MASTER'S THESIS

Lilla Gula

Information Data Management for the Future of Communication

Robin Norberg

Master of Science in Engineering Technology
Computer Science and Engineering

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering

ABSTRACT

Data mining is an area where computer science, machine learning and statistics meet and where the goal is to discover and extract information such as relations and patterns that's hidden in the data. Depending on the method used and the data set being analysed, it's possible to detect for example fraudulent bank transactions, predict shopping habits, categorize large collections of e-books or discover variable associations. The main objective in this thesis is to get a deeper knowledge of the data mining area and evaluate methods to automatically classify the data in structured personal registries. In this report I'll describe different methods and algorithms used in data mining, apply a few selected popular algorithms on two data sets and then discuss the result.

Contents

CHAPTER 1
Introduction

This thesis for the degree of Master in Science and Engineering at Luleå University of Technology was made at the company Ågrenshuset Prepress IT AB in Örnsköldsvik. This particular topic about analysing data was selected after a discussion with Ågrenshuset about their current needs. In this thesis, I investigated ways to automatically identify what kind of data was stored in structured data records. The data types investigated was a few of the ones stored in two records over customers that included names, addresses, phone and postal numbers as well as some other attributes.

1.1 Background

Every day worldwide, enormous amounts of data is being produced and stored at an amazing pace. Meanwhile, the steadily dropping prices for storage devices and the increasing capacity of each unit have made it into a simple choice to postpone the decision what to do with all this data being produced. In today's modern society where we're constantly interaction with our environment through computers in everyday life and each choice we make is being processed in one way or another. May it be due to reading the newspaper online, having a voice call through our mobile phone, writing a post on facebook or shopping groceries with our credit card. Each of our action is recorded and stored in massive records of data. While some just want statistics on how their services are being used or are tracking your choices due to services you pay for, other will try to find meaningful patterns in your choices and try to benefit from them. This can for example be done either by trying to target advertisements online due to your search preferences or by giving you discount prices on groceries in your local grocery shop due to your earlier shop history and what you'd be likely to buy. But our digital-footprints on the Internet is not the only thing generating data.

The Library of Congress in the United States, being the biggest library in the world had 285 terabytes of web archive data stored in January 2012 with a steady growth of about 5 terabytes per month[1]. That alone being an impressive figure, but even more impressive statistics from the year 2009 stated that companies in the United States with more than 1000 employees in 14 of 17 sectors(such as Government, Banking, Health care providers etc) was storing more data per company then compared to the volumes the Library of Congress was storing in the beginning this year[2]. It was estimated that 1.8 trillion gigabytes of information would be created during the year 2011 and that the amount of data created each year would grow around 50 times todays number before the year 2020 since the amount of data created keeps doubling every two years[3]. The problems that rises for companies other than the obvious storage issue is the ability to use all this data for something meaningful. Manually analysing this kind of information simply takes too much time so a great deal of time have been spent in last couple of decades to develop computer methods that are able to analyse the relevant data automatically.

But in todays IT society, it's not only large corporations that is analysing their data, even small companies benefit from using data analysing methods to increase productivity in their everyday tasks. In this thesis I'm going to present an overview how data analysis is being performed, take a look at how some of the basic and popular algorithms work and then compare a handful of them using a small specific problem in the domain of data analysis.

1.2 Problem Description

The main problem discussed in this thesis was ways to automatically classify text data grouped in columns. The idea was that if a structured data source with tables of data was presented, a machine should be able to automatically analyze the structure and content of data in each column and try to classify the data depending on how similar data had been classified earlier by the algorithm. The classification algorithm should after being subjected to training data, where the machine had access to whether the classification was correct or not, be able to draw conclusions from the data analysed and construct a classification model which would be able to classify similar data.

This was the first step for a future project where the goal is to be able to join two unknown data sources together based on how the tables are classified so that the content of each column is kept consistent even after the merge.

1.3 Project goal

The goal with this thesis can be split up as following.

- **Concept:** Understand the concept of data analysis and how these methods can be applied on small specific problems such as structured data classification.
- **Method:** After the initial research, select a handful of promising methods and compare their performance when classifying data.
- **Evaluate:** Find a method that are able to correctly classify a data set of four different classes (firstname, lastname, address and city) with a success rate of at least 70% per class. For this requirement the method should have access to a larger data set with the same classes to construct some sort of classification model before trying to classify the smaller data set.

1.4 Motivation

Ågrenshuset Prepress IT AB is one of 5 companies that's part of a corporate group called Grafisk kompetens i norr, all located in the vicinity of Örnsköldsvik. These companies communicate internally, between each other and to customers through the Internet, sharing information and data. A long term goal for this corporate group is the ability to quickly be able to modify data records by adding, removing or joining data records without in beforehand having to format the data manually to a certain structure.

The reason why they wanted this investigated is to gain knowledge how that kind of computer program can be built, benefits and drawbacks of the most common algorithms and possible limitations of the concept.

1.5 Delimitation

The initial limitations chosen for the above problem was to classify structured textual data on column basis in tables over customer records. To keep the size of the classification problem down, only Swedish data was analysed since different countries sometimes have different types of structures for addresses etc. The main data types this thesis focused on classifying is first names, last names, addresses and city names. All in plain text format.

CHAPTER 2

Theory

This chapter start by introducing the basic fundamentals of analyzing structured data. It then takes a closer look at a couple of the popular algorithms used when doing data analysis on data sets to see if any of those might work well for the problem description given.

2.1 Data mining

The need for more automated methods to analyze data was formed during the 80's since more and more data was being produced in the dawn of the computer age. It was also in 1980 that the database language Structured query language, more known to people as SQL, was released on the market and instantly became a huge success which boosted the amount of data being stored. As more and more businesses started to work with computers in their administration to store for example financial records, the databases grew so much that the traditional methods, where they performed statistical analysis by hand, couldn't keep up with the new requirements of efficiency so new techniques had to be developed. This lead to companies pooling money into research and new advances in areas such as artificial intelligence, neural networks, machine learning and statistics was made. In the beginning of 1990, the association for the advancement of artificial intelligence released a press book called Knowledge Discovery in Databases that described the method how to unlock hidden knowledge in data[4]. Data mining, which nowadays is a popular buzzword for the whole area of data analysis, can be seen as a modern version of the analysis step of the Knowledge discovery in databases process proposed in that book.

Data mining is used to describe the process of analyzing data to find patterns & relations, classify data and predict results in large data sets of structured data. It can be seen as an area where computer science, machine learning and statistics meet on common grounds. Different techniques that also fit in the this category include association learning, data classification and clustering as well as regression.

2.2 Text mining

If data mining is the main area when dealing with structured data in databases, text mining refers to the process of analysing and detecting knowledge in unstructured data in the form of text. Some popular areas in which this is used is automatic text-classification, clustering and summarization. The main problem in text mining is that the data in text form is written using grammatical rules to make it readable by humans, so to be able to analyse the text it first needs to be preprocessed with techniques such as stemming which removes suffixes.

An even older method to analyse unstructured text is called Natural language processing(NLP). The research around NLP began in 1950, and is more focused on linguistics where the position of each words related to each other is of importance when you're for example trying to determine the content of an sentence.

2.3 Machine learning

Machine learning is a learning technique in the area of artificial intelligence which deals with making intelligent decisions and drawing conclusions based on features of the input data rather than preset rules. The idea is to be able to solve problems where the algorithm is unknown, or the problem becomes unsolvable due to constantly changing requirements. By analyzing existing data you'll be able to detect patterns and relations in the data. From the conclusions made with existing data, the algorithm can then "learn" to make accurate predictions when encountering new data for the same problem by producing general rules that can be used. Machine learning can be divided into two different categories: supervised learning and unsupervised learning. Supervised learning deals with methods such as classification, prediction and regression for which the user is required to supervise the algorithms during the learning phase and tweak the settings to optimize the solution. With unsupervised learning which instead deals with methods for clustering, compression and rule learning, the algorithms themselves handle the tweaking.

2.4 The basic representation

Data structure
The most common structure for input data when it comes to data mining algorithms consist of a table like structure where data is stored in rows and columns. Each row consist of data related to each other, divided up in different columns where each column can have it's own data type.

An example of such a table is:

First name	**Last name**	**Address**	**City**	...
Sven	Svensson	Kungsgatan 22	Örnsköldsvik	...
Ben	Bengtsson	Fiskgatan 11	Umeå	...
Hans	Hansson	Bagarvägen 12	Husum	...

Database
A database can be seen as a collection of data tables, where the data table store data in rows and columns as described above.

2.5 The trivial solution

Depending on the application for which data needs to be analysed, the trivial solution might sometimes be sufficient. Here's a couple of methods that can be used for small scale implementations where you only have to test a single data type.

Ways to trivially classify a string of text for a specific problem

- With the help of regular expressions(regex), that is, the structure of the text. An email address will almost always have the following structure: LocalAdress-Name@DomainName.SuffixWord and can hence be tested with a regex string such as the following: `\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b`

- Iterating the words and trying to find a match against a "dictionary" of words made beforehand.

- The column name of the data set, if available.

However, this thesis won't go any further into these types of solutions since it's too specific.

2.6 Classification

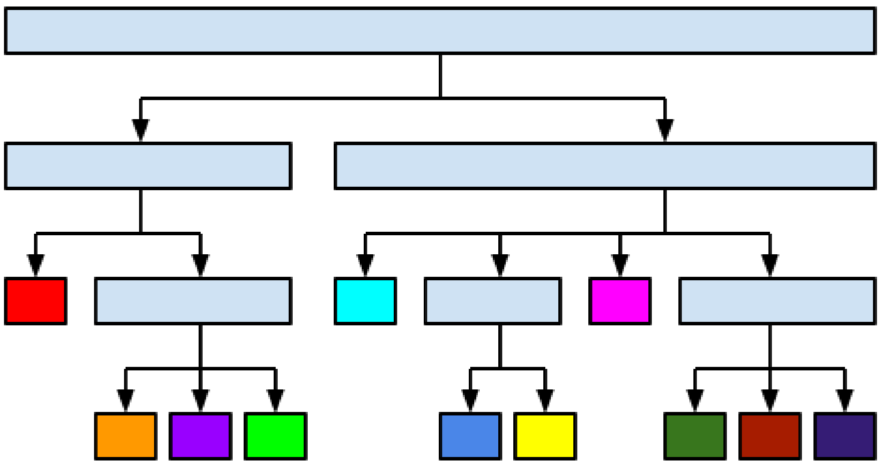


Figure 2.1: One type of classification, here in the form of a decision tree.

Classification in data mining and machine learning is a technique used to predict what categorical class a data instance belongs to. An example of classification can be seen in figure 2.1 where data is being broken down into different parts until a specific classification can be found. The data instance to be classified can be anything between a single data point to a 500 pages long book. Usually this technique is based on supervised learning. Supervised learning means that by analyzing one or several sets of labeled training data, it's possible to construct a classification model using the unique features of the data that seems to identify the correct class. New data that's similar to the training data can then be used in this classification model to predict what class they belong to. Some popular methods for the classification techniques include Bayes/Naive Bayes, Decision trees, Neural networks, Rule-based methods as well as Support Vector Machines.

2.7 Clustering

Clustering is the technique to divide a set of objects into groups, called clusters. In the domain of data mining and machine learning, clustering is a unsupervised learning technique. The idea is that you can use clustering algorithms on a data set without having to construct a model for the algorithm using training data, instead the algorithm will try to group data together on pre-existing similarities in the data itself.

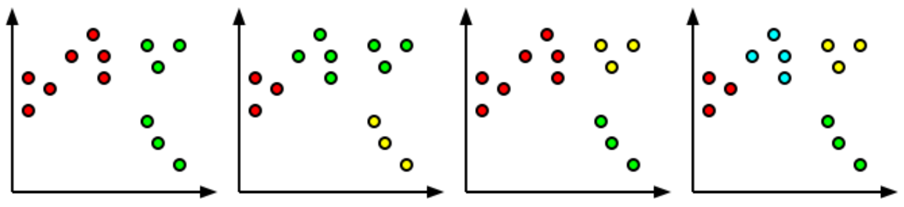


Figure 2.2: A few examples how a data set can be clustered.

As seen in the figure above, clusters can be detected in a various ways. Neither solution is more correct than another in a general sense since each one can be beneficial depending on the situation where it's implemented.

2.8 Algorithms

This section describes a few of the most common algorithms used in data mining.

2.8.1 Statistical probability

Lets start with a simple statistical problem explaining the statistics behind a coin game called heads or tail. Heads or tail is played by two players, often to solve a dispute, by flipping a coin into the air and then looking what side of the coin is facing upwards after landing or being caught. The winner is the player who guessed right before hand, or while the coin was airborne.

After the coin has been flipped and landed on a surface, it will show one of two outcomes, meaning that the side of the coin displayed will either show heads or tail. A special case also exist where the coin lands on its side, or leaning against another object, but in those cases the throw is repeated.

Lets say one want to know the probability to get heads 3 times in a row.

Lets denote the probability of the coin after each throw showing heads as P(A) and tails as P(B). Each outcome has equal chance of occurring so the probability is 50%. In statistical terms this is denoted 0.5 since P(A) E [0,1].

The probability to get heads in the first row: $P(A) = 0.5 = 50\%$

The probability to get heads 3 times in a row:

$$P(A) = 0.5^3 = 0.125 = 12.5\% \tag{2.1}$$

The specific outcome of all possible outcomes is called an "event", for example the outcome of the coin showing heads after a toss. Here follows a couple of event types:

Disjoint events

$$P(A \cup B) = 0 \tag{2.2}$$

Two events are disjoint if the different outcomes A and B cannot both occur at the same time. An example of this is a coin toss since it will never show head and tails at the same time.

Independent events

$$P(A \cup B) = P(A) * P(B) \tag{2.3}$$

Two events are independent if both can occur at the same time, but the occurrence of one won't affect the occurrence of the other. For example the outcome of a dice roll will not affect the outcome if the dice is tossed once more.

When it comes to calculating independent events, all that's needed to do is multiply the probabilities together.

For example, the probability to roll a 5 with a dice will be 1/6 since there's six different outcomes.

The probability to roll a 5 and then a 4 will be 1/36 since there is 36 possible combinations two dices can show.

$$P(C \cup D) = P(C) * P(D) = \frac{1}{6} \cdot \frac{1}{6} = \frac{1}{36} \tag{2.4}$$

Dependent Events
Two events are dependent if the first outcome will affect the probabilities of a second outcome. Lets say if you have 2 black and 2 red marbles in a bag. If you randomly remove a marble from the bag and it happened to be red, the odds of picking another red marble from the 3 marbles that's left will decrease.

Unlike Independent events, it's a bit trickier to calculate the probability of multiple dependent events, especially if the number of dependent events is high. To explain this, the Bayes theorem can be used.

2.8.2 Bayes theorem

Named after the English mathematician Thomas Bayes who studied probability in the 18th century and who constructed this theorem and was the first to use it to influence belief. This kind is called conditional probability where the probability of certain event A is calculated, given the probability of another event B.

A classical problem to describe Bayes Theorem is spam filtering. There are 70 new emails in a email inbox with the following properties:

43 of the 70 emails are spam
  35 of the emails that are spam have the word “viagra” in the content.
  8 of the emails that are spam doesn’t mention the word.
27 of the 70 emails are valid emails
  4 of the valid emails have the word “viagra” in the content.
  23 of the valid emails doesn’t mention the word.

What is the probability that a new email is spam if the content mention the word “viagra” if the above information is known? Since these events are dependent, it’s possible to use Bayes theorem to calculate the answer.

P(spam): The probability of an email being spam
P(viagra): The probability of an email mentioning the word “viagra”

$$P(spam|viagra) = \frac{P(spam \cap viagra)}{P(viagra)} = \frac{(P(viagra j spam) * P(spam))}{P(viagra)} \tag{2.5}$$

$$P(spam|viagra) = \frac{\frac{35}{43} \cdot \frac{43}{70}}{\frac{39}{70}} = \frac{0.5}{0.557} = 0.8976 \tag{2.6}$$

As seen in equation 2.6, the probability of an mail being spam if the content mention the word “viagra” is about 89.8%

Bayes Theorem also support more than two dependent events. Lets expand the example above by adding the word “enlarge”. It’s now possible to calculate the probability that an email where the content include both words “viagra and “enlarge” can be classified as spam.

$$P(spam|viagra, enlarge) = \frac{P(spam) * P(viagra|spam) * P(viagra|spam, enlarge)}{P(viagra) * P(viagra|enlarge)} \tag{2.7}$$

The problem with adding more dependent events is that it the equation rather quickly becomes complex as seen in equation 2.7. In big data sets, the number of dependent variables could easily become hundreds.

2.8.3 Naive Bayes classifier

While Bayes theorem calculates the probability of one event occurring given that another event have already occurred, Naive Bayes modifies the method and "naively" assume that each event is conditionally independent of each other.

By using independent events, the following equation can be used instead:

$$P(spam|viagra, enlarge) = \frac{P(spam) * P(viagra|spam) * P(enlarged|spam)}{P(viagra) * P(enlarge)} \tag{2.8}$$

The good thing here is that the equation only grow linearly for each additional variable so it's possible to process large quantities of data at the same time without any problems.

Advantage
Naive Bayes simplicity makes it a fast and scalable algorithm that perform surprisingly well compared to the more complex models as long as your data set doesn't grow too much.

Drawback
Naive Bayes will run into a problem if you encounter data with a variable having zero probability since it will ruin your equation when multiplied with the other variables. However, this can be fixed if you smooth the data beforehand where zero probabilities is removed.

2.8.4 Rules and decisions

Lets say one needs to take a decision to go to the beach or not. Our mind is able to analyze all the different pro or cons of going in a split of a second while if a machine was asked to make the choice for us it would require a set of rules to determine the most favorable outcome for ourselves. To simplify the problem the machine will consider 3 different variables (weather, air temperature and friends are going) to calculate the outcome.

For the computer to make sense of this data, the following rules are used to create a relation between the different variables. The rules are read from top to bottom where the conditions of each set of rules is checked against the variables of the row in the table above. When the requirements for a rule succeeds the final value will be set and the remaining rules will not be controlled.

Scenario ID	Weather	Air temperature	Friends are going
1	Sunny	20 °C	Yes
2	Cloudy	23 °C	Yes
3	Rainy	17 °C	No
....	...	...	...
n	Rainy	26 °C	Yes

Table 2.1: A couple of scenarios used to decide if you want to go to the beach or not.

Here's an example of a ruleset for the table 2.1.
If Weather = Sunny and Friends are going = Yes then Will enjoy the beach = Yes
Else If Weather = Sunny and Air temperature $\geq$ 20 °C then Will enjoy the beach = Yes
Else If Weather = Cloudy and Friends are going = Yes then Will enjoy the beach = Yes
Else If Weather = Rainy then Will enjoy the beach = No
Else Will enjoy the beach = No

2.8.5 Decision Table

A problem with a normal if-then-else ruleset is that it's hard to get a good overview of the solution if the number of variables increase, if the ruleset themselves become complex or if you want to use the same types of variables for more than one type of activity.

What you can do then is to construct a decision table where all the combinations of the variables are entered under Conditions. The ruleset is used once to calculate the answer for the different combinations of input and then fill in the output answer for each "question" under activity. An example of this can be seen in table 2.2.

Conditions					
Weather	Sunny	Cludy	Rainy	...	Sunny
Air temperature	28 °C	24 °C	18 °C	...	18 °C
Friends are coming	Yes	No	Yes	...	No
...	...	...	...	...	...
Water temperature	22 °C	16 °C	22 °C	...	24 °C
Activity					
Will enjoy the beach	Yes	Yes	No	...	No
Will enjoy a swim	Yes	No	Yes	...	Yes
Will enjoy shopping	No	Yes	Yes	...	Yes

Table 2.2: A decision table used to decide the outcome for a couple of scenarios depending on the prerequisites.

2.8.6 Decision Trees

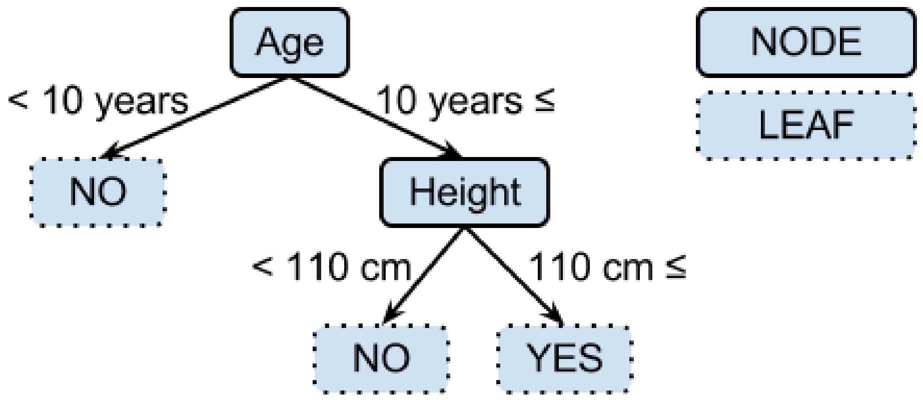


Figure 2.3: A simple example of a decision tree.

A decision tree is another way to model a ruleset graphically and is always constructed with a start node from which other nodes are attached. Its built up using the “divide-and-conquer” technique where a big ruleset is split up into a network of nodes, where a node can be seen as a rule that only checks a single variable. A node is not limited to two children nodes, but can have as many as required by the variable tested. Integer values are usually compared with equal, greater or less then logic creating a two or three way split while symbolic variables like constants normally create one path per possible value. As one traverse down the network you’ll eventually reach a leaf node, that is a node without any children. The leafnode will present the output value for all the different scenarios that will reach this state.

The decision tree above(Figure 2.3) is a simple example modeling the following if-else-then ruleset that checks if a child is allowed to enter a specific rollercoaster:

If Age = 10 years and Height = 110 cm then Can ride the rollercoaster = Yes Else Can ride the rollercoaster = No

The advantages of a decision tree instead of a ruleset is that you can easily visualize the different path a complex problem can take. Another thing is that it's easy to extend the decision tree with more rules when needed. A drawback however is that complex solutions could potentially create multiple subgraphs in the decision tree that are identical. Another thing is scenarios where a variable with a missing value would get stuck in the tree if the data haven't been prepared beforehand, or a special solution for missing values have been implemented.

2.8.7 Inferring rudimentary rules

Inferring rudimentary rules, or also called 1R for the 1-rule concept, is a simple but remarkable usable algorithm proposed by Robert C. Holte in 1993[5]. The idea of the algorithm is based on observations from real world sets of data that classifications can be predicted fairly well with only one or a few of the variables from a data set. The algorithm itself works as following. For each variable in the training set, construct a one level decision tree with branches corresponding to the values of that variable. Classify each branch based on how often it have occurred in the training data for the values within and then calculate the error rate for this 1-rule. The rule generated from the 1R-decision tree with the smallest error rate will be the rule used to classify new data.

While the algorithm method above works fairly well for symbolic values, a problem could raise for numeric values if each number would produce a branch each. That would create overly complex rules for that attribute which would work good on the training set itself, but perform poorly on live data. To create a more general ruleset when dealing with numeric values, the values are split into groups of intervals instead. Lets describe this using a similar example as the one seen in table 2.1. As an example, we're given rules that as can be seen in figure 2.3.

The first step to generalize this is to first divide the sets of values into intervals for each step when the value changes from No to Yes and vice versa, seen below in table 2.4.

Although this will create intervals with pure output, the ruleset is still too complex and

°C	15	16	17	18	19	20	22	23	23	24	26	28	29	30	30
Y/N	No	Yes	No	Yes	Yes	No	Yes	Yes	Yes	No	Yes	No	No	Yes	No

Table 2.3: A table for the Air Temperature-attribute related to the output value of go to the beach are displayed. Note that some temperatures such as 23 °C might have different outputs due to the influence of other attributes from the data set.

°C	15	16	17	18, 19	20	22, 23, 23	24	26	28, 29	30, 30
Y/N	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes

Table 2.4: The values have now been divided.

won't perform well. The idea is to simplify the intervals further by introducing error to this data set while creating a more general solution which will perform better in the long run. Holte suggest to require a minimum amount of values per interval expect the rightmost one. In his studies he recommended a minimum of 6 values for big data sets and 3 values for smaller ones. Depending on the algorithm used to select intervals while optimizing the error rate, it might look something like table 2.5 below:

°C	15, 16, 17	18, 19, 20, 22, 23, 23	24 ,26, 28, 29	30, 30
Y/N	No	Yes	No	Yes

Table 2.5: An example of the final result.

After this simplification, only a classification error rate of 3/15 is introduced even though the number of intervals used went down from 10 to 4.

One should note that while this algorithm performs fairly well in some situations, this is not a suitable algorithm for text analysis since a single rule can't keep up to classify data sets with a high number of unique text strings without overfitting the data into a specific class.

2.8.8 Hidden Markov model

To explain the Hidden Markov mode, also called HMM, one needs to start to explain the Markov process. A Markov process is a memoryless stochastic process with finite states(possible outcomes), where the outcome of every state change is dependent only on the the last state visited and where the probabilities doesn't change over time. As an simple example of the Markov process, imagine studies being made on shopping habits for new mobile phones. 35% of the people who bought a new phone last year does not buy another phone this year. Meanwhile, 25% of the people who didn't buy a new phone last year will buy one this year. If 40000 people buys a phone and 100,000 doesn't buy one a certain year, how will the distribution look like the year after that? By using ekuation 2.9 below, it's possible to calculate this.

b1 = people that will buying a new phone that year
b2 = people who won't buy one that year

$$Ax = b \tag{2.9}$$

A = $\begin{bmatrix} .65 & .25 \\ .35 & .75 \end{bmatrix}$
x = $\begin{bmatrix} 40000 \\ 100000 \end{bmatrix}$
b = $\begin{pmatrix} b1 \\ b2 \end{pmatrix}$

$$\begin{bmatrix} .65 & .25 \\ .35 & .75 \end{bmatrix} * \begin{bmatrix} 40000 \\ 100000 \end{bmatrix} = \begin{bmatrix} 51000 \\ 89000 \end{bmatrix} \tag{2.10}$$

As seen in equation 2.10, 51000 people will buy a new phone that year while 89000 won't be buying one.

If one want to calculate the distribution after 5, 20 or n years, all one need to do is replace x with b and use equation 2.11.

$$Ab = A^n x \tag{2.11}$$

The Hidden Markov model is much like the Markov process, except that the HMM hides its current states for the observer. An example of the Hidden Markov model can be seen in figure 2.4 below. Only the output from each state will be visible.

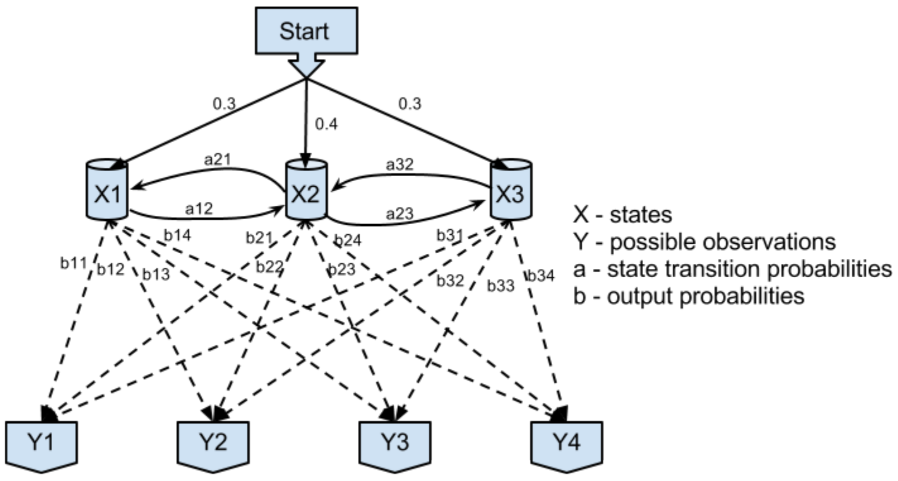


Figure 2.4: An example how the Hidden Markov model can look like.

A popular problem to describe the HMM concept is the Urn problem. In a room there's a finite number of urns(states), each with a known collection of colored balls(observations).

Someone selects a urn and randomly pick up one of the colored balls from the active urn. The color of the selected ball is documented on a paper and the ball is put back into the urn it came from. This is then repeated finite number of times. At any time t, you're allowed to look on the paper where you can see the generated output so far. The process of selecting the urn may depend on the last selection, each possible transition to another urn have a fixed probability and the total transition probability between any couple of urns must sum to one.

2.8.9 k-means Clustering

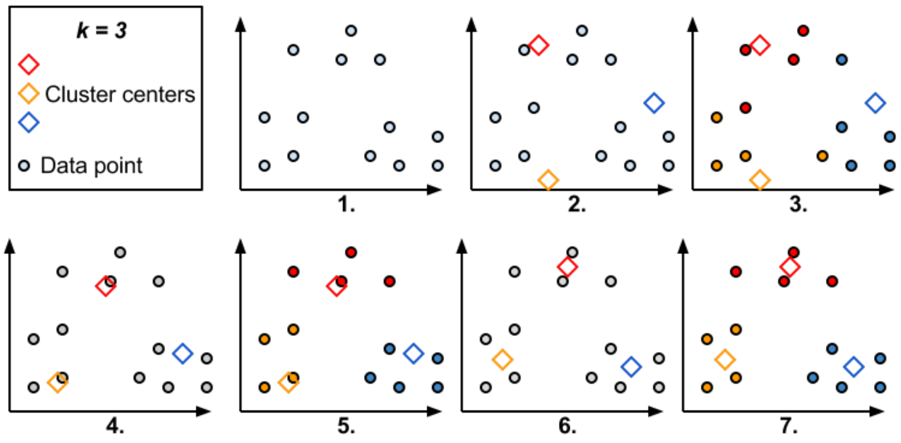


Figure 2.5: A classification of a new data point will depend on the selected size of k.

The concept of the k-means clustering algorithm is to partition data points into k preset numbers of clusters. An example of this can be seen in figure 2.5.
For your given data set, select k number of clusters you want the data divided in.

1. Arbitrarily create k points in the data space as initial cluster centers.
2. For each point in the data set, assign that point to the cluster center that's closest, measured in Euclidean distance.
3. Compute the average position of all the points in each cluster, and then move the cluster center to that location.
4. Redo steps 2 and 3 until the cluster centers have stabilized at one location.
5. When the clusters have stabilized (or after a fixed number of iterations) the set of data points in each cluster is returned.

The idea of this algorithm is pretty simple but one should have in mind that with the method described, the solution found will usually be a local minimum and that there's no guarantee that the global minimum is found. To have a better chance at finding a global minimum, a common technique is to run this algorithm several times using different positions for the initial cluster centers and then selecting the solution with the smallest square distance in all clusters. This algorithm is one of the most popular due to the good performance even though it has a low complexity. A drawback however is that the user need to take careful consideration of what data is being classified when selecting the number of clusters to divide in. There's a high risk of potentially overfitting/underfitting the data. Due to this, this algorithm is more suitable for supervised learning algorithms where the user can tweak the algorithm to better fit the data.

2.8.10 k-Nearest Neighbor

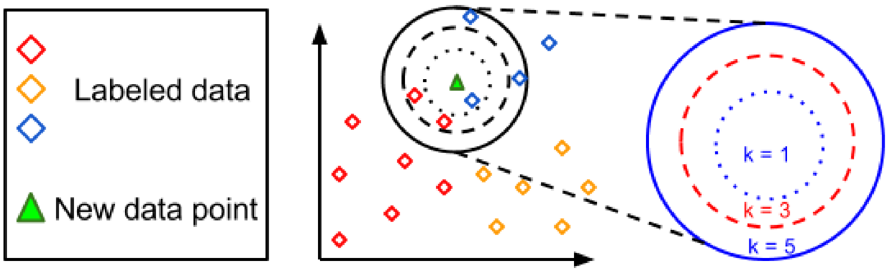


Figure 2.6: Here one can see the process of the cluster centers moving across the space and classifying points before stabilizing and returning the found clusters.

The k-nearest neighbor, also called k-NN, is a popular algorithm used in Data Mining due to its simplicity. The concept is that new data points will be classified based on what the majority of the nearest neighbor points have been labeled as in the training data, an example of this can be seen in figure 2.6. The amount of neighbor points being evaluated is determined by the input variable k, which usually is an odd number to avoid cases where a new data point is surrounded by an equal distribution of two classes. For some situations, it could be beneficial to weigh the "vote" for the neighbors depending on their distance so that closer points will contribute more to classification than distant ones even though they still are neighbors. One should also keep in mind that the efficiency of the value k is highly dependent on the data set being analyzed. The classification algorithm will benefit from a high k value if the data is noisy, but could also be exposed to overfitting due to this. The k-NN is also an lazy learning algorithm, that means it doesn't create a generalized model during the learning phase which most of the other classifiers have to do. Instead it relies on having access to the whole training data set when performing classification on the new data. Another thing is that the algorithm doesn't make any assumptions of the data distribution which could come in handy when analyzing handmade data from real world applications.

The drawback of being extremely fast in the training phase is that this lazy algorithm becomes computationally heavy when performing the actual classification of the data. It also requires a lot of memory to process if the training set is large since it need direct access to all the data in order to classify.

2.8.11 Support Vector Machines

Support vector machines, also called SVM, is a popular type of learning machine that use supervised learning models to analyze and classify data. One of the main areas of usage is to construct an optimal model that can distinguish new data points into one of two different classes. First presented by Vapnik and Lerner in 1963[6], the main concept is to construct a hyperplane as the separator of the two classes. The clever thing about hyperplanes is that it can be with ease be applied in higher dimensions as well which makes them ideal to general solutions.

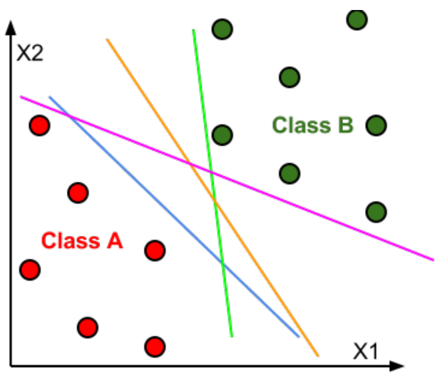


Figure 2.7: A set of labeled data points from the test data that are linearly separable. The data points have been separated by 4 hyperplanes which will classify them correctly.

When dealing with a linearly separable data set, there can be up to infinite ways to construct a hyperplane to correctly divide a data set into two classes as seen in figure 2.7. However, this is where SVM excel since the method presented by Vapnik will present an optimal hyperplane such as the one seen below in figure 2.8.

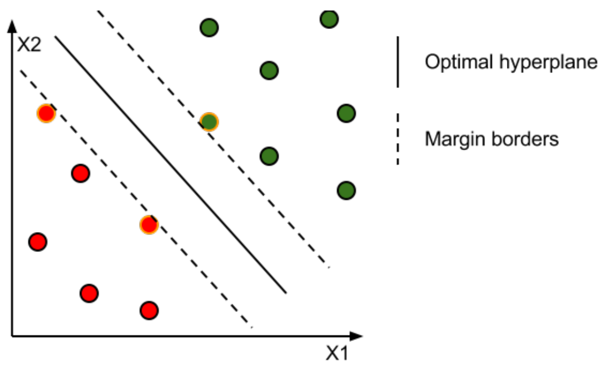


Figure 2.8: A optimal placement of the hyperplane that divides the two categories of the test data while maximizing the size of the margin.

It's called the optimal hyperplane in the sense that it's constructed so that the space between the point closest to the hyperplane and the hyperplane itself is maximized, this distance is called the margin. An maximized margin will increase the likelihood of a new data point to be classified correctly, even if subjected to some noise. A closer look at how the margin is calculated can be seen in figure 2.9. SVM will convert the data points to m-dimensional vectors, and hence the hyperplane constructed to divide them will be (m-1)-dimensional. Due to this, the data points that touch the maximized margin is called support vectors.

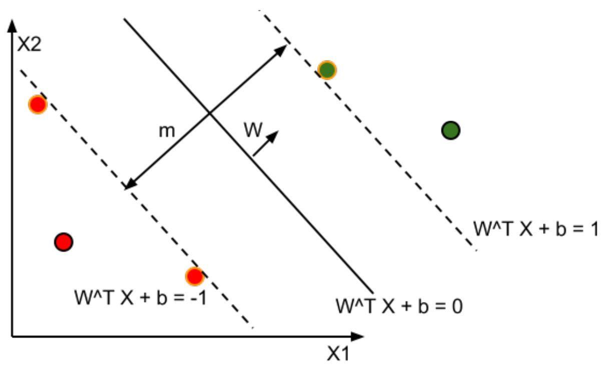


Figure 2.9: An enlarged view of Figure 2.8 above.

The mathematical problem to find the optimal hyperplane can be described as seen in equation 2.12.

$$Minimize\frac{1}{2}\left\|w\right\|^2 \text{subject to } y_i\left(w^Tx_i+b\right)\geq 1 \tag{2.12}$$

where $\{x_1, x_2, ...x_n\}$ is the data set and y_i, the class label $\{-1, 1\}$

This however is still a rather complex problem since it's dependent on w but with the help of the Lagrangian duality theorem, this problem can be simplified into a function of the support vectors instead. The new problem then becomes the following as seen in equation 2.13:

$$MaximizeL\left(\alpha\right)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}y_i\alpha_iy_j\alpha_jx_i^Tx_j \tag{2.13}$$

subjected to $\alpha_i \geq 0$ for i $= \{1, 2, ..., n\}$ and $\sum_{i=1}^{n}\alpha_iy_i = 0$

The optimal hyperplane can then be created from the solution using equation 2.13.

$$\sum_{i=1}\alpha_iy_ix_i \tag{2.14}$$

where the support vectors is the corresponding x_i and α is greater than zero.

Non-linear separable

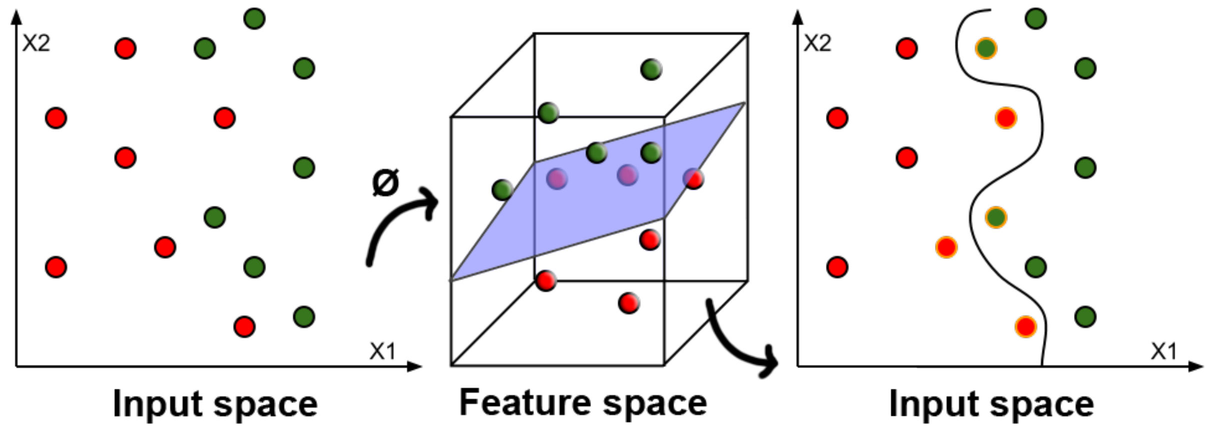


Figure 2.10: Left: A non-linearly separable data set in the input space. Middle: The same input in a feature space where a linear classification is found. Right: The input in the feature space is then transformed back into the input space where the support vectors found is displayed with a bright glow around them.

Not all data sets are linearly separable as seen in figure 2.10, in fact that is usually the case when dealing with real data. A clever trick is that the above algorithm to maximize the margin still works in higher dimensions, so if you're faced with non-linear separable points in the x space, you can do a nonlinear transformation into a much higher dimensional space and solve the problem there with the linear SVM method. When you have your solution, you can transform the linear hyperplane back into the x space where the linear hyperplane will be displayed as a "snake" that's separating the points. The support vectors in the higher dimension will be the ones in x space with a positive alpha.

Boser et al [7] suggested to apply a kernel trick [8] to the maximum margin problem to make the SVM capable transformations into infinite dimensions without having to pay the computational cost of the transformation or the cost for calculate the inner product.

The kernel trick is a method where you use a kernel function that represent $x^T x$ in a non-specified higher dimensional, this without having to pay the computational cost for the transformation.

Multiclass SVM

One of the drawbacks of the original SVM is that it can only divide the data into two classes. While it does this separation very well, data can usually belong to a high number of classes which would make the usage for this algorithm very limited if there wasn't a way around this. Two of the most popular ways around this is Winner Takes All-SVM(WTA-SVM) or the Max Win Voting (MWV-SVM) [9].

WTA-SVM will construct N number of classifiers where each classifier will test it's label against all of the other labels at the same time. The classifier with the higher output function will be the classifier that assign the class.

MWV-SVM on the other hand will construct one classifier for each possible pair of labels. The label that "wins" the most classifications will be the one that assign the class.

2.9 Summary

Here's a summary of the algorithms above.

Statistical probability & Bayes theorem
The fundamental statistics used in algorithms such as the Naive Bayes.

Naive Bayes classifier
The Naive Bayes classifier is a lightweight, fast and scalable algorithm that perform surprisingly well compared to the more complex models as long as the data set doesn't grow too big. A drawback is that the algorithm will run into a problem if you encounter data with a variable having zero probability since it will ruin your equation when multiplied with the other variables. This can however be fixed if you smooth the data beforehand by eliminating those values.

Rules and decisions
Inferring rudimentary rules is amongst the most lightweight of the classification algorithms that will perform unusally well given small specific problems. However, it's not suitable for classification problems where you have a high number of different attributes and values since it relies on a single rule. For text classification, there can be several thousands of different words that needs to be classified and hence the inferring rudimentary rules classifier will most likely always perform worse than classifying by chance. Yet, this algorithm is a good example that even complex problems can have a simple solution if you're allowed to introduce error.

Decision Table
A decision table is another way to present a set of rules in a more human readable format.

Decision Trees
This is a pretty straight forward way to construct a classifier since it initialize the tree with a single rule, and as you progress down in the tree by constantly making decisions what branch to continue downwards in, you'll eventually reach the end node being a leaf where the final classification answer for your instance lies. If this solution is implemented in an algorithm that can optimize the rules at each branch by removing paths that hold none, or very little valuable information and hence maximizing the number of instances you can classify in the final leaf without overfitting, then you can get pretty good generalized results.

Hidden Markov model
This is an interesting algorithm that works using the "black box" principle. You always know the initial input as well as what goes out at time t, but what actually happends inside is a bit of a mystery. Not knowing what goes on inside the model can for some applications be a big drawback.

k-means Clustering
An interesting clustering algorithm with a straight forward implementation that can also be used for classification as long as the k-number of classes is the same as the number of classes existing in the data set.

k-Nearest Neighbor
This algorithm is similar to k-means clustering since it classifies values based on the classification of the majority of the pre-labeled data points in the closest cluster with size k. It has good performance and due to it's lazy nature it doesn't require any time to generate a classification data model. The drawback of this is the memory as well as time complexity during the classification process quickly increases if you're using a big training-data set.

Support Vector Machines
An fairly old idea for a classification algorithm that was brought back in a revised version during the 90's with superb performance. It uses a clever way to separate data points into two classes using vectors. This can also be applied on classifications where you have more than two classes by running the algorithm several times, testing all the 1 vs 1 class-combinations and then finally classifying the data using the support vector machine classifier that had the highest performance.

CHAPTER 3

Data Analysis

This chapter discuss how a method to analyse data can be constructed.

3.1 Data classification using machine learning

The basic concept of data classification is to identify what kind of class a certain data point belongs to based on the features that the point possess. This is possible by comparing it to the features known for each of the possible classes, and then the data is classified as the class whose features match the most. An requirement is of course that information about the different classes have been collected beforehand. This is done by "training" a classifier model using a data set where the data points in it have been classified by hand into different classes.

Training the classifier

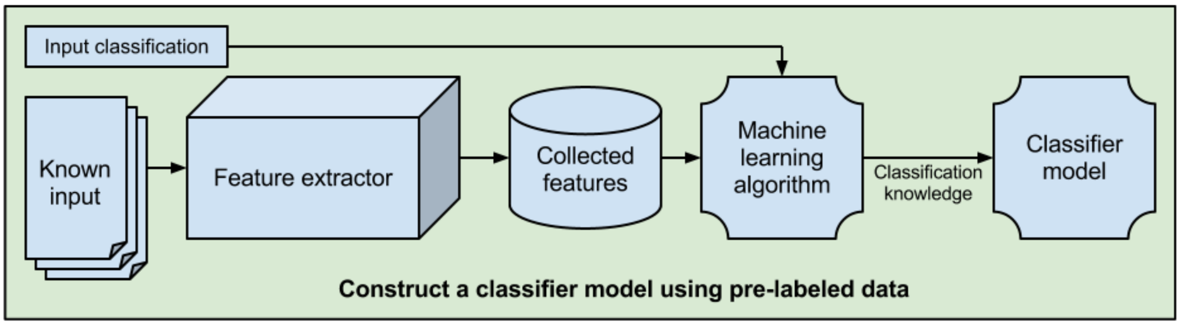


Figure 3.1: A model how to construct a classification model.

To train the classifier, first each of the pre-classified data points from the data set is run through a feature extractor that analyse the data and store a number of attributes that can identify this particular data point. Since it's known what kind of classification this data point has, the features collected is inserted into a machine learning algorithm which then tries to draw conclusions based on all the classified features collected and then constructs a model based on these that can be used to classify unlabeled data. A visual example of this can be seen in figure 3.1.

Testing data with the classifier

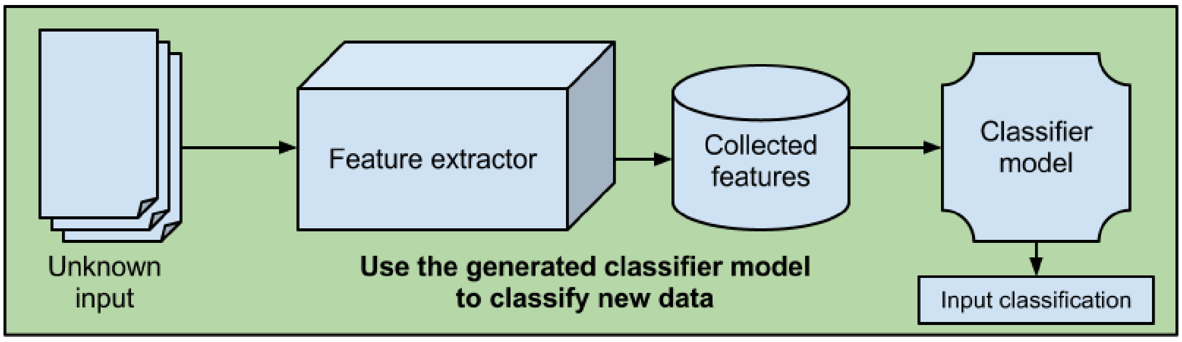


Figure 3.2: Unknown data is classified using a pre-generated classifier model.

The classifier model which was generated using the pre-labeled data can then be used to classify unknown input on the same principle. Each data point in the data set was run through a feature extractor which was then sent to the classifier model. These unlabeled features was then compared to the features in the classifier model and then classified as the class it was most similar to. A visual example of this can be seen in figure 3.2.

In theory, using a big data set to construct the classifier model will increase the performance when classifying new data since it would be easier to construct a more general model and hence finding a suitable match for your data. But this only true up to a specific point. Using too much data can cause the classifier to become very slow either when constructing the model since there's a lot of optimizations needed to make the model general, or when performing the actual classification of new data since there's just too many special cases to compare your data against.

The optimal size of the data set used to construct the classifier model is dependent on a number of things such as the size of the classification problem, the classifier algorithm used and the quality of the data set.

3.2 The method used to classify data

As seen in the problem definition in chapter 1.2 as well as the delimitation in chapter 1.4, the classification problem was about classifying structured data of persons that belonged to one of the following 4 classes: first name, last name, home address and city.

Large non-generated data sets with usable data in the form of personal data records was hard to find for free, however during this thesis I received access to two relatively large registries to analyse.

Data set A
A record over visitors to a local event in Örnsköldsvik, Sweden.

- 1324 complete entries of Swedish persons.
- Each entry was divided into 4 columns consisting of first name, last name, address and city name.
- This gave a total of 5296 instances.

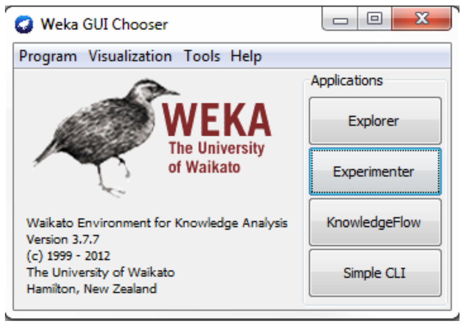
Data set B
A record over private persons in the area of Örnsköldsvik, Sweden.

- 20702 complete entries of Swedish persons.
- Each entry was divided into 4 columns consisting of first name, last name, address and city name.
- This gave a total of 82808 instances.

The reason why i chose to use the following two data sets was that they are based on actual real life data and that they are both relatively large. This gave a good starting point to be able to draw conclusions from the results generated in a later state. The idea was not to implement some of the algorithms discussed in chapter 2, and then try to evaluate their performance on these two data sets. The goal was to see how well the different algorithms performed, not just by comparing the number of correct classifications, but also by lookomg into the time required to construct the classification model depending on the size of the input data and number features used of as well as the time required to classify a data set using the generated classification model. It was entirely possible to implement these algorithms into classifiers from scratch since there was a lot of documentations describing them. But since this thesis was about finding a suitable algorithm rather than implementing them from scratch, the decision was made to spend the limited time and look into ways to test their performance using pre-existing solutions.

The test environment The tests performed in this thesis was made on a virtual machine running the Microsoft Windows Server 2012 Datacenter operative system with an Intel Xeon E5649 @ 2.53 GHz CPU running 8 cores and having access to 32 GB of RAM.

3.3 Waikato Environment for Knowledge Analysis



Waikato Environment for Knowledge Analysis
Version 3.7.7
(c) 1999 - 2012
The University of Waikato
Hamilton, New Zealand

Figure 3.3: The WEKA GUI chooser. From this it was possible to start one of the 3 different graphical interfaces or to run the fourth option which was terminal based.

There are many tools available for data mining and machine learning, but this thesis I chose to use the open source software suite WEKA which stands for Waikato Environment for Knowledge Analysis. The main reason why I selected to use WEKA was because of its versatility. WEKA is a popular tool used for data analysis, machine learning and predictive modeling that was developed by the University of Waikato in New Zealand using the programming language JAVA.

It can be seen as a big workbench for data analysis and machine learning where a majority of the most common algorithms used in data mining have been implemented and are ready to be used. The WEKA suite is divided into 3 different graphical interfaces as seen in figure 3.3, namely the Explorer, Experimenter and the Knowledge Flow. There's also a text based terminal mode where you can call the different methods with function calls directly. In the Explorer, the user gain quick access to all the features in WEKA and can freely analyse the data. The Experimenter on the other hand is more focused on setting up machine learning experiments where you can more easily compare different algorithms against each other. And lastly, in the Knowledge flow the user can use the same methods as in the Explorer, but instead of applying a certain function one at a time the user can set up complex flows that does the whole chain from reading the data to plotting the result in a graph. A big benefit of using WEKA is the fact that its easy to try different algorithms and filters and when a suitable method was found, it is quite easy to implement this into a stand alone java program since the source code is freely available.

Some of WEKAs main features are the following:

Data preprocessing - WEKA supports a couple of popular text file formats such as CSV, JSON and Matlab ASCII files to import data along with their own file format ARFF. They also have support to import data from databases through JDBC. Beside importing data, they have a wide collection of supervised as well as unsupervised filters to apply on your data to facilitate further analysis.

Data classification - A huge collection of algorithms have been implemented to perform classification on data sets. These include Bayesian algorithms, mathematical functions such as support vector machines, lazy classifiers implementing nearest-neighbour calculations, meta based algorithms as well as rule and tree-based classifiers.

Data clustering - A couple of algorithms for clustering exist such as variations of the k-mean method as well as density and hierarchical based clustering algorithms.

Attribute association - Methods to analyse data using association rule learners. Association rules can be seen as rules describing relations between attributes in a data set.

Attribute selection - Methods to evaluate which attribute contribute the most when predicting an outcome.

Data visualization - Depending on the methods used to analyse the data, this view can to plot data against suitable variables as well as give tools to analyze specific points further.

3.4 The WEKA file format

The main file format used in WEKA is their own called Attribute Relationship File Format, or ARFF in short. It is basically a normal text file with the following structure:

% This is a comment in WEKA
@RELATION `should_i_drink_coffee` %the data set name

% the attributes that occur in the data set is declared one after another
@ATTRIBUTE `thirsty` {yes, no} % nominal values
@ATTRIBUTE `cups_already_had` numeric % numeric values
@ATTRIBUTE `time date` "HH:mm:ss" % date and time
@ATTRIBUTE `coffee_type` string % string values
@ATTRIBUTE `drink` {yes, no} % the last attribute is the "label" class by default which is used when training the data to know if it was classified correctly

% @DATA means the start of the real data. Each row is one entry, and the values was comma separated. The values are entered in the order that the attributes are declared above. Also note that string and date values have to be quoted since they can include whitespaces.

@DATA
yes, 1, "08:05:22", 'expresso', yes
yes, 8, "23:23:00", 'regular', no
no, 3, "13:50:00", 'regular', no
. . . .
no, 0, "06:30:00", 'regular', yes

3.5 Convert data to the ARFF format

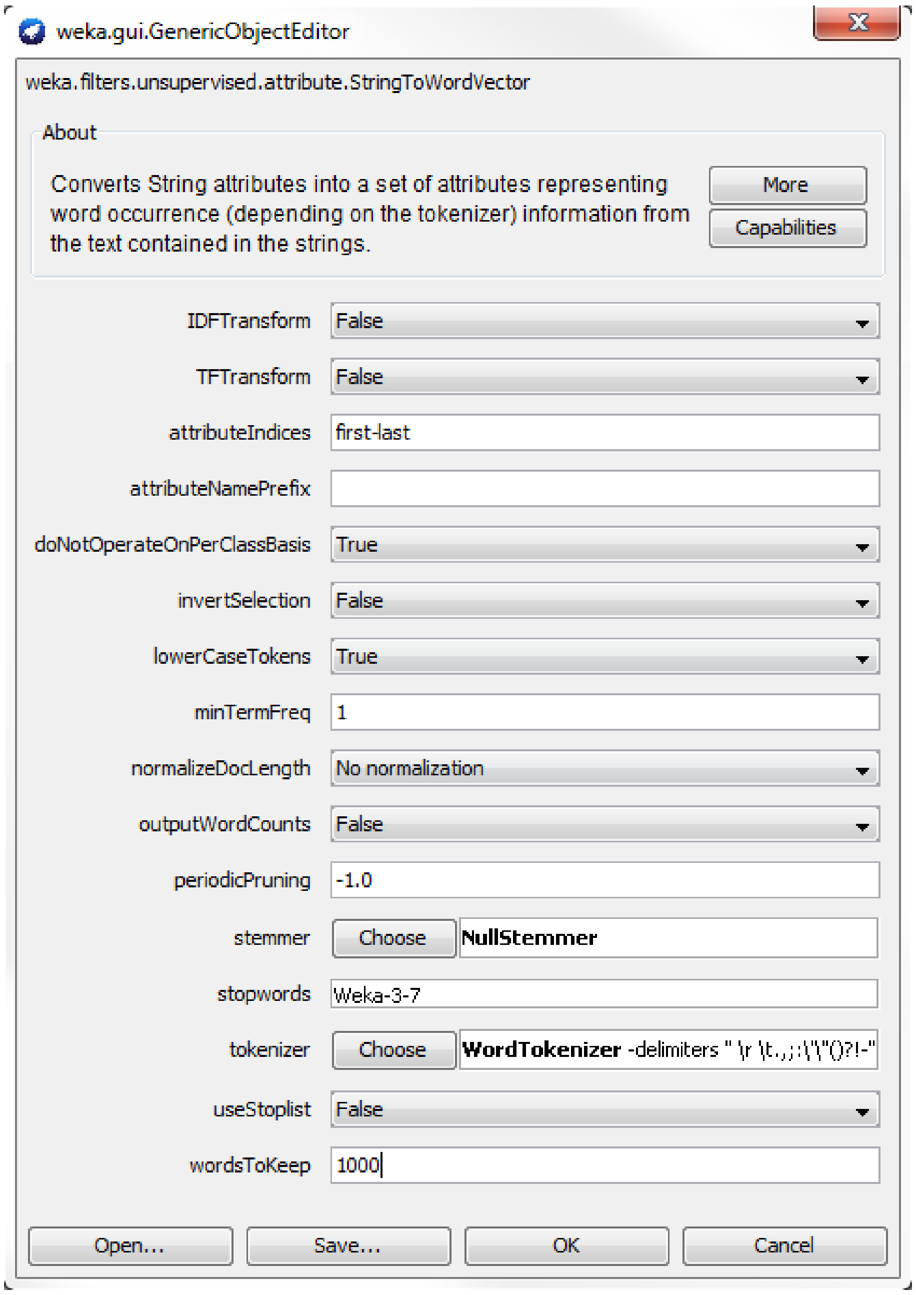
The data in data set A and B had to be cleaned manually where entries with missing or very strange values was removed along with non-interesting columns. The remaining data was then divided into first name, last name, home address and city before being labeled and converted into the WEKA .arff format using the following structure.

```
@relation mixed_labels
@attribute data string
@attribute conclusion {firstname,lastname,address,city}

@data
'Lilian',firstname
'Aalto',lastname
'Vesslevägen 17',address
'ÖRNSKÖLDSVIK',city
'Asyf',firstname
'Abassy',lastname
'Sörbyvägen',address
'ÖRNSKÖLDSVIK',city
'Muhammad',firstname
'Abassy',lastname
'Murargränd 6 A',address
'ORSA',city
...
'Sven',firstname
'Östman',lastname
'Husbyn 142',address
'HUSUM',city
```

3.6 Preprocessing data

A majority of the classifiers are built to analyze numeric and nominal values. This was a problem since the data in the data sets consist of string attributes only, so the method used had to transform this into something useful. Lucky enough, WEKA had a filter for doing just this. Using the StringToWordVector-filter, it was possible to convert each string entry in our data into a set of attributes describing the string as well as information about its occurrence in the data set.



weka.filters.unsupervised.attribute.StringToWordVector

About

Converts String attributes into a set of attributes representing word occurrence (depending on the tokenizer) information from the text contained in the strings.

IDFTransform	False
TFTransform	False
attributeIndices	first-last
attributeNamePrefix	
doNotOperateOnPerClassBasis	True
invertSelection	False
lowerCaseTokens	True
minTermFreq	1
normalizeDocLength	No normalization
outputWordCounts	False
periodicPruning	-1.0
stemmer	NullStemmer
stopwords	Weka-3-7
tokenizer	WordTokenizer -delimiters " \r \t.,;:\'\"()?!-"
useStoplist	False
wordsToKeep	1000

Figure 3.4: The settings used when analysing the data. Note that different values for 'wordsTo-Keep' was tested to find a suitable efficiency.

The StringToWordVector-filter had 16 different settings as seen in figure 3.4 that you could change to weak your classification. Here's a few of them:

- IDFT/TF-transform: These settings will change the way word frequencies are calculated depending on the number of documents you are analysing. Since I was only interested in a single data set, I set these to false.
- Attribute indices: This specifies what data range to use. Everything in this case.
- Do not operate on per class basis, hwn set to yes the Words to keep value will mean the amount of words in total, instead of per type of class as default. I set this to "true" due to a bug in WEKA that made "false" mean the same.
- Lower case tokens: Transforms all strings in the data to lower case. This was used to generalize the data.
- Min term frequency: With this its possible tweak the solution by focusing on popular words that repeat over and over in your data set. Since I didn't know how future data sets used are constructed, I left this at one.
- Normalize doc length: This can be used if you want to regulate the text length in those cases when you're analysing multiple documents at once.
- Output word counts: Set this to yes if you are interested in the word count of a certain word when classifying a data set instead of just knowing if it's present or not.
- Stop words: With this you can set up a dictionary of words to be discarded. This is especially important when classifying unstructured texts to remove words such as "the" that exists in all documents so it doesn't end up being a factor when classifying a text later on. Since I was analysing structured text in column form, this was disabled.
- Tokenizer: This comes in handy when you're dealing with fulltext classifications since theres a lot of special characters that doesn't bring any value to the classification. What it does is that it separates the string into an additional word for each of these characters. In this case when dealing with personal data records, I was only interested in the "textual" words in the string so any characters such as punctuation, commas etc was discarded.
- Words to keep: From here on also called WTK. It's the number of words to store in the dictionary. There's a tradeoff for the size of the dictionary since a big dictionary will increase the performance of your classification but in the other hand the running time to classify something will increase as well.

3.7 The selected WEKA algorithms

These are the selected WEKA algorithms I chose to analyze since whey where implemented in the WEKA suite and ready to use directly. The decision to use the following algorithms was based on the efficiencies seen in the reports I read about data classification. I tried to pick at least one type of classifier from each of the major classifier groups and ended up with the following:

- Based on the Bayes Theorem - Naive Bayes
- Function oriented - SMO
- Lazy learners - IBk
- Rules - DecisionTable & JRip
- Trees - J48 & REPTree

Here's a small description of them as well as references for further reading.

Naive Bayes - A classic implementation of the Naive Bayes algorithm. For more information on this Naive Bayes implementation, see George H, John and Pat Langley (1995)[10].

SMO - A implementation of John Platt's[11] sequential minimal optimization algorithm that uses support vectors. Has built in support to handle multiple classes using pairwise classification.

IBk - A classic implementation of the k-nearest neighbours classifier. Note that I settled with using k=1 for my analysis since the model didn't get any additional performance by raising this. Note that this algorithm is of the lazy type which does all the calculations from memory while classifying, so by having a low k the running time decreased to a somewhat reasonable level which the results later on will show. This implementation is based on Aha and Kibler (1991)[12].

J48 - A open source implementation of the C4.5 algorithm[13] that builds a decision tree using information entropy. That means that when building the decision tree, C4.5 will at each node select the attribute that most successfully splits its set of samples seen to the difference in entropy that the selected subtree generates.

REPTree - A quick decision tree classifier that use information gain as well as pruning to classify the data. Implemented by Eibe Frank [14].

DecisionTable - A classifier that uses a simple decision table. Based on the work of Ron Kohavi (1995), stating that decision table classifiers can in fact perform well[15].

JRip - This is a propositional rule learner that will start with a big ruleset and then keep simplifying the rule by removing conditions that creates the largest reduction of error. The implementation is based on the report of William W. Cohen on fast effective rule induction[16].

3.8 The testbench constructed in WEKA

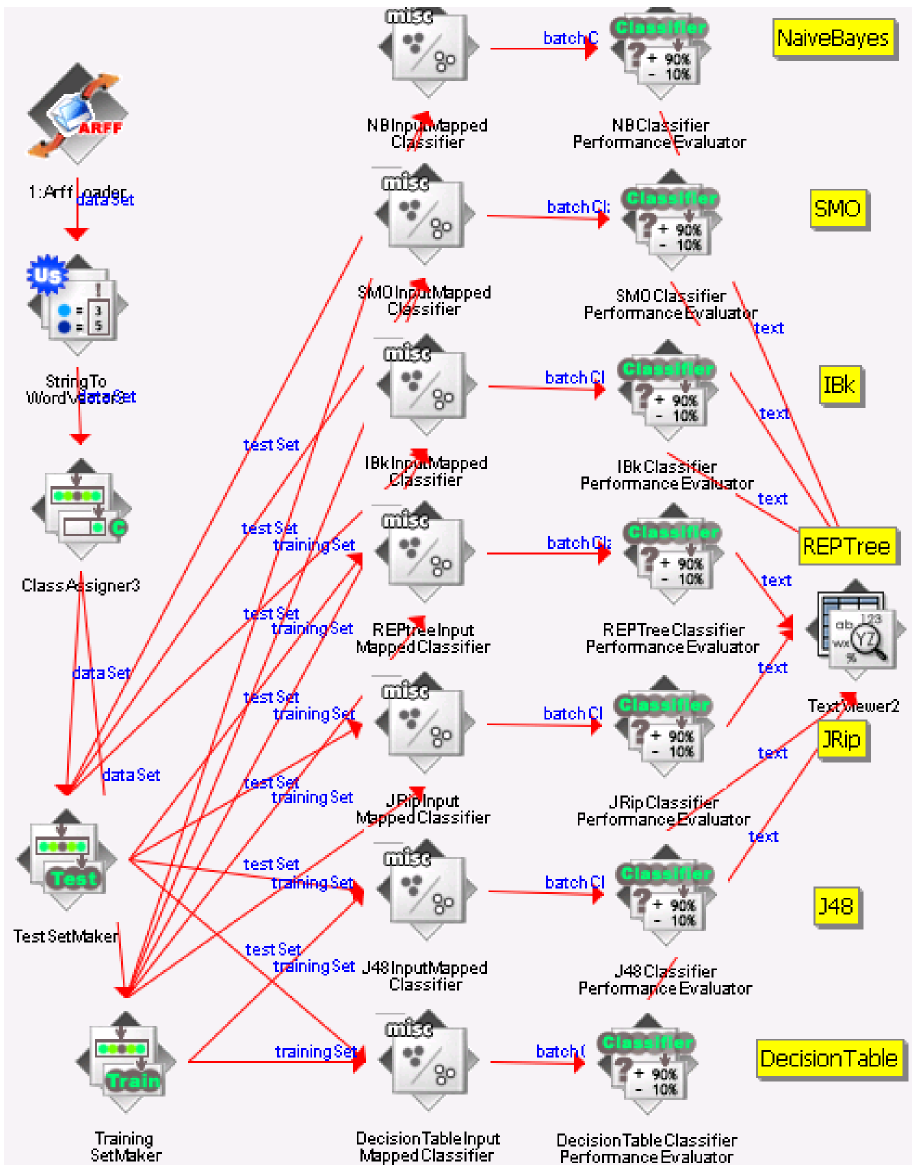

1:ArffLoader
dataSet
StringToWordVector
dataSet
ClassAssigner3
dataSet
dataSet
TestSetMaker
TrainingSetMaker
test Set
trainingSet
NBInputMappedClassifier
SMOInputMappedClassifier
IBkInputMappedClassifier
REPTreeInputMappedClassifier
JRipInputMappedClassifier
J48InputMappedClassifier
DecisionTableInputMappedClassifier
batchCl
NBClassifierPerformanceEvaluator
SMOClassifierPerformanceEvaluator
IBkClassifierPerformanceEvaluator
REPTreeClassifierPerformanceEvaluator
JRipClassifierPerformanceEvaluator
J48ClassifierPerformanceEvaluator
DecisionTableClassifierPerformanceEvaluator
text
TextViewer2
NaiveBayes
SMO
IBk
REPTree
JRip
J48
DecisionTable

Figure 3.5: The knowledge flow model used to test the classifiers in WEKA.

The testbench constructed in WEKA's knowledge flow can be seen in figure 3.5. To summarize this model, the data set that's converted to the ARFF file format was loaded into WEKA using the Arff loader and then the data was converted using the StringToWordVector filter into a set of attributes that could be analysed. The next step was to tell the model to use the conclusion attribute seen in chapter 3.5 as the attribute that dictates what kind of class each data string belongs to so it would know if the classification was correct or not.

The data in vector form was then loaded up into the TrainingSetMaker which feeds the data to each classifier which in hand constructs a classification model. When the classification model was completed, the data to be classified went through the TestSetMaker, passing the InputMappedClassifier and was then classified in the ClassifierPerformanceEvaluator using the generated model. The result from this classification was then displayed in the TextViewer.

By using this model, the classification was performed in all seven classifiers in parallel using the same data set with the same value of WTK in the StringToWordVector-filter settings. This saved a lot of time since I only had to initialize the algorithms once, instead of testing the algorithms one by one.

3.9 Classifier settings

3.9.1 Optimizing the classifier parameters

I tried to optimize the classifier performance by finding a more suitable input value for the classifiers using WEKA's CVParameterSelection-filter. It works as such that after you've loaded a data set as usual, you run CVParamterSelection with the classifier to be optimized selected along with the input value and what interval to be tested.

For example, to try to optimize the -C parameter for the SMO classifier by testing the following C values {1,2,3,...,9} can look like this:

=== Run information ===

```
Scheme: weka.classifiers.meta.CVParameterSelection -P ''C 1.0
    9.0 9.0'' -X 10 -S 1 -W weka.classifiers.functions.SMO -- -C
    1.0 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K ''weka.
    classifiers.functions.supportVector.PolyKernel -C 250007 -E
    1.0''
```

=== Result ===

Cross-validated Parameter selection.
Classifier: weka.classifiers.functions.SMO

Cross-validation Parameter: '-C' ranged from 1.0 to 8.0 with 8.0 steps

```
Classifier Options: -C 1 -L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K
    ''weka.classifiers.functions.supportVector.PolyKernel'' -C
    250007 -E 1.0
```

And the classifier option in the result will give you the C parameter that the CVParameterSelection-filter suggests that you use. In this case -C = 1, which is the default value. After testing different parameters with this filter on the selected classifiers, I decided to stick to the default values since the changes in the result I was seeing was either none or extremely small. Perhaps parameter optimiziation works better for other types of classifications and data sets then the ones I was using.

3.9.2 The classifier options used

Here's the default options used for each classifier tested. These values remained the same for all the different tests performed in this thesis.

DecisionTable

```
Options: -I -trim -W weka.classifiers.rules.DecisionTable -- -X
    1 -S ''weka.attributeSelection.BestFirst -D 1 -N 5''
```

IBk

```
Options: -I -trim -W weka.classifiers.lazy.IBk -- -K 1 -W 0 -A '
    'weka.core.neighboursearch.LinearNNSearch -A \''weka.core.
    EuclideanDistance -R first-last\''''
```

J48

```
Options: -I -trim -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
```

JRip

```
Options: -I -trim -W weka.classifiers.rules.JRip -- -F 3 -N 2.0
    -O 2 -S 1
```

Naive Bayes

```
Options: -I -trim -W weka.classifiers.bayes.NaiveBayes --
```

REPTree

```
Options: -I -trim -W weka.classifiers.trees.REPTree -- -M 2 -V
    0.0010 -N 3 -S 1 -L -1 -I 0.0
```

SMO

```
Options: -I -trim -W weka.classifiers.functions.SMO -- -C 1.0 -L
     0.0010 -P 1.0E-12 -N 0 -V -1 -W 1 -K ''weka.classifiers.
    functions.supportVector.PolyKernel -C 250007 -E 1.0''
```

CHAPTER 4

Evaluation

This chapter discuss how different algorithms perform when classifying data.

A note about the performance seen in this chapter!
The time seen to build and classify the tests below should not be seen as a normal value compared to if the same classifier had been tested by itself, but rather as a ratio compared to the other classifiers for the same test. This is because testing each classifier for a specific data set one at a time would require an unreasonable long time(constantly running for weeks) as the results will display, and that is why all the classification algorithms was run in parallel for each test performed. Even though the parallel classifications was run on 8 CPU cores, the actual running time will most likely be shorter than shown if a specific classifier would have been tested by itself.

4.1 The initial performance analysis using data set A

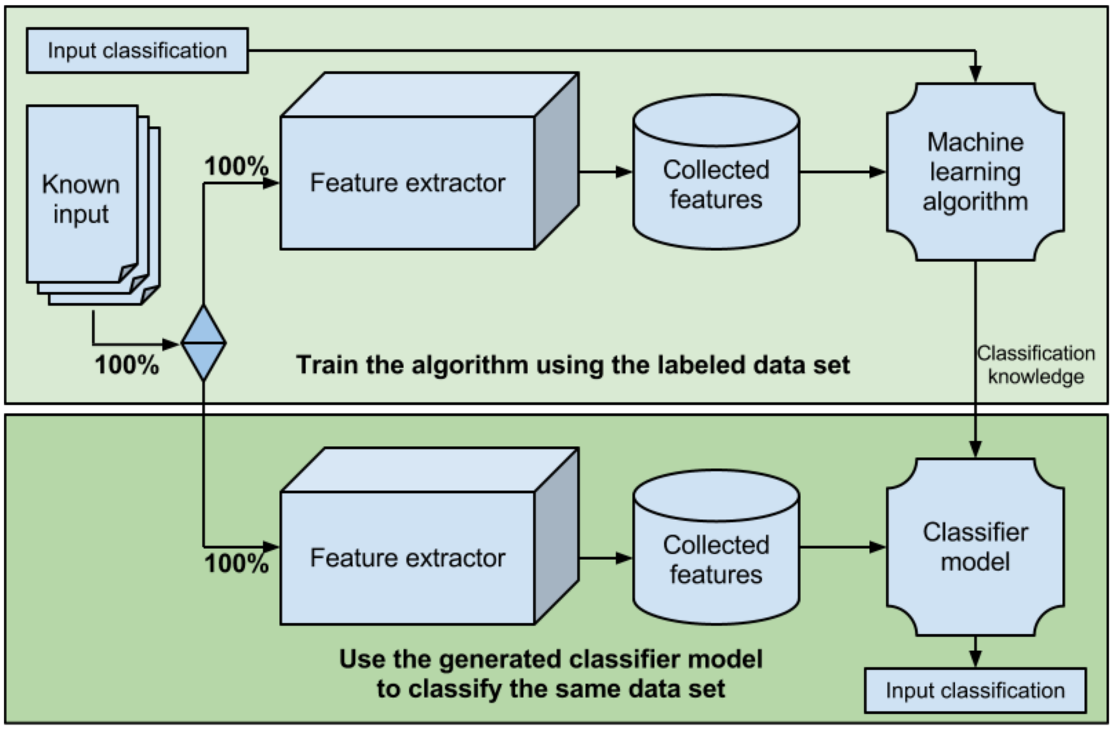
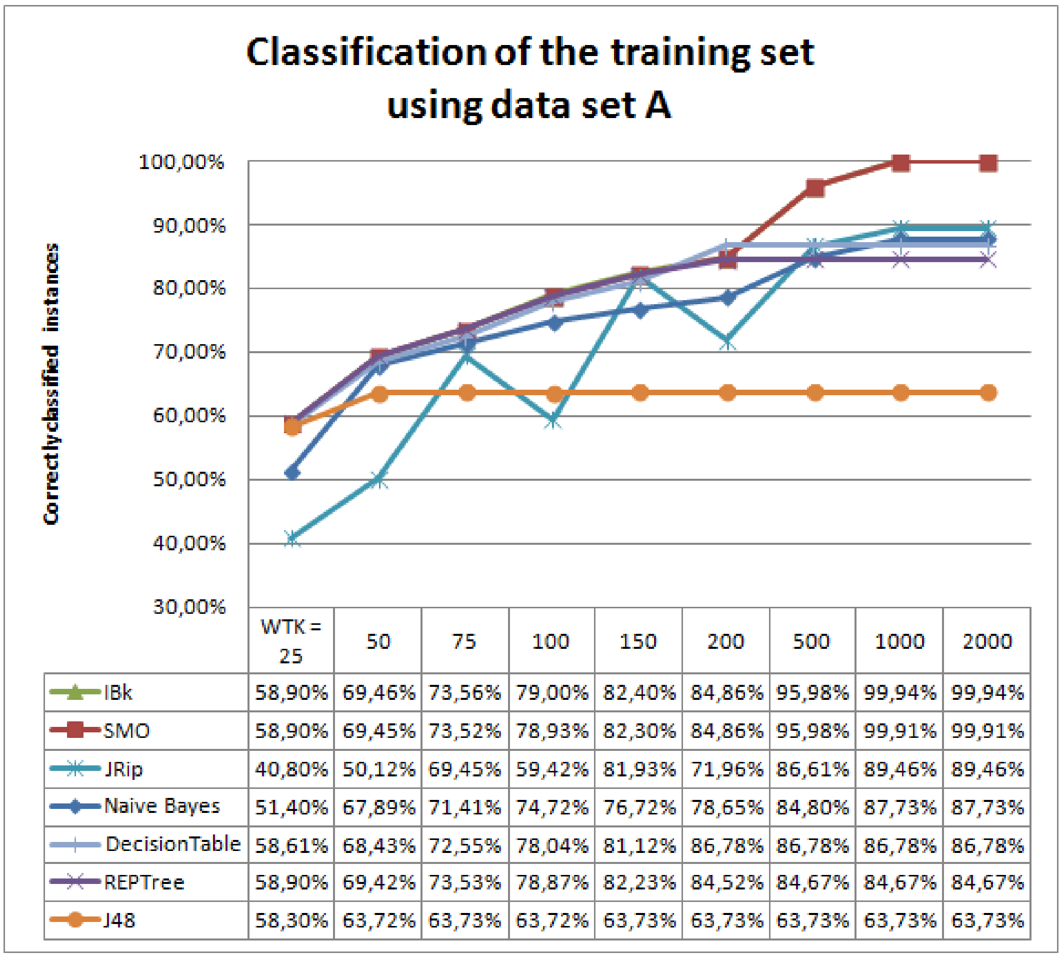


Figure 4.1: A model how the analysis was performed.

To get an initial indicator of the performance for the selected algorithms, each algorithm was trained on the data set and then the generated model was used to classify the same data set once again. The method used for this is described in section 3.1, but this time training the classifier and testing the classifier on data was done in the same process as can be seen in figure 4.1.

The optimal result from this test would be to correctly classify 100% of the data since our model was trained for exaclt this input. This was however not the case in practice since most algorithms tend to optimize their model using generalization to be able to classify similar data as well. 100% was also highly unlikely since the StringToWordVector filter only keept a limited number of words from the training set, so it was always possible to miss some unique strings that might be spreading at some direction when training the model. Since this was the first analysis of the data made in this thesis, a suitable value for the "Words to keep"-setting in the StringToWordVector filter had to be found.

Test mode: Evaluate the algorithms on the same data set that was used to train the model
Training data: Data set A with the StringToWordVector-filter applied.
Testing data: Data set A with the StringToWordVector-filter applied.
WTK: Testing values from 25 up to 2000 to evaluate performance of the classifiers.

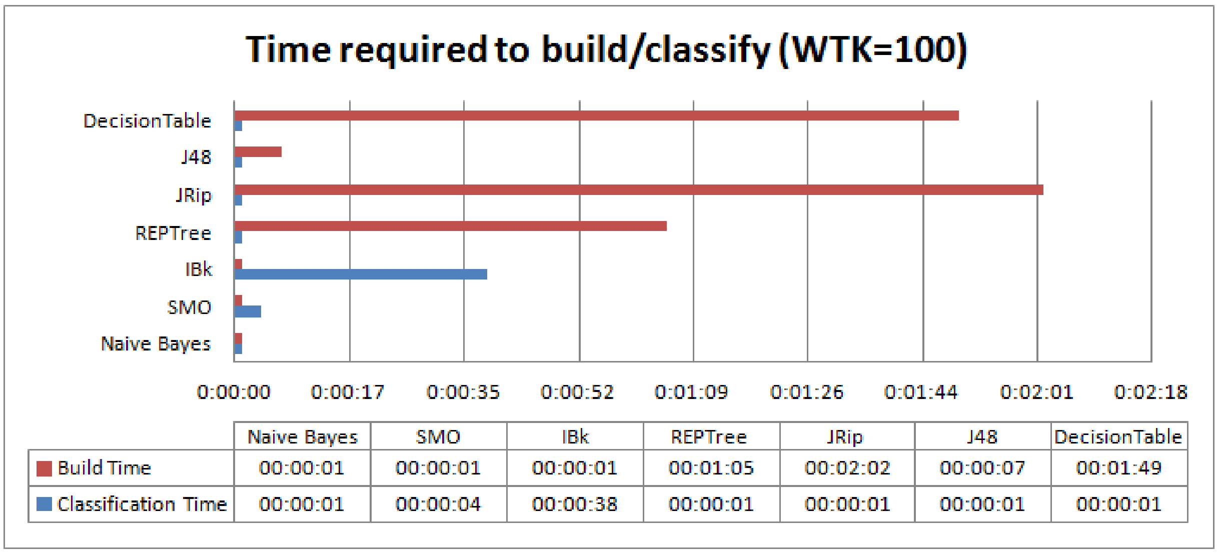


	WTK = 25	50	75	100	150	200	500	1000	2000
IBk	58,90%	69,46%	73,56%	79,00%	82,40%	84,86%	95,98%	99,94%	99,94%
SMO	58,90%	69,45%	73,52%	78,93%	82,30%	84,86%	95,98%	99,91%	99,91%
JRip	40,80%	50,12%	69,45%	59,42%	81,93%	71,96%	86,61%	89,46%	89,46%
Naive Bayes	51,40%	67,89%	71,41%	74,72%	76,72%	78,65%	84,80%	87,73%	87,73%
DecisionTable	58,61%	68,43%	72,55%	78,04%	81,12%	86,78%	86,78%	86,78%	86,78%
REPTree	58,90%	69,42%	73,53%	78,87%	82,23%	84,52%	84,67%	84,67%	84,67%
J48	58,30%	63,72%	63,73%	63,72%	63,73%	63,73%	63,73%	63,73%	63,73%

Figure 4.2: The result from the initial classification test.

As seen in figure 4.2, the IBk and SMO algorithms was performing very well and started to peak close to 100%. The decision tree based algorithm REPTree as well as the DecisionTable classifier also performed well but reached their peak around 200 WTK where Naive Bayes and JRip soon meet up. The rule based algorithm JRip fluctuated quite much before leveling out around 90%.

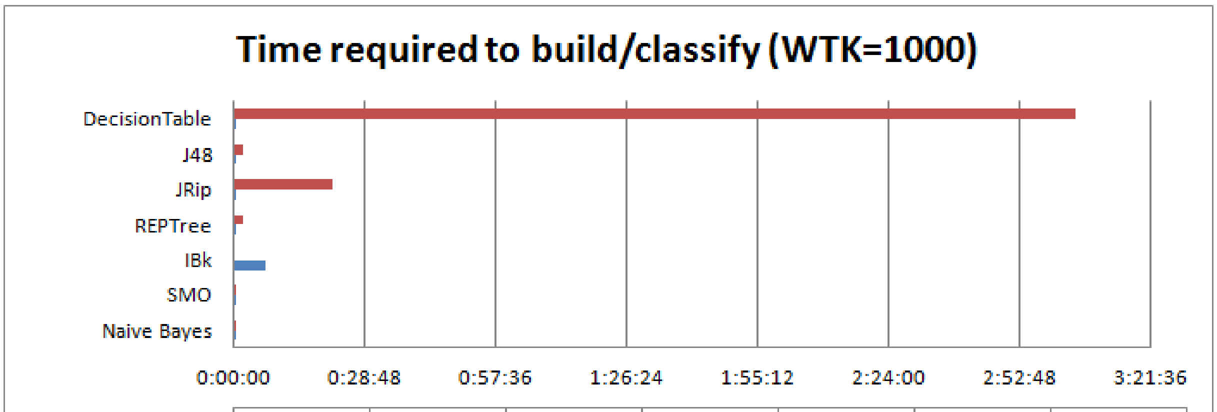
An interesting thing was that the J48 classifier peaked already at 50 WTK and did not improve by being subjected to a higher number of words. By looking at the model it produces which can be seen in the appendix(A2, WTK = 50), it constructs a tree with 61 nodes and the rules doesn't change for higher WTK values then 50. I tried to optimize the results from the J48 classifier by running it with different settings for this data set, but was unable to get better results.



	Naive Bayes	SMO	IBk	REPTree	JRip	J48	DecisionTable
🟥 Build Time	00:00:01	00:00:01	00:00:01	00:01:05	00:02:02	00:00:07	00:01:49
🟦 Classification Time	00:00:01	00:00:04	00:00:38	00:00:01	00:00:01	00:00:01	00:00:01

Figure 4.3: The time required to build the classification model as well as the time required to analyse the test data using the generated model. This was measured using WTK = 100.

As seen in figure 4.3, most of the classifier algorithms was able to classify the data set very quickly after creating the classifier model. The reason why IBk was so slow at classifying compared to the other algorithms was because it didn't rely on a pre-built classification model, but instead keept the whole labeled data set used to train the classifiers in the memory and made direct calculations for each comparison as described earlier. This was also the main reason why this algorithm performed so well when classifying the data set it was "trained" on since close to no optimizations was made.



	Naive Bayes	SMO	IBk	REPTree	JRip	J48	DecisionTable
🟥 Build Time	00:00:05	00:00:17	00:00:00	00:01:47	00:21:32	00:01:51	03:04:52
🟦 Classification Time	00:00:19	00:00:04	00:06:41	00:00:03	00:00:02	00:00:02	00:00:01

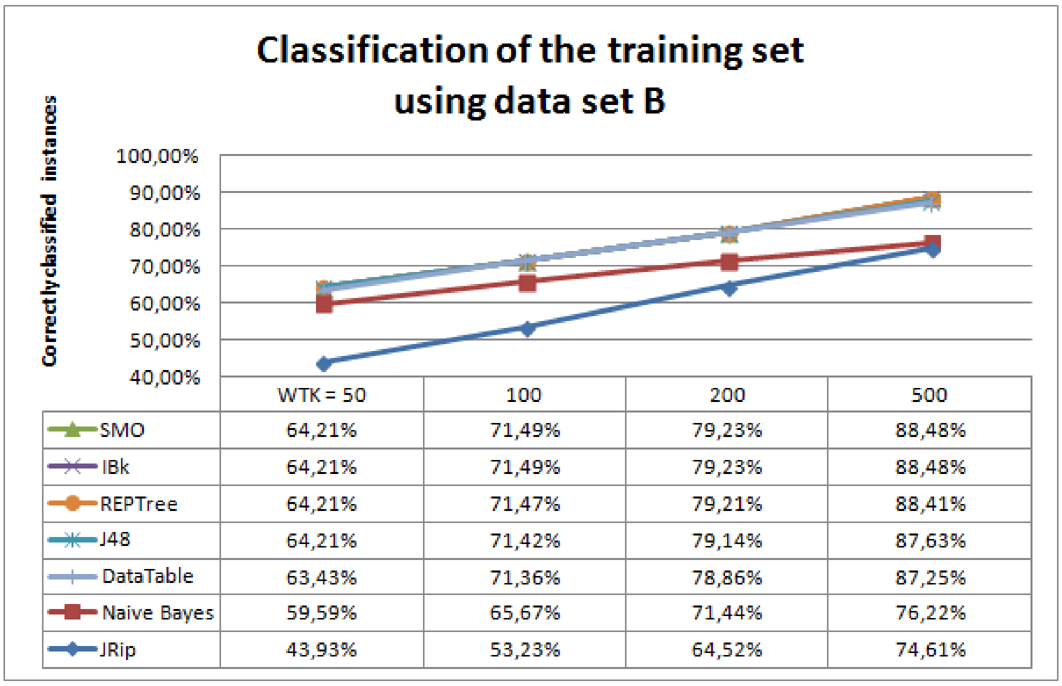
Figure 4.4: The time required to build the classification model as well as the time required to analyse the test data using the generated model. This was measured using WTK = 1000.

As seen in figure 4.4, the algorithms was able to classify the data very quickly once the classifier model had been built. The IBk drawback of a slow classification was pretty bad back when WTK was 100, and it only got worse as WTK increased. Another thing this graph shows was that the time required to construct a classification model for the DecisionTable gets almost exponentially worse when increasing the number of words to keep. Since the DecisionTable performance peaked at WTK=200, this was also an entirely unnecessary to run it on values this high.

4.2 The initial performance analysis using data set B

The same kind of test performed in chapter 4.1 was used on data set B as well.

Test mode: Evaluate the algorithms on the same data set that was used to train the model
Training data: Data set B with the StringToWordVector-filter applied.
Testing data: Data set B with the StringToWordVector-filter applied.
WTK: The following WTK values of 50, 100, 200 & 500 was used to evaluate the performance of the classifiers.

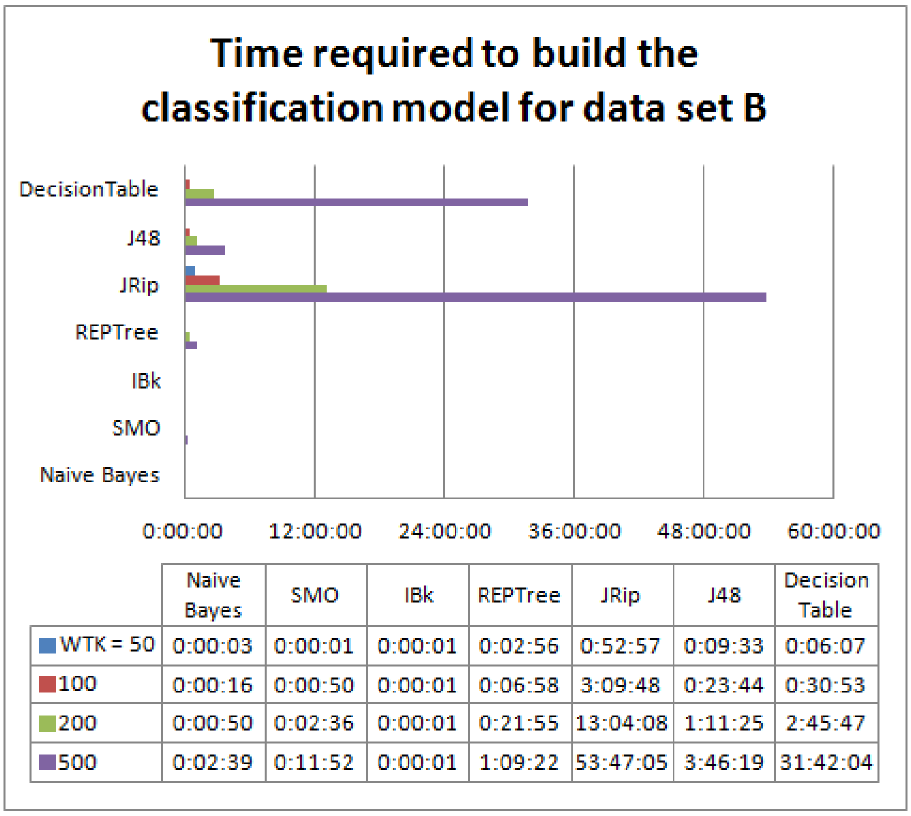


	WTK = 50	100	200	500
SMO	64,21%	71,49%	79,23%	88,48%
IBk	64,21%	71,49%	79,23%	88,48%
REPTree	64,21%	71,47%	79,21%	88,41%
J48	64,21%	71,42%	79,14%	87,63%
DataTable	63,43%	71,36%	78,86%	87,25%
Naive Bayes	59,59%	65,67%	71,44%	76,22%
JRip	43,93%	53,23%	64,52%	74,61%

Figure 4.5: The performance of the algorithms on data set B.

Since data set B was a pretty large data set with 82808 instances, only 4 tests was performed using a WTK of 50, 100, 200 and 500.

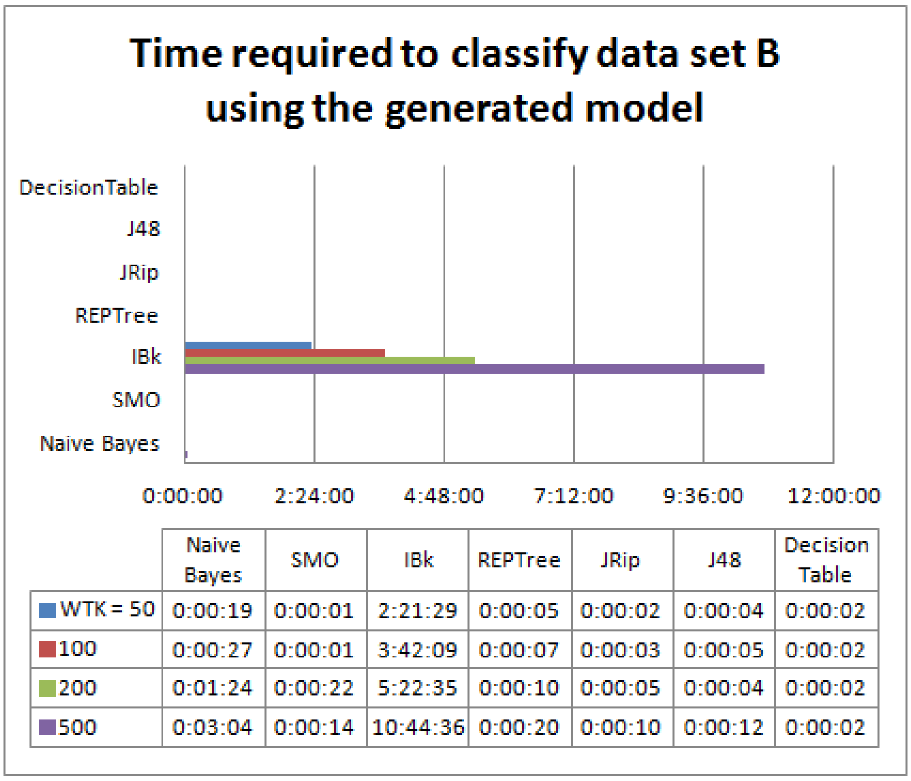
As seen in figure 4.5, five of the seven algorithms reached a classification success rate above 87% which was around how all except the J48 classifier performed in chapter 4.1 using a WTK value of 500. It's also worth noticing that J48 was performing considerably better using this data set then data set A used in chapter 4.1.



	Naive Bayes	SMO	IBk	REPTree	JRip	J48	Decision Table
🟦 WTK = 50	0:00:03	0:00:01	0:00:01	0:02:56	0:52:57	0:09:33	0:06:07
🟥 100	0:00:16	0:00:50	0:00:01	0:06:58	3:09:48	0:23:44	0:30:53
🟩 200	0:00:50	0:02:36	0:00:01	0:21:55	13:04:08	1:11:25	2:45:47
🟪 500	0:02:39	0:11:52	0:00:01	1:09:22	53:47:05	3:46:19	31:42:04

Figure 4.6: The time it took to construct the classification model using data set B.

As seen in figure 4.6, the time required to construct a classification model for both the DecisionTable and the JRip classifier became exponentially worse as the WTK value increased. This was however to be expected since they both where rule based algorithms and as the number of possible values increases, so must the number of rules to be able to keep up.



	Naive Bayes	SMO	IBk	REPTree	JRip	J48	Decision Table
🟦 WTK = 50	0:00:19	0:00:01	2:21:29	0:00:05	0:00:02	0:00:04	0:00:02
🟥 100	0:00:27	0:00:01	3:42:09	0:00:07	0:00:03	0:00:05	0:00:02
🟩 200	0:01:24	0:00:22	5:22:35	0:00:10	0:00:05	0:00:04	0:00:02
🟪 500	0:03:04	0:00:14	10:44:36	0:00:20	0:00:10	0:00:12	0:00:02

Figure 4.7: The time it took to classify data set B using the generated model

As expected, IBk spendt all the time during the classification state instead of when constructing the classifier model.

4.3 Further analysis using 10-fold cross validation

To get a better picture how the algorithms performed, a 10-fold data split was constructed as seen in figure 4.8. This means the classifier constructed 10 identical instances of the data set and then split the data in each of these instances in 10 % for training and 90% for testing. Each of these 10 instances was trained/tested with a unique split. In the end, the result from each of these instances was combined into a final result. By using this kind of 10-fold cross validation split, it was possible to get a pretty realistic result since 10 combinations of 10% of the data have been used to classify the data. A more graphical explanation of a 10-Fold split can be seen in figure 4.9.

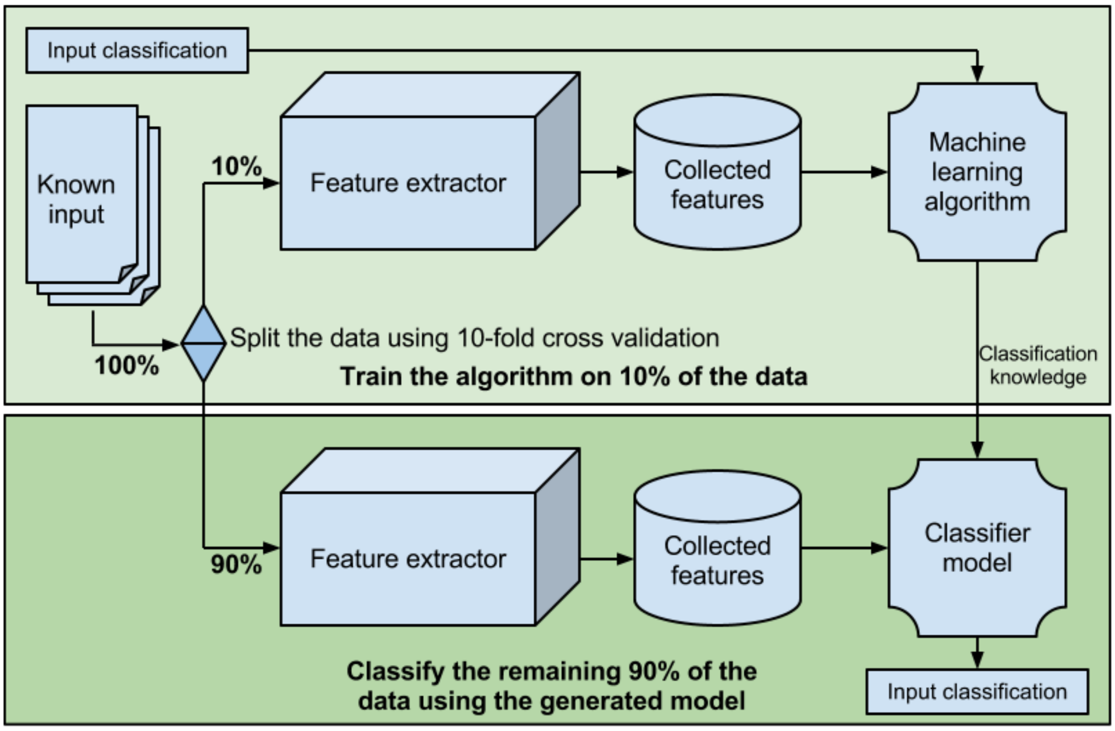

Input classification
Known input
10%
Feature extractor
Collected features
Machine learning algorithm
Split the data using 10-fold cross validation
100%
Train the algorithm on 10% of the data
Classification knowledge
90%
Feature extractor
Collected features
Classifier model
Classify the remaining 90% of the data using the generated model
Input classification

Figure 4.8: The model for analysing the algorithms using a 10-fold data split.

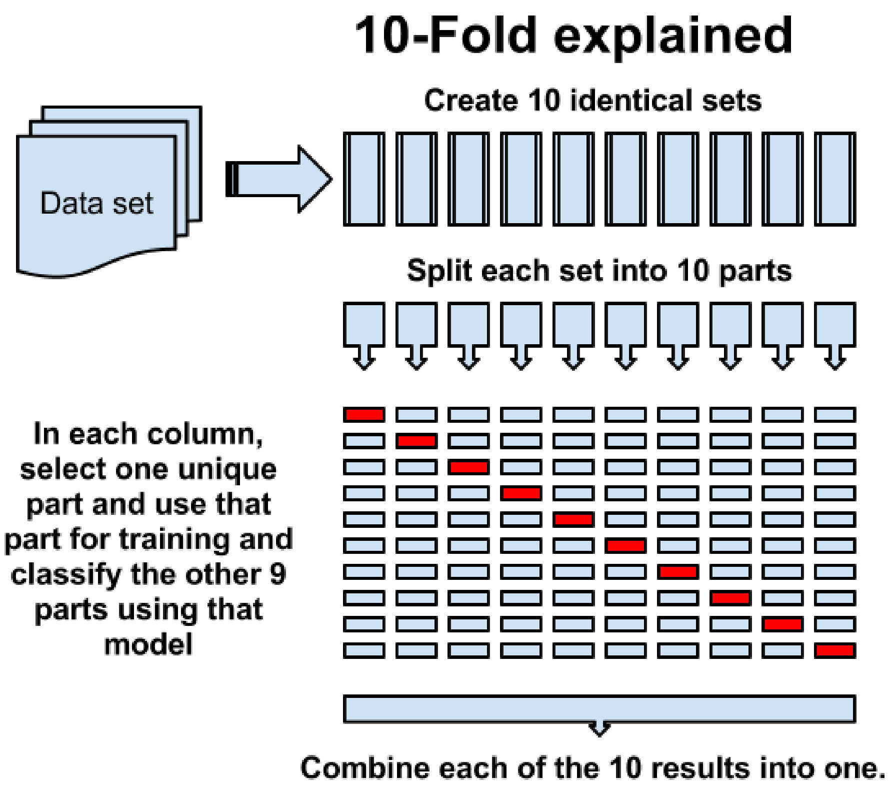
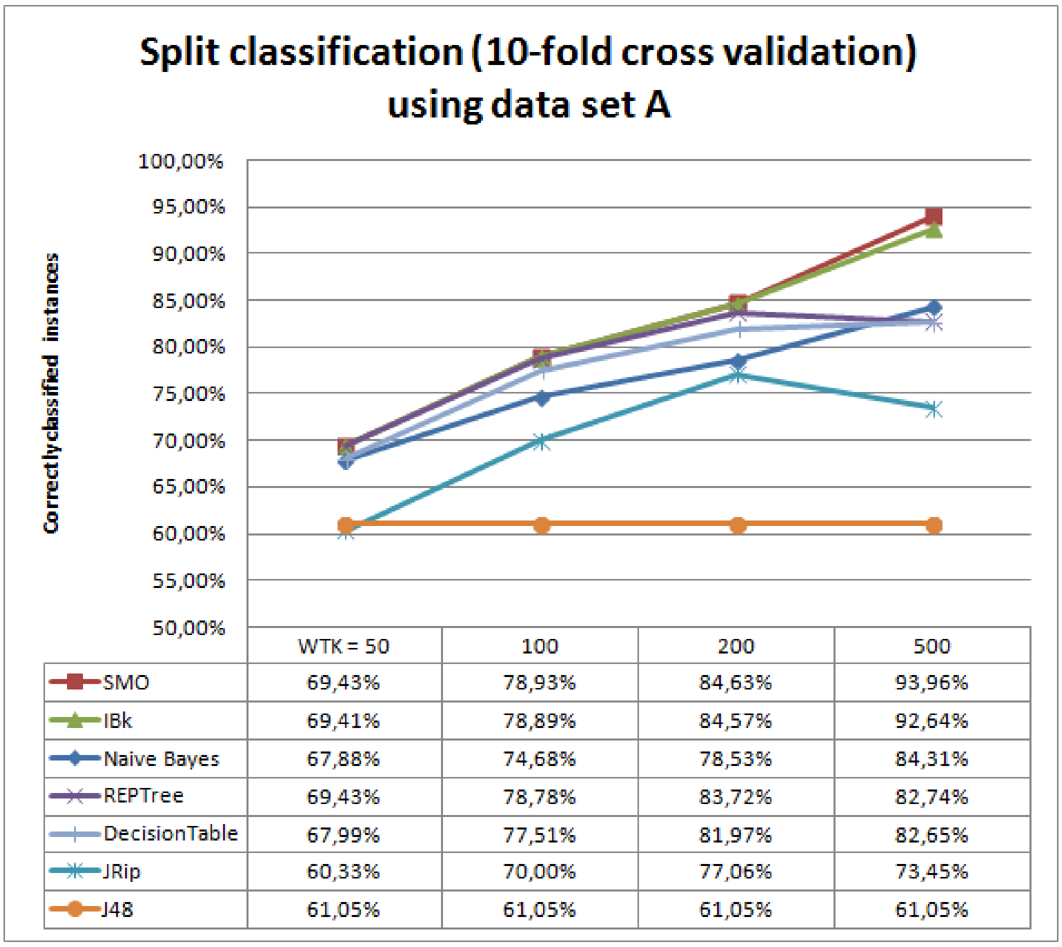

10-Fold explained
Create 10 identical sets
Data set
Split each set into 10 parts
In each column, select one unique part and use that part for training and classify the other 9 parts using that model
Combine each of the 10 results into one.

Figure 4.9: A model how the actual 10-fold is constructed.

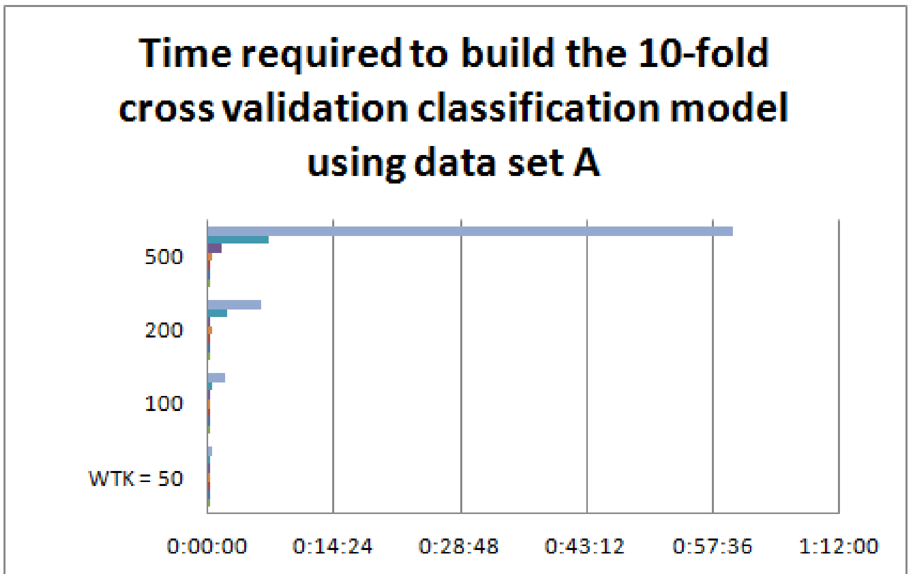
Test mode: Evaluate data set A using a 10-fold cross validation
Training data: Data set A.
Testing data: Data set A.
WTK: The following WTK values of 50, 100, 200 & 500 was tested.



	WTK = 50	100	200	500
SMO	69,43%	78,93%	84,63%	93,96%
IBk	69,41%	78,89%	84,57%	92,64%
Naive Bayes	67,88%	74,68%	78,53%	84,31%
REPTree	69,43%	78,78%	83,72%	82,74%
DecisionTable	67,99%	77,51%	81,97%	82,65%
JRip	60,33%	70,00%	77,06%	73,45%
J48	61,05%	61,05%	61,05%	61,05%

Figure 4.10: The result of the classification when using a 10-fold data split.

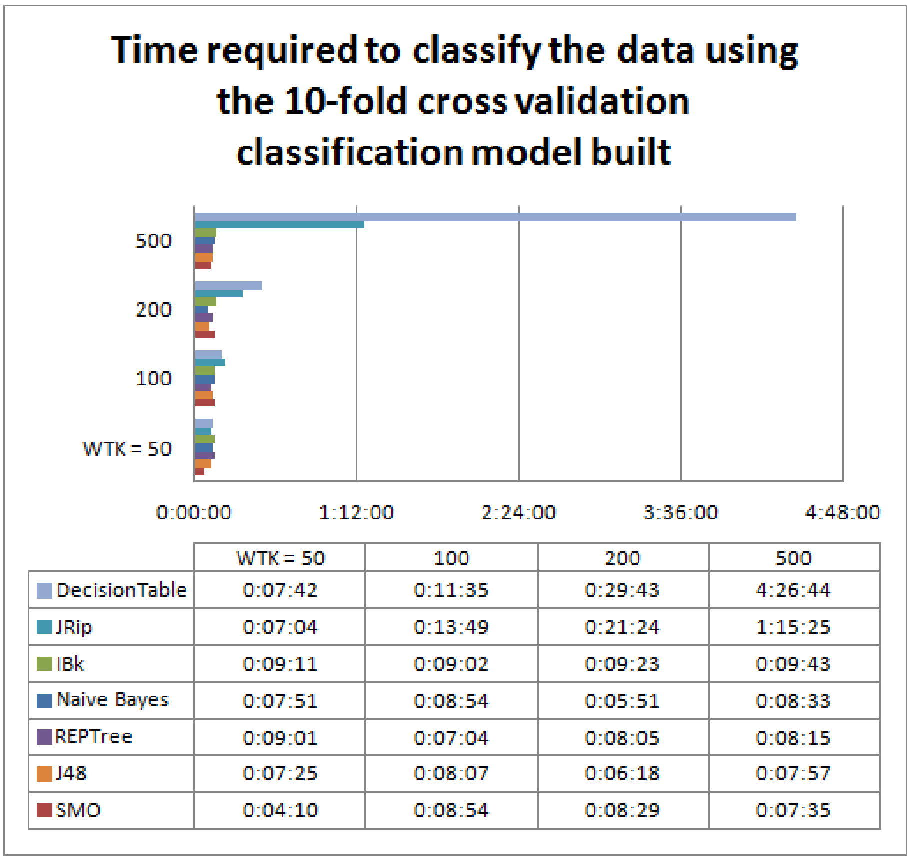
As seen in figure 4.10, the SMO and IBk classifiers remained in the lead for the WTK values tested. REPTree, JRip and the Decision tree started to reach their peak and even decline a little after reaching WTK of 200. Another interesting thing is that the J48 classifier didn't improve its performance even after increasing the WTK value. Specific tests shows that J48 reached its peak somewhere between 25 and 50 WTK for this data set when using a 10-fold cross validation.



	WTK = 50	100	200	500
🟦 DecisionTable	0:00:34	0:02:05	0:06:02	0:59:53
🟦 JRip	0:00:10	0:00:25	0:02:09	0:07:00
🟪 REPTree	0:00:07	0:00:13	0:00:05	0:01:33
🟧 J48	0:00:07	0:00:03	0:00:35	0:00:30
🟥 SMO	0:00:01	0:00:03	0:00:02	0:00:17
🟦 Naive Bayes	0:00:01	0:00:01	0:00:05	0:00:07
🟩 IBk	0:00:01	0:00:01	0:00:01	0:00:01

Figure 4.11: The time required to build the 10-fold classification model for WTK steps of 50, 100, 200 and 500.

As seen in figure 4.11, the Decision tree and JRip classifier need a lot more time to construct the classifier models compared to the other algorithms.



	WTK = 50	100	200	500
🟦 DecisionTable	0:07:42	0:11:35	0:29:43	4:26:44
🟦 JRip	0:07:04	0:13:49	0:21:24	1:15:25
🟩 IBk	0:09:11	0:09:02	0:09:23	0:09:43
🟦 Naive Bayes	0:07:51	0:08:54	0:05:51	0:08:33
🟪 REPTree	0:09:01	0:07:04	0:08:05	0:08:15
🟧 J48	0:07:25	0:08:07	0:06:18	0:07:57
🟥 SMO	0:04:10	0:08:54	0:08:29	0:07:35

Figure 4.12: The time required to classify the 10-fold data set using the classification models built. This was measured in WTK steps of 50, 100, 200 and 500.

As seen in figure 4.12, even reasonably small sets such as data set A made the Decision tree & JRip classifier algorithms need a lot more time to classify the data compared to the other classifiers for WTK values higher than 100. This was interesting since IBk which didn't even built a classifier model due to it's lazy nature was faster than the these two rule based classifiers.

4.4 The final test

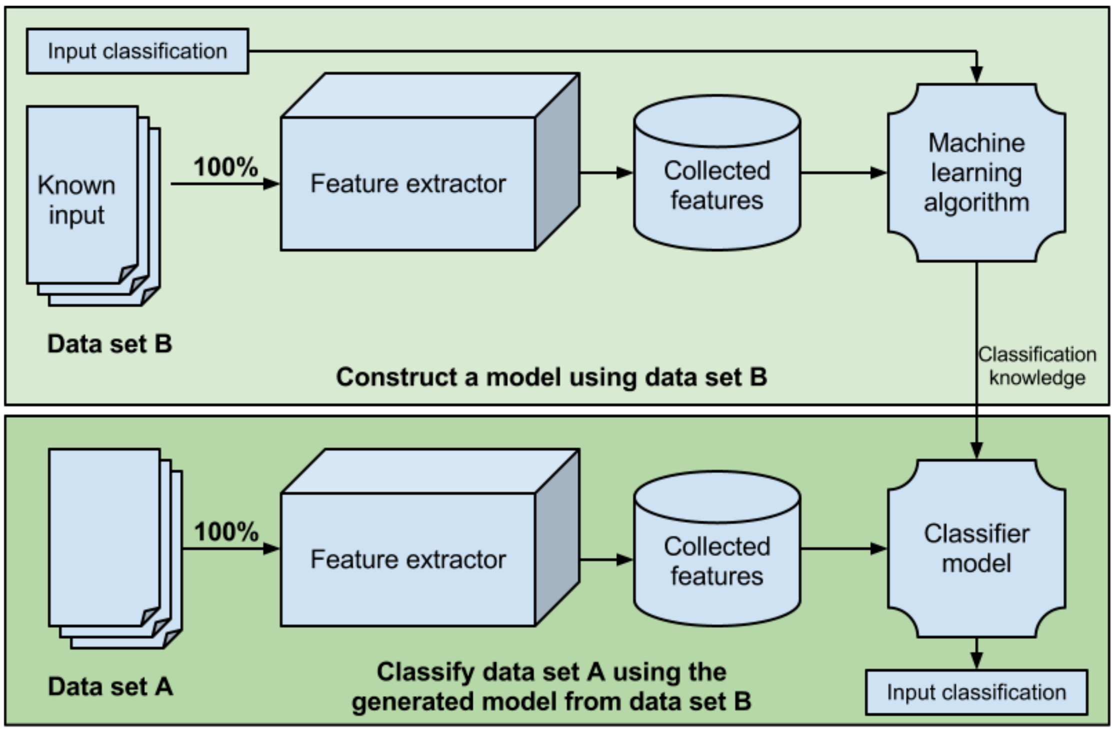
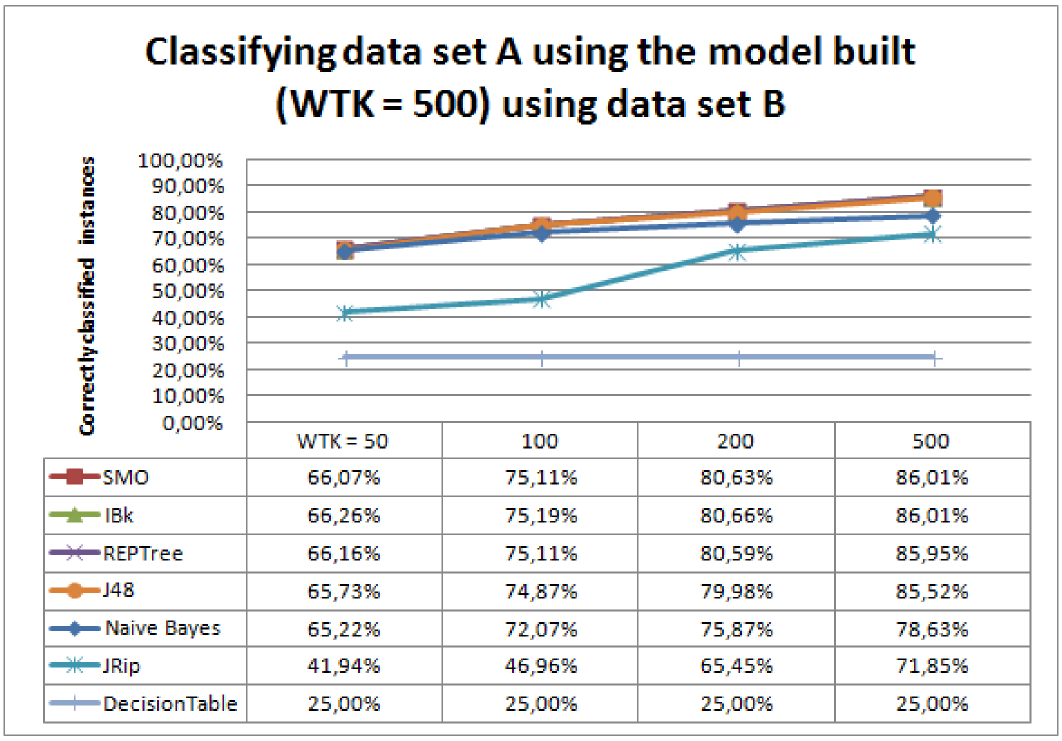


Figure 4.13: A model of how the final test is performed where classifier model was trained using data set B and then used to classify data set A.

As seen in figure 4.13, the final test used to measure the performance of the algorithms was constructed by training a classification model using the bigger data set B and then classified data set A using the model generated. Since data set B had more values then data set A, the idea was that the model constructed would be more generalized and perform better even when different data sets was being classified with it. The classifier models used for each classifier algorithm was the same models that was generated in the test performed in chapter 4.2 with WTK = 500.

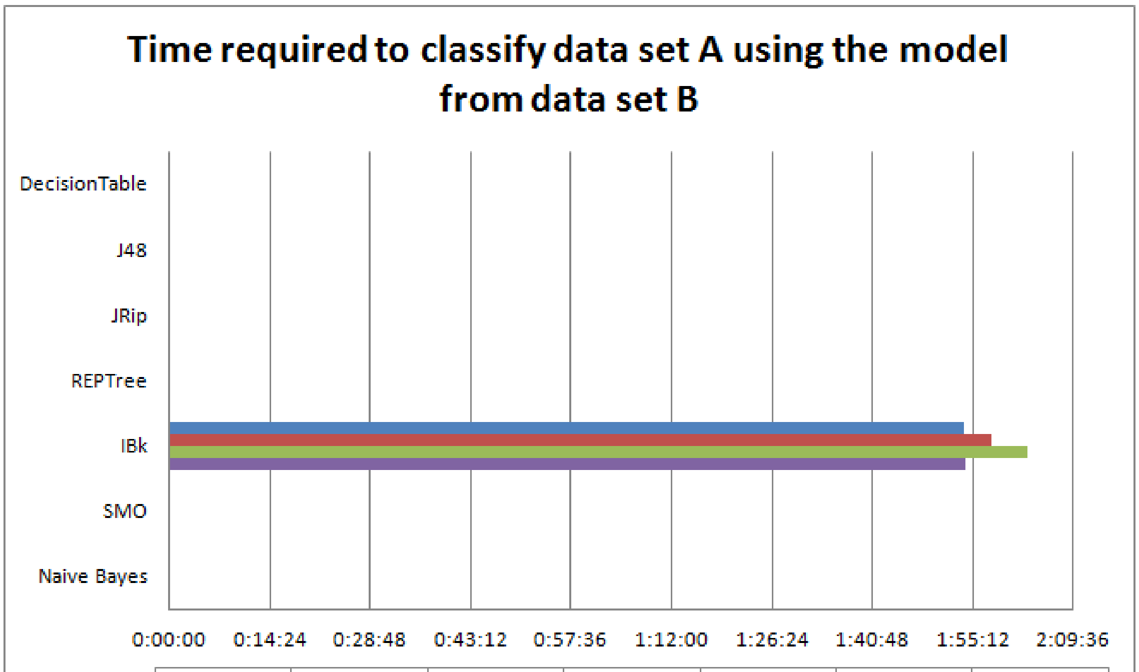
Test mode: Classifying data set A using the model generated from data set B.
Training data: 100% of data set B. 82808 instances divided equally on 4 classes.
Testing data: 100% of data set A. 5296 instances divided equally on 4 classes.
WTK: The classifier models was tested against data set A using WTK = 50, 100, 200 and 500.



	WTK = 50	100	200	500
■ SMO	66,07%	75,11%	80,63%	86,01%
▲ IBk	66,26%	75,19%	80,66%	86,01%
✕ REPTree	66,16%	75,11%	80,59%	85,95%
● J48	65,73%	74,87%	79,98%	85,52%
◆ Naive Bayes	65,22%	72,07%	75,87%	78,63%
✳ JRip	41,94%	46,96%	65,45%	71,85%
+ DecisionTable	25,00%	25,00%	25,00%	25,00%

Figure 4.14: The classification results when classifying data set A using the model built using data set B.

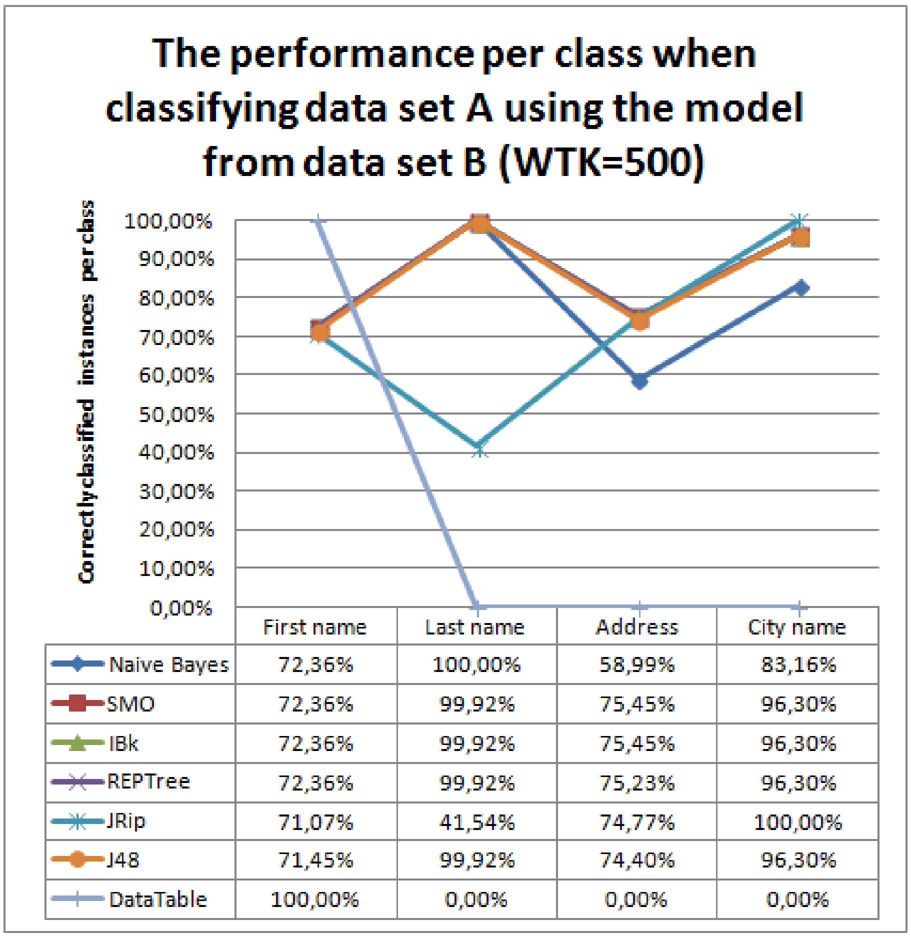
As seen in figure 4.14, a clear majority the algorithms performed well. Especially the SMO, IBk, REPTree and J48 classifiers which all manage to classify more than 85% of the data instances correctly. The DecisionTable classifier performed extremely poorly compared how well it managed in the previous tests. The reason why it only reached 25% will be explained later.



	Naive Bayes	SMO	IBk	REPTree	JRip	J48	DecisionTable
🟦 WTK= 50	0:00:02	0:00:01	1:53:53	0:00:02	0:00:01	0:00:02	0:00:02
🟥 WTK=100	0:00:03	0:00:01	1:57:56	0:00:02	0:00:01	0:00:01	0:00:02
🟩 WTK= 200	0:00:03	0:00:02	2:02:54	0:00:02	0:00:01	0:00:01	0:00:02
🟪 WTK= 500	0:00:06	0:00:02	1:54:05	0:00:01	0:00:01	0:00:01	0:00:02

Figure 4.15: The time to build the classification model as well as the time it took to classify data set A.

As always when it came to the classification step, only the IBk classifier needed more than a couple of seconds to classify the data set.



	First name	Last name	Address	City name
Naive Bayes	72,36%	100,00%	58,99%	83,16%
SMO	72,36%	99,92%	75,45%	96,30%
IBk	72,36%	99,92%	75,45%	96,30%
REPTree	72,36%	99,92%	75,23%	96,30%
JRip	71,07%	41,54%	74,77%	100,00%
J48	71,45%	99,92%	74,40%	96,30%
DataTable	100,00%	0,00%	0,00%	0,00%

Figure 4.16: The performance per class for each algorithm when classifying data set A using the model built using data set B and WTK=500.

As seen in figure 4.16, most classifiers except JRip and DecisionTable performed well even for the specific classes. For some reason the classification model created by DecisionTable decided that all instances belonged to the first class. Since this was really strange, several seperate classifications with the DecisionTable classifier was performed where different data sets was tested. I wasn't sure if this was a bug, but if the data set classified wasn't the same as the data set used to train the classifier model, the DecisionTable classifier labeled all the instances as the first class.

Chapter 5

Discussion

This section discusses the goals made in the problem definition in chapter 1.2.

5.1 Data analysing

Back when I started this thesis I didn't know much more about data classifications other than comparing text strings against dictionaries, nor had I ever heard the word data mining before. During these months that the work with this thesis have progressed, I've come to get a good understanding of the concept and principles of how data analysing can be performed as well as how companies regardless of size can benefit from analysing their data since there's always more information to discover than meets the eye.

5.2 Classifying data

As seen in chapter 4, it was without doubt possible to classify a data set given that a classification model have been constructed using a data set with the same type of classes. To summarize the 7 classification algorithms analysed in this thesis:

Naive Bayes
This was a very simple classifier that perfored decent and should be easy to implement regardless of language used. The drawback was that it wasn't in the top when it came to classifying instances correctly. This was however not a big drawback since it was quick at constructing a classification model, as well as classifying data.

SMO
This support vector machine perfored in the top of the classifiers tested in this thesis, and the concept of support vectors is very powerful once you understand the math behind it. A small drawback was that the algorithm had to perform 1 vs 1 comparison between all the classes to classify data. While this hasn't been a problem for these data setstested, it might become computational complex in the future if the classifier needs to be able to

detect a high number of different classes.

IBk
This was the fastest classifier since no data model is constructed. However, there was a huge drawback to this. The classifier has very high memory constraints since the whole training set has to be kept in memory constantly, and since no generalization was made to the classifier model, a big data set will also require quite some time to be classified since each data point has to be evaluated against the whole training data set.

REPTree
This was a classifier that was able to classify data almost as good as classifiers such as SMO even though it was decision tree based. The drawback of using such a tree structure which is basically made up of rules which wasbeing simplified, was that the performance of algorithm might suffer when the number of classes becomes high since either the tree needs to grow quite a bit and require more time to classify, or the rules in the data structure will be simplified to the point that it becomes unable to correctly classify data as well as you would like it to.

JRip
An decent classifier that performed okay even though I had higher expectations of this rule learner due to the reports I saw where it had been used. The drawback of this classifier was that it requires an extremely long time to construct a classification model for big data sets when using a high WTK value to a point where it becomes useless. For example, it required almost 54 hours to construct a classification model for data set B in chapter 4.2 meanwhile the Naive Bayes classifier managed to do the same in under 3 minutes.

J48
A very efficient decision tree classifier that perfored on pair with SMO and IBk. The drawback of this algorithm was the time it required to construct the classification model since it was decision tree based.

DecisionTable
A classifier that was implemented using a decision table that performed very well for using such a simple method when classifying data it had been trained on. The biggest drawback with this classifier was its inability to classify other data sets using the classification model generated. I'm not sure if this was a bug since I didn't have time to look more closely into this classifier other then running a couple of test using different data sets and observing the same poor performance when using a generated model to classify data. Another drawback wasthe long time required to build a classifier model using big data sets with a high number of words to keep.

5.3 Reaching the goal

The problem definition stated that the classification algorithm must be able to correctly classify a data set it haven't been trained on, with a classification success rate of at least 70% per class. Did we reach that kind of performance?

Looking at the classification results in chapter 4.4, all the classifiers except the DecisionTable managed to get an correct classification over 70% in average, but only SMO, IBk, REPTree and J48 managed to reach this goal per class basis. It was great to see that more than half of the classifiers tested performed this well.

A few thoughts about the classification performance per class seen in chapter 4.4, we can see that first names as well as the address had an successful classification rate of around 70-75% each while last names and city names had a much higher rating. I took a deeper look into the data sets and found that there existed anomalies such as first names in the last name column as well as addresses with the city names in them. Some of these is due to human error when these data sets was created, but there's also logical explanations for them such as the following address "Örnsköldsvik Airport" which is understandable. Since the StringToWordVector filter splits the strings into words, this could be classified as a city even though it was an address.

Chapter 6

Conclusion and future works

There's a few more variables to consider before making the final decision, but from the performance seen in chapter 4, the proposed solution for how Ågrenshuset should tackle the problem of classifying structured data in data sets is to implement a solution where a support vector machine is used to classify each column. The reason why SMO is proposed instead of the other 3 candidates REPTree, J48 and IBk that also managed to reach the goal of a positive classification of 70% per class is as following.

REPTree and J48, both being tree based classifiers suffer from taking a long time when constructing a classifier model for big data sets. IBk being the total opposite of REPTree and J48 since it requires close to no time at all to set up the classifier model. Instead it suffer due to having to store the whole data set in the memory since no optimizations is performing to the classification model. Another thing is that the running time to classify data is related to the size of the data set used as model, so a big data set will make this classifier unusable if a user is actively awaiting the result from the classification due to the time complexity involved. Regarding the SMO classifier which implements a support vector machine, it's one of the fastest to construct the classifier model as well as one of the fastest when it comes to classify data using the model constructed. Memory and storage wise, it is excellent since only the support vectors are stored in the classifier model. It also happens to be the classifier that along with IBk always had the highest number of correctly classified instances. Due to this, in my opinion the SMO is without doubt the classifier that comes out in top in this thesis.

For future works, it would be interesting to see if the support vector machine algorithms such as SMO could keep the high performance seen in this thesis if the data set was extended with a high number of different classes.

This thesis was meant as the first initial step towards creating an application that are able to merge two data sets without having prior knowledge of the structure in each set but instead evaluate the data in each column using data classification algorithms and then based on the result merge the columns together. However, there's still some important aspects to look into before this application can become reality. The application must take into consideration that some data sets have combined data types such as first and last names in the same column. It also needs an interface where it can ask the user if the classification is correct before merging the data. Another important part is to test a wide variety of data types including common ones such as zip codes, tele & mobile-phone numbers as well as email addresses. There's also question marks such as how to be able to detect if you run across a new kind of data class that your model haven't trained for without misclassifying it as a known class.

Overall, data analysis is an interesting area that i enjoyed working with in this thesis and an area i will be looking into further in the future. I think the goals I made in the beginning have been fulfilled as far as this thesis goes and it will be interesting to see if the conclusions made will result in further work towards a working application developed at Ågrenshuset.

CHAPTER 7

Acknowledgements

I would like to thank the internal supervisor Josef Hallberg at Luleå University of Technology, Johan Pålsson and the external supervisor Per-Håkan Westman at Ågrenshuset for their contributions towards the completion of this thesis.

REFERENCES

[1] The Library of Congress, "The library of congress - web archiving faqs." `http://www.loc.gov/webarchiving/faq.html`. Accessed: 2012-09-25.

[2] James Manyika et al for McKinsey Global Institute, "Big data: The next frontier for innovation, competition, and productivity." `http://www.mckinsey.com/insights/mgi/research/technology_and_innovation/big_data_the_next_frontier_for_innovation`. Accessed: 2012-09-25.

[3] EMC^2, "Digital universe." `http://www.emc.com/leadership/programs/digital-universe.htm`. Accessed: 2012-09-25.

[4] U. F. et al, *From Data Mining to Knowledge Discovery in Databases*, pp. 37–54. AI Magazine, 17 ed., 1991. 123.

[5] R. C. Holte, *Machine Learn Vol 11 - Very Simple Classification Rules Perform Well on Most Commonly Used Datasets*, pp. 63–90. Kluwer Academic Publishers-Plenum Publishers, 1 ed., 1993. Available at: `http://dx.doi.org/10.1023/A:1022631118932`.

[6] V. V. et al, *Automation and Remote Control - Pattern recognition using generalized portrait method*, p. 24. Springer, 1 ed., 1963.

[7] B. et al, *Proceedings of the 5th Annual Workshop on Computational Learning Theory COLT'92 - A Training Algorithm for Optimal Margin Classifiers*, pp. 144–152. ACM Press, 1 ed., 1992.

[8] M. A. et al, *Automation and Remote Control - Theoretical foundations of the potential function method in pattern recognition learning*, pp. 821–837. Springer, 1 ed., 1964.

[9] K.-B. D. et al, *Multiple classifier systems : 6th international workshop - Which Is the Best Multiclass SVM Method? An Empirical Study*, pp. 278–285. Berlin : Springer, 1 ed., 2005.

[10] G. H. John and P. Langley, *Eleventh Conference on Uncertainty in Artificial Intelligence - Estimating Continuous Distributions in Bayesian Classifiers*, pp. 338–345. Morgan Kaufmann, 1 ed., 1995.

[11] J. Platt, "Advances in kernel methods, support vector learning - fast training of support vector machines using sequential minimal optimization." `http://research.microsoft.com/apps/pubs/?id=68391`. Accessed: 2012-09-25.

[12] D. Aha and D. Kibler, *Machine Learning - Instance-based learning algorithms*, pp. 37–66. Kluwer Academic Publishers, 1 ed., 1991.

[13] R. Quinlan, *C4.5: Programs for Machine Learning*, pp. –. Morgan Kaufmann Publishers, 1993.

[14] E. Frank, "The weka classifier reptree." `http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/REPTree.html`. Accessed: 2012-12-30.

[15] R. Kohavi, *8th European Conference on Machine Learning - The Power of Decision Tables*, pp. 174–189. Springer, 1 ed., 1995.

[16] W. W. Cohen, *Twelfth International Conference on Machine Learning - Fast Effective Rule Induction*, pp. 115–123. Morgan Kaufmann, 1 ed., 1995.

APPENDIX A

Classification results from chapter 4.1

A.1 WTK = 25

DecisionTable

```
Correctly Classified Instances          3104     58.6103 %
Incorrectly Classified Instances        2192     41.3897 %
Kappa statistic                            0.4481
Mean absolute error                        0.2376
Root mean squared error                    0.3403
Relative absolute error                   63.3506 %
Root relative squared error               78.5877 %
Coverage of cases (0.95 level)            99.9434 %
Mean rel. region size (0.95 level)        79.3523 %
Total Number of Instances               5296

=== Confusion Matrix ===

   a    b    c    d   <-- classified as
 119 1205    0    0 |    a = firstname
   0 1324    0    0 |    b = lastname
  15  713  593    3 |    c = address
   0  256    0 1068 |    d = cityname
```

IBk

```
Correctly Classified Instances          3119     58.8935 %
Incorrectly Classified Instances        2177     41.1065 %
Kappa statistic                            0.4519
Mean absolute error                        0.2229
Root mean squared error                    0.3338
Coverage of cases (0.95 level)            99.9434 %
```

```
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  119 1205    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  713  608    3 |    c = address
    0  256    0 1068 |    d = cityname
```

J48

```
Correctly Classified Instances          3087     58.2893 %
Incorrectly Classified Instances        2209     41.7107 %
Kappa statistic                            0.4439
Mean absolute error                        0.2315
Root mean squared error                    0.3402
Relative absolute error                   61.737  %
Root relative squared error               78.5729 %
Coverage of cases (0.95 level)            99.9434 %
Mean rel. region size (0.95 level)        74.9906 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  119 1205    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  745  576    3 |    c = address
    0  256    0 1068 |    d = cityname
```

JRip

```
Correctly Classified Instances          2161     40.8044 %
Incorrectly Classified Instances        3135     59.1956 %
Kappa statistic                            0.2107
Mean absolute error                        0.3111
Root mean squared error                    0.3944
Relative absolute error                   82.949  %
Root relative squared error               91.0764 %
Coverage of cases (0.95 level)           100      %
Mean rel. region size (0.95 level)        88.1467 %
```

```
Total Number of Instances                5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  119    0    0 1205 |    a = firstname
    0  110    0 1214 |    b = lastname
    0    0  608  716 |    c = address
    0    0    0 1324 |    d = cityname
```

Naive Bayes

```
Correctly Classified Instances           3119     58.8935 %
Incorrectly Classified Instances         2177     41.1065 %
Kappa statistic                             0.4519
Mean absolute error                         0.2702
Root mean squared error                     0.3558
Coverage of cases (0.95 level)             99.9434 %
Total Number of Instances                5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  119 1205    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  713  608    3 |    c = address
    0  256    0 1068 |    d = cityname
```

REPTree

```
Correctly Classified Instances           3119     58.8935 %
Incorrectly Classified Instances         2177     41.1065 %
Kappa statistic                             0.4519
Mean absolute error                         0.2229
Root mean squared error                     0.3338
Relative absolute error                    59.4397 %
Root relative squared error                77.0971 %
Coverage of cases (0.95 level)             99.9434 %
Mean rel. region size (0.95 level)         72.9796 %
Total Number of Instances                5296

=== Confusion Matrix ===
    a    b    c    d   <-- classified as
  119 1205    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  713  608    3 |    c = address
    0  256    0 1068 |    d = cityname
```

SMO

```
Correctly Classified Instances          3119     58.8935 %
Incorrectly Classified Instances        2177     41.1065 %
Kappa statistic                            0.4519
Mean absolute error                        0.3035
Root mean squared error                    0.3859
Coverage of cases (0.95 level)            95.1662 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  119 1205    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  713  608    3 |    c = address
    0  256    0 1068 |    d = cityname
```

A.2 WTK = 50

DecisionTable

```
Correctly Classified Instances          3624     68.429  %
Incorrectly Classified Instances        1672     31.571  %
Kappa statistic                            0.5791
Mean absolute error                        0.2014
Root mean squared error                    0.309
Relative absolute error                   53.7182 %
Root relative squared error               71.3582 %
Coverage of cases (0.95 level)            99.9434 %
Mean rel. region size (0.95 level)        77.162  %
Total Number of Instances               5296

=== Confusion Matrix ===
    a    b    c    d   <-- classified as
  343  981    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
   13  500  807    4 |    c = address
    0  174    0 1150 |    d = cityname
```

IBk

```
Correctly Classified Instances        3678    69.4486 %
Incorrectly Classified Instances      1618    30.5514 %
Kappa statistic                          0.5926
Mean absolute error                      0.1773
Root mean squared error                  0.2977
Coverage of cases (0.95 level)          99.9245 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  343  981    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  459  861    4 |    c = address
    0  174    0 1150 |    d = cityname
```

J48

```
Correctly Classified Instances        3375    63.7273 %
Incorrectly Classified Instances      1921    36.2727 %
Kappa statistic                          0.5164
Mean absolute error                      0.2086
Root mean squared error                  0.3229
Relative absolute error                 55.6227 %
Root relative squared error             74.5806 %
Coverage of cases (0.95 level)          99.9056 %
Mean rel. region size (0.95 level)      70.8837 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  238 1086    0    0 |    a = firstname
```

```
    0  1324     0     0 |     b = lastname
    1   632   687     4 |     c = address
    0   198     0  1126 |     d = cityname
```

JRip

```
=== Evaluation result ===

Scheme: InputMappedClassifier
Options: -I -trim -W weka.classifiers.rules.JRip -- -F 3 -N 2.0
   -O 2 -S 1
Relation: mixed_labels-weka.filters.unsupervised.attribute.
   StringToWordVector-R1-W51-prune-rate-1.0-N0-L-stemmerweka.
   core.stemmers.NullStemmer-M1-O-tokenizerweka.core.tokenizers.
   WordTokenizer -delimiters "␣\r␣\t.,;:\'\"()?!-"


J48 pruned tree
---------------

   rnskldsvik      <= 0
|   arn svall <= 0
|   |   domsj  <= 0
|   |   |   sj levad <= 0
|   |   |   |   bj sta <= 0
|   |   |   |   |   bredbyn <= 0
|   |   |   |   |   |   husum <= 0
|   |   |   |   |   |   |         verhrns      <= 0
|   |   |   |   |   |   |   |   box <= 0
|   |   |   |   |   |   |   |   |   2 <= 0
|   |   |   |   |   |   |   |   |   |   1 <= 0
|   |   |   |   |   |   |   |   |   |   |   6 <= 0
|   |   |   |   |   |   |   |   |   |   |   |   4 <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   3 <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   7 <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |
   nygatan <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
   15 <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
   |   12 <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   11 <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   20 <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   5 <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   |   8 <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   |   |   18 <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   |   |   |   anders <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   |   |   |   |   lars <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   |   |   |   |   |   ulf <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   |   |   |   |   |   |   jan <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   |   |   |   |   |   |   |   per <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   |   |   |   |   |   |   |   |   peter <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   |   |   |   |   |   |   |   |   |   erik <=
    0: lastname (3240.0/1916.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   |   |   |   |   |   |   |   |   |   erik > 0:
     firstname (25.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   |   |   |   |   |   |   |   |   peter > 0:
    firstname (31.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   |   |   |   |   |   |   |   per > 0:
    firstname (31.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   |   |   |   |   |   |   jan > 0: firstname
    (33.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   |   |   |   |   |   ulf > 0: firstname (37.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    |   |   |   |   |   |   |   |   lars > 0: firstname (37.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
   |   |   |   |   |   |   |   anders > 0: firstname (45.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
   |   |   |   |   |   |   18 > 0: address (25.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
   |   |   |   |   |   8 > 0: address (28.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
   |   |   |   |   5 > 0: address (28.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
   |   |   |   20 > 0: address (28.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
   |   |   11 > 0: address (28.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
   |   12 > 0: address (33.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
   15 > 0: address (34.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |
   nygatan > 0: address (35.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   7 > 0:
   address (36.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   3 > 0:
   address (37.0)
|   |   |   |   |   |   |   |   |   |   |   |   4 > 0: address
   (43.0)
|   |   |   |   |   |   |   |   |   |   |   6 > 0: address
   (45.0)
|   |   |   |   |   |   |   |   |   |   1 > 0: address (49.0)
|   |   |   |   |   |   |   |   |   2 > 0: address (70.0)
|   |   |   |   |   |   |   |   box > 0: address (168.0)
|   |   |   |   |   |   |   verhrns > 0: cityname
   (26.0/1.0)
|   |   |   |   |   |   husum > 0: cityname (33.0)
|   |   |   |   |   bredbyn > 0: cityname (45.0)
|   |   |   |   bj sta > 0: cityname (53.0)
|   |   |   sj levad > 0: cityname (66.0)
|   |   domsj > 0: cityname (105.0/2.0)
|   arn svall > 0: cityname (93.0)
   rnskldsvik > 0: cityname (709.0/1.0)

Number of Leaves : 31
```

```
Size of the tree :   61

Correctly Classified Instances        2654    50.1133 %
Incorrectly Classified Instances      2642    49.8867 %
Kappa statistic                          0.3348
Mean absolute error                      0.271
Root mean squared error                  0.3681
Relative absolute error                 72.2595 %
Root relative squared error             85.0056 %
Coverage of cases (0.95 level)          99.9622 %
Mean rel. region size (0.95 level)      81.1367 %
Total Number of Instances             5296

=== Confusion Matrix ===

   a    b    c    d   <-- classified as
 343    0    0  981 |    a = firstname
   0  134    0 1190 |    b = lastname
   2    0  853  469 |    c = address
   0    0    0 1324 |    d = cityname
```

Naive Bayes

```
Correctly Classified Instances        3595    67.8814 %
Incorrectly Classified Instances      1701    32.1186 %
Kappa statistic                          0.5718
Mean absolute error                      0.2644
Root mean squared error                  0.3484
Coverage of cases (0.95 level)          99.9434 %
Total Number of Instances             5296

=== Confusion Matrix ===

   a    b    c    d   <-- classified as
 343  981    0    0 |    a = firstname
   0 1324    0    0 |    b = lastname
   1  460  860    3 |    c = address
   0  256    0 1068 |    d = cityname
```

REPTree

```
Correctly Classified Instances        3677    69.4298 %
```

```
Incorrectly Classified Instances        1619     30.5702 %
Kappa statistic                            0.5924
Mean absolute error                        0.1774
Root mean squared error                    0.2978
Relative absolute error                   47.298  %
Root relative squared error               68.7736 %
Coverage of cases (0.95 level)            99.9245 %
Mean rel. region size (0.95 level)        64.7234 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  343  981    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  460  860    4 |    c = address
    0  174    0 1150 |    d = cityname
```

SMO

```
Correctly Classified Instances          3678     69.4486 %
Incorrectly Classified Instances        1618     30.5514 %
Kappa statistic                            0.5926
Mean absolute error                        0.2882
Root mean squared error                    0.3679
Coverage of cases (0.95 level)            96.7145 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  343  981    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  459  861    4 |    c = address
    0  174    0 1150 |    d = cityname
```

A.3 WTK = 75

DecisionTable

```
Correctly Classified Instances          3842     72.5453 %
```

```
Incorrectly Classified Instances      1454    27.4547 %
Kappa statistic                          0.6339
Mean absolute error                      0.1878
Root mean squared error                  0.2955
Relative absolute error                 50.0704 %
Root relative squared error             68.237  %
Coverage of cases (0.95 level)          99.9434 %
Mean rel. region size (0.95 level)      76.8599 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  486  838    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
   16  452  852    4 |    c = address
    0  144    0 1180 |    d = cityname
```

IBk

```
Correctly Classified Instances        3896    73.565  %
Incorrectly Classified Instances      1400    26.435  %
Kappa statistic                          0.6475
Mean absolute error                      0.1548
Root mean squared error                  0.2782
Coverage of cases (0.95 level)          99.9245 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  486  838    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  414  906    4 |    c = address
    0  144    0 1180 |    d = cityname
```

J48

```
Correctly Classified Instances        3375    63.7273 %
Incorrectly Classified Instances      1921    36.2727 %
Kappa statistic                          0.5164
Mean absolute error                      0.2086
```

```
Root mean squared error                  0.3229
Relative absolute error                 55.6227 %
Root relative squared error             74.5806 %
Coverage of cases (0.95 level)          99.9056 %
Mean rel. region size (0.95 level)      70.8837 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  238 1086    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  632  687    4 |    c = address
    0  198    0 1126 |    d = cityname
```

JRip

```
Correctly Classified Instances        3678    69.4486 %
Incorrectly Classified Instances      1618    30.5514 %
Kappa statistic                          0.5926
Mean absolute error                      0.1799
Root mean squared error                  0.2999
Relative absolute error                 47.9634 %
Root relative squared error             69.2556 %
Coverage of cases (0.95 level)          97.2054 %
Mean rel. region size (0.95 level)      52.7379 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1324    0    0    0 |    a = firstname
 1054  270    0    0 |    b = lastname
  416    0  904    4 |    c = address
  144    0    0 1180 |    d = cityname
```

Naive Bayes

```
Correctly Classified Instances        3782    71.4124 %
Incorrectly Classified Instances      1514    28.5876 %
Kappa statistic                          0.6188
Mean absolute error                      0.2622
```

```
Root mean squared error                  0.3457
Coverage of cases (0.95 level)          99.9434 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  486  838    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  416  904    3 |    c = address
    0  256    0 1068 |    d = cityname
```

REPTree

```
Correctly Classified Instances        3894     73.5272 %
Incorrectly Classified Instances      1402     26.4728 %
Kappa statistic                          0.647
Mean absolute error                      0.155
Root mean squared error                  0.2784
Relative absolute error                 41.323  %
Root relative squared error             64.2829 %
Coverage of cases (0.95 level)          99.9245 %
Mean rel. region size (0.95 level)      59.7243 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  486  838    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  416  904    4 |    c = address
    0  144    0 1180 |    d = cityname
```

SMO

```
Correctly Classified Instances        3894     73.5272 %
Incorrectly Classified Instances      1402     26.4728 %
Kappa statistic                          0.647
Mean absolute error                      0.2831
Root mean squared error                  0.3611
Relative absolute error                 75.5035 %
Root relative squared error             83.382  %
```

```
Coverage of cases (0.95 level)          97.281  %
Mean rel. region size (0.95 level)      75      %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  486  838    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  416  904    4 |    c = address
    0  144    0 1180 |    d = cityname
```

A.4 WTK = 100

DecisionTable

```
Correctly Classified Instances        4133    78.04   %
Incorrectly Classified Instances      1163    21.96   %
Kappa statistic                          0.7072
Mean absolute error                      0.1673
Root mean squared error                  0.274
Relative absolute error                 44.6189 %
Root relative squared error             63.2723 %
Coverage of cases (0.95 level)          98.6971 %
Mean rel. region size (0.95 level)      65.2615 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  661  663    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
   19  411  890    4 |    c = address
    0   66    0 1258 |    d = cityname
```

IBk

```
Correctly Classified Instances        4184    79.003  %
Incorrectly Classified Instances      1112    20.997  %
Kappa statistic                          0.72
Mean absolute error                      0.1293
```

```
Root mean squared error                    0.2542
Coverage of cases (0.95 level)            98.6782 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  661  663    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  379  941    4 |    c = address
    0   66    0 1258 |    d = cityname
```

J48

```
Correctly Classified Instances          3375      63.7273 %
Incorrectly Classified Instances        1921      36.2727 %
Kappa statistic                            0.5164
Mean absolute error                        0.2086
Root mean squared error                    0.3229
Relative absolute error                   55.6227 %
Root relative squared error               74.5806 %
Coverage of cases (0.95 level)            99.9056 %
Mean rel. region size (0.95 level)        70.8837 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  238 1086    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  632  687    4 |    c = address
    0  198    0 1126 |    d = cityname
```

JRip

```
Correctly Classified Instances          3147      59.4222 %
Incorrectly Classified Instances        2149      40.5778 %
Kappa statistic                            0.459
Mean absolute error                        0.2321
Root mean squared error                    0.3407
Relative absolute error                   61.8975 %
Root relative squared error               78.675  %
```

```
Coverage of cases (0.95 level)         99.9622 %
Mean rel. region size (0.95 level)     74.155  %
Total Number of Instances            5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  661    0    0  663 |    a = firstname
    0  240    0 1084 |    b = lastname
    2    0  922  400 |    c = address
    0    0    0 1324 |    d = cityname
```

Naive Bayes

```
Correctly Classified Instances       3957     74.7168 %
Incorrectly Classified Instances     1339     25.2832 %
Kappa statistic                         0.6629
Mean absolute error                     0.2608
Root mean squared error                 0.3437
Relative absolute error                69.5366 %
Root relative squared error            79.372  %
Coverage of cases (0.95 level)         99.9434 %
Mean rel. region size (0.95 level)     86.0366 %
Total Number of Instances            5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  661  663    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  416  904    3 |    c = address
    0  256    0 1068 |    d = cityname
```

REPTree

```
Correctly Classified Instances       4177     78.8708 %
Incorrectly Classified Instances     1119     21.1292 %
Kappa statistic                         0.7183
Mean absolute error                     0.1299
Root mean squared error                 0.2549
Relative absolute error                34.6477 %
Root relative squared error            58.8623 %
```

```
Coverage of cases (0.95 level)          98.6782 %
Mean rel. region size (0.95 level)      45.119  %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  661  663    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  386  934    4 |    c = address
    0   66    0 1258 |    d = cityname
```

SMO

```
Correctly Classified Instances        4180      78.9275 %
Incorrectly Classified Instances      1116      21.0725 %
Kappa statistic                          0.719
Mean absolute error                      0.2756
Root mean squared error                  0.3505
Relative absolute error                 73.502  %
Root relative squared error             80.946  %
Coverage of cases (0.95 level)          98.7538 %
Mean rel. region size (0.95 level)      75      %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  661  663    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  383  937    4 |    c = address
    0   66    0 1258 |    d = cityname
```

A.5 WTK = 150

DecisionTable

```
Correctly Classified Instances        4296      81.1178 %
Incorrectly Classified Instances      1000      18.8822 %
Kappa statistic                          0.7482
Mean absolute error                      0.1564
```

```
Root mean squared error                  0.2612
Relative absolute error                 41.7147 %
Root relative squared error             60.3166 %
Coverage of cases (0.95 level)          99.1503 %
Mean rel. region size (0.95 level)      66.2812 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  751  573    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
   30  350  939    5 |    c = address
    0   42    0 1282 |    d = cityname
```

IBk

```
Correctly Classified Instances        4364      82.4018 %
Incorrectly Classified Instances       932      17.5982 %
Kappa statistic                          0.7654
Mean absolute error                      0.1073
Root mean squared error                  0.2316
Coverage of cases (0.95 level)          99.1314 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  755  569    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  316 1003    5 |    c = address
    0   42    0 1282 |    d = cityname
```

J48

```
Correctly Classified Instances        3375      63.7273 %
Incorrectly Classified Instances      1921      36.2727 %
Kappa statistic                          0.5164
Mean absolute error                      0.2086
Root mean squared error                  0.3229
Relative absolute error                 55.6227 %
Root relative squared error             74.5806 %
```

```
Coverage of cases (0.95 level)          99.9056 %
Mean rel. region size (0.95 level)      70.8837 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  238 1086    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  632  687    4 |    c = address
    0  198    0 1126 |    d = cityname
```

JRip

```
Correctly Classified Instances        4339     81.9298 %
Incorrectly Classified Instances       957     18.0702 %
Kappa statistic                          0.7591
Mean absolute error                      0.1248
Root mean squared error                  0.2498
Relative absolute error                 33.2855 %
Root relative squared error             57.6936 %
Coverage of cases (0.95 level)          99.0937 %
Mean rel. region size (0.95 level)      46.5068 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  751  573    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    2  335  982    5 |    c = address
    0   42    0 1282 |    d = cityname
```

Naive Bayes

```
Correctly Classified Instances        4063     76.7183 %
Incorrectly Classified Instances      1233     23.2817 %
Kappa statistic                          0.6896
Mean absolute error                      0.2596
Root mean squared error                  0.3421
Relative absolute error                 69.2244 %
Root relative squared error             79.0081 %
```

```
Coverage of cases (0.95 level)          99.9434 %
Mean rel. region size (0.95 level)      86.0366 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  755  569    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  404  916    3 |    c = address
    0  256    0 1068 |    d = cityname
```

REPTree

```
Correctly Classified Instances          4355     82.2319 %
Incorrectly Classified Instances         941     17.7681 %
Kappa statistic                           0.7631
Mean absolute error                       0.1095
Root mean squared error                   0.234
Relative absolute error                  29.2084 %
Root relative squared error              54.0448 %
Coverage of cases (0.95 level)           99.1314 %
Mean rel. region size (0.95 level)       42.145  %
Total Number of Instances              5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  755  569    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  325  994    5 |    c = address
    0   42    0 1282 |    d = cityname
```

SMO

```
Correctly Classified Instances          4359     82.3074 %
Incorrectly Classified Instances         937     17.6926 %
Kappa statistic                           0.7641
Mean absolute error                       0.2711
Root mean squared error                   0.344
Relative absolute error                  72.2894 %
Root relative squared error              79.4338 %
```

```
Coverage of cases (0.95 level)          99.2069 %
Mean rel. region size (0.95 level)      75      %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  755  569    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  321  998    5 |    c = address
    0   42    0 1282 |    d = cityname
```

A.6 WTK = 200

DecisionTable

```
Correctly Classified Instances        4596     86.7825 %
Incorrectly Classified Instances       700     13.2175 %
Kappa statistic                          0.8238
Mean absolute error                      0.1375
Root mean squared error                  0.237
Relative absolute error                 36.663  %
Root relative squared error             54.7349 %
Coverage of cases (0.95 level)          99.3769 %
Mean rel. region size (0.95 level)      67.8106 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  926  398    0    0 |    a = fornamn
    0 1324    0    0 |    b = efternamn
   35  232 1052    5 |    c = adress
    0   30    0 1294 |    d = postort
```

IBk

```
Correctly Classified Instances        4494     84.8565 %
Incorrectly Classified Instances       802     15.1435 %
Kappa statistic                          0.7981
Mean absolute error                      0.0948
```

```
Root mean squared error                  0.2176
Coverage of cases (0.95 level)          99.2069 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  845  479    0    0 |    a = firstname
    0 1322    0    2 |    b = lastname
    0  278 1041    5 |    c = address
    0   38    0 1286 |    d = cityname
```

J48

```
Correctly Classified Instances        3375     63.7273 %
Incorrectly Classified Instances      1921     36.2727 %
Kappa statistic                          0.5164
Mean absolute error                      0.2086
Root mean squared error                  0.3229
Relative absolute error                 55.6227 %
Root relative squared error             74.5806 %
Coverage of cases (0.95 level)          99.9056 %
Mean rel. region size (0.95 level)      70.8837 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  238 1086    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  632  687    4 |    c = address
    0  198    0 1126 |    d = cityname
```

JRip

```
Correctly Classified Instances        3811     71.96   %
Incorrectly Classified Instances      1485     28.04   %
Kappa statistic                          0.6261
Mean absolute error                      0.168
Root mean squared error                  0.2898
Relative absolute error                 44.7939 %
Root relative squared error             66.9282 %
```

```
Coverage of cases (0.95 level)          99.1692 %
Mean rel. region size (0.95 level)      51.435  %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  836    0  488    0 |    a = firstname
    0  372  950    2 |    b = lastname
    2    0 1317    5 |    c = address
    0    0   38 1286 |    d = cityname
```

Naive Bayes

```
Correctly Classified Instances        4165     78.6443 %
Incorrectly Classified Instances      1131     21.3557 %
Kappa statistic                          0.7153
Mean absolute error                      0.2591
Root mean squared error                  0.3414
Relative absolute error                 69.09    %
Root relative squared error             78.8519 %
Coverage of cases (0.95 level)          99.9434 %
Mean rel. region size (0.95 level)      86.0366 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  845  479    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  392  928    3 |    c = address
    0  256    0 1068 |    d = cityname
```

REPTree

```
Correctly Classified Instances        4476     84.5166 %
Incorrectly Classified Instances       820     15.4834 %
Kappa statistic                          0.7936
Mean absolute error                      0.0972
Root mean squared error                  0.2205
Relative absolute error                 25.9193 %
Root relative squared error             50.911  %
```

```
Coverage of cases (0.95 level)          99.2069 %
Mean rel. region size (0.95 level)      40.2568 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  845  479    0    0 |    a = firstname
    0 1322    0    2 |    b = lastname
    0  296 1023    5 |    c = address
    0   38    0 1286 |    d = cityname
```

SMO

```
Correctly Classified Instances        4494      84.8565 %
Incorrectly Classified Instances       802      15.1435 %
Kappa statistic                          0.7981
Mean absolute error                      0.2682
Root mean squared error                  0.3397
Relative absolute error                 71.5173 %
Root relative squared error             78.4558 %
Coverage of cases (0.95 level)          99.2825 %
Mean rel. region size (0.95 level)      75       %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  845  479    0    0 |    a = firstname
    0 1322    0    2 |    b = lastname
    0  278 1041    5 |    c = address
    0   38    0 1286 |    d = cityname
```

A.7 WTK = 500

DecisionTable

```
Correctly Classified Instances        4596      86.7825 %
Incorrectly Classified Instances       700      13.2175 %
Kappa statistic                          0.8238
Mean absolute error                      0.1375
```

```
Root mean squared error                  0.237
Relative absolute error                 36.663  %
Root relative squared error             54.7349 %
Coverage of cases (0.95 level)          99.3769 %
Mean rel. region size (0.95 level)      67.8106 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  926  398    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
   35  232 1052    5 |    c = address
    0   30    0 1294 |    d = cityname
```

IBk

```
Correctly Classified Instances        5083     95.9781 %
Incorrectly Classified Instances       213      4.0219 %
Kappa statistic                          0.9464
Mean absolute error                      0.028
Root mean squared error                  0.1182
Coverage of cases (0.95 level)          99.6601 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1170  154    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0   40 1280    4 |    c = address
    0   15    0 1309 |    d = cityname
```

J48

```
Correctly Classified Instances        3375     63.7273 %
Incorrectly Classified Instances      1921     36.2727 %
Kappa statistic                          0.5164
Mean absolute error                      0.2086
Root mean squared error                  0.3229
Relative absolute error                 55.6227 %
Root relative squared error             74.5806 %
```

```
Coverage of cases (0.95 level)          99.9056 %
Mean rel. region size (0.95 level)      70.8837 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  238 1086    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  632  687    4 |    c = address
    0  198    0 1126 |    d = cityname
```

JRip

```
Correctly Classified Instances          4587     86.6125 %
Incorrectly Classified Instances         709     13.3875 %
Kappa statistic                            0.8215
Mean absolute error                        0.0974
Root mean squared error                    0.2207
Relative absolute error                   25.9792 %
Root relative squared error               50.9698 %
Coverage of cases (0.95 level)            99.6035 %
Mean rel. region size (0.95 level)        44.1654 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  819  505    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    2  182 1135    5 |    c = address
    0   15    0 1309 |    d = cityname
```

Naive Bayes

```
Correctly Classified Instances          4491     84.7998 %
Incorrectly Classified Instances         805     15.2002 %
Kappa statistic                            0.7973
Mean absolute error                        0.2577
Root mean squared error                    0.3396
Relative absolute error                   68.7181 %
Root relative squared error               78.4169 %
```

```
Coverage of cases (0.95 level)          99.9434 %
Mean rel. region size (0.95 level)      86.0366 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1170  154    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  391  929    3 |    c = address
    0  256    0 1068 |    d = cityname
```

REPTree

```
Correctly Classified Instances        4484     84.6677 %
Incorrectly Classified Instances       812     15.3323 %
Kappa statistic                          0.7956
Mean absolute error                      0.1023
Root mean squared error                  0.2262
Relative absolute error                 27.2904 %
Root relative squared error             52.2403 %
Coverage of cases (0.95 level)          99.5279 %
Mean rel. region size (0.95 level)      42.315  %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  815  509    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  277 1042    5 |    c = address
    0   21    0 1303 |    d = cityname
```

SMO

```
Correctly Classified Instances        5083     95.9781 %
Incorrectly Classified Instances       213      4.0219 %
Kappa statistic                          0.9464
Mean absolute error                      0.2545
Root mean squared error                  0.3189
Relative absolute error                 67.8541 %
Root relative squared error             73.6389 %
```

```
Coverage of cases (0.95 level)          99.7168 %
Mean rel. region size (0.95 level)      75      %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1170  154    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0   40 1280    4 |    c = address
    0   15    0 1309 |    d = cityname
```

A.8 WTK = 1000

DecisionTable

```
Correctly Classified Instances        4596    86.7825 %
Incorrectly Classified Instances       700    13.2175 %
Kappa statistic                          0.8238
Mean absolute error                      0.1375
Root mean squared error                  0.237
Relative absolute error                 36.663  %
Root relative squared error             54.7349 %
Coverage of cases (0.95 level)          99.3769 %
Mean rel. region size (0.95 level)      67.8106 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  926  398    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
   35  232 1052    5 |    c = address
    0   30    0 1294 |    d = cityname
```

IBk

```
Correctly Classified Instances        5293    99.9434 %
Incorrectly Classified Instances         3     0.0566 %
Kappa statistic                          0.9992
Mean absolute error                      0.0006
```

```
Root mean squared error                  0.0155
Coverage of cases (0.95 level)          99.9811 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1324    0    0    0 |    a = firstname
    0 1323    1    0 |    b = lastname
    0    0 1322    2 |    c = address
    0    0    0 1324 |    d = cityname
```

J48

```
Correctly Classified Instances        3375     63.7273 %
Incorrectly Classified Instances      1921     36.2727 %
Kappa statistic                          0.5164
Mean absolute error                      0.2086
Root mean squared error                  0.3229
Relative absolute error                 55.6227 %
Root relative squared error             74.5806 %
Coverage of cases (0.95 level)          99.9056 %
Mean rel. region size (0.95 level)      70.8837 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  238 1086    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  632  687    4 |    c = address
    0  198    0 1126 |    d = cityname
```

JRip

```
Correctly Classified Instances        4738     89.4637 %
Incorrectly Classified Instances       558     10.5363 %
Kappa statistic                          0.8595
Mean absolute error                      0.0797
Root mean squared error                  0.1996
Relative absolute error                 21.2561 %
Root relative squared error             46.1043 %
```

```
Coverage of cases (0.95 level)          99.6035 %
Mean rel. region size (0.95 level)      42.754  %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  881  443    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    2   92 1224    6 |    c = address
    0   15    0 1309 |    d = cityname
```

Naive Bayes

```
Correctly Classified Instances        4646      87.7266 %
Incorrectly Classified Instances       650      12.2734 %
Kappa statistic                          0.8364
Mean absolute error                      0.2574
Root mean squared error                  0.3392
Relative absolute error                 68.6515 %
Root relative squared error             78.3386 %
Coverage of cases (0.95 level)          99.9434 %
Mean rel. region size (0.95 level)      86.0366 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1324    0    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  390  930    3 |    c = address
    0  256    0 1068 |    d = cityname
```

REPTree

```
Correctly Classified Instances        4484      84.6677 %
Incorrectly Classified Instances       812      15.3323 %
Kappa statistic                          0.7956
Mean absolute error                      0.1023
Root mean squared error                  0.2262
Relative absolute error                 27.2904 %
Root relative squared error             52.2403 %
```

```
Coverage of cases (0.95 level)        99.5279 %
Mean rel. region size (0.95 level)    42.315  %
Total Number of Instances           5296

=== Confusion Matrix ===

   a    b    c    d   <-- classified as
 815  509    0    0 |    a = firstname
   0 1324    0    0 |    b = lastname
   0  277 1042    5 |    c = address
   0   21    0 1303 |    d = cityname
```

SMO

```
Correctly Classified Instances        5291     99.9056 %
Incorrectly Classified Instances         5      0.0944 %
Kappa statistic                          0.9987
Mean absolute error                      0.2501
Root mean squared error                  0.312
Relative absolute error                 66.696  %
Root relative squared error             72.049  %
Coverage of cases (0.95 level)         100      %
Mean rel. region size (0.95 level)      75      %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1324    0    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0    2 1319    3 |    c = address
    0    0    0 1324 |    d = cityname
```

A.9 WTK = 2000

DecisionTable

```
Correctly Classified Instances        4596     86.7825 %
Incorrectly Classified Instances       700     13.2175 %
Kappa statistic                          0.8238
Mean absolute error                      0.1375
```

```
Root mean squared error                  0.237
Relative absolute error                 36.663  %
Root relative squared error             54.7349 %
Coverage of cases (0.95 level)          99.3769 %
Mean rel. region size (0.95 level)      67.8106 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  926  398    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
   35  232 1052    5 |    c = address
    0   30    0 1294 |    d = cityname
```

IBk

```
Correctly Classified Instances        5293     99.9434 %
Incorrectly Classified Instances         3      0.0566 %
Kappa statistic                          0.9992
Mean absolute error                      0.0006
Root mean squared error                  0.0155
Coverage of cases (0.95 level)          99.9811 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1324    0    0    0 |    a = firstname
    0 1323    1    0 |    b = lastname
    0    0 1322    2 |    c = address
    0    0    0 1324 |    d = cityname
```

J48

```
Correctly Classified Instances        3375     63.7273 %
Incorrectly Classified Instances      1921     36.2727 %
Kappa statistic                          0.5164
Mean absolute error                      0.2086
Root mean squared error                  0.3229
Relative absolute error                 55.6227 %
Root relative squared error             74.5806 %
```

```
Coverage of cases (0.95 level)          99.9056 %
Mean rel. region size (0.95 level)      70.8837 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  238 1086    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  632  687    4 |    c = address
    0  198    0 1126 |    d = cityname
```

JRip

```
Correctly Classified Instances        4738     89.4637 %
Incorrectly Classified Instances       558     10.5363 %
Kappa statistic                          0.8595
Mean absolute error                      0.0797
Root mean squared error                  0.1996
Relative absolute error                 21.2561 %
Root relative squared error             46.1043 %
Coverage of cases (0.95 level)          99.6035 %
Mean rel. region size (0.95 level)      42.754  %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  881  443    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    2   92 1224    6 |    c = address
    0   15    0 1309 |    d = cityname
```

Naive Bayes

```
Correctly Classified Instances        4646     87.7266 %
Incorrectly Classified Instances       650     12.2734 %
Kappa statistic                          0.8364
Mean absolute error                      0.2574
Root mean squared error                  0.3392
Relative absolute error                 68.6515 %
Root relative squared error             78.3386 %
```

```
Coverage of cases (0.95 level)         99.9434 %
Mean rel. region size (0.95 level)     86.0366 %
Total Number of Instances            5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1324    0    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  390  930    3 |    c = address
    0  256    0 1068 |    d = cityname
```

REPTree

```
Correctly Classified Instances        4484     84.6677 %
Incorrectly Classified Instances       812     15.3323 %
Kappa statistic                          0.7956
Mean absolute error                      0.1023
Root mean squared error                  0.2262
Relative absolute error                 27.2904 %
Root relative squared error             52.2403 %
Coverage of cases (0.95 level)          99.5279 %
Mean rel. region size (0.95 level)      42.315  %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  815  509    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  277 1042    5 |    c = address
    0   21    0 1303 |    d = cityname
```

SMO

```
Correctly Classified Instances        5291     99.9056 %
Incorrectly Classified Instances         5      0.0944 %
Kappa statistic                          0.9987
Mean absolute error                      0.2501
Root mean squared error                  0.312
Relative absolute error                 66.696  %
Root relative squared error             72.049  %
```

```
Coverage of cases (0.95 level)          100      %
Mean rel. region size (0.95 level)       75      %
Total Number of Instances              5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1324    0    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0    2 1319    3 |    c = address
    0    0    0 1324 |    d = cityname
```

Appendix B

Classification results from chapter 4.2

B.1 WTK = 50

DecisionTable

```
Correctly Classified Instances        52526    63.4311 %
Incorrectly Classified Instances      30282    36.5689 %
Kappa statistic                           0.5124
Mean absolute error                       0.2059
Root mean squared error                   0.3197
Relative absolute error                  54.917  %
Root relative squared error              73.821  %
Coverage of cases (0.95 level)           98.377  %
Mean rel. region size (0.95 level)       56.2844 %
Total Number of Instances             82808

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
  2543 18140    19     0 |     a = firstname
     5 20697     0     0 |     b = lastname
    33 10770  9889    10 |     c = address
     1  1304     0 19397 |     d = cityname
```

IBk

```
Correctly Classified Instances        53172    64.2112 %
Incorrectly Classified Instances      29636    35.7888 %
Kappa statistic                           0.5228
Mean absolute error                       0.1936
Root mean squared error                   0.3111
Coverage of cases (0.95 level)           98.3287 %
```

```
Total Number of Instances              82808

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
 20643    40    19     0 |     a = firstname
 17487  3215     0     0 |     b = lastname
 10776     0  9916    10 |     c = address
  1304     0     0 19398 |     d = cityname
```

J48

```
Correctly Classified Instances         53171    64.21    %
Incorrectly Classified Instances       29637    35.79    %
Kappa statistic                            0.5228
Mean absolute error                        0.2018
Root mean squared error                    0.3177
Relative absolute error                   53.8184 %
Root relative squared error               73.361  %
Coverage of cases (0.95 level)            98.3407 %
Mean rel. region size (0.95 level)        55.3171 %
Total Number of Instances              82808

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
 20643    40    19     0 |     a = firstname
 17487  3215     0     0 |     b = lastname
 10776     0  9915    11 |     c = address
  1304     0     0 19398 |     d = cityname
```

JRip

```
Correctly Classified Instances         36375    43.9269 %
Incorrectly Classified Instances       46433    56.0731 %
Kappa statistic                            0.2524
Mean absolute error                        0.2997
Root mean squared error                    0.3871
Relative absolute error                   79.9252 %
Root relative squared error               89.4009 %
Coverage of cases (0.95 level)            99.9143 %
Mean rel. region size (0.95 level)        85.7396 %
```

```
Total Number of Instances              82808

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
  2543    40    19 18100 |     a = firstname
     5  3215     0 17482 |     b = lastname
     6     0  9916 10780 |     c = address
     0     0     1 20701 |     d = cityname
```

Naive Bayes

```
Correctly Classified Instances         49345    59.5897 %
Incorrectly Classified Instances       33463    40.4103 %
Kappa statistic                            0.4612
Mean absolute error                        0.2793
Root mean squared error                    0.3604
Coverage of cases (0.95 level)           100       %
Total Number of Instances              82808

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
 20643    40    19     0 |     a = firstname
 17487  3215     0     0 |     b = lastname
 10803     0  9898     1 |     c = address
  5113     0     0 15589 |     d = cityname
```

REPTree

```
Correctly Classified Instances         53172    64.2112 %
Incorrectly Classified Instances       29636    35.7888 %
Kappa statistic                            0.5228
Mean absolute error                        0.1936
Root mean squared error                    0.3111
Relative absolute error                   51.6168 %
Root relative squared error               71.8449 %
Coverage of cases (0.95 level)            98.3287 %
Mean rel. region size (0.95 level)        53.775  %
Total Number of Instances              82808

=== Confusion Matrix ===
      a      b      c      d   <-- classified as
  20643     40     19      0 |     a = firstname
  17487   3215      0      0 |     b = lastname
  10776      0   9916     10 |     c = address
   1304      0      0  19398 |     d = cityname
```

SMO

```
 Correctly Classified Instances       53171    64.21   %
 Incorrectly Classified Instances     29637    35.79   %
Kappa statistic                           0.5228
Mean absolute error                       0.2933
Root mean squared error                   0.3748
Relative absolute error                  78.2101 %
Root relative squared error              86.5672 %
Coverage of cases (0.95 level)           98.4253 %
Mean rel. region size (0.95 level)       75      %
Total Number of Instances             82808

=== Confusion Matrix ===

      a      b      c      d   <-- classified as
  20643     40     19      0 |     a = firstname
  17487   3215      0      0 |     b = lastname
  10776      0   9915     11 |     c = address
   1304      0      0  19398 |     d = cityname
```

B.2 WTK = 100

DecisionTable

```
 Correctly Classified Instances       59088    71.3554 %
 Incorrectly Classified Instances     23720    28.6446 %
Kappa statistic                           0.6181
Mean absolute error                       0.1754
Root mean squared error                   0.2933
Relative absolute error                  46.7699 %
Root relative squared error              67.7419 %
Coverage of cases (0.95 level)           99.5761 %
Mean rel. region size (0.95 level)       53.6056 %
```

```
Total Number of Instances             82808

=== Confusion Matrix ===

    a     b     c     d   <-- classified as
 6307 14376    19     0 |     a = firstname
    8 20694     0     0 |     b = lastname
   70  8899 11691    42 |     c = address
    2   304     0 20396 |     d = cityname
```

IBk

```
Correctly Classified Instances        59198     71.4883 %
Incorrectly Classified Instances      23610     28.5117 %
Kappa statistic                           0.6198
Mean absolute error                       0.1553
Root mean squared error                   0.2787
Coverage of cases (0.95 level)           99.5085 %
Total Number of Instances             82808

=== Confusion Matrix ===

    a     b     c     d   <-- classified as
 6307 14376    19     0 |     a = firstname
    8 20694     0     0 |     b = lastname
    5  8856 11799    42 |     c = address
    0   304     0 20398 |     d = cityname
```

J48

```
Correctly Classified Instances        59145     71.4243 %
Incorrectly Classified Instances      23663     28.5757 %
Kappa statistic                           0.619
Mean absolute error                       0.1709
Root mean squared error                   0.2923
Relative absolute error                  45.5656 %
Root relative squared error              67.5023 %
Coverage of cases (0.95 level)           99.5592 %
Mean rel. region size (0.95 level)       51.815  %
Total Number of Instances             82808

=== Confusion Matrix ===
     a     b     c     d   <-- classified as
  6307 14376    19     0 |     a = firstname
     8 20694     0     0 |     b = lastname
     6  8905 11746    45 |     c = address
     0   304     0 20398 |     d = cityname
```

JRip

```
Correctly Classified Instances         44078     53.2292 %
Incorrectly Classified Instances       38730     46.7708 %
Kappa statistic                            0.3764
Mean absolute error                        0.2627
Root mean squared error                    0.3624
Relative absolute error                   70.0587 %
Root relative squared error               83.7011 %
Coverage of cases (0.95 level)            99.8901 %
Mean rel. region size (0.95 level)        78.7439 %
Total Number of Instances              82808

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
  6307    56    19 14320 |     a = firstname
     8  5322     0 15372 |     b = lastname
     6     0 11749  8947 |     c = address
     0     0     2 20700 |     d = cityname
```

Naive Bayes

```
Correctly Classified Instances         54379     65.6688 %
Incorrectly Classified Instances       28429     34.3312 %
Kappa statistic                            0.5423
Mean absolute error                        0.2765
Root mean squared error                    0.3568
Relative absolute error                   73.7236 %
Root relative squared error               82.4093 %
Coverage of cases (0.95 level)           100      %
Mean rel. region size (0.95 level)        88.5603 %
Total Number of Instances              82808

=== Confusion Matrix ===
     a     b     c     d   <-- classified as
  6307 14376    19     0 |     a = firstname
     8 20694     0     0 |     b = lastname
    15  8897 11789     1 |     c = address
     0  5113     0 15589 |     d = cityname
```

REPTree

```
Correctly Classified Instances       59184     71.4714 %
Incorrectly Classified Instances     23624     28.5286 %
Kappa statistic                          0.6196
Mean absolute error                      0.1555
Root mean squared error                  0.2788
Relative absolute error                 41.4555 %
Root relative squared error             64.3859 %
Coverage of cases (0.95 level)          99.4868 %
Mean rel. region size (0.95 level)      48.5418 %
Total Number of Instances            82808

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
  6307 14376    19     0 |     a = firstname
     8 20694     0     0 |     b = lastname
    11  8862 11785    44 |     c = address
     0   304     0 20398 |     d = cityname
```

SMO

```
Correctly Classified Instances       59198     71.4883 %
Incorrectly Classified Instances     23610     28.5117 %
Kappa statistic                          0.6198
Mean absolute error                      0.2833
Root mean squared error                  0.3613
Relative absolute error                 75.5424 %
Root relative squared error             83.4285 %
Coverage of cases (0.95 level)          99.6329 %
Mean rel. region size (0.95 level)      75      %
Total Number of Instances            82808

=== Confusion Matrix ===
     a      b      c      d   <-- classified as
  6307  14376     19      0 |       a = firstname
     8  20694      0      0 |       b = lastname
     5   8856  11799     42 |       c = address
     0    304      0  20398 |       d = cityname
```

B.3 WTK = 200

DecisionTable

```
Correctly Classified Instances        65303     78.8607 %
Incorrectly Classified Instances      17505     21.1393 %
Kappa statistic                           0.7181
Mean absolute error                       0.1457
Root mean squared error                   0.2654
Relative absolute error                  38.8509 %
Root relative squared error              61.2874 %
Coverage of cases (0.95 level)           99.7029 %
Mean rel. region size (0.95 level)       50.5543 %
Total Number of Instances             82808

=== Confusion Matrix ===

     a      b      c      d   <-- classified as
 10909   9771     22      0 |       a = firstname
    11  20691      0      0 |       b = lastname
   137   7316  13206     43 |       c = address
     2    202      1  20497 |       d = cityname
```

IBk

```
Correctly Classified Instances        65612     79.2339 %
Incorrectly Classified Instances      17196     20.7661 %
Kappa statistic                           0.7231
Mean absolute error                       0.118
Root mean squared error                   0.2429
Coverage of cases (0.95 level)           99.593  %
Total Number of Instances             82808

=== Confusion Matrix ===
     a     b     c     d   <-- classified as
 10917  9763    22     0 |     a = firstname
    11 20691     0     0 |     b = lastname
     6  7148 13505    43 |     c = address
     0   202     1 20499 |     d = cityname
```

J48

```
Correctly Classified Instances       65531     79.1361 %
Incorrectly Classified Instances     17277     20.8639 %
Kappa statistic                          0.7218
Mean absolute error                      0.138
Root mean squared error                  0.2627
Relative absolute error                 36.7947 %
Root relative squared error             60.6586 %
Coverage of cases (0.95 level)          99.6752 %
Mean rel. region size (0.95 level)      47.9594 %
Total Number of Instances            82808

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
 10909  9772    21     0 |     a = firstname
    11 20691     0     0 |     b = lastname
     7  7216 13432    47 |     c = address
     0   202     1 20499 |     d = cityname
```

JRip

```
Correctly Classified Instances       53430     64.5228 %
Incorrectly Classified Instances     29378     35.4772 %
Kappa statistic                          0.527
Mean absolute error                      0.2154
Root mean squared error                  0.3281
Relative absolute error                 57.4297 %
Root relative squared error             75.7824 %
Coverage of cases (0.95 level)          99.8466 %
Mean rel. region size (0.95 level)      70.2402 %
Total Number of Instances            82808

=== Confusion Matrix ===
     a     b     c     d   <-- classified as
 10909    89    19  9685 |     a = firstname
    11  8432     0 12259 |     b = lastname
     5     0 13390  7307 |     c = address
     0     0     3 20699 |     d = cityname
```

Naive Bayes

```
Correctly Classified Instances       59160     71.4424 %
Incorrectly Classified Instances     23648     28.5576 %
Kappa statistic                          0.6192
Mean absolute error                      0.2747
Root mean squared error                  0.3547
Relative absolute error                 73.265  %
Root relative squared error             81.903  %
Coverage of cases (0.95 level)         100      %
Mean rel. region size (0.95 level)      88.5207 %
Total Number of Instances            82808

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
 10920  9763    19     0 |     a = firstname
    11 20691     0     0 |     b = lastname
    38  8703 11960     1 |     c = address
     0  5113     0 15589 |     d = cityname
```

REPTree

```
Correctly Classified Instances       65594     79.2122 %
Incorrectly Classified Instances     17214     20.7878 %
Kappa statistic                          0.7228
Mean absolute error                      0.1182
Root mean squared error                  0.2431
Relative absolute error                 31.5127 %
Root relative squared error             56.1362 %
Coverage of cases (0.95 level)          99.5616 %
Mean rel. region size (0.95 level)      42.7755 %
Total Number of Instances            82808

=== Confusion Matrix ===
     a     b     c     d   <-- classified as
 10917  9763    22     0 |     a = firstname
    11 20691     0     0 |     b = lastname
    13  7156 13487    46 |     c = address
     0   202     1 20499 |     d = cityname
```

SMO

```
Correctly Classified Instances       65611     79.2327 %
Incorrectly Classified Instances     17197     20.7673 %
Kappa statistic                          0.7231
Mean absolute error                      0.2749
Root mean squared error                  0.3495
Relative absolute error                 73.3083 %
Root relative squared error             80.7063 %
Coverage of cases (0.95 level)          99.7561 %
Mean rel. region size (0.95 level)      75      %
Total Number of Instances            82808

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
 10917  9763    22     0 |     a = firstname
    11 20691     0     0 |     b = lastname
     6  7148 13504    44 |     c = address
     0   202     1 20499 |     d = cityname
```

B.4 WTK = 500

DecisionTable

```
Correctly Classified Instances       72247     87.2464 %
Incorrectly Classified Instances     10561     12.7536 %
Kappa statistic                          0.83
Mean absolute error                      0.1043
Root mean squared error                  0.2198
Relative absolute error                 27.8007 %
Root relative squared error             50.7562 %
Coverage of cases (0.95 level)          99.7561 %
Mean rel. region size (0.95 level)      48.9092 %
```

```
Total Number of Instances              82808

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
 14488  6195    19     0 |     a = firstname
    16 20686     0     0 |     b = lastname
   462  3677 16533    30 |     c = address
     5   157     0 20540 |     d = cityname
```

IBk

```
Correctly Classified Instances         73270    88.4818 %
Incorrectly Classified Instances        9538    11.5182 %
Kappa statistic                            0.8464
Mean absolute error                        0.0709
Root mean squared error                    0.1883
Coverage of cases (0.95 level)            99.622  %
Total Number of Instances              82808

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
 14514  6176    12     0 |     a = firstname
    16 20670    16     0 |     b = lastname
    13  3120 17539    30 |     c = address
     0   155     0 20547 |     d = cityname
```

J48

```
Correctly Classified Instances         72568    87.634  %
Incorrectly Classified Instances       10240    12.366  %
Kappa statistic                            0.8351
Mean absolute error                        0.093
Root mean squared error                    0.2156
Relative absolute error                   24.7961 %
Root relative squared error               49.7957 %
Coverage of cases (0.95 level)            99.7271 %
Mean rel. region size (0.95 level)        43.6929 %
Total Number of Instances              82808

=== Confusion Matrix ===
    a     b     c     d   <-- classified as
14403  6283    16     0 |     a = firstname
   16 20671    15     0 |     b = lastname
    7  3699 16949    47 |     c = address
    0   157     0 20545 |     d = cityname
```

JRip

```
Correctly Classified Instances        61785     74.6124 %
Incorrectly Classified Instances      21023     25.3876 %
Kappa statistic                           0.6615
Mean absolute error                       0.165
Root mean squared error                   0.2872
Relative absolute error                  43.995  %
Root relative squared error              66.3287 %
Coverage of cases (0.95 level)           99.8333 %
Mean rel. region size (0.95 level)       62.6621 %
Total Number of Instances             82808

=== Confusion Matrix ===

    a     b     c     d   <-- classified as
14375    94    17  6216 |     a = firstname
   16  9503     0 11183 |     b = lastname
    7     0 17209  3486 |     c = address
    0     0     4 20698 |     d = cityname
```

Naive Bayes

```
Correctly Classified Instances        63120     76.2245 %
Incorrectly Classified Instances      19688     23.7755 %
Kappa statistic                           0.683
Mean absolute error                       0.2738
Root mean squared error                   0.3534
Relative absolute error                  73.0018 %
Root relative squared error              81.6047 %
Coverage of cases (0.95 level)          100      %
Mean rel. region size (0.95 level)       88.5038 %
Total Number of Instances             82808

=== Confusion Matrix ===
     a     b     c     d   <-- classified as
 14507  6176    19     0 |     a = firstname
    16 20686     0     0 |     b = lastname
    43  8320 12338     1 |     c = address
     0  5113     0 15589 |     d = cityname
```

REPTree

```
Correctly Classified Instances       73208     88.4069 %
Incorrectly Classified Instances      9600     11.5931 %
Kappa statistic                          0.8454
Mean absolute error                      0.0715
Root mean squared error                  0.1891
Relative absolute error                 19.0701 %
Root relative squared error             43.6693 %
Coverage of cases (0.95 level)          99.5918 %
Mean rel. region size (0.95 level)      36.2003 %
Total Number of Instances            82808

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
 14502  6179    21     0 |     a = firstname
    16 20670    16     0 |     b = lastname
    12  3155 17489    46 |     c = address
     0   155     0 20547 |     d = cityname
```

SMO

```
Correctly Classified Instances       73270     88.4818 %
Incorrectly Classified Instances      9538     11.5182 %
Kappa statistic                          0.8464
Mean absolute error                      0.2631
Root mean squared error                  0.3321
Relative absolute error                 70.1467 %
Root relative squared error             76.689  %
Coverage of cases (0.95 level)          99.8128 %
Mean rel. region size (0.95 level)      75      %
Total Number of Instances            82808

=== Confusion Matrix ===
     a     b     c     d   <-- classified as
 14514  6176    12     0 |     a = firstname
    16 20670    16     0 |     b = lastname
    13  3120 17539    30 |     c = address
     0   155     0 20547 |     d = cityname
```

Appendix C

Classification results from chapter 4.3

C.1 WTK = 50

DecisionTable

```
Correctly Classified Instances         3601    67.9947 %
Incorrectly Classified Instances       1695    32.0053 %
Kappa statistic                           0.5733
Mean absolute error                       0.2041
Root mean squared error                   0.3105
Relative absolute error                  54.4312 %
Root relative squared error              71.7084 %
Coverage of cases (0.95 level)           99.9434 %
Mean rel. region size (0.95 level)       78.4366 %
Total Number of Instances              5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  342  982    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
   28  507  785    4 |    c = address
    0  174    0 1150 |    d = cityname
```

IBk

```
Correctly Classified Instances         3676    69.4109 %
Incorrectly Classified Instances       1620    30.5891 %
Kappa statistic                           0.5921
Mean absolute error                       0.1775
Root mean squared error                   0.2979
```

```
Relative absolute error                 47.3293 %
Root relative squared error             68.8036 %
Coverage of cases (0.95 level)          99.9245 %
Mean rel. region size (0.95 level)      64.7611 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  343  981    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  460  859    4 |    c = address
    0  174    0 1150 |    d = cityname
```

J48

```
Correctly Classified Instances        3233     61.0461 %
Incorrectly Classified Instances      2063     38.9539 %
Kappa statistic                          0.4806
Mean absolute error                      0.22
Root mean squared error                  0.3321
Relative absolute error                 58.6625 %
Root relative squared error             76.6888 %
Coverage of cases (0.95 level)          99.9245 %
Mean rel. region size (0.95 level)      72.9088 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  206 1118    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  720  600    4 |    c = address
    0  221    0 1103 |    d = cityname
```

JRip

```
Correctly Classified Instances        3195     60.3285 %
Incorrectly Classified Instances      2101     39.6715 %
Kappa statistic                          0.471
Mean absolute error                      0.221
Root mean squared error                  0.3325
```

```
Relative absolute error                 58.9334 %
Root relative squared error             76.7913 %
Coverage of cases (0.95 level)          98.9992 %
Mean rel. region size (0.95 level)      68.2307 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  636  199  292  197 |    a = firstname
  359  379  352  234 |    b = lastname
  144   91  998   91 |    c = address
   56   37   49 1182 |    d = cityname
```

Naive Bayes

```
Correctly Classified Instances        3595    67.8814 %
Incorrectly Classified Instances      1701    32.1186 %
Kappa statistic                          0.5718
Mean absolute error                      0.2654
Root mean squared error                  0.3494
Relative absolute error                 70.7662 %
Root relative squared error             80.685  %
Coverage of cases (0.95 level)          99.9434 %
Mean rel. region size (0.95 level)      86.2963 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  343  981    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  460  860    3 |    c = address
    0  256    0 1068 |    d = cityname
```

REPTree

```
Correctly Classified Instances        3677    69.4298 %
Incorrectly Classified Instances      1619    30.5702 %
Kappa statistic                          0.5924
Mean absolute error                      0.1774
Root mean squared error                  0.2978
```

```
Relative absolute error                  47.3084 %
Root relative squared error              68.7854 %
Coverage of cases (0.95 level)           99.9245 %
Mean rel. region size (0.95 level)       64.7234 %
Total Number of Instances              5296


=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  343  981    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  460  860    4 |    c = address
    0  174    0 1150 |    d = cityname
```

SMO

```
Correctly Classified Instances         3677     69.4298 %
Incorrectly Classified Instances       1619     30.5702 %
Kappa statistic                           0.5924
Mean absolute error                       0.2883
Root mean squared error                   0.3681
Relative absolute error                  76.8672 %
Root relative squared error              85.0018 %
Coverage of cases (0.95 level)           96.6767 %
Mean rel. region size (0.95 level)       75      %
Total Number of Instances              5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  343  981    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  460  860    4 |    c = address
    0  174    0 1150 |    d = cityname
```

C.2 WTK = 100

DecisionTable

```
Correctly Classified Instances         4105     77.5113 %
```

```
Incorrectly Classified Instances        1191     22.4887 %
Kappa statistic                            0.7002
Mean absolute error                        0.1708
Root mean squared error                    0.2762
Relative absolute error                   45.5487 %
Root relative squared error               63.7856 %
Coverage of cases (0.95 level)            98.6971 %
Mean rel. region size (0.95 level)        66.5644 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  655  669    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
   24  428  868    4 |    c = address
    0   66    0 1258 |    d = cityname
```

IBk

```
Correctly Classified Instances          4178     78.8897 %
Incorrectly Classified Instances        1118     21.1103 %
Kappa statistic                            0.7185
Mean absolute error                        0.1298
Root mean squared error                    0.2549
Relative absolute error                   34.6192 %
Root relative squared error               58.868  %
Coverage of cases (0.95 level)            98.6782 %
Mean rel. region size (0.95 level)        45.1378 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  661  663    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  384  935    4 |    c = address
    0   66    0 1258 |    d = cityname
```

J48

```
Correctly Classified Instances          3233     61.0461 %
```

```
Incorrectly Classified Instances      2063    38.9539 %
Kappa statistic                          0.4806
Mean absolute error                      0.22
Root mean squared error                  0.3321
Relative absolute error                 58.6625 %
Root relative squared error             76.6888 %
Coverage of cases (0.95 level)          99.9245 %
Mean rel. region size (0.95 level)      72.9088 %
Total Number of Instances             5296

=== Confusion Matrix ===

   a    b    c    d   <-- classified as
 206 1118    0    0 |    a = firstname
   0 1324    0    0 |    b = lastname
   0  720  600    4 |    c = address
   0  221    0 1103 |    d = cityname
```

JRip

```
Correctly Classified Instances        3707    69.9962 %
Incorrectly Classified Instances      1589    30.0038 %
Kappa statistic                          0.5999
Mean absolute error                      0.181
Root mean squared error                  0.3012
Relative absolute error                 48.2649 %
Root relative squared error             69.5597 %
Coverage of cases (0.95 level)          99.0559 %
Mean rel. region size (0.95 level)      58.7897 %
Total Number of Instances             5296

=== Confusion Matrix ===

   a    b    c    d   <-- classified as
 803  267    0  254 |    a = firstname
 202  699    0  423 |    b = lastname
  85  165  927  147 |    c = address
  13   33    0 1278 |    d = cityname
```

Naive Bayes

```
Correctly Classified Instances        3955    74.679  %
```

```
Incorrectly Classified Instances        1341     25.321  %
Kappa statistic                            0.6624
Mean absolute error                        0.262
Root mean squared error                    0.345
Relative absolute error                   69.8647 %
Root relative squared error               79.666  %
Coverage of cases (0.95 level)            99.9434 %
Mean rel. region size (0.95 level)        86.2679 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  661  663    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  418  902    3 |    c = address
    0  256    0 1068 |    d = cityname
```

REPTree

```
Correctly Classified Instances          4172     78.7764 %
Incorrectly Classified Instances        1124     21.2236 %
Kappa statistic                            0.717
Mean absolute error                        0.1305
Root mean squared error                    0.2556
Relative absolute error                   34.7933 %
Root relative squared error               59.0222 %
Coverage of cases (0.95 level)            98.6782 %
Mean rel. region size (0.95 level)        45.1945 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  661  663    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  391  929    4 |    c = address
    0   66    0 1258 |    d = cityname
```

SMO

```
Correctly Classified Instances          4180     78.9275 %
```

```
Incorrectly Classified Instances       1116    21.0725 %
Kappa statistic                           0.719
Mean absolute error                       0.2757
Root mean squared error                   0.3507
Relative absolute error                  73.5314 %
Root relative squared error              80.9823 %
Coverage of cases (0.95 level)           98.716  %
Mean rel. region size (0.95 level)       75      %
Total Number of Instances              5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  661  663    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  383  937    4 |    c = address
    0   66    0 1258 |    d = cityname
```

C.3 WTK = 200

DecisionTable

```
Correctly Classified Instances         4341    81.9675 %
Incorrectly Classified Instances        955    18.0325 %
Kappa statistic                           0.7596
Mean absolute error                       0.1561
Root mean squared error                   0.2591
Relative absolute error                  41.6357 %
Root relative squared error              59.8414 %
Coverage of cases (0.95 level)           99.0937 %
Mean rel. region size (0.95 level)       68.2213 %
Total Number of Instances              5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  821  503    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
   48  354  917    5 |    c = address
    0   45    0 1279 |    d = cityname
```

IBk

```
Correctly Classified Instances          4479     84.5733 %
Incorrectly Classified Instances         817     15.4267 %
Kappa statistic                            0.7943
Mean absolute error                        0.0961
Root mean squared error                    0.2197
Relative absolute error                   25.635  %
Root relative squared error               50.7389 %
Coverage of cases (0.95 level)            99.1881 %
Mean rel. region size (0.95 level)        40.0538 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  845  479    0    0 |    a = firstname
    0 1322    0    2 |    b = lastname
    1  292 1026    5 |    c = address
    0   38    0 1286 |    d = cityname
```

J48

```
Correctly Classified Instances          3233     61.0461 %
Incorrectly Classified Instances        2063     38.9539 %
Kappa statistic                            0.4806
Mean absolute error                        0.22
Root mean squared error                    0.3321
Relative absolute error                   58.6625 %
Root relative squared error               76.6888 %
Coverage of cases (0.95 level)            99.9245 %
Mean rel. region size (0.95 level)        72.9088 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  206 1118    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  720  600    4 |    c = address
    0  221    0 1103 |    d = cityname
```

JRip

```
Correctly Classified Instances        4081     77.0582 %
Incorrectly Classified Instances      1215     22.9418 %
Kappa statistic                          0.6941
Mean absolute error                      0.1443
Root mean squared error                  0.2701
Relative absolute error                 38.4871 %
Root relative squared error             62.3775 %
Coverage of cases (0.95 level)          99.2636 %
Mean rel. region size (0.95 level)      51.487  %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  862  340   50   72 |    a = firstname
  205  914  100  105 |    b = lastname
   62  195 1038   29 |    c = address
    4   49    4 1267 |    d = cityname
```

Naive Bayes

```
Correctly Classified Instances        4159     78.531  %
Incorrectly Classified Instances      1137     21.469  %
Kappa statistic                          0.7137
Mean absolute error                      0.2605
Root mean squared error                  0.343
Relative absolute error                 69.4752 %
Root relative squared error             79.212  %
Coverage of cases (0.95 level)          99.9434 %
Mean rel. region size (0.95 level)      86.2632 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  845  479    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  398  922    3 |    c = address
    0  256    0 1068 |    d = cityname
```

REPTree

```
Correctly Classified Instances        4434    83.7236 %
Incorrectly Classified Instances       862    16.2764 %
Kappa statistic                          0.783
Mean absolute error                      0.1046
Root mean squared error                  0.2288
Relative absolute error                 27.8975 %
Root relative squared error             52.834  %
Coverage of cases (0.95 level)          99.1692 %
Mean rel. region size (0.95 level)      41.6163 %
Total Number of Instances             5296

=== Confusion Matrix ===

   a    b    c    d   <-- classified as
 827  497    0    0 |    a = firstname
   0 1322    0    2 |    b = lastname
   0  319 1000    5 |    c = address
   0   39    0 1285 |    d = cityname
```

SMO

```
Correctly Classified Instances        4482    84.6299 %
Incorrectly Classified Instances       814    15.3701 %
Kappa statistic                          0.7951
Mean absolute error                      0.2686
Root mean squared error                  0.3404
Relative absolute error                 71.6348 %
Root relative squared error             78.6054 %
Coverage of cases (0.95 level)          99.2258 %
Mean rel. region size (0.95 level)      75      %
Total Number of Instances             5296

=== Confusion Matrix ===

   a    b    c    d   <-- classified as
 845  479    0    0 |    a = firstname
   0 1322    0    2 |    b = lastname
   0  288 1031    5 |    c = address
   0   40    0 1284 |    d = cityname
```

C.4 WTK = 500

DecisionTable

```
Correctly Classified Instances         4377    82.6473 %
Incorrectly Classified Instances        919    17.3527 %
Kappa statistic                           0.7686
Mean absolute error                       0.1508
Root mean squared error                   0.2573
Relative absolute error                  40.2072 %
Root relative squared error              59.4163 %
Coverage of cases (0.95 level)           99.0937 %
Mean rel. region size (0.95 level)       68.226  %
Total Number of Instances              5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  847  477    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
   43  349  927    5 |    c = address
    0   45    0 1279 |    d = cityname
```

IBk

```
Correctly Classified Instances         4906    92.636  %
Incorrectly Classified Instances        390    7.364  %
Kappa statistic                           0.9018
Mean absolute error                       0.0438
Root mean squared error                   0.1569
Relative absolute error                  11.6896 %
Root relative squared error              36.224  %
Coverage of cases (0.95 level)           99.6224 %
Mean rel. region size (0.95 level)       31.9675 %
Total Number of Instances              5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1145  179    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    2  189 1128    5 |    c = address
```

```
      0    15     0 1309 |     d = cityname
```

J48

```
Correctly Classified Instances        3233     61.0461 %
Incorrectly Classified Instances      2063     38.9539 %
Kappa statistic                          0.4806
Mean absolute error                      0.22
Root mean squared error                  0.3321
Relative absolute error                 58.6625 %
Root relative squared error             76.6888 %
Coverage of cases (0.95 level)          99.9245 %
Mean rel. region size (0.95 level)      72.9088 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  206 1118    0    0 |     a = firstname
    0 1324    0    0 |     b = lastname
    0  720  600    4 |     c = address
    0  221    0 1103 |     d = cityname
```

JRip

```
Correctly Classified Instances        3890     73.4517 %
Incorrectly Classified Instances      1406     26.5483 %
Kappa statistic                          0.646
Mean absolute error                      0.1565
Root mean squared error                  0.2833
Relative absolute error                 41.7341 %
Root relative squared error             65.4271 %
Coverage of cases (0.95 level)          99.5279 %
Mean rel. region size (0.95 level)      55.0085 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  793  317   54  160 |     a = firstname
  226  768  122  208 |     b = lastname
   48  170 1051   55 |     c = address
```

```
    4    40      2  1278  |       d = cityname
```

Naive Bayes

```
Correctly Classified Instances          4465     84.3089 %
Incorrectly Classified Instances         831     15.6911 %
Kappa statistic                            0.7908
Mean absolute error                        0.2596
Root mean squared error                    0.3417
Relative absolute error                   69.2237 %
Root relative squared error               78.918  %
Coverage of cases (0.95 level)            99.9434 %
Mean rel. region size (0.95 level)        86.2632 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a     b     c     d   <-- classified as
 1150   174     0     0  |     a = firstname
    0  1324     0     0  |     b = lastname
    1   397   923     3  |     c = address
    0   256     0  1068  |     d = cityname
```

REPTree

```
Correctly Classified Instances          4382     82.7417 %
Incorrectly Classified Instances         914     17.2583 %
Kappa statistic                            0.7699
Mean absolute error                        0.1119
Root mean squared error                    0.2396
Relative absolute error                   29.8307 %
Root relative squared error               55.3347 %
Coverage of cases (0.95 level)            99.3202 %
Mean rel. region size (0.95 level)        43.8822 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a     b     c     d   <-- classified as
  776   548     0     0  |     a = firstname
    0  1323     0     1  |     b = lastname
    0   329   990     5  |     c = address
    0   31     0 1293 |     d = cityname
```

SMO

```
Correctly Classified Instances         4978     93.9955 %
Incorrectly Classified Instances        318      6.0045 %
Kappa statistic                           0.9199
Mean absolute error                       0.257
Root mean squared error                   0.3228
Relative absolute error                  68.5213 %
Root relative squared error              74.5394 %
Coverage of cases (0.95 level)           99.679  %
Mean rel. region size (0.95 level)       75      %
Total Number of Instances              5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1145  179    0    0 |     a = firstname
    0 1324    0    0 |     b = lastname
    0  119 1200    5 |     c = address
    0   15    0 1309 |     d = cityname
```

APPENDIX D

Classification results from chapter 4.4

D.1 WTK = 50

DecisionTable

```
Correctly Classified Instances       1324    25      %
Incorrectly Classified Instances     3972    75      %
Kappa statistic                         0
Mean absolute error                     0.375
Root mean squared error                 0.433
Coverage of cases (0.95 level)        100        %
Total Number of Instances            5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1324    0    0    0 |   a = firstname
 1324    0    0    0 |   b = lastname
 1324    0    0    0 |   c = address
 1324    0    0    0 |   d = cityname
```

IBk

```
Correctly Classified Instances       3509    66.2576 %
Incorrectly Classified Instances     1787    33.7424 %
Kappa statistic                         0.5501
Mean absolute error                     0.1944
Root mean squared error                 0.3105
Coverage of cases (0.95 level)         99.9245 %
Total Number of Instances            5296

=== Confusion Matrix ===
    a    b    c    d   <-- classified as
  343  981    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  628  692    4 |    c = address
    0  174    0 1150 |    d = cityname
```

J48

```
Correctly Classified Instances        3481    65.7289 %
Incorrectly Classified Instances      1815    34.2711 %
Kappa statistic                          0.5431
Mean absolute error                      0.2113
Root mean squared error                  0.3191
Coverage of cases (0.95 level)          99.9245 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  343  981    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  656  664    4 |    c = address
    0  174    0 1150 |    d = cityname
```

JRip

```
Correctly Classified Instances        2221    41.9373 %
Incorrectly Classified Instances      3075    58.0627 %
Kappa statistic                          0.2258
Mean absolute error                      0.3006
Root mean squared error                  0.4066
Coverage of cases (0.95 level)         100      %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  111    0    0 1213 |    a = firstname
    0  134    0 1190 |    b = lastname
    0    0  652  672 |    c = address
    0    0    0 1324 |    d = cityname
```

Naive Bayes

```
Correctly Classified Instances         3454    65.219   %
Incorrectly Classified Instances       1842    34.781   %
Kappa statistic                           0.5363
Mean absolute error                       0.2762
Root mean squared error                   0.3587
Coverage of cases (0.95 level)           99.9434 %
Total Number of Instances              5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  343  981    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  634  686    3 |    c = address
    0  223    0 1101 |    d = cityname
```

REPTree

```
Correctly Classified Instances         3504    66.1631 %
Incorrectly Classified Instances       1792    33.8369 %
Kappa statistic                           0.5488
Mean absolute error                       0.2036
Root mean squared error                   0.3123
Coverage of cases (0.95 level)           99.9245 %
Total Number of Instances              5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  343  981    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  633  687    4 |    c = address
    0  174    0 1150 |    d = cityname
```

SMO

```
Correctly Classified Instances         3499    66.0687 %
Incorrectly Classified Instances       1797    33.9313 %
Kappa statistic                           0.5476
Mean absolute error                       0.2937
Root mean squared error                   0.3754
```

```
Coverage of cases (0.95 level)          96.6579 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  343  981    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
   10  628  682    4 |    c = address
    0  174    0 1150 |    d = cityname
```

D.2 WTK = 100

DecisionTable

```
Correctly Classified Instances          1324     25      %
Incorrectly Classified Instances        3972     75      %
Kappa statistic                            0
Mean absolute error                        0.375
Root mean squared error                    0.433
Coverage of cases (0.95 level)           100       %
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1324    0    0    0 |    a = firstname
 1324    0    0    0 |    b = lastname
 1324    0    0    0 |    c = address
 1324    0    0    0 |    d = cityname
```

IBk

```
Correctly Classified Instances          3982     75.1888 %
Incorrectly Classified Instances        1314     24.8112 %
Kappa statistic                            0.6692
Mean absolute error                        0.1491
Root mean squared error                    0.2728
Coverage of cases (0.95 level)            98.4517 %
Total Number of Instances               5296
```

```
=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  661  663    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  569  751    4 |    c = address
    0   78    0 1246 |    d = cityname
```

J48

```
Correctly Classified Instances        3965     74.8678 %
Incorrectly Classified Instances      1331     25.1322 %
Kappa statistic                          0.6649
Mean absolute error                      0.1708
Root mean squared error                  0.2861
Coverage of cases (0.95 level)          98.4517 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  661  663    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  586  734    4 |    c = address
    0   78    0 1246 |    d = cityname
```

JRip

```
Correctly Classified Instances        2487     46.96    %
Incorrectly Classified Instances      2809     53.04    %
Kappa statistic                          0.2928
Mean absolute error                      0.2808
Root mean squared error                  0.3918
Coverage of cases (0.95 level)         100        %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  140    0    0 1184 |    a = firstname
    0  295    0 1029 |    b = lastname
    0    0  728  596 |    c = address
```

```
     0     0     0 1324 |     d = cityname
```

Naive Bayes

```
Correctly Classified Instances          3817     72.0733 %
Incorrectly Classified Instances        1479     27.9267 %
Kappa statistic                            0.6276
Mean absolute error                        0.2735
Root mean squared error                    0.3551
Coverage of cases (0.95 level)            99.9434 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  661  663    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  589  731    3 |    c = address
    0  223    0 1101 |    d = cityname
```

REPTree

```
Correctly Classified Instances          3978     75.1133 %
Incorrectly Classified Instances        1318     24.8867 %
Kappa statistic                            0.6682
Mean absolute error                        0.1559
Root mean squared error                    0.2739
Coverage of cases (0.95 level)            98.4517 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  661  663    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  573  747    4 |    c = address
    0   78    0 1246 |    d = cityname
```

SMO

```
Correctly Classified Instances          3978     75.1133 %
Incorrectly Classified Instances        1318     24.8867 %
```

```
Kappa statistic                          0.6682
Mean absolute error                      0.2822
Root mean squared error                  0.3598
Coverage of cases (0.95 level)          98.4705 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  661  663    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    4  569  747    4 |    c = address
    0   78    0 1246 |    d = cityname
```

D.3 WTK = 200

DecisionTable

```
Correctly Classified Instances        1324     25       %
Incorrectly Classified Instances      3972     75       %
Kappa statistic                          0
Mean absolute error                      0.375
Root mean squared error                  0.433
Coverage of cases (0.95 level)         100       %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1324    0    0    0 |    a = firstname
 1324    0    0    0 |    b = lastname
 1324    0    0    0 |    c = address
 1324    0    0    0 |    d = cityname
```

IBk

```
Correctly Classified Instances        4272     80.6647 %
Incorrectly Classified Instances      1024     19.3353 %
Kappa statistic                          0.7422
Mean absolute error                      0.1155
Root mean squared error                  0.2412
```

```
Coverage of cases (0.95 level)          98.886  %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  832  492    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  473  846    5 |    c = address
    0   54    0 1270 |    d = cityname
```

J48

```
Correctly Classified Instances        4236    79.9849 %
Incorrectly Classified Instances      1060    20.0151 %
Kappa statistic                          0.7331
Mean absolute error                      0.1455
Root mean squared error                  0.2638
Coverage of cases (0.95 level)          98.886  %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  822  502    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  499  820    5 |    c = address
    0   54    0 1270 |    d = cityname
```

JRip

```
Correctly Classified Instances        3466    65.4456 %
Incorrectly Classified Instances      1830    34.5544 %
Kappa statistic                          0.5393
Mean absolute error                      0.2041
Root mean squared error                  0.3266
Coverage of cases (0.95 level)         100      %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
```

```
  822    0    0  502 |    a = firstname
    0  497    0  827 |    b = lastname
    0    0  823  501 |    c = address
    0    0    0 1324 |    d = cityname
```

Naive Bayes

```
Correctly Classified Instances        4018     75.8686 %
Incorrectly Classified Instances      1278     24.1314 %
Kappa statistic                          0.6782
Mean absolute error                      0.272
Root mean squared error                  0.3531
Coverage of cases (0.95 level)          99.9434 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  832  492    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    1  559  761    3 |    c = address
    0  223    0 1101 |    d = cityname
```

REPTree

```
Correctly Classified Instances        4268     80.5891 %
Incorrectly Classified Instances      1028     19.4109 %
Kappa statistic                          0.7412
Mean absolute error                      0.1223
Root mean squared error                  0.243
Coverage of cases (0.95 level)          98.886  %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  832  492    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    0  477  842    5 |    c = address
    0   54    0 1270 |    d = cityname
```

SMO

```
Correctly Classified Instances        4270    80.6269 %
Incorrectly Classified Instances      1026    19.3731 %
Kappa statistic                          0.7417
Mean absolute error                      0.2754
Root mean squared error                  0.3501
Coverage of cases (0.95 level)          98.9237 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  832  492    0    0 |    a = firstname
    0 1324    0    0 |    b = lastname
    2  473  844    5 |    c = address
    0   54    0 1270 |    d = cityname
```

D.4 WTK = 500

DecisionTable

```
Correctly Classified Instances        1324    25      %
Incorrectly Classified Instances      3972    75      %
Kappa statistic                         0
Mean absolute error                      0.375
Root mean squared error                  0.433
Coverage of cases (0.95 level)         100      %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 1324    0    0    0 |    a = firstname
 1324    0    0    0 |    b = lastname
 1324    0    0    0 |    c = address
 1324    0    0    0 |    d = cityname
```

IBk

```
Correctly Classified Instances        4555    86.0083 %
Incorrectly Classified Instances       741    13.9917 %
Kappa statistic                          0.8134
```

```
Mean absolute error                      0.0865
Root mean squared error                  0.2099
Coverage of cases (0.95 level)          98.9804 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  958  366    0    0 |    a = firstname
    0 1323    1    0 |    b = lastname
    0  320  999    5 |    c = address
    0   45    4 1275 |    d = cityname
```

J48

```
Correctly Classified Instances        4529     85.5174 %
Incorrectly Classified Instances       767     14.4826 %
Kappa statistic                          0.8069
Mean absolute error                      0.1106
Root mean squared error                  0.2325
Coverage of cases (0.95 level)          98.9804 %
Total Number of Instances             5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  946  378    0    0 |    a = firstname
    0 1323    1    0 |    b = lastname
    0  334  985    5 |    c = address
    0   49    0 1275 |    d = cityname
```

JRip

```
Correctly Classified Instances        3805     71.8467 %
Incorrectly Classified Instances      1491     28.1533 %
Kappa statistic                          0.6246
Mean absolute error                      0.1766
Root mean squared error                  0.299
Coverage of cases (0.95 level)         100      %
Total Number of Instances             5296

=== Confusion Matrix ===
    a     b     c     d   <-- classified as
  941     0     0   383 |     a = firstname
    0   550     0   774 |     b = lastname
    0     0   990   334 |     c = address
    0     0     0  1324 |     d = cityname
```

Naive Bayes

```
 Correctly Classified Instances        4164     78.6254 %
 Incorrectly Classified Instances      1132     21.3746 %
Kappa statistic                           0.715
Mean absolute error                       0.2713
Root mean squared error                   0.3521
 Coverage of cases (0.95 level)          99.9434 %
 Total Number of Instances             5296

=== Confusion Matrix ===

    a     b     c     d   <-- classified as
  958   366     0     0 |     a = firstname
    0  1324     0     0 |     b = lastname
    1   539   781     3 |     c = address
    0   223     0  1101 |     d = cityname
```

REPTree

```
 Correctly Classified Instances        4552     85.9517 %
 Incorrectly Classified Instances       744     14.0483 %
Kappa statistic                           0.8127
Mean absolute error                       0.091
Root mean squared error                   0.2102
 Coverage of cases (0.95 level)          98.9804 %
 Total Number of Instances             5296

=== Confusion Matrix ===

    a     b     c     d   <-- classified as
  958   366     0     0 |     a = firstname
    0  1323     1     0 |     b = lastname
    0   323   996     5 |     c = address
    0    49     0  1275 |     d = cityname
```

SMO

```
Correctly Classified Instances          4555     86.0083 %
Incorrectly Classified Instances         741     13.9917 %
Kappa statistic                            0.8134
Mean absolute error                        0.2683
Root mean squared error                    0.3399
Coverage of cases (0.95 level)            99.0181 %
Total Number of Instances               5296

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  958  366    0    0 |    a = firstname
    0 1323    1    0 |    b = lastname
    0  320  999    5 |    c = address
    0   45    4 1275 |    d = cityname
```