After we found the root, we need to exclude all the checked data. Figure 9 shows how the simplified dataset is reduced.



	CHERRY	APPLE	PURE	BERRY	Grade
Wine 1	1	1	0	1	90+
Wine 2	0	0	1	1	90+
Wine 3	1	0	0	1	90+
Wine 4	0	1	1	1	90-
~~Wine 5~~	~~1~~	~~0~~	~~0~~	~~0~~	~~90~~
Wine 6	0	1	0	1	90-

	CHERRY	APPLE	PURE	Grade
Wine 1	1	1	0	90+
Wine 2	0	0	1	90+
Wine 3	1	0	0	90+
Wine 4	0	1	1	90-
Wine 6	0	1	0	90-

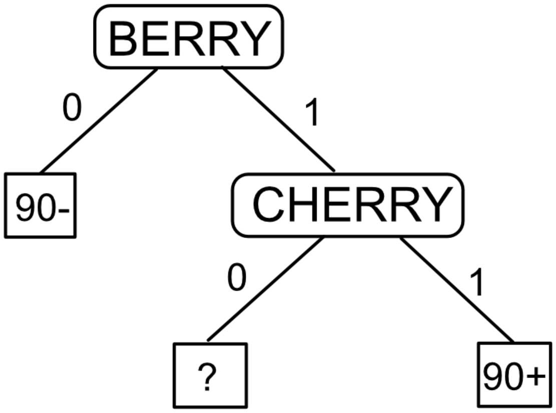
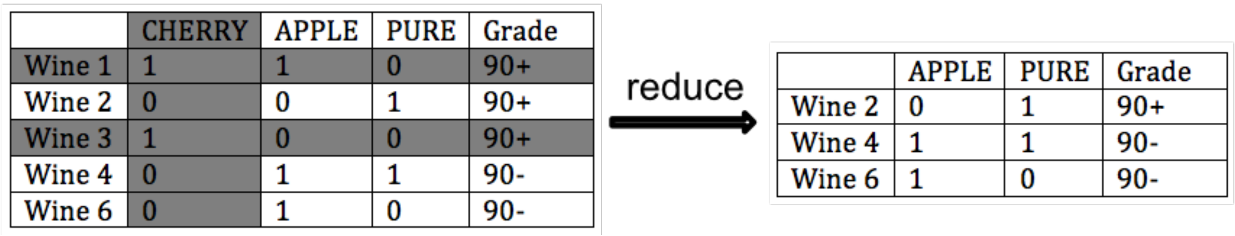
Figure 9 The simplified dataset after the first step of Decision Tree

BERRY attribute becomes the root, so we need to remove it. Wine 5 is the one that has "BERRY = 0", and it need to be removed as well. Next, the process is repeated until the tree is no longer expanded, and table 6 shows how it is completed.

<table>
<tr><th>Step</th><th>Gain Calculations</th><th>The decision tree after each step</th></tr>
<tr><td>2</td><td>$Info(D)=I(3,2)=-\frac{3}{5}\log_2\frac{3}{5}-\frac{2}{5}\log_2\frac{2}{5}=0.97$

$Info_{CHERRY}(D)=\frac{3}{5}I(1,2)+\frac{2}{5}I(2,0)=0.459$
$Info_{PURE}(D)=\frac{3}{5}I(1,2)+\frac{2}{5}I(2,0)=0.459$
$Info_{APPLE}(D)=\frac{3}{5}I(2,1)+\frac{2}{5}I(1,1)=0.792$

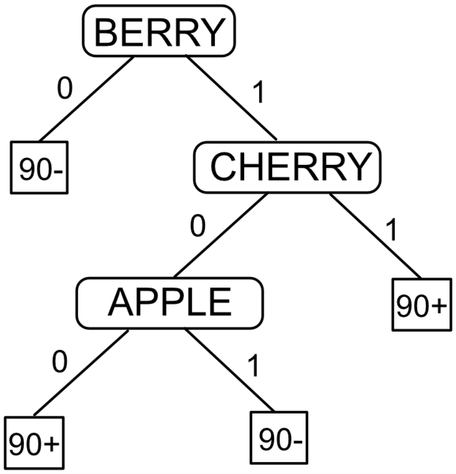
$Gain_{CHERRY}(A)=Info(D)-Info_{CHERRY}(D)=0.97-0.459=0.511$
$Gain_{PURE}(A)=Info(D)-Info_{PURE}(D)=0.97-0.459=0.511$
$Gain_{APPLE}(A)=Info(D)-Info_{APPLE}(D)=0.97-0.792=0.178$

=> **CHERRY** and **APPLE** have the highest gain, choose one of them to become the next note</td><td> </td></tr>
<tr><td>2.5</td><td colspan="2">
<table>
<tr><th></th><th>CHERRY</th><th>APPLE</th><th>PURE</th><th>Grade</th></tr>
<tr><td>Wine 1</td><td>1</td><td>1</td><td>0</td><td>90+</td></tr>
<tr><td>Wine 2</td><td>0</td><td>0</td><td>1</td><td>90+</td></tr>
<tr><td>Wine 3</td><td>1</td><td>0</td><td>0</td><td>90+</td></tr>
<tr><td>Wine 4</td><td>0</td><td>1</td><td>1</td><td>90-</td></tr>
<tr><td>Wine 6</td><td>0</td><td>1</td><td>0</td><td>90-</td></tr>
</table>

<table>
<tr><th></th><th>APPLE</th><th>PURE</th><th>Grade</th></tr>
<tr><td>Wine 2</td><td>0</td><td>1</td><td>90+</td></tr>
<tr><td>Wine 4</td><td>1</td><td>1</td><td>90-</td></tr>
<tr><td>Wine 6</td><td>1</td><td>0</td><td>90-</td></tr>
</table></td></tr>
<tr><td>3</td><td>$Info(D)=I(1,2)=-\frac{1}{3}\log_2\frac{1}{3}-\frac{2}{3}\log_2\frac{2}{3}=0.886$

$Info_{APPLE}(D)=\frac{1}{3}I(1,0)+\frac{2}{3}I(2,0)=0$
$Info_{PURE}(D)=\frac{1}{3}I(0,1)+\frac{2}{3}I(1,1)=0.333$

$Gain_{APPLE}(A)=Info(D)-Info_{APPLE}(D)=0.886-0=0.886$
$Gain_{PURE}(A)=Info(D)-Info_{PURE}(D)=0.886-0.333=0.553$

=>**APPLE** has the highest gain, choose it to become the next note</td><td>

BERRY
0
1
90-
CHERRY
0
1
APPLE
90+
0
1
90+
90-

</td></tr>
</table>

Table 6 Process of making decision tree on simplified wine dataset

After eliminating the BERRY attribute, the other three including: APPLE, PURE, and CHERRY once again are calculated to find the new gains. In step 2, CHERRY and APPLE both get the highest gain. Based on our algorithm implementation, in case there is more than one highest gain, it will always pick the first attribute; therefore, CHERRY becomes the next node in the tree. Because all the rows that have "CHEERY = 1" belong to class "90+", it becomes pure and cannot expand anymore. Next, the dataset needs to be reduced, and the step 2.5 shows that the CHERRY attribute is cut down along with Wine 1 and Wine 3, which have "CHERRY = 1". The whole process one again is repeated in step 3, and APPLE gets the highest gain, so it becomes the next node. All the rows that have "APPLE = 0" belongs to class "90+", and all the rows that have "APPLE = 1" belongs to class "90-". At this point, both branches of the tree become pure; therefore, the tree can no longer expand and the algorithm stops.

After we have the model, we can apply it to several test wines and predict their labels. To predict them, based on the value of attributes of the test wine, the model will follow the decision tree paths from top to bottom to make a decision. Considering two test wines in the table below:

	CHERRY	APPLE	PURE	BERRY	Grade
Test wine 1	0	1	1	1	?
Test wine 2	1	1	0	0	?

Table 7 A simplified testing dataset (Decision Tree)

For test wine 1, figure 10 shows how the model follows the tree and makes decision. Because the BERRY attribute of test wine 1 has value of 1, the model follows the BERRY path 1 and reaches the CHERRY. Next, it checks the CHERRY value of test wine 1, since the value is 0, the model follow the CHERRY path 0 and reach the APPLE.

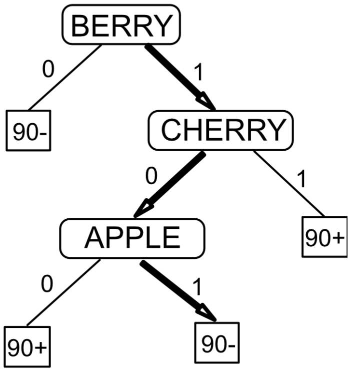


Figure 10 Decision tree path of test wine 1

Finally, the APPLE attribute of test wine 1 has value of 1, the model predicts that this wine will have label "90-". For test wine 2, since its "BERRY = 0", the model follows the tree and decides that test wine 1 will have label "90-".

4.3 Results

We apply 5 – fold cross validation to 1000 wine dataset, and table 8 is the results

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
90+	95%	87%	83%	82%	83%
90-	26%	17%	13%	5%	15%
Accuracy	60.5%	52%	48%	43.5%	49%
Average	50.6%				

Table 8 Results from 5-CV Test for Decision Tree

The average accuracy just barely passes 50%. Because we want the model to predict between two labels: "90+", and "90-", the accuracy is just better than guessing "heads" or "tails" when we flip a coin. Notice that there is a significantly lower percentage of 90- wines that were predicted versus the 90+ wines. This could be due to the fact that 90- wines do not have as many of the attributes listed as the 90+ wines, which would cause problems with classifying by an attribute. As mentioned above, depending on the datasets, some classification algorithms will generate high accuracy predictions, some will not; and decision tree is not suitable for wine dataset. As a result, it gives us motivation to try and test more classification models, and k- nearest neighbor is our next choice.

Chapter 5: k- nearest neighbor

5.1 Definition

k-Nearest Neighbors (k-NN) is "a non-parametric method used for classification and regression. In both cases, the input consists of the k closet training examples in the feature space" (Altman). Different from Decision Tree where the algorithm builds models and predicts the accuracy, k-NN classification is a type of instance-based learning (lazy learning). The output of the algorithm is a class membership, and "an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k > 0)" (Sutton). In other words, k-NN does not build any model. k values are chosen, and the algorithm calculates distances between instances and then predicts labels directly. Even though it is lazy learning, one of the advantages of k-NN is that we can weight the contribution of the attributes. Our Computational Wine Wheel has one column about weight, where attributes are weighted differently (1,2, and 3) based on their importance. As a result, k-NN is a perfect method to test how attribute weights affect the final accuracy.

5.2 k-NN and wine example

The purpose of the KNN algorithm is to "use a database in which the data points are separated into several separate classes to predict the classification of a new sample point." (Sutton) For our wine dataset, the prediction of a test wine is based on the majority label vote of its k "nearest" wines. In other words, the algorithm chooses k wines that are the most similar to the test wine, and counts how many of them are "90+" and "90-", then predicts the test wine label based on the majority vote. Another way to express how k-NN works on wine dataset can be seen in this example with simplified data:.

	LONG FINISH	APPLE	PURE	Grade
Wine 1	0	1	1	90+
Wine 2	1	0	0	90+
Wine 3	1	1	1	90-
Wine 4	0	1	0	90-
Wine 5	0	0	1	90+

Table 9. A simplified training wine dataset (k-NN)

Dataset in table 9 has 5 wines and 3 attributes. Among 5 wines, there are three wines have label "90+", and the other two are "90-". The k-NN algorithm has three main steps.

Step 1: Choosing k value

Before executing the k-NN algorithm, we need to decide the value of k. In a wine dataset, k present number of nearest wines compared to the test wine. Because the wine dataset has two classes, the majority vote decides if a wine label will be 90+ or 90-. As a result, k is better to be an odd number and positive to prevent an equal voting. k also cannot be a large number, since it might decrease the performance of k-NN algorithm.

Step 2: Measure how similarity between two wines

The "distances" between a testing wine and all training wines will be calculated, then choose k nearest wines with the testing wine. Wine dataset is special because it is binary, which means it only contained 0 and 1. For that reason, Jaccard's distance formula is used. Even though the equation is called "Jaccard's distance," it is used to measure the similarity between two wines in wine dataset. The smaller the value is, the more similar the two wines are.

Jaccard's distance formula: $J = \dfrac{Q+R}{P+Q+R}$ (4)

Q: positive in wine 1 but not in wine 2

R: positive in wine 2 but not it wine 1

P: positive in both wine 1 and wine 2.

A test wine sample on table 10 is used by k-NN classification to predict its grade.

	LONG FINISH	APPLE	PURE	Grade
Test wine	1	0	1	?

Table 10. A test wine sample for k-NN

Apply Jaccard's distance formula on (4), we have

The distance between Test wine and Wine 1 is:

$$J_{T1} = \frac{1+1}{1+1+1} = \frac{2}{3}$$

The distance between Test wine and Wine 2 is:

$$J_{T2} = \frac{1}{1+1} = \frac{1}{2}$$

The distance between Test wine and Wine 3 is:

$$J_{T3} = \frac{1}{2+1} = \frac{1}{3}$$

The distance between Test wine and Wine 4 is:

$$J_{T4} = \frac{1+2}{1+2} = \frac{3}{3}$$

The distance between Test wine and Wine 5 is:

$$J_{T5} = \frac{1}{1+1} = \frac{1}{2}$$

Step 3: Predict the label of the test wine base on the majority vote

After all the distances between the test wine and all wines in the training dataset are calculated, the k smallest distance results will be chosen to vote and predict which class the test wine belong to. For example:

k = 1: J_{T3} is the smallest distance. Because the label of wine 3 is “90-”, k-NN predicts Test wine is 90-

k = 3: J_{T3}, J_{T2}, J_{T5} are the three smallest distance. Base on their label, there are two “90+” and one “90-” wine. As a result, the test wine is predicted “90+”.

5.3 Weight attributes

In our Computational Wine Wheel, one of the descriptions of attribute is weight. In our wine dataset, there are 304 different attributes, but they are not equal in terms of their importance. To further increase the quality of the dataset, we assign weights for each attribute based on its appearance frequency in the review and expert opinions. Weight has three values: 1, 2, and 3.

“1” is the least important attribute: non-flavor descriptions (PURE, BEAUTY, WONDERFUL, etc.)

“2” is the semi-important attribute: non-flavor wine characteristics (TANNINS, ACIDITY, BODY, etc.)

“3” is the most important attribute: food wine characteristics (specific fruit, woods, flavors, etc.)

The simplified dataset below shows how we describe weights for each attribute:

	LONG FINISH	APPLE	PURE	Grade
Weight	2	3	1	
Wine 1	0	1	1	90+
Wine 2	1	0	0	90+
Wine 3	1	1	1	90-
Wine 4	0	1	0	90-
Wine 5	0	0	1	90+

Table 11. A simplified wine dataset with weight (k-NN)

The data in table 11 is the same as table 9, but now we have one more row called "Weight". LONG FINISH will be weighted 2, APPLE is weighted 3, and PURE is weight 1. k-NN algorithm that applies to Weight wine dataset is the same as the one apply to Wine dataset without weight, except for step 2; when we measure the similarity between two wines. Because each attribute has its own weight now, the Jaccard's distance needs to be adjusted to include it:

$$J = \frac{weight_q \times Q + weight_r \times R}{weight_q \times Q + weight_r \times R + weight_p \times P} \qquad (5)$$

$weight_q$: weight of Q
$weight_r$: weight of R
$weight_p$: weight of P

Using the test wine in table 5.2.2, and applying the formula (5) to calculate the distance between the test wine and all training wines on table 5.3, the results are:

The distance between Test Wine and Wine 1 is:

$$J_{T1} = \frac{1 \times 2 + 1 \times 3}{1 \times 2 + 1 \times 3 + 1} = \frac{5}{6}$$

The distance between Test Wine and Wine 2 is:

$$J_{T2} = \frac{1}{1+1\times 2} = \frac{1}{3}$$

The distance between Test Wine and Wine 3 is:

$$J_{T3} = \frac{1\times 3}{1\times 2+1+1\times 3} = \frac{3}{6} = \frac{1}{2}$$

The distance between Test Wine and Wine 4 is:

$$J_{T4} = \frac{1\times 3+1\times 2}{1\times 3+1\times 2} = \frac{5}{5}$$

The distance between Test Wine and Wine 5 is:

$$J_{T5} = \frac{1\times 2}{1\times 2+1} = \frac{2}{3}$$

If k = 1, the algorithm will predict test wine belongs to 90+ since its "nearest" wine is wine 2. If k = 3, the algorithm will predict test wine also belongs to 90+ since its three "nearest" wines are wine 2, 3 , and 5 (two of them have 90+ label).

5.4 Results

We modified our k parameter from 1 to 21. Because wine dataset has 2 class labels (90+ and 90-), k must be an odd number to prevent equal voting.

5.4.1 The result of the wine dataset without weight.

Table 12 shows the results of KNN for each fold and its average when k = 1 to 21. Fold 1 to fold 5 represent the accuracy for each fold, then we take the average among them, and get the accuracy.

	KNN without weight					
k	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Accuracy
1	70	77	71	68.5	67.5	70.8
3	79.5	83	75.5	68	65	74.2
5	81.5	83.5	75	67.5	66.5	74.8
7	78	82.5	74	67.5	66.5	73.7
9	81.5	82.5	71.5	67	67	73.9
11	83.5	86	72	64	66	74.3
13	85	85.5	71	66	64	74.3
15	82.5	86.5	73.5	66	64	74.5
17	83.5	88	73.5	64	63.5	74.5
19	84	90	72	63.5	65	74.9
21	81.5	89.5	71	64	66	74.4

Table 12. The results of k-NN without weight (k-NN)

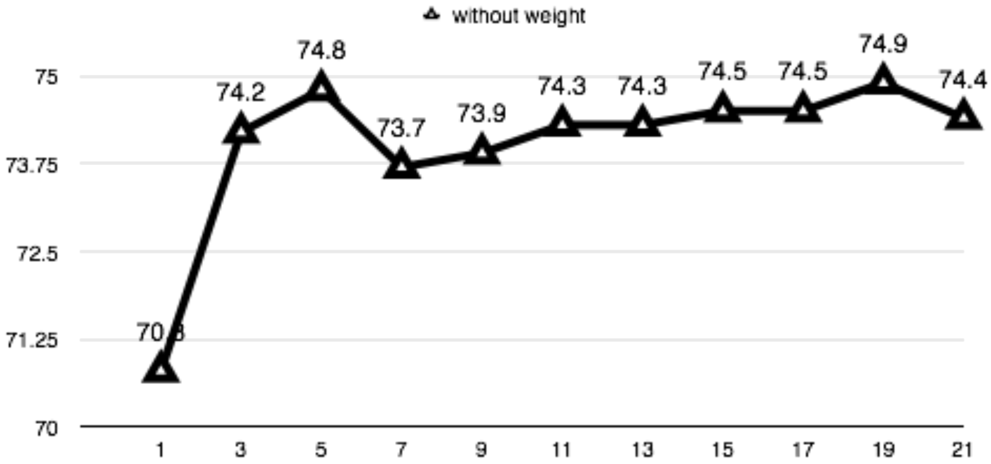
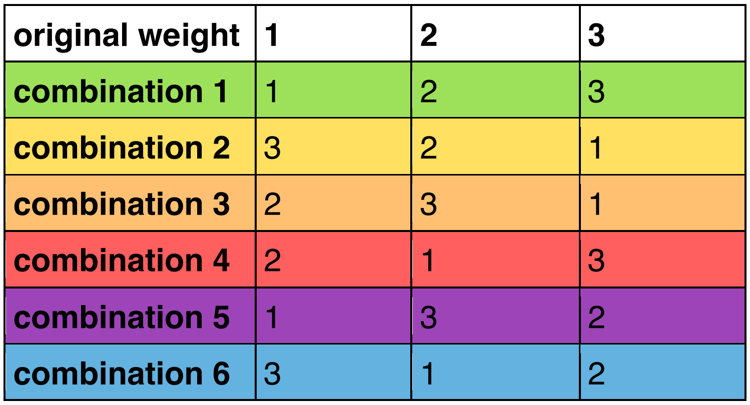


Figure 11. The averages accuracy of k from 1 to 21 (k-NN)

The highest accuracy is 74.9% (k = 19), which is much better than Decision Tree result (50.6%). Overall, the accuracy results of KNN are similar to each other except when k = 1. Because KNN predicted the result base on majority vote, with k = 1, the algorithm will be completely based on the label of only one wine. As a result, it leads to bias when the algorithm does not consider more instances to vote. As a result, the result of k = 1 is an outlier.

5.4.2 The result of the wine dataset with weight.

As mentioned above, weight has 3 values. 1 to 3 corresponds from the least to the most important attributes. We assume that many attributes are the least important because they express feeling; and feeling is not something we can measure exactly. For that reason, to prevent all the assumption and bias, we switch the values of weights between attributes, and create all possible combinations between them. Table 13 shows how it is done.



original weight	1	2	3
combination 1	1	2	3
combination 2	3	2	1
combination 3	2	3	1
combination 4	2	1	3
combination 5	1	3	2
combination 6	3	1	2

Table 13. All combinations of weight

There are 6 different combinations between the weights. For example, combination 2 will assign weight "3" for the attribute that has original weight is "1". Weight "2" is the same, and weight "1" is assigned to he attribute that has original weight is "3"

Next, k-NN algorithm is applied to all 6 combinations. Please notice that Jaccard's distance is calculated based on the formula (5) in section 5.3, and table 14 shows the accuracy of all combinations after measuring with 5-fold cross validation.

	Accuracy					
k	Combination 1	Combination 2	Combination 3	Combination 4	Combination 5	Combination 6
1	68	70.7	67.1	74.1	68.5	71.2
3	67.9	72	69.1	73.2	71.7	72.7
5	69.9	73.1	71.3	75.4	72.6	73.6
7	70.3	73.8	70.1	76.3	73.7	74.1
9	70.8	73.1	70.2	75.4	74.3	73.8
11	70.1	72.7	69.8	75.9	73	74.4
13	69.6	71.6	69.9	75.9	73	74.9
15	69.8	72.8	69.4	75.8	73.3	75.2
17	69.9	72.5	69.8	76.5	73.3	75.2
19	70.4	72.2	70.3	75.3	73.8	75.3
21	70.6	72	70.1	75.6	73.9	74.2

Table 14. The accuracy of 6 combinations with k from 1 to 21 (k-NN)

Among 6 combinations, combination 3 gives the lowest accuracy (67.1%) while combination 4 gives the highest result (76.5%). Figure 12 shows the comparison between them and the result of the dataset without weight.

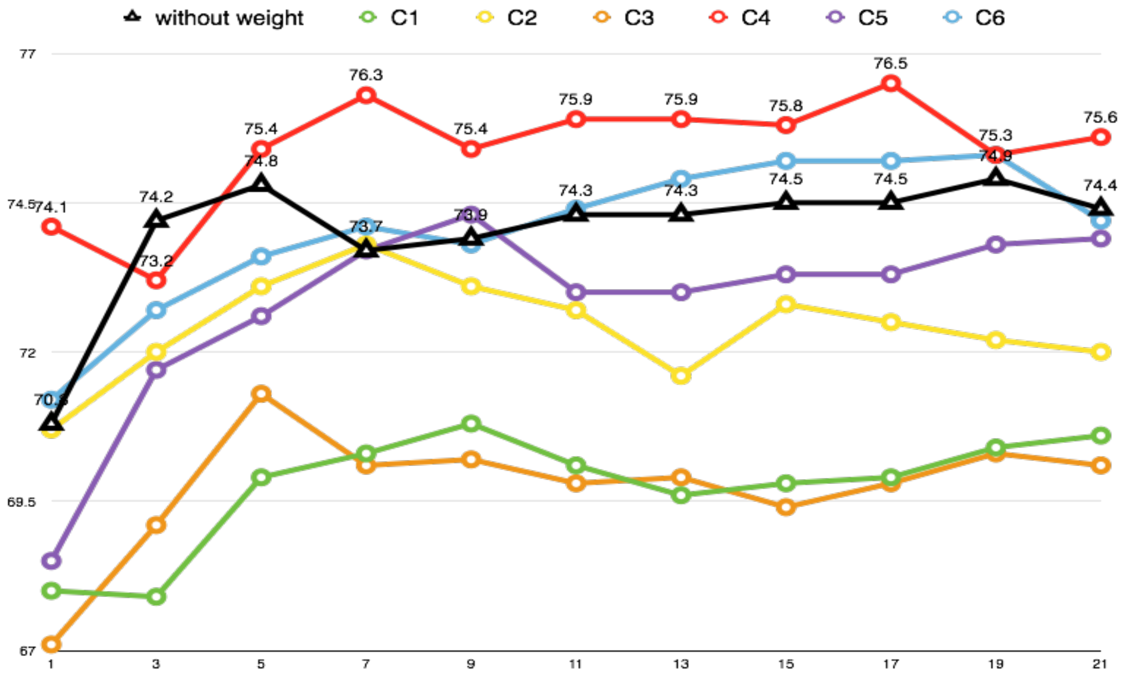


Figure 12. The accuracy comparison chart of all weight combinations and without weight

Compared to the highest accuracy without weight (74.9%), the original weight (combination 1) generates lower accuracy with the highest result being 70.6%. Combinations 4 and 6 are the two that perform better than the dataset without weight:

original weight	**1**	**2**	**3**
combination 4	2	1	3
combination 6	3	1	2

Based on the result of the experience, combination 4, which is generated the best accuracy among all, suggests that attributes with weight 3 are kept the same; but attributes that are weighted 1 are actually more important that those attributes that are weighted 2, so we need to switch them. Combination 6 follows the same routine, but it says the attributes weighted 1 are the most important. In both cases, even though there is a conflict between the original weight 1 and 3, all combinations agree that the attributes that are weighted 2 should be the least important attribute.

Chapter 6: Naïve Bayes

6.1 Definition

“A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model”. In other words, a naive Bayes classifier assumes that the presence (or absence) of an instance of a class is unrelated to the presence (or absence) of any other instance. (Russell and Norvig) For example, a wine may be considered to be a “90+” wine if it has BLUE BERRY, APPLE, and LONG FINISH. Even if these attributes depend on each other or on other attributes, when a naïve Bayes classifier generates the probability of the wine, it considers all of these attributes independently. As a result, depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. (Russell and Norvig). Bayes formula is used to calculate the probability of a testing instance:

Bayes theorem $$P(H|X)=\frac{P(X|H)P(H)}{P(X)} \qquad (6)$$

P(H|X): the probability that the hypothesis holds given the observed data sample X.

P(H): (prior probability), the initial probability.

P(X): probability that sample data is observed

P(X|H) (posteriori probability), the probability of observe the sample X, given that the hypothesis holds. (B. Chen, CSCI 4370/5370 Data Mining)

In other to explain how Naïve Bayes works on wine dataset, an example with simplified data is provided:

6.2 Naïve Bayes and wine example

	APPLE	ALMOND	LIME	STYLE	Label
Wine 1	1	0	0	0	90+
Wine 2	0	0	1	0	90+
Wine 3	1	0	0	0	90+
Wine 4	0	1	1	0	90-
Wine 5	1	0	0	1	90-
Wine 6	1	1	1	0	90-

Table 15 A simplified wine training dataset (Naïve Bayes)

A simplified dataset in table 15 has six wines with three of them belonging to class “90+”, and the other three belonging to class “90-”. There are four different attributes, and since we use Naïve Bayes, we assume that all these attributes are independent from each other. First, the initial probability of each label is calculated:

P (X): P (90+) = 3/6 = 0.5

P(90-) = 3/6 = 0.5

Because among six wines, each label has three, there will be 50% that a test wine is “90+”, and 50% it is “90-”. Next, probability for each attribute of each class is computed by following formula (6).

APPLE

P (APPLE = 0 | 90+) = 1/3

P (APPLE = 0 | 90-) = 1/3

P (APPLE = 1 | 90+) = 2/3

P (APPLE = 1| 90-) = 2/3

P (APPLE = 0 | 90+) = 1/3 means that the probability of a 90+ wine has attribute APPLE = 0 is 0.333%.

P (APPLE = 0 | 90-) = 1/3 means that the probability of a 90- wine has attribute APPLE = 0 is 0.333%.

P(APPLE = 1 | 90+) = 2/3 means that the probability of a 90+ wine has attribute APPLE = 1 is 0.666%.

P(APPLE = 1 | 90-) =2/3 means that the probability of a 90- wine has attribute APPLE = 1 is 0.666%.

The same explanations for **ALMOND**, **LIME**, and **STYLE**

ALMOND

P (ALMOND = 0 | 90+) = 3/3

P (ALMOND = 0 | 90-) = 1/3

P (ALMOND = 1 | 90+) = 0/3

P (ALMOND = 1 | 90-) = 2/3

LIME

P (LIME = 0 | 90+) = 2/3

P (LIME = 0 | 90-) = 1/3

P(LIME = 1 | 90+) = 1/3

P(LIME = 1 | 90-) = 2/3

STYLE

P(STYLE = 0 | 90+) = 3/3

P(STYPE = 0 | 90-) = 2/3

P(STYLE = 1 | 90+) = 0/3

P(STYLE = 1 | 90-) = 1/3

After Naïve Bayes builds the classifier, a test wine is provided to predict its label.

	APPLE	ALMOND	LIME	STYLE	Label
Wine T1	1	0	1	0	?

Table 16 The simplified test wine number 1 (Naïve Bayes)

The test wine 1 (Wine T1) has APPLE = 1, ALMOND = 0, LIME = 1, and STYLE = 0, therefore posteriori probabilities of test wine 1 with the hypothesis it belongs to class "90+" and "90-" is:

P (Wine T1 | 90+)

= P(**APPLE = 1** | 90+) * P(**ALMOND = 0** | 90+) * P(**LIME = 1** | 90+) * P(**STYLE 0** | 90+)

= 2/3 * 3/3 * 1/3 * 3/3 = 18/81 = 0.222

P (Wine T1 | 90-)

= P(**APPLE = 1** | 90-) * P(**ALMOND = 0** | 90-) * P(**LIME = 1** | 90-) * P(**STYLE 0** | 90-)

= 2/3 * 1/3 * 2/3 * 2/3 = 8/81 =0.098

Next, the probability that the hypothesis holds given the observed test wine 1 is

P (90+ | Wine T1) * P (90+) = 18/81 * 1/2 = 18/162 = 0.111

P (90- | Wine T1) * P (90-) = 8/81 * 1/2 = 8 /162 = 0.049

Since P (90+ | Wine T1) is greater than P (90- | Wine T1), Naive Bayes classification predicts that Test wine 1 belongs to label "90+".

6.3 Zero frequency problem

Zero frequency problem happens when none of the training instances have the same value as testing instances; therefore, the result will equal zero, and ignore all the effects of other instances. For example: apply Naive Bayes on the training dataset on 1.3, with this testing wine.

	APPLE	ALMOND	LIME	STYLE	Label
Wine T2	0	1	1	1	?

Table 17 The simplified test wine number 2

Wine T2 = (APPLE = 0, ALMOND = 1, LIME = 1, STYLE = 1)

Posteriori probabilities of test wine 2 are:

P (Wine T2 | 90+)

= P(**APPLE = 0** | 90+) * P(**ALMOND = 1** | 90+) * P(**LIME = 1** | 90+) * P(**STYLE 1** | 90+)

= 1/3 * **0/3** * 1/3 * **0/3** = 0

P (Wine T2 | 90-)

= P(**APPLE = 0** | 90-) * P(**ALMOND = 1** | 90-) * P(**LIME = 1** | 90-) * P(**STYLE 1** | 90-)

= 1/3 * 1/3 * 2/3 * 1/3 = 2/81

Next, P(Ci | Wine 2) * P (Ci) are

P (Wine T2 | 90+) * P (90+) = 0 * 1/2 = 0

P (Wine T2 | 90-) * P (90-) = 2/81 * 1/2 = 2 /162

Since P (90+ | Wine T2) < P (90- | Wine T2), the classifier predicts wine T2 belongs to label "90-". As the result shows, because ALMOND and STYLE make P (Wine T2 | 90+) equal to zero, it ignores the values of APPLE and LIME; therefore, the final results might be effected, and lead to the wrong conclusion. There are several solutions to minimize the effect of zero frequency problems, and we can apply Add Penalty and Laplace methods to see how the wine dataset result is modified.

6.3.1 Add penalty

With the Add Penalty method, when computing the probabilities, it substitutes each 0 with:

$\left(\frac{1}{k}\right)^n$. As a result, Bayes theorem formula (1) will be modified to:

$$P(H|X)=\left(\frac{1}{k}\right)^n \times \frac{P(X|H)P(H)}{P(X)} \qquad (7)$$

n: numbers of 0 value

k: a parameter ($k \neq 0$).

With this, we can manipulate the result by changing k. The larger k, the smaller the P (H|X) is, so if k increase to ∞, P(H|X) will become 0. In order to explain how the Add Penalty method effects the prediction, the same test wine in table 17 is used

	APPLE	ALMOND	LIME	STYLE	Label
Wine T2	0	1	1	1	?

Wine T2 = (APPLE = 0, ALMOND = 1, LIME = 1, STYLE = 1)

P (Wine T2 | 90+) = 1/3 * **0/3** * 1/3 * **0/3** = 0

P (Wine T2 | 90-) = 1/3 * 1/3 * 2/3 * 1/3 = 2/81

Because P (Wine T2 | 90+) has two zero value, formula (7) is used and those two zero will be substituted by $\frac{1}{k}$.

P (Wine T2 | 90+) = 1/3 * $\frac{1}{k}$ * 1/3 * $\frac{1}{k} = \frac{1}{3\times k\times 3\times k}$

Since k is a parameter, it can be any value (except 0 due to divide by zero problem). If k = 2, then P (Wine T2 | 90+) = $\frac{1}{3\times 2\times 3\times 2}$ = 1/36, and P (90+ | Wine T2) = 1/36*1/2= 1/72.

As a result, P (90+ | Wine T2) is no longer = 0, and compare to P(90- | Wine T2), it larger (1/ 72 > 2/162). Therefore, wine T2 is predict “90+”.

6.3.2 Laplace

Laplace is a smoothing data technique, the purpose is manipulate the value of the data at the beginning, so Navie Bayes classification will never have zero frequency problem. (Except when parameter k = 0). Bayes theorem formula (1) will be modified to:

$$P(H|X)=\frac{P(X|H)P(H)+k}{P(X)+b-1} \qquad (8)$$

b: number of instances in dataset

k: the parameter

With Laplace method, initial probability outcomes of each class P(H) also needs to be modified to:

$$P(H)=\frac{a+k}{b-1+k\times c} \qquad (9)$$

c: number of classes

a: number of instances in one class

Applying Laplace equations to wine dataset, we have:

$$P(90+)=\frac{num\_pos+k}{num\_wines-1+k\times num\_class}$$

$$P(90-)=\frac{num\_neg+k}{num\_wines-1+k\times num\_class}$$

$$P(X|90+)=\frac{count\_pos+k}{num\_pos+num\_wines-1}$$

$$P(X|90-)=\frac{count\_neg+k}{num\_neg+num\_wines-1}$$

num_pos: numbers of positive wines in the dataset

count_pos: numbers of 0 or 1 of each attribute in 90+ wines

num_neg: numbers of negative wines in the dataset

count_neg: numbers of 0 or 1 of each attribute in 90- wines

num_wines: numbers of wines in the dataset (num_wines = num_pos + num_neg)

num_class: numbers of classes in the dataset

k: the parameter.

In order to compare the differences between Laplace and other methods, the same training dataset in table 15 and testing wine 2 in table 17 are used:

Training dataset

	APPLE	ALMOND	LIME	STYLE	Label
Wine 1	1	0	0	0	90+
Wine 2	0	0	1	0	90+
Wine 3	1	0	0	0	90+
Wine 4	0	1	1	0	90-
Wine 5	1	0	0	1	90-
Wine 6	1	1	1	0	90-

Testing wine

	APPLE	ALMOND	LIME	STYLE	Label
Wine T2	0	1	1	1	?

With parameter k = 2, we have:

Formula (9) is used to compute P(H):

P (90+) = (3 + 2)/(6-1+2*2) = 5/9

P (90-) = (3 + 2)/(6-1+2*2) = 5/9

Compute P (X|H) for each attribute by using formula (8)

APPLE

P (**APPLE** = 0 | 90+) = (1 + 2)/(3 + 6 -1) = 3/8

P (**APPLE** = 0 | 90-) = (1 + 2)/(3 + 6 -1) = 3/8

P (**APPLE** = 1 | 90+) = (2 + 2)/(3 + 6 -1) = 4/8

P (**APPLE** = 1| 90-) = (2 + 2)/(3 + 6 -1) = 4/8

ALMOND

P (**ALMOND** = 0 | 90+) = (3 + 2)/ (3 + 6 -1) = 5/8

P (**ALMOND** = 0 | 90-) = (1 + 2)/(3 + 6 -1) = 3/8

P (**ALMOND** = 1 | 90+) = (0 +2)/ (3 + 6 -1) = 2/8

P (**ALMOND** = 1 | 90-) = (2 + 2)/(3 + 6 -1) = 4/8

LIME

P (**LIME** = 0 | 90+) = (2 + 2)/(3 + 6 -1) = 4/8

P (**LIME** = 0 | 90-) = (1 + 2)/(3 + 6 -1) = 3/8

P(**LIME** = 1 | 90+) = (1 + 2)/(3 + 6 -1) = 3/8

P(**LIME** = 1 | 90-) = (2 + 2)/(3 + 6 -1) = 4/8

STYLE

P(**STYLE** = 0 | 90+) = (3 + 2)/ (3 + 6 -1) = 5/8

P(**STYPE** = 0 | 90-) = (2 + 2)/(3 + 6 -1) = 4/8

P(**STYLE** = 1 | 90+) = (0 +2)/ (3 + 6 -1) = 2/8

P(**STYLE** = 1 | 90-) = (1 + 2)/(3 + 6 -1) = 3/8

As the result show, ALMOND and STYLE no long generate zero results.

Using test wine 2 to predict its label, we have

Wine T2 = (**APPLE** = 0, **ALMOND** = 1, **LIME** = 1, **STYLE** = 1)

P (Wine T2 | 90+) = 3/8 * 2/8 * 3/8 * 2/8 = 36/4096

P (Wine T2 | 90-) = 3/8 * 3/8 * 5/8 * 3/8 = 135/4096

P (Wine T2 | 90+) * P (positive) = 36/4096 * 5/9 = 180/36846

P (Wine T2 | 90-) * P (negative) = 135/4096* 5/9 = 675/26846

Since 180/36846 < 675/26846 => Wine T2 belongs to “90-”

6.4 Results

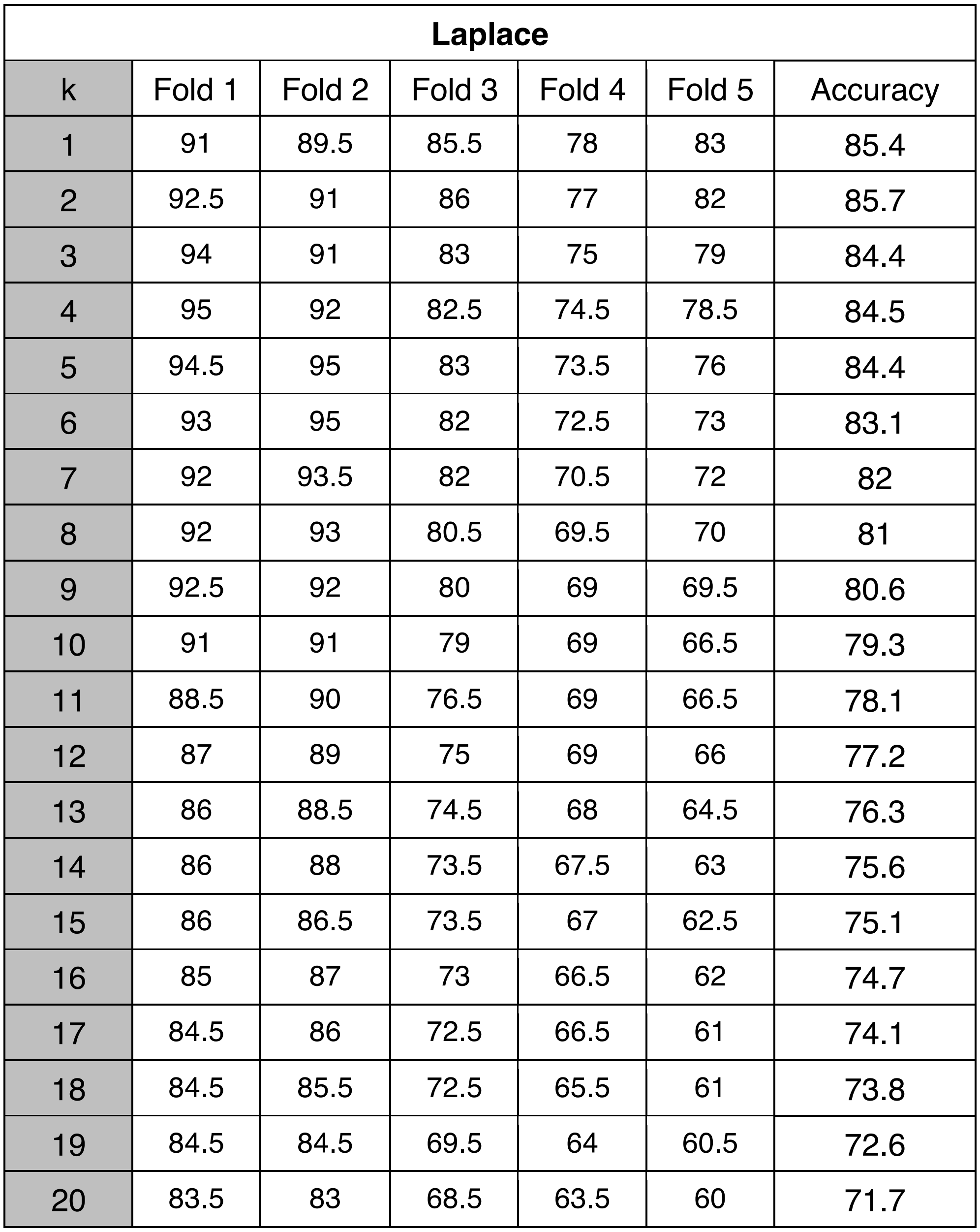
When applying Add penalty and Laplace methods, we manipulate the value of k from 1 to 20. After we apply 5 fold cross validations, the results of all three methods are shown below:

Include 0					
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Accuracy
76	82	80	78	82	79.6

Table 18 The results of include 0 values methods (Naïve Bayes)

Add penalty						
k	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Accuracy
1	64.5	67.5	69.5	65	63	65.9
2	65	68.5	71.5	66	63	66.8
3	65	70.5	71.5	66	63	67.2
4	65.5	71	71.5	66	63	67.4
5	66.5	71	71.5	66	63	67.6
6	67	71	71.5	66	63	67.7
7	67.5	72	71.5	66	63.5	68.1
8	68	72	71.5	66	63.5	68.2
9	69	72	71.5	66	63.5	68.4
10	69	72	72	66	63.5	68.5
11	69.5	72.5	72	66.5	63.5	68.8
12	69.5	73.5	72	66.5	64	69.1
13	70	73.5	72	66.5	65	69.4
14	70.5	74	71.5	67	65	69.6
15	70.5	74	71.5	67	65.5	69.7
16	70.5	74	71.5	67	65.5	69.7
17	71	74.5	71.5	67	65.5	69.9
18	71	74.5	71.5	67	66	70
19	72	75	71.5	67	66	70.3
20	72	75	72	67	66	70.4

Table 19 The results of Add Penalty method with k from 1 to 20 (Naïve Bayes)



Laplace						
k	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Accuracy
1	91	89.5	85.5	78	83	85.4
2	92.5	91	86	77	82	85.7
3	94	91	83	75	79	84.4
4	95	92	82.5	74.5	78.5	84.5
5	94.5	95	83	73.5	76	84.4
6	93	95	82	72.5	73	83.1
7	92	93.5	82	70.5	72	82
8	92	93	80.5	69.5	70	81
9	92.5	92	80	69	69.5	80.6
10	91	91	79	69	66.5	79.3
11	88.5	90	76.5	69	66.5	78.1
12	87	89	75	69	66	77.2
13	86	88.5	74.5	68	64.5	76.3
14	86	88	73.5	67.5	63	75.6
15	86	86.5	73.5	67	62.5	75.1
16	85	87	73	66.5	62	74.7
17	84.5	86	72.5	66.5	61	74.1
18	84.5	85.5	72.5	65.5	61	73.8
19	84.5	84.5	69.5	64	60.5	72.6
20	83.5	83	68.5	63.5	60	71.7

Table 19 The results of Laplace method with k from 1 to 20 (Naïve Bayes)

Overall, Naive Bayes generates very good results. The accuracy is better than 80%, which is quite high for a real dataset. For Naive Bayes classification without penalty, since there is no k parameter in the formula, there is only one accuracy result = **79.6%.** Add penalty achieve the highest accuracy of **70.4%** when k = 20. Add penalty achieve the highest accuracy of **85.7%** when k = 2. Figure 13 is provided to display the comparisons.

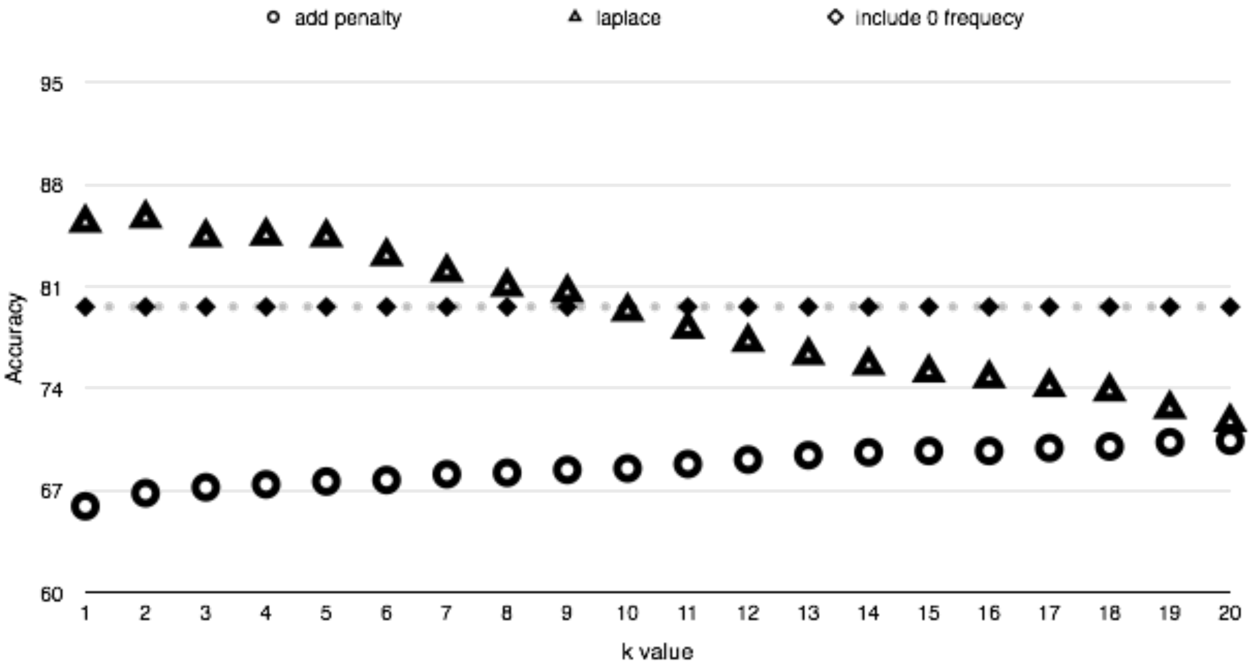


Figure 13 The comparison between three methods: include 0 frequency, Add penalty, Laplace.

With Add Penalty method, at first the accuracy increased rapidly until k = 15; after that, the result is still increasing, but much slower and bound around 70%. Laplace on the other hand, shows that accuracy decreases when k increases. To explain why the wine data behaves like that, we need to examine how Add Penalty and Laplace work.

For Add penalty: $P(H|X) = \left(\frac{1}{k}\right)^n \times \frac{P(X|H)P(H)}{P(X)}$ (formula 7)

For Laplace: $P(H|X) = \frac{P(X|H)P(H)+k}{P(X)+b-1}$ (formula 8)

Observe the equations above and notice that for the Add Penalty method because k is in denominator, the greater k, the smaller P(H|X) will be. If k is increased to ∞, P (H|X) will become zero. In other hand, Laplace has k in numerator, so the greater k, the further P (H|X) from zero.

In conclusion, zero frequency problem usually needs to be prevented when applying Naive Bayes classification, but for wine informatics dataset, zero value makes the accuracy increase. We apply two different methods to resolve zero frequency problem, and the results of both methods show that the nearer P(H|X) is to zero, the greater the accuracy will be.

Chapter 7: Comparisons

All of the classification models that we have covered so far (Decision Tree, k-NN, and Naïve Bayes) are white box testing, which "is a method of testing software that tests internal structures or working." (William) In other words, we can analyze the prediction accuracy and draw out useful information from how the models react to the database. Opposite with white box testing is black box testing, which tests the algorithm functionality. It works fast and usually generates better results than white box testing, but we will not be able to explain how it gets the conclusions. We apply Support Vector Machine, which is one of the most popular black box testing methods, to the wine dataset. Its results will be set as a benchmark to compare with our white box testing methods.

7.1 Support Vector Machines (SVM)

"SVM are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis." (Cortes and Vapnik). SVM for classification will based on the training data, build a model by constructing "a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier." (Press, Teukolsky and Vetterling). After having the model, a test data is used to predict the accuracy. SVM usually gives very high accuracy, but it is a black box technique. Not like white box techniques such as Decision Tree, k-NN, or Naïve Bayes where we can look at the analyzed data and figure out the reasons why the accuracy is high or low. We are not be able do the same thing with SVM, even though some detailed examples are provided in books and articles such as: *"An Introduction to Support Vector Machines*" of N. Cristianini and J. Shawe-Taylo*,* or *"Learning with Kernels*" of B. Schölkopf and A. J. Smola.

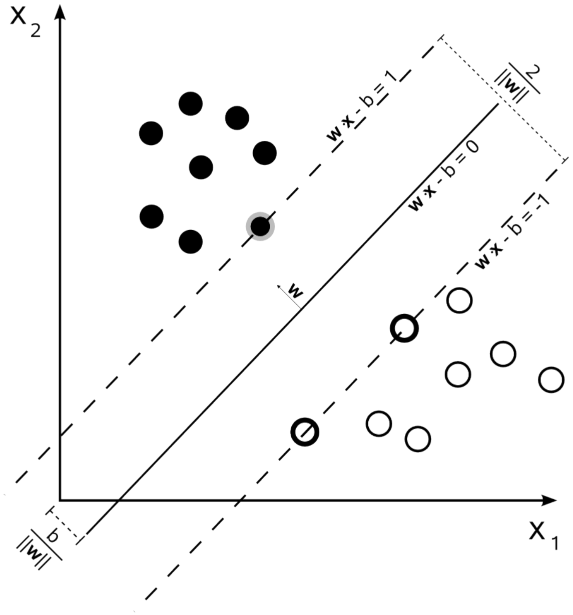


Figure 14. 2 dimensional hyperplane of Support Vector Machine.(Cyc)

In this figure, a simple dataset with 2 dimensional samples from two classes is used to demonstrate how linear SVM works. Because wine dataset has 304 attributes, SVM needs to build model of 304 dimensional spaces. As a result, the analyzed data is so complicated that it is impossible for us to look through and understand the meaning behind it. For that reason, the SVM results are used as a benchmark, and we will know how good our models are compared to it.

There are different methods to improve the accuracy of SVM, and we will use two of them called: scale dataset and choose the best parameter. (Hsu, Chang and Lin). Scaling dataset is to "to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges" and "avoid numerical difficulties during the calculation" (Hsu, Chang and Lin). As the paper suggests, we apply linearly scaling each wine attribute to the range [-1, +1]. For the best parameter method, SVM used grid search to scans through the whole dataset and tried to pick the best C and γ (C is penalty parameter, and γ is kernel parameters). Table 21 shows how SVM generates the accuracy.

Support vector machine methods	Accuracy
SVM	81.9%
SVM Scale	86.1%
SVM Parameter	88%

Table 21. Prediction accuracy of 3 support vector machine methods

7.2 Compare the accuracy of Decision Tree, k-NN, Naïve Bayes, and SVM.

As section 4.3 indicates, Decision Tree did not perform very well in the 5-CV test with an average prediction estimate of 50.6%. When we compare Decision Tree against Naïve Bayes and KNN, these two performed much better, and Naïve Bayes Laplace generated the highest performance among them all. To further analyze the successful performance of Naïve Bayes, we compared our results against those from a previous research project going on using wine sensory data. In the previous works, we used several approaches but the one we focused on was the associated rule for predicting. Table 7.2 indicates the results from our research for Decision Tree, Naïve Bayes, and KNN, SVM as well as Dr. Chen's results for Associated Rule where we all used the same data.

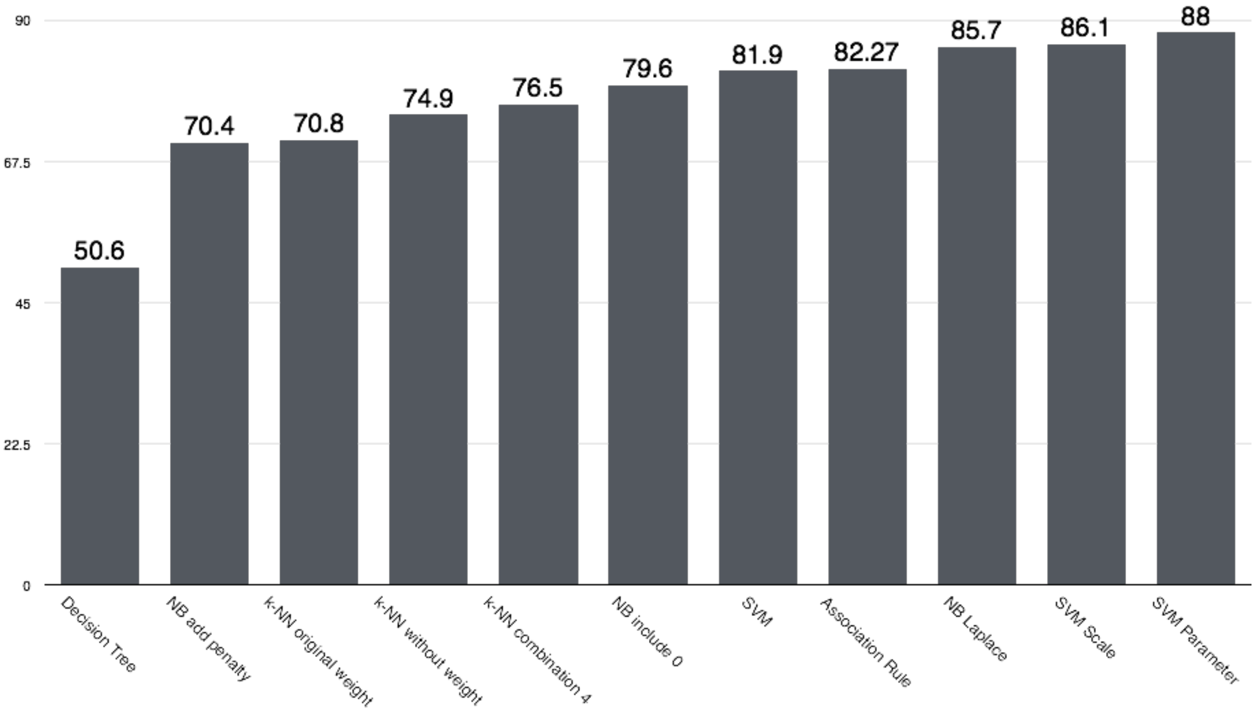


Figure 15. The comparison chart between Association Rule, Decision Tree, Naïve Bayes (3 methods), k-NN (3 methods), and Support Vector Machine (3 methods) classifications.

Decision Tree achieves the lowest accuracy with the prediction of only 50.6%, and that is just better than guessing heads or tails when flipping a coin. For all other methods, the accuracy results are above 70%, which is acceptable. Among our implementation algorithms, Naïve Bayes Laplace archives the highest accuracy of 85.7%. Compare the Support Vector Machine; the accuracy of Naïve Bayes Laplace beat the original SVM method (85.5% compares to 81.9%). However, the other two SVM methods generated even better results, especially SVM Parameter with an accuracy of 88%. Compared to our previous work, the results of Association Rule use a 1% support with 90% confidence that includes 61% coverage (61% of the testing data is used). In this work, all the results include 100% coverage. That means that more of the wines were predictable by the Naïve Bayes algorithm than by the Associated Rule, and Naïve Bayes Laplace also gets the better results. With an accuracy of 85.7%, it is a successful achievement.

Chapter 8: Conclusion & Future Work

8.1 Final remark

In this research, we have accomplished the following task: (1) We introduced classifying wines by grade with the sensory data provided from well-respected, established wine review sources. (2) We examined how to process attributes from a wine review and normalize them with the Computational Wine Wheel. (3) We implemented Decision Tree to learn the attribute paths of the training dataset. (4) With k-nearest neighbor algorithms, we applied it not only to plain attributes but also to weight attributes. As a result, we know how weight effects to the final results. (5) We analyzed Naïve Bayes classifier with three different methods: include 0 frequency, add penalty, and Laplace to learn the impact of zero probability and smooth data technique to the training dataset. (6) We applied Support Vector Machine on the wine dataset to set its results as benchmarks, and compare our accuracy results to them. As mentioned in introduction section, our purpose is using other data mining classifications to further analyzing the dataset. The comparisons in table 7.2 shows that we are able to discover a classification technique that generates better accuracy than Association rule, which is the original method used in Wine informatics paper (Chen, Rhodes and Crawford). We resolved those results into a readable format that suggests the Naïve Bayes classifier performed the best with this type of data because it successfully classified initially into two classifications of “90+” and “90-”. The Naïve Bayes algorithm made use of a parameter k that takes the place of the probability of 0 in order to strengthen our prediction accuracy. Even though Support Vector Machine is able to generate the better results, Naïve Bayes Laplace still archived a very high accuracy of 85.7%. Furthermore, all of our classifications are white box techniques, so we can analyze the result to get value conclusions. As a result, we supported our introductory concept that wine can be classified successfully using only wine sensory data.

As the consumption of wine continues to gain popularity, more people will seek explanations for what differs in a wine that makes it excellent or good. A layperson reading a wine review better understands words like CHERRY or TART versus the results of a physiochemical analysis. Research supporting the validity of classifying wines based on sensory data alone will greatly benefit wine growers as well as reinforcing the wine reviewer.

8.2 Future work

As there is not a great collection of research published about using wine sensory data to classify wines, there are some future works to consider furthering support and enhancing the new data science of Wine Informatics.

1.Dimentional selection

For now, we have 304 attributes in our dataset, and they are not equal in term of importance. As a result, an attribute classifier will help us focus more on important attributes. Even though weighting attributes is considered as a method that classifies which attributes are more important than others, a classifier that can choose the number of attributes might help increase the prediction accuracy results. For example: we might just want to choose the top 100 of the most important attributes to build the model.

2. Multi-Source Data Set

Our testing included using one source, *Wine Spectator* magazine wine reviews. There is more than one reliable source for wine reviews. A single wine can be reviewed by more than one source. We suggest using a data set that includes each wine having multiple sources for its review data.

3. Multi-Approach Testing with Focus on Classification

Our research centered on using three different algorithms, the Decision Tree, k- nearest neighbor and Naïve Bayes. We suggest that the next step is combining them with clustering to further analyze the accuracy, especially with Decision Tree. It generates the lowest accuracy of 50.6%, and mainly because it predicts incorrectly the "90-" wine instances. Therefore, with clustering techniques involved, the accuracy might greatly increase.

4. Differing Data Sets

Our research used one data set. We suggest using more than one data set to verify research results to rule out skewed data regarding why one approach performs better over another. *Wine Spectator* magazine has a huge repository of wine review data. We could make use of a larger quantity of data and preprocessed to have a minimum number of attributes so wines with lower classifications will still predict with a higher average than found in our research.

5. Use different methods to weight attributes

In chapter 5, we use different weight combinations, and we concluded that other combinations generate better accuracy results than the original combination. Since feeling is not something that can be measured exactly, we assume the attributes that express feelings are the least important, and this might not be true. As a result, we need further testing to weight attributes better when doing data pre-processing. We suggest the method called "keyword extraction using word cooccurrence statistical information.” The method scans through wine reviews and ranges keywords by the most frequent of appearance. With this approach, the attributes will be weighted based solely on statistical information and might give a better evaluation.

.

List of References

Altman, N. S. "An introduction to kernel and nearest-neighbor nonparametric regression." The American Statistician (1992): 46.

Antonie, Maria, Luizaiane Osmar R. Za and Alexandru Coman. "Application of Data Mining Techniques for Medical Image Classification." ACM SIGKDD conference. San Francisco: Multimedia Data Mining (MDM/KDD'2001, 2001.

Chen, B. GSU. 2 1 2015 <http://www.cs.gsu.edu/~cscbecx/Wine%20Informatics/Wine_Wheel_01242014.dat>.

Chen, B. www.cs.gsu.edu. 10 2 2014 <http://www.cs.gsu.edu/~cscbecx/Wine%20Informatics.htm. File: Wine_Wheel_01242014.dat>.

Chen, Bernard, et al. Wine Informatics: Applying Data Mining on Wine Sensory Review. Prod. Manuscript submitted for publication. Conway, 10 10 2013.

Chowdhury, G. Natrual language processing. Annual Review of Information Science and Technology, 2003.

Cortes, C. and V. Vapnik. "Support-vector networks." Machine Learning (1995): 20.

eRobertParker. e Robert Parker. 10 2 2014 < http://www.erobertparker.com/info/glossary.asp.>.

Furey, Terrence S. "Support vector machine classification and validation of cancer tissue samples using microarray expression data." Bioinformatics (2000): 906-914.

Foods & Wines from Spain. 22 3 2013 <http://www.winesfromspain.com/icex/cda/controller/pageGen/0,3346,1549487_6763472_6778161_0,00.html>.

International Organisation of Vine and Wine. OIV. 20 3 2013 <http://www.oiv.int/oiv/cms/index?lang=en.>.

Huang, Cheng-Lung, Mu-Chen Chen and Chieh-Jen Wang. Credit scoring with a data mining approach based on support vector machines. Elsevier, 2007.

Hagan, Martin T., Howard B. Demuth and Mark H. Beale. Neural network design. Boston: Pws Pub, 1996.

Han, Jiawei, Micheline Kamber and Jian Pei. Data Mining Concepts and Technology. Vol. 3rd. n.d.

Hand, D. J., Heikki Mannila and Padhraic Smyth. Principles of Data Mining. MIT Press, 2011.

Hipp, MLA, et al. "Algorithms for association rule mining—a general survey and comparison." CM sigkdd explorations newsletter (2000): 58-64.

Hsu, Chih-Wei, Chih-Chung Chang and Chih-Jen Lin. A Practical Guide to Support Vector Classification. Taipei, 19 March 2015.

Kohavi, Ron. "A study of cross-validation and bootstrap for accuracy estimation and model selection." Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. San Mateo, n.d. 1137-1143.

Lavrač, Nada. "Selected techniques for data mining in medicine." Artificial intelligence in medicine (1999): 3-23.

Lee, Wenke, Salvatore J. Stolfo and Kui W. Mok. A Data Mining Framework for Building Intrusion Detection Models. Columbia University . Columbia University , n.d.

Liaw, Andy and and Matthew Wiener. "Classification and regression by randomForest." R news (2002): 18-22.

Loukides, Mike. What is Data Science? O'Reilly Media, 2011.

Noble, Ann C. The Wine Aroma Wheel. 9 2 2015 <http://winearomawheel.com>.

McCallum, Andrew and Kamal Nigam. "A comparison of event models for naive bayes text classificatio." AAAI-98 workshop on learning for text categorization. 1998.

Menard, Scott. Applied logistic regression analysis. Vol. 106. Sage, 2002.

Oberholster, Anita. Chemical analysis for the winery Practical aspects. 30 12 2014.

Quinlan, J. R. Induction of Decision Trees. Machine Learning. Kluwer Academic Publishers, 1986.

Qi, Yanjun, Ziv Bar-Joseph and Judith Klein-Seetharama. NIH Public Access Author Manuscript. 4 January 2012. 21 Feburary 2015 <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3250929/>.

Pachet, François, Gert Westermann and and Damien Laigre. "Musical data mining for electronic music distribution." Web Delivering of Music, 2001. Proceedings. First International Conference on, 2001.

Press, William H., et al. Numerical Recipes: The Art of Scientific Computing. New York: Cambridge University Press, 2007.

Sun, et al. ""Classification of wine samples by means of artificial neural networks and discrimination analytical methods."." Journal of analytical chemistry 359.2 (1997): 143-149.

Sutton, Oliver. "Introduction to k Nearest Neighbour Classification and Condensed Nearest Neighbour Data Reduction ." University lectures. 2012.

Safavian, S. Rasoul, and David Landgrebe. A survey of decision tree classifier methodology. 1990.

Sarawagi, Sunita. Information Extraction. Vol. 1. Foundations and Trends in Database, 2007.

Shanken, Marcin R. and Thomas Matthews. Why We Taste Blind. 1 1 2015 <http://images.winespectator.com/wso/pdf/WShowwetasteLTR.pdf>.

Spectator., Wine Spectator Home | Wine. Wine Spectator Home | Wine Spectator. 7 4 2014 <http://www.winespectator.com>.

Steinbach, Michael. "Data mining for the discovery of ocean climate indices." Proc of the Fifth Workshop on Scientific Data Mining, 2002.

Russell, Stuart and Peter Norvig. "Artificial Intelligence: A Modern Approach." Prentice Hall (2003).

Villiers, André De, et al. "International Journal of Wine Marketing." International Journal of Wine Marketing (1989).

Yang, Nan. Quality differentiation in wine markets. Wasington DC: Ann Arbor , 2010.