MASTER'S THESIS | LUND UNIVERSITY 2016

Machine Learning for Categorizing Companies in Sweden

Christian Frid

Department of Computer Science
Faculty of Engineering LTH

ISSN 1650-2884
LU-CS-EX 2016-08

Machine Learning for Categorizing Companies in Sweden

(A Study for Decision Making Support in Customer Relationship Management)

Christian Frid
christian.frid90@gmail.com

April 12, 2016

Master's thesis work carried out at Lundalogik AB.

Supervisors: Pierre Nugues, Pierre.Nugues@cs.lth.se
Anders Pålsson, anders.pålsson@lundalogik.se

Examiner: Mathias Haage, Mathias.Haage@cs.lth.se

Abstract

The number of companies in Sweden has shown a significant increase over the last five years (`www.bolagsverket.se`). Data about these companies is an important asset for the customer relationship management market. In most business areas, users are in search for new customers. Companies, such as Lundalogik AB, provide a service, where the analysts can look for customers in an interactive environment that provides useful data about other companies.

However, the majority of company names is largely unknown even to analysts. This makes it hard for them to make quick decisions about which companies that could be future customers. It is an ineffective and time consuming activity to scout through large amounts of data in search of interesting companies. This is why there is a need for a tool that compares companies to one another (within the same line of business and company form). This way the user can get quicker insights about companies.

Machine learning techniques work well within customer relationship management (Glas, 2015). This Master's thesis uses techniques in machine learning to categorize companies in regards to size and economy. It also shows how to make predictive models that foretell the category of any previously unknown company.

The results I obtained show that the companies can be clustered and labeled with meaningful descriptions. With a sufficiently large number of instances, these labels can in turn be used to create a supervised learner model with great predictive ability.

Keywords: CRM, Machine Learning, Clustering, Data Mining, Decision Trees, C4.5, K-means, Principal Components Analysis (PCA)

Acknowledgements

I would like to thank my supervisor Pierre Nugues for his technical inputs and guidance throughout this study. Without his knowledge of things it wouldn't have been possible to finish in time. I also would like to thank all the nice people at the Lime-Go team at Lundalogik AB for their support and good time they've provided during my time as a thesis worker at the company.

Contents

Chapter 1
Introduction

1.1 Customer Relationship Management

Customer relationship management (CRM) is a business technique that about tending to the needs of current or future customers to the utmost extent. The goal is to understand more about what a customer is worth to the company and their needs. It is also desired to keep customers as well as attract loyal customers. Data is collected about customers that is used in order to draw conclusions about their needs and interests. It can also be used to calculate if a relation is going to be valuable in the long run.

Customer satisfaction is a really big factor when discussing quality. In a competitive world where good customer experience is crucial, it is increasingly important to manage relations with customers. This is becoming more and more apparent for companies, as mentioned by Kostojohn et al. (2011). Communication is key in many aspects of everyday life, so is the ability to really understand the interest of a customer according to CRM. In order to increase customer satisfaction, much focus is directed towards marketing, sales and service. By doing this, a company can increase the quality of the whole product.

1.2 Lime Go

The work presented in this Master's thesis is meant to help analysts to more effectively search for customers in Lime Go. This is done by categorizing companies and giving them descriptive labels so analysts can get quicker insights about potential customers.

Lime Go (screenshot in Fig. 1.1) is an interactive environment used in internet browsers. With this tool an analyst can look up company information about economy, locations, contact information etc. It features *twin matching*: a service that presents similar companies to the one the analyst searched for. Also, call lists can be made for interesting customers to the analyst, meetings can be booked and contacts can be saved.

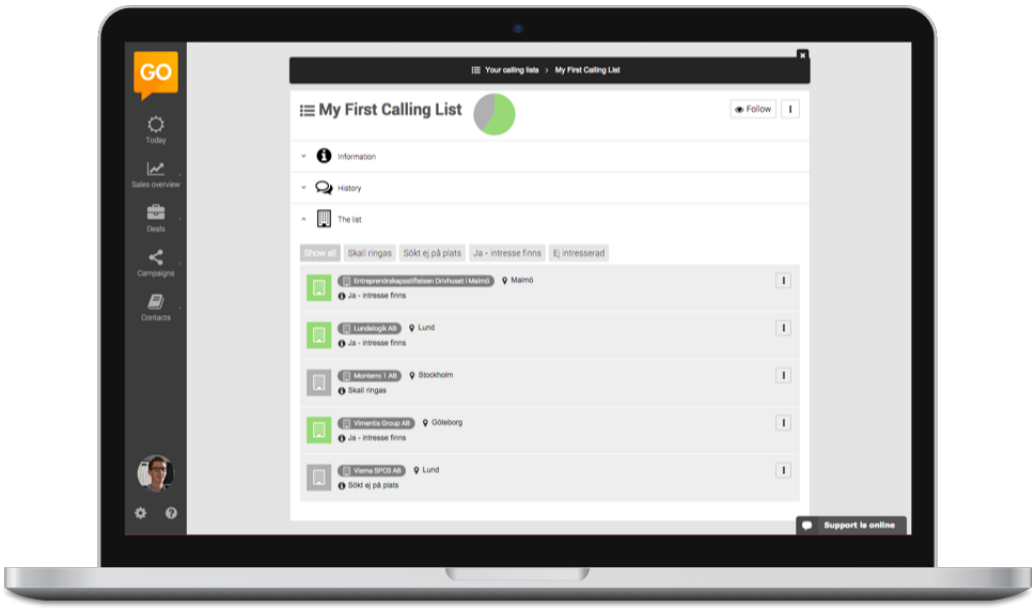

Figure 1.1: Screenshot of Lime Go `http://www.lime-go.se/tour/`.

1.3 Extracting Company Insights from Data

The techniques used in my work have been used in other contexts before but, to the best of my knowledge, they are outside the boundaries of my scope; to find groups that describe the diversity of companies in Sweden with regards to company size and economy. An economically educated person has the knowledge to, with numbers and math, say something about a company in the same regards. However, not as fast nor with as great emphasis on what's most relevant for that particular business and company form. This is what the chosen techniques present. Machine learning is a powerful tool when it comes to bringing understanding of huge amounts of data that can be hard to interpret.

Classification is a concept within machine learning. It includes a set of algorithms that given large amounts of data can describe the present, create models with predictive abilities and even, given a reward or punishment, learn how actions affects the environment. With the right algorithms supplied, a program can look at a population of instances and divide them into groups. It can then learn what characterizes a group and place an out of context instance into either of these groups.

There are two main categories in machine learning:

Supervised learning uses (preferably) large data sets where the instances have known labels. These are used to create predictive models where an unlabeled instance can be assigned a label with great precision. This category contains methods of inducting *decision trees*.

Unsupervised learning is the category that contains methods such as *clustering*, *cluster analysis* or *dimensionality reduction*. The first ones are used in segmentation discovery where homogeneous groups are located in the data.

Both of which are used in this work. In this thesis, we will analyse what algorithms are suitable for this kind of problem with respect to performance as well as reasonable estimations and satisfying results. To do this, the following questions will be answered:

- Which algorithms are best suited for this problem?
- What estimations can be considered reasonable?
- What are satisfying results?

1.4 Data Requirements

For the required methods to work, there's a need for large quantities of data. Machine learning techniques get more accurate when the size of the data sets increase. This is general among all machine learning techniques and it's also important that the data is correct. The more data the better the model and the data has to be related to the problem in some way. To few data samples may cause *over fitting*. When this happens the model has poor generalization and only works for the selected population and is useless for anything else. We want the model to be as general as possible and to work in as many cases as possible.

I extracted data from the database at Lundalogik AB. The content in the database consists of data about every company in Sweden (those that are registered at the tax agency). My data set contained data about companies with regards to two different characteristics; company size and company economy. Each of these have their own features. The features used when researching company size are features such as turnover, turnover per employee, the number of offices and more. The features regarding company economics where financial key figures like equity ratio, quick ratio, net income etc.

Some values are missing for some companies as they haven't been reported to the Swedish tax agency. Imputation has been done to fix this issue. The number zero (0) has been put in places where the values where missing for a neither positive nor negative impact on further data treatment such as the principal components analysis.

1.5 Related Work

Domingos (2012) discusses different good-to-know things about machine learning. He also mention common pitfalls and how to avoid them, important issues to focus on and common questions.

Tibshirani et al. (2001) describes how to better estimate the number of clusters to look for when using the k-means clustering algorithm.

Machine learning in combination with CRM have been used before by Glas (2015) in his work to establish a recommendations engine for CRM purposes. The methods used were clustering and decision trees for classification tasks. In his work he proves that these methods work well as tools for recommendations within the domain of CRM.

Bratel (2015) writes about integrating clustering and decision trees for the purpose of greater insights within the customer database of the IT company Narrative AB. In this

thesis, he carries out and compares many different algorithms that is of great value to my work. The tasks are related this way.

1.6 Structure of the report

After this introduction, Chapter 2 will describe the general approach to the problem at hand. It will describe how the pipeline of algorithms are set up and how different measures are used for critically judging the results. Chapter 3 will describe the algorithms used, how they work and an evaluation of the results in each step in the pipeline of algorithms. Chapter 5 will be used to discuss different matters regarding the results and explain uncertainties. Chapter 6 will summarise the results and present future work.

Chapter 2

Approach

2.1 CRISP-DM

The process is based upon the *Cross-Industry Standard Process for Data Mining* (CRISP-DM). It is an iterative six-stage process as seen in Fig. 2.1 and a standard in data-mining projects (Marbán et al., 2009). It is used by data-mining experts when tackling various problems within projects. The process is circular and good for projects where continuous improvement is a necessity.

The labels in Fig. 2.1 describe different development stages in this project. *Business Understanding* and *Data Understanding* were quickly treated thanks to competent people working at Lundalogik AB. These two stages consisted of choosing the right features as well as picking a population of company instances to work on. During *Data preparation* and *Modeling*, the data was filtered from extreme values, normalized and modeled using k-means clustering and C4.5 decision tree induction. The resulting clusters and decision trees were evaluated during *Evaluation*. If these results were unsatisfying in any way, (unbalanced trees, too few/many clusters etc.) changes were implemented so they affected the outcome in the most desirable way.

2.2 Preprocessing of data

To make a good model the right data has to be extracted and treated. The data is normalized so that the range of values becomes zero to one. This hinders too large or too small values from standing out too much since values from different features with different means will be plotted against each other. Normalization keeps the relation between the values intact so that the relative difference between them are the same as when they weren't normalized.

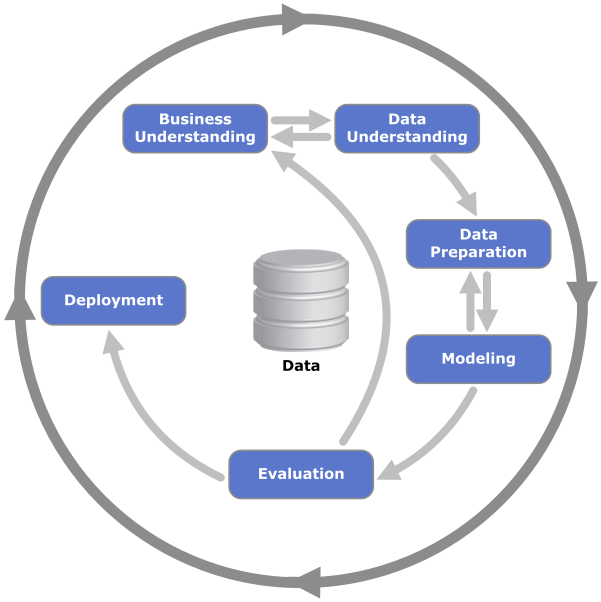

Figure 2.1: The CRISP-DM Diagram

2.3 Use of Algorithms

There are numerous algorithms within machine learning that are suitable for this kind of problem and therefore a selected few were chosen to determine which ones work the best. This is done by measuring their performance in their respective sub tasks as the basis for evaluation and selection. The types of learning that have been used are *supervised learning* and *unsupervised learning*.

2.3.1 Supervised Learning

Supervised learning teaches a program to classify instances by training a predictive model out of a labeled data set.

There are mainly two types of supervised learning; **classification** and **regression analysis**. During regression analysis, a data set is used to train a predictive model that predicts a future value. This is useful in fields such as the stock market where the price of stock is uncertain. Another common application of using supervised learning is by determining the output label (class) of an input instance x_i. The whole process is called classification and the model that is created through the training data is called a **classifier**. When the model is considered good enough it is used on real world instances. The program is fed with a number of desired outputs y_1, y_2, ... and the goal of the algorithm is to produce the correct output given a new input x_1, x_2, ... It does this by using a data set of desired outputs to train a predictive model. It represents the relationship between the output classes and it's features. This model is called predictive because it predicts the classes of unclassified instances.

2.3.2 Unsupervised Learning

Unsupervised learning does not posses any predictive value nor feedback from the environment to increase the quality of the model. Though, it is very useful when it comes to finding patterns in data sets and creating representations of this data for future decision making (Ghahramani, 2004). The two most common examples of unsupervised learning are **clustering** and **dimensionality reduction**. The latter of these is useful when a problem requires clustering of instances spanning values over multiple axes where the number of axes are large. The principal components analysis (PCA) does exactly this.

2.4 Handling many features

In this work, a data instance of a company has many features i.e different economic figures. These figures makes up a large number of dimensions that exceeds anything that can be treated by clustering algorithms or visualized in a plot. Therefore, clustering wont be possible and the number of dimensions has to be reduced to something that clustering algorithms can work on.

Reduction of dimensions can be achieved using the *Principal components analysis* (PCA). When applying the PCA it is important to keep the axes for which the variance (level of information expressed by the axes) is the greatest. This enables for sparser data points when the amount of dimensions are reduced and keeps the most variance expressed by all the original axes.

2.5 Determining the number of categories

k means requires an initial estimation of k; the number of clusters the algorithm is going to look for. This parameter can be guessed depending on the case but since we're researching multiple combinations of businesses and company forms there's a huge variety of different data sets with different characteristics and therefore no room for guessing. This means k has to be estimated.

There are a number of methods for doing this and the ones used in this thesis are called the **the elbow method** and the **GAP statistic**.

2.6 Evaluation of results

When comparing results of learner models there is a set of statistic measures that are used for evaluation purposes. These range from comparison of the accuracy of the full model to specific measures regarding smaller parameters or characteristics within the algorithms. The estimation of the k in k means is just for the parameter itself while as for example the confusion matrix is used for evaluating the full performance of the C4.5 decision tree algorithm.

Computer power in terms of speed come second in hand because computers now days have performance that exceeds the needs of the program. Here better results are valued more than speed.

2.6.1 Confusion matrix

A confusion matrix is an evaluation method for the performance of a classification algorithm. It shows a matrix with all predicted classes in one dimension and the actual classes in the other dimension. The purpose of the method is to see if the algorithm confuses two different classes, hence the name. The more the matrix is like a diagonal matrix the better the algorithm performs. A diagonal matrix means that predicted instances are classified as the their actual instances and then all of them are *positive*. The rest are *negative*. The following example (in Fig. 2.2) of the confusion matrix shows the typical relationship between the *true/false positives* and the *true/false negatives*.

True positive Correctly classified instance of interest

True negative Correctly classified instance of no interest

False positive Incorrectly classified instance of interest

False negative Incorrectly classified instance of no interest

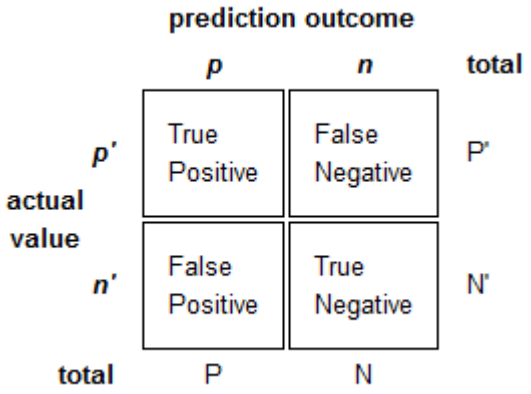

Figure 2.2: Confusion matrix

As an example, imagine 27 animals in a picture. There are 8 cats, 6 dogs and 13 rabbits. A classifier is trained to recognize the animals in the picture and the resulting confusion matrix looked like Fig. 2.3. This means that this hypothetical classifier predicted that:

5 animals were classified as cats (8 actual). The remaining actual cats were wrongfully classified as dogs.

3 animals were classified as dogs (6 actual). The remaining actual dogs were wrongfully classified as 2 cats and 1 rabbit.

11 animals were classified as rabbits (6 actual). The remaining actual rabbits were wrongfully classified as dogs.

		Predicted		
		Cat	Dog	Rabbit
Actual class	Cat	5	3	0
	Dog	2	3	1
	Rabbit	0	2	11

Figure 2.3: A hypothetical example of a confusion matrix portraying the classification of animals from a picture.

2.6.2 Precision and recall

When evaluating how well a classifier performs there are some measures of extra importance. The measures determine how good the classification algorithm could recreate the original data set and, therefore, how great it is at labeling a new instance the correct way.

Precision is the fraction of correctly classified instances out of all positive instances. In other words the fraction of positive instance relative to the amount of truly positive. This ratio is calculated as follows:

$$precision = \frac{TP}{TP + FP} \tag{2.1}$$

Recall is the fraction of positively classified instances found out of all actual positive instances. It is directly related to the percentage of positives the model will yield for any new input. Recall is defined as follows:

$$recall = \frac{TP}{TP + FN} \tag{2.2}$$

2.6.3 Evaluation of clustering

When evaluating the performance of clustering, the distribution of points within each cluster was used as a measure for useful results. Each point represent a company.

As an example: If 4 clusters are identified and one cluster contains almost all points, the algorithm suggests that there is no significant diversity between the companies. In other words, they are all one homogeneous mass. However, when the results show a clear distribution of points among the clusters, it gives a more descriptive representation of the data. These results are far more useful.

Chapter 3

Algorithms

The algorithms I evaluated in this thesis were **principal components analysis**, **the elbow method**, **the GAP statistic**, ***k* means clustering** and the **C4.5 decision tree algorithm**. The last one is well known and widely used for all kinds of classification problems. It is easy to understand and is a very powerful classification tool. The ones mentioned before are examples from the library of **unsupervised learning** algorithms and are used as the basic foundation from which the **C4.5** gets it's input from. Therefore the pre work before the clustering is evaluated with great care to produce accurate labels for a qualitative training set for the classifier.

This chapter will explain the algorithms in a depth, the implementation of them and the evaluation methods and results of each. The pipeline of the project, starting at the database, is described in Fig. 3.1 and it will be explained in this chapter.

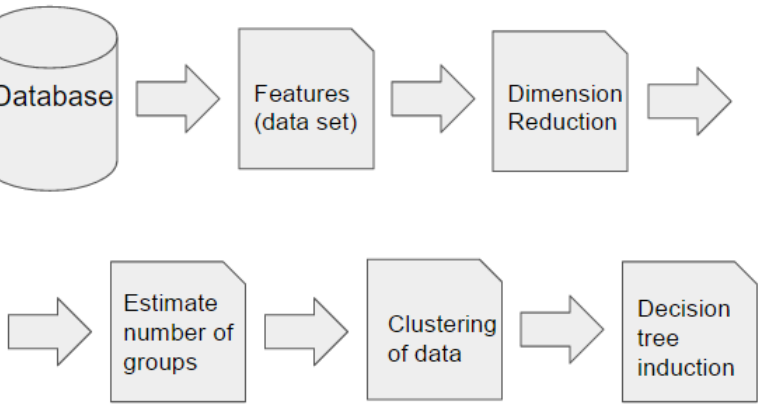

Figure 3.1: The pipeline of algorithms.

Size		
Feature	**Type**	**Description**
FinancialInfoNumberOfSubsidaries	discrete	The number of subsidiaries
FinancialInfoNumberOfEmployees	continuous	The number of employees
FinancialInfoResultBeforeTax	continuous	The net income
BasicInfoTotalTurnover	continuous	The total turnover
FinancialInfoTurnoverGrowth	continuous	The turnover growth
EmployeeGrowth	continuous	The growth of the employee count
Nbr_Offices	discrete	The total number of offices

Table 3.1: This table shows the features used when researching the size of companies.

Economy		
Feature	**Type**	**Description**
FinancialInfoQuickRatio	continuous	The quick ratio
FinancialInfoEquityRatio	continuous	The equity ratio
FinancialInfoTurnoverGrowth	continuous	The turnover growth
TurnoverPerEmployee	continuous	The turnover per employee
Profit_Margin	continuous	The profit margin
Result_Before_Tax_Growth	continuous	The growth of the net income

Table 3.2: This table shows the features used when researching the economy of companies.

3.1 Features

The features chosen were those of greatest interest to Lundalogik AB. The ones used are expressed in tables 3.1 and 3.2.

3.2 Data preparation

Before clustering any data set and using it to create a predictive model one must treat it and get rid of instances that doesn't supply a useful contribution to the representation of a generalized model. The goal is to have an as qualitative data set as possible.

3.2.1 Filtering the data

Let's start with an example; if 95 percent of the population of interest have a turnover ranging between 0 - 10 000 000 SEK and is uniformly distributed, will the last 5 percent with turnovers starting at 20 000 000 add any value to the model? No because they will be classified as "big turnover" either way. We call these instances either *outliers* or *extreme values* and these are to be removed. A data point x should reside in the intervals (3.1) or (3.2) and points outside the intervals gets deleted. This check is conducted in both axes. An example is shown in Fig. 3.2.

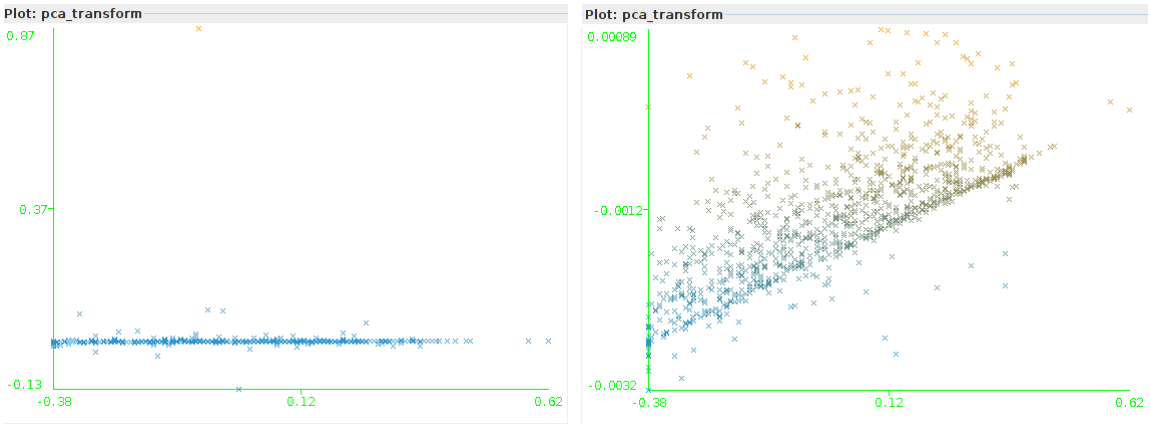

Figure 3.2: This screen shot shows how the unfiltered plot (leftmost) differs from the filtered plot (rightmost). The axes have been cut at the lower end of the x-axes and both ends of the y-axes. This way, the rightmost plot becomes a zoom of the leftmost plot.

$$Q_3 + OF * IQR < x <= Q_3 + EVF * IQR \tag{3.1}$$

$$Q_1 - EVF * IQR <= x < Q_1 - OF * IQR \tag{3.2}$$

where Q_1 is the 25% quartile, Q_3 is the 75% quartile, IQR is the Interquartile Range, difference between Q_1 and Q_3, OF is the Outlier Factor and EVF is the Extreme Value Factor.

3.3 *k*-means

This is a very popular algorithm for identifying clusters and is a member in the library of *unsupervised learning* algorithms. It is old, yet both powerful and simple. It was first thought of by Stuart Lloyd in 1957 as it is stated in Lloyd (1982). However, the name "k-means" was first uttered in MacQueen (1967).

It is used on sets of unlabeled, numeric data in search for k clusters. It is an iterative method consisting of a few steps that stops when the method converges i.e. all points belong to their closest **centroid** and the **centroids** don't move anymore. The algorithm can be described as follows:

1. **Initialization:** k-means always begins with initial guesses of where the starting centroids should be located. These guesses are initialized by a function of random described in 3.3.1.

   A centroid is the geometric mean position of all data points within a certain cluster.

2. **Assign points to clusters:** With the clusters so far, the distance (e.g. Euclidean) from each point to each centroid is calculated. Each point is assigned to the nearest centroid.

3. **Update position of centroids:** As some points might have changed to another cluster during the previous step, the old centroids are not the actual centroids anymore. Therefore, the new centroids are calculated and their positions changed. Now a choice has to be made; either repeat from step 2 or the algorithm is done and we don't need to do anything more. If the previously assigned points didn't change the position of the cluster centers all is well. If not, centers where obviously changed and some points might not be assigned to their nearest centroid anymore. Therefore, the steps are done over again starting from step 2.

3.3.1 Implementation

The input values to *k*-means in this project were vectors spanning a 2-dimensional space. It was the two components (i.e. vectors) from the *Principal Components Analysis* that expressed the highest variance ratio, giving the highest chance of more distinct resulting clusters. Each of the vectors contained normalized values ranging between 0 and 1 and every pair of coordinates represents the data of a company, transformed by the *Principal Components Analysis*. The result of the PCA transform is shown in Fig. 3.3.

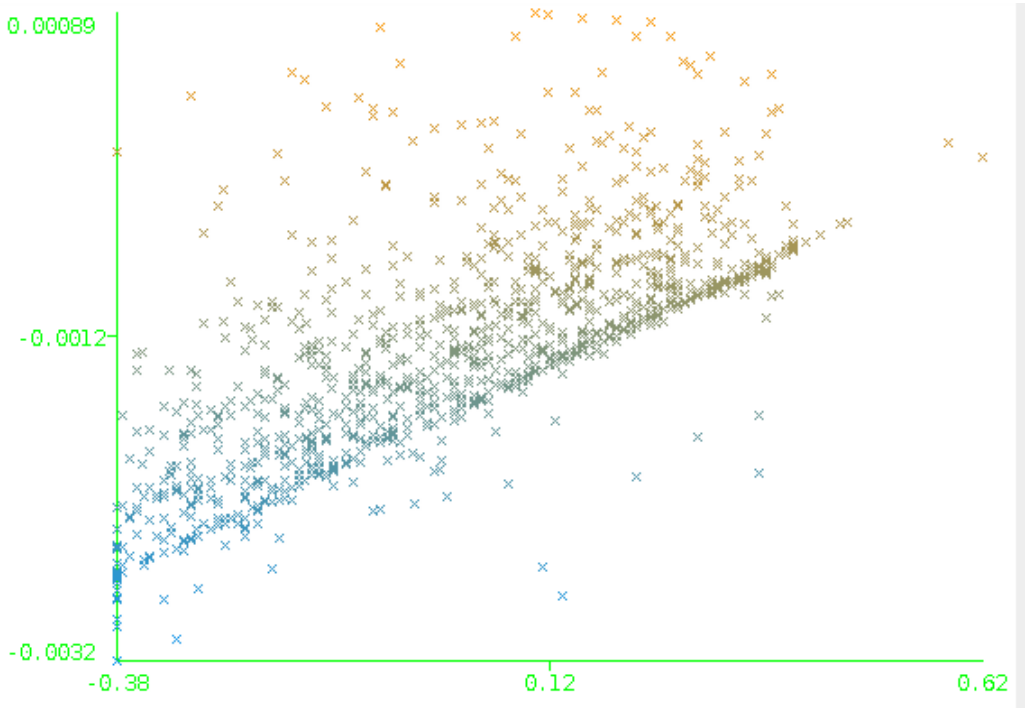

Figure 3.3: This is an screen shot the PCA transform of share companies within the IT business in Sweden. The original features of every point are all key figures regarding that particular company's economy.

There are a few initialization methods to choose from and the ones that were explored in this thesis were *random*, *MacQueen* and *Forgy* (James et al., 2013). Rather than choosing just *k* random points as initial centroids (as other methods do) *random* involves all points. It starts by assigning every point to a random label out of the *k* and computes the

initial centroid of each initial cluster. *MacQueen* initializes the algorithm by choosing k random points as initial centroids, assigning the nearest points to each centroid and then recalculates the initial centroids. *Forgy* just uses k random points as the initial centroids.

Each of these initialization methods all use the same formulas for calculating the distance between points. The distance from one point P to another Q is defined in (3.3). Here p_i and q_i are the coordinates of the points P and Q.

$$distance(P, Q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2} \tag{3.3}$$

The position of a the kth centroid that is used during the 3rd step in Sect. 3.3 is calculated as coordinates consisting of the means of each dimensional vector. In (3.4) the centroid for cluster k is calculated with $|C_K|$ as the number of points and c_{ij} as a point where i is marks the dimensional vector and j is the jth element of the vector.

$$mean(C_k) = (\frac{1}{|C_k|}\sum_{j}^{|C_k|} c_{1j}, \frac{1}{|C_k|}\sum_{j}^{|C_k|} c_{2j}) \tag{3.4}$$

There is, however, a problem with *randomization* of initial cluster centroids. Depending of which initialization method is used, they will appear on different places. So, which initialization method is the best? When talking about which clusters are the "best" there's a measurement called the *within-cluster variation*. This variation that tells how well packed a cluster really is. The smaller the within-cluster variance, the lesser the difference between the data points. The Euclidean distance is the usual approach when determining the *within-cluster variation* It is defined in (3.5) as $W(C_k)$.

$$W(C_k) = \frac{1}{|C_k|}\sum_{i,i' \in C_k}\sum_{j \in F}(f_{ij} - f_{i'j})^2 \tag{3.5}$$

The C_k denotes the kth cluster and F denotes the set of coordinates for each point. The formula calculates the mean distance out of all the pairwise squared Euclidean distances within the cluster k.

To get measure of the results of one initialization method (that can be compared to other runs when different methods were used), all variations for all k are summarized. This is viewed as the total measure $V(C)$ and is compared with different measures from different runs initialized by different initialization methods. The total variance is described in (3.6).

$$V(C) = \sum_{k=1}^{k} V(C_k) \tag{3.6}$$

C is the set of all clusters from one run: $C = \{C_1, C_2, C_3, C_4, .., C_k\}$. The initialization method that results in the lowest variance $V(C)$ is the best.

3.3.2 Principal Components Analysis

PCA is a form of **dimensionality reduction** popular when data instances are to be clustered but they posses multiple attributes larger than 3. The database containing all companies in Sweden each have multiple information attributes such as *total turnover*, *net income*, *equity ratio*, *quick ratio* and many more.

PCA is an algorithm that uses orthogonal transformation to transform a set of correlated data points to a set of uncorrelated data points called the **principal components**. The word "component" in this context is equal to "dimension" or "axes". The definition states that the first component has the largest possible variance of all other components. In other words the first component posses the most "information". A data set with higher variance is more informative compared to one with lesser variance because the sparser the points the easier to tell which points differ from other points. If a data set has lesser variance it might imply that a subset of points can be viewed as one and the same by the program and they will contribute the total information of one point to the algorithm instead of many. The succeeding components also have the highest possible variance and have to be orthogonal to the preceding components.

3.4 Estimating *k*

The estimation of k is a crucial part of the program since it is here the number of groups in the data is decided.

3.4.1 The Elbow method

The **Elbow method** plots the variance explained by the number of clusters against the number of clusters in search for "the elbow". This is the point where the marginal gain of information (percentage of variance) will drop, giving it an angle in the graph. This is "the elbow" marking the estimated k at the bottom axes. For every k the sum of intra-cluster distances between points in a given cluster C_k containing n_k points:

$$D_k = \sum_{x_i \in C_k} \sum_{x_j \in C_k} ||x_i - x_j||^2 = 2n_k \sum_{x_i \in C_k} ||x_i - \mu_k||^2 \tag{3.7}$$

Adding the normalized intra-cluster sums of squares gives a measure of the compactness of our clustering:

$$W_k = \sum_{k=1}^{K} \frac{1}{2n_k} D_k \tag{3.8}$$

W_k is the percentage of variance explained by k clusters and the result might look like the curve in Fig. 3.4.

3.4.2 The Gap statistic

The **Gap statistic** describes the differences between $\log W_k$ and an estimated $\log {W_k}^*$ obtained from B reference data sets extracted from the original data set. These reference data sets are random outtakes (with a size lesser than the original) that all together make a general representation of the original set. For k clusters the gap is defined as (Fig 3.5):

$$Gap(k) = \frac{1}{B} \sum_{b=1}^{B} \log W_{kb}^* - \log W_k \tag{3.9}$$

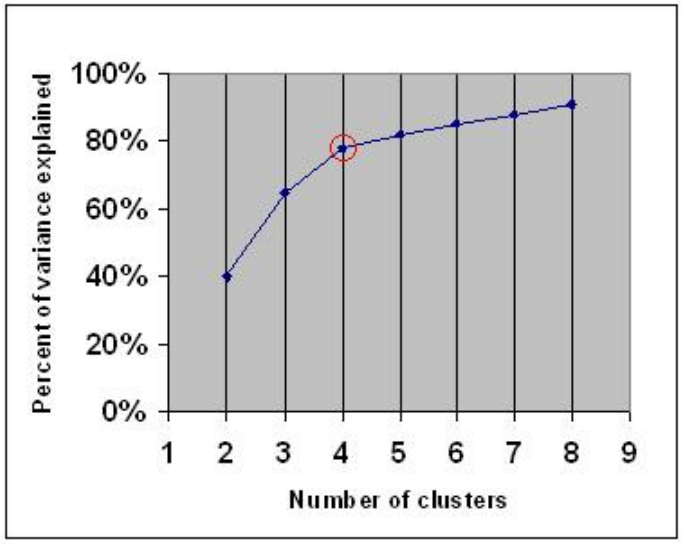

Figure 3.4: In this unrelated example, the number of clusters should be 4

Then the standard deviation $sd(k) = \sqrt{\frac{1}{B}\sum_b (\log W^*_{kb} - \bar{w})^2}$ with $\bar{w} = \frac{1}{B}\sum_b \log W^*_{kb}$ is calculated to define $s_k = \sqrt{1 + \frac{1}{B}} \cdot sd(k)$.

The number of clusters is chosen as the smallest k such that

$$Gap(k) \geq Gap(k+1) - s_{k+1} \tag{3.10}$$

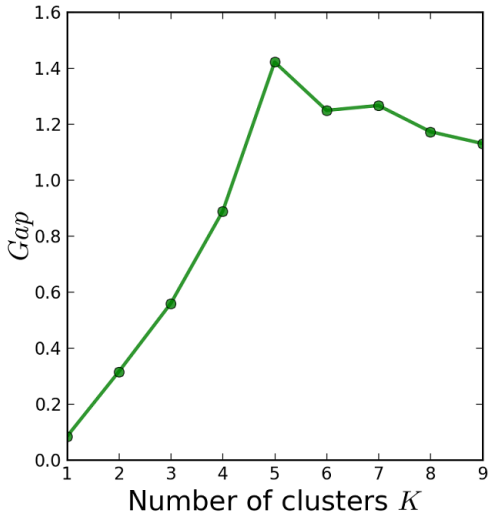

Figure 3.5: In this unrelated example the number of clusters should be 5.

Day#	Outlook	Temperature	Humidity	Wind	Play Golf
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

Table 3.3: This is an example consisting of labeled training data about weather (Quinlan, 1986).

3.5 C4.5 decision tree

The C4.5 is an extended version of the original ID3 decision tree algorithm (Quinlan, 1993) and comes from the library of *supervised learning* algorithms. It is different because (unlike the ID3) it can handle both continuous and discrete attributes, missing values, values of different costs and prune the trees. This makes it more receivable for many different kinds of attributes and is widely popular in tasks that requires machine learning.

Decision trees are predictive and descriptive models that are very useful for understanding large sets of data. However, the created model is only as good as the data allows it to be. When exposed to too much missing data or too few features it makes for a bad generalization (Geurts, 2002).

They work by first using a training set of *labeled* data, a key, to learn what mostly characterizes a *label*. These *training sets* are large amounts of instances with descriptive *features* about every instance. After training is done, the algorithms constructs a tree structure where every splitting point marks one *feature* for which the data has a number of distinct subsets that differ on a certain predicate p. The predicate can be greater/lesser than a numeric value, a nominal value (like descriptive text) or any value that applies for one branch and not the other ones. The branches that extend a splitting point are unique in the sense that p applies for all instances down one branch and $\neg p$ for the other branches.

The leafs of the tree (where the tree ends) marks one of the different *classes* or *labels* i.e. one of those used in the training set. The tree works as a test for any new data instance that follows the path of predicates and splitting points until it reaches a leaf node. When this happens a *label* for the new instance has been predicted or *classified*. Models using decision trees are called *classifiers*.

In table 3.3 there's an example of what a training set could look like. It consists of weather data collected over 14 days and the *class* of that particular day (in this case, if we're going to play golf or not). When the data is fed to the algorithm it induces a decision

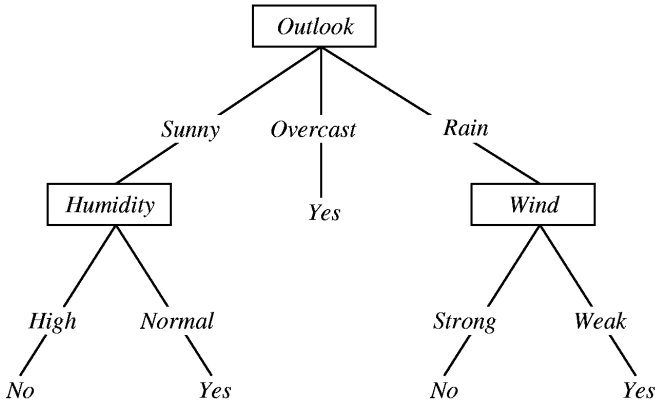

Figure 3.6: This is the resulting decision tree from the training data in table 3.3 (Quinlan, 1986).

tree as demonstrated in 3.6. This is a predictive model of whether we're going to play golf or not given what weather it is today. If we would look outside (or the weather channel) and take notes about what weather it is today, it would maybe look like this:

Outlook rain

Temperature mild

Humidity high

Wind weak

This particular day we would go and play golf since the path chosen would be the rightmost in Fig 3.6.

3.5.1 Implementation

As mentioned before, the algorithm used was the **C4.5** (the extension of **ID3**) (Quinlan, 1993). Due to some values being missing and, more or less, continuous (like turnovers) it was the optimal choice as the normal ID3 does not treat these kinds of attributes. A very small number of values are actually missing. This is because the data, which originates from the Swedish tax agency, doesn't present anything in some columns of the database. This is simply because some companies haven't declared anything there. As the C4.5 originates from ID3 and for practically being an extension they work by the same ground rules. These are explained in Algorithm 1.

The algorithm uses a divide-and-conquer approach. It recursively divides the data instances into subsets until all instances within every recursive call consists of a subset representing a *class* or if the subset is empty (Quinlan, 1993). Before each recursive call the ID3 needs to determine what *decision* the *tree* is going to present next. In other words, an attribute has to be chosen that splits the current set into subsets where the predicate p is valid for one and $\neg p$ for the others. It does this by looking at the *information gain* of each attribute in the current set. The one with the highest becomes the next splitting point. There is a method for determining the order at which the attributes of the data instances

Algorithm 1 ID3 decision tree algorithm (Quinlan, 1993)

procedure ID3(*Dataset*,*Attributes*)
  Create a node for the tree
  if all in Dataset are positive **then**
    return the single-node tree Root, with label = +
  if all in Dataset are negative **then**
    return the single-node tree Root, with label = -
  if Attributes is empty **then**
    return the single node tree Root with most frequent class of *Dataset*
  $A \leftarrow maxGain(Dataset, Attributes)$ ▷ The attribute with the largest info gain
  $\{c_i | i = 1, 2, ..., n\} \leftarrow c_i$ is every possible class of A
  $\{S_i | i = 1, 2, ..., n\} \leftarrow S_i$ only contains instances with class c_i for the attribute A
  return a node that splits on attribute A with extending branches S_i, connecting $ID3(S_1, Attributes - \{A\}), ID3(S_2, Attributes - \{A\}), .., ID3(S_n, Attributes - \{A\})$

(in each subset) are split. It is by descending order of their *information gain*. The attribute with highest gain gets split first. For every attribute A described in the set S the gain is calculated as:

$$infoGain(A) = I(T) - E(A) \tag{3.11}$$

where $I(T)$ is the **entropy** of the attributes. It is defined as

$$I(T) = -\sum_{i=1}^{n}(t_i \log_2(t_i)) \tag{3.12}$$

where T denotes the probabilities t_i from the probability distribution of all possible classes. This distribution is shown in (3.13).

$$T = \left\{ \frac{x_1}{x_1 + .. + x_n}, \frac{x_2}{x_1 + .. + x_n}, .., \frac{x_n}{x_1 + .. + x_n} \right\} \tag{3.13}$$

$I(T)$ only applies to the current set of attributes and will have to be recalculated after each branching since the distribution T will also change. This is because the number of data instances went down the other branches. It results in a lesser amount of instances down each of the other branches. T is a set of probabilities of each possible outcome of the current subset where x_i denotes the number of instances with one unique label. The variable m denotes the number of possible labels. In other words, the sum of the elements of T should be 1.

So far we have something that tells us about **entropy** of a set of instances i.e. something common among all attributes within the set. What is left is $E(A)$ that tells us something unique about an attribute A in the set S and together they will define the *information gain*.

$E(A)$ is the weighted average of the possible values v_i the attribute A can define. If we would split on the attribute A, there will be as many branches as values of A i.e. the set of possible values $\{v_1, v_2, v_3, .., v_i\}$ will split the set S into i sets $\{S_1, S_2, S_3, .., S_i\}$. This means that the information needed to determine the class of an instance in the set S_i would then be

$$T_i = I(\left\{ \frac{x_{1i}}{x_{1i} + .. + x_{ni}}, \frac{x_{2i}}{x_{1i} + .. + x_{ni}}, .., \frac{x_{ni}}{x_{1i} + .. + x_{ni}} \right\}). \tag{3.14}$$

With T_i, $E(A)$ can finally be defined as

$$E(A) = \sum_{i=1}^{v}((\frac{\sum_j^n x_{ij}}{x_1 + .. + x_n})I(T_i)) \tag{3.15}$$

where x_{ij} denotes the number of instances where the class j takes on the value v_i and n is the number of different classes. Now the *information gain* can be calculated and the attribute with the highest gain becomes the next splitting point.

Chapter 4
Results

4.1 k-means

In clustering, one can't compare results to an expected result as in decision trees. This also comes from the fact that any categorization like this hasn't been done before i.e. there is no definite already-made result to compare with. That's why something else needs to represent measurement of good clustering.

If there's a clear distribution of cluster members in the data, the result can be considered reliable. It means that it is a better generalization that is easier for the classifier to work on. If the distribution would show that one class would over-represent all others, something would not be right. That would probably mean that a huge amount of instances were classified as one big mass and a few other instances were classified as others (with the possibility of being outliers).

The distributions shown in Fig. 4.1 an Fig. 4.2 are screen shots from the project.

4.2 C4.5 decision tree induction

The resulting measures show that the classifier has a high precision and recall values i.e. the model has high predictive accuracy. The results described here are produced from running the algorithm over a training set of company data that has been filtered and labeled (the target company population are share companies in the IT business). The number of data points within each cluster make up a good distribution of classes as each class contains a significant number. This way, no class has too few instances (which otherwise would make a class insignificant). This makes the resulting classifier generalize better. The important measures to note are (as previously mentioned in Sect. 2.6.2) *precision*, *recall* and *F-measure*. The latter is a harmonic mean of the two previous and is defined as:

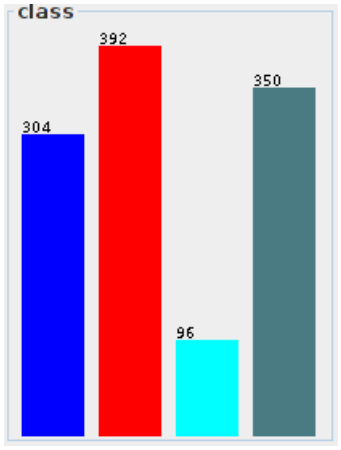

Figure 4.1: This is an unsorted resulting distribution of cluster members when researching company economy. Each color is a different class.

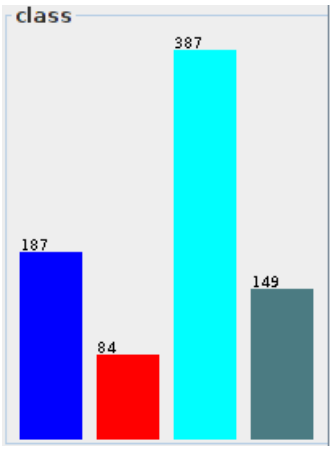

Figure 4.2: This is an unsorted resulting distribution of cluster members when researching company size. Each color is a different class.

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4.1}$$

The performance results can be viewed in table 4.1 and 4.2. These all contain high values of each measure, indicating a classifier that was trained well and performed well over the test data. This is also a measure of how good the tree can recreate the original labels of each instance from an identical, unlabeled set.

The confusion matrices $Confusion_{economy}$ and $Confusion_{size}$ show that the vast majority of instances were classified as their actual classes. The element on $(row, column) = (n, n)$ show how many instances that were correctly classified for the nth class. The more like a diagonal matrix it looks, the better the classification. It is also interesting to observe the number of *true/false positives* and *true/false negatives*. They give an observable insight of how many instances that were incorrectly classified. They are defined in Fig. 2.2. The matrices $Confusion_{size}$ and $Confusion_{economy}$ show that the vast majority of instances

Class#	Precision	Recall	F-measure
0	0.936	0.914	0.925
1	0.930	0.982	0.955
2	0.750	0.813	0.780
3	0.917	0.857	0.886

Table 4.1: Decision tree results of Fig. 4.4 with features regarding company economy.

Class#	Precision	Recall	F-measure
0	0.881	0.974	0.912
1	0.954	0.988	0.971
2	0.971	0.946	0.958
3	1.0	0.953	0.976

Table 4.2: Decision tree results of Fig. 4.3 with features regarding company size.

lie in the diagonal elements. In table. 4.3 the percentage of correctly classified instances is shown as the the sum of the diagonal elements divided by the sum of all elements.

$$Confusion_{size} = \begin{bmatrix} 170 & 0 & 15 & 2 \\ 0 & 83 & 1 & 0 \\ 12 & 4 & 371 & 0 \\ 7 & 0 & 0 & 142 \end{bmatrix}$$

$$Confusion_{economy} = \begin{bmatrix} 281 & 8 & 0 & 15 \\ 0 & 383 & 0 & 9 \\ 6 & 0 & 80 & 10 \\ 8 & 16 & 10 & 316 \end{bmatrix}$$

This means that the classifiers managed to learn how to generalize and classify in good ways. however, it is worth to mention that the total number of instances in the training set aren't the same for both classifiers (as can be viewed in table 4.3). To address this, we can begin with the fact that both populations come from the same original group of companies (share companies in the IT business). When filtering each data set, some points were labeled as outliers or extreme values in one population but not the other. Depending on values in different features, the principal components (Sect. 3.3.2) have different appearances in each population. This is why the total number of instances differ.

The resulting decision trees are shown in Fig. 4.4 and Fig. 4.3. At the bottom of every path, there's a leaf node with figures. In these, the class is firstly denoted. Then follows the number of instances that fall under that particular leaf node inside the parenthesis. If

Population	Total number of Instances	Correctly classified instances
Size	807	94.9195 %
Economy	1142	92.8196 %

Table 4.3: Percentage of correctly classified instances.

Size Class	Desc.
0	Average sized, slow growth
1	Big, very significant growth
2	Big, significant growth
3	Negative growth

Table 4.4: This shows the general description of each class in company size.

Class	Economy Desc.
0	Not so stable, slow growth
1	Very stable, rich
2	Stable, very rich
3	Stable, no significant growth

Table 4.5: This shows the general description of each class in company economy.

there were any incorrectly classified instances down a path, there's a slash sign inside the parenthesis. The number to the right of the slash sign denotes the number of incorrectly classified instances.

When describing each class, it is most relevant to look at what leaf node has the most instances. It means that the path down to that node describes the characteristics of those companies the most. Effectively, the general description of the classes can be shown in table 4.5 and 4.4.

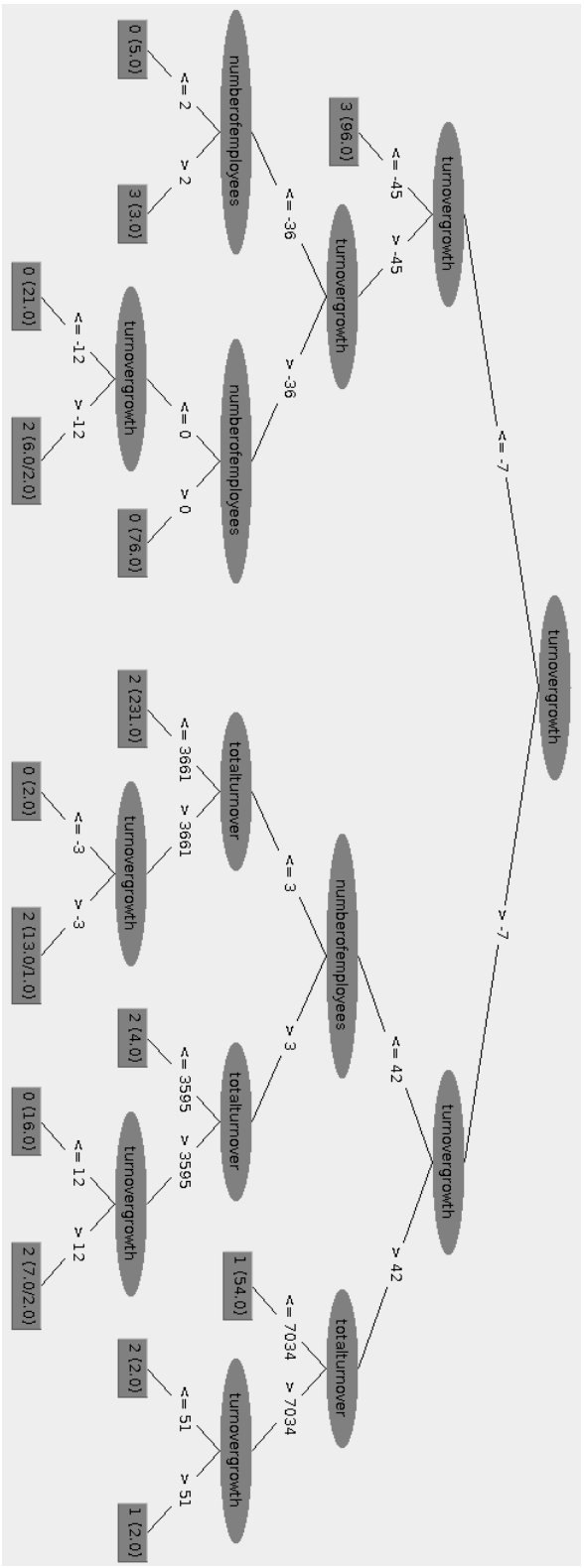

Figure 4.3: This decision tree describe the categorization of companies in regards to size.

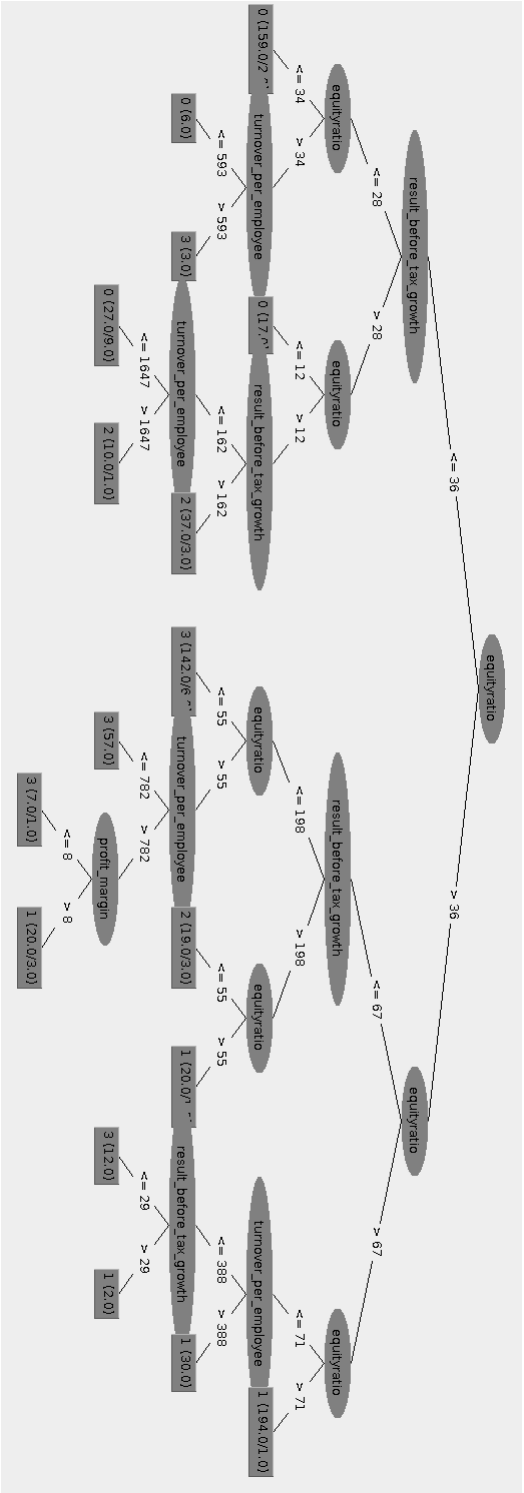

Figure 4.4: This decision tree describe the categorization of companies in regards to economy.

Chapter 5

Discussion

5.1 Extending the algorithms to other data sets

This project was developed with a data set consisting of public share companies in the IT business. This was due to the large diversity in size and economy within this group. The database from Lundalogik AB could supply data of about 1400 companies in this group. It meant that this was a fair amount and it worked well when creating a model. It also meant that the model only touched share companies in the IT business. However, if the target population would have been smaller there would be a risk of *over fitting* the model. Over fitting means that the model is so good for the targeted population, that it's practically useless for any new instance with values that would differ to much from any value in the training set. That's why there's a need for diversity in the data.

If the data set is large and diverse enough, my way of processing and using data from the database works well. For smaller groups of companies it would be harder to form valuable clusters and create predictive models. Even if the number of instances reach the hundreds we must take into account that the data must be filtered from outliers and extreme values, thus reducing the total number of instances. Using share companies in the IT business resulted in between 807-1142 instances of data from the original 1400. This is a maximum reduction of almost a third that resulted in useful data instances by the filter.

To tackle the problem of too few instances one could, if possible, extend the population with additional instances from neighboring countries like Denmark, Norway or Finland because it would be of greater use if it came from countries that resemble Sweden. For example it could mean countries that remind of Sweden culturally, structurally or socioeconomically.

5.2 Other features

The features used when researching companies aren't permanent in any way. Here, features (database columns) have been chosen so that they resemble the field of interest the most (size or economy). Features can be changed as seen fit and the database will grow over time. If they would be changed, it would change the outcome of the k-means clustering algorithm and the C4.5. As of now, the features chosen were those of greatest interest to Lundalogik AB.

5.3 Possible Improvements

5.3.1 *k*-means results

The number of clusters is greatly dependent on what the data looks like as k is estimated. I chose two different methods for estimating k. One of them is a more sophisticated method (the GAP statistic) and is known for preciser estimates. The other one, the elbow method, is more naive and will give a good estimate even if the data looks homogeneous. The problem with the GAP statistic (despite being praised as the preciser one) is that when presented with a homogeneous mass it will suggest one cluster i.e. $k = 1$. This homogeneity comes from the data being too continuous for the GAP statistic. This is where the elbow method performs better, often resulting in a $k = 4$. A k like that is far more useful than $k = 1$.

5.3.2 Decision tree results

When describing a class in the model it sometimes splits on the same attribute several times. This contradicts the usual behaviour of decision trees. However, this is the **C4.5**. It handles continuous values (which is a prerequisite for this problem since the data isn't discrete). It does this by establishing thresholds over the different attributes to produce distinct predicates for the decision tree. As mentioned in 3.5 a tree doesn't repeat past decisions. This is true as the tree might repeat an attributes but it doesn't treat any past thresholds i.e. predicates/decisions.

When interpreting the results of the tree the main point is to get a general picture of what mostly characterizes a certain group of companies. This is not necessarily done by absolute values down the tree's branches but rather described as those companies that are greater/worse in some senses. The groups are generally described as companies with larger/smaller values of certain features and together they make up a table describing 4 groups with certain important characteristics that are significant for these groups.

Chapter 6
Conclusion

The final result of the program proves to be useful and tell something about what characterizes different groups of companies in Sweden. With regards to the original scope of this master's thesis it can be said that it is actually possible to find clusters in the data and it is actually possible to create a predictive and descriptive model using decision trees.

6.1 Solution

The algorithms used for this thesis have been proven to be very effective at their respective tasks. Due to the fact that some algorithms have a lot of parameters that can be altered to affect the outcome, they were explored and tried to create the best outcome possible. Best in the context means the following:

A set of successfully created clusters with a, more or less, even distribution of cluster member points.

An acceptable number of clusters that resonates with a common and useful way of grouping things. From a good distribution of companies with a good variance, one would expect an acceptable set of predicted groups. Too many groups or too few groups don't present as much descriptive value as the natural way of dividing things into groups. A popular way of recognizing size can be: "small", "medium-small", "medium-large", "large" and as the methods predict; the number of clusters became 4 for this population. It might become something else depending on which company form and line of business the target population has. I this Master's thesis i researched upon companies in the IT business.

Descriptive results by the decision tree that makes it possible to interpret a clear view over the different company groups and what characterizes them.

To connect to the original questions at issue, the *best suited algorithms for this problem* are those that produce *satisfying results* using *reasonable estimations*. The best algorithms would be those suited for the task and that can give some sort of description of every group. Satisfying results would be those that give many interpretable insights that are general for a whole group. It would, however, be a sought after feature to view all groups in the same parameters. For example, some groups talk about growth while others do not. Reasonable estimations would be those that are clearly motivated and seem plausible. When talking about *optimal* performance, it is harder to evaluate what this means. There are more algorithms (especially in supervised learning) that have not been tried out due to the lack of time. Therefore, it is not possible at this stage to know if the solution is optimal or not. Optimal would mean that the solution give an as detailed description as possible about every company group. The number of groups would also have to be an acceptable figure.

6.2 Future work

This thesis depends largely on what different features that are used from the database. So far, features that seem correlated have been chosen in cooperation between me, my supervisors and the Lime Go team of Lundalogik AB. The features are extracted from their database and it changes over time as more columns are added. This means that, in the future, more features might be added to the model and thus the results might show something different than now.

Another thing that might be of future use is a **correlation matrix**. This is a statistical model showing the correlation between two random variables or two data sets. It seems like something that could be useful in the future when the number of features will increase and the need to see any possible correlation between features in use and features of possible future use.

When evaluating the clusters, a useful check would be the *Calinski Harabarsz Index* in combination with the distribution of cluster members. It would also be interesting to see the results of using k-means++ instead of just the k-means. Subsequently, I found that the k-means has been deemed inferior to the k-means++ by Arthur and Vassilvitskii (2007).

There's a number of different supervised learning algorithms used for classification that can serve the same purpose as the **C4.5** and would be interesting to try. Such would be *artificial neural networks* (ANN) or *support vector machines* (SVM). They are a bit more complex but have proven to be very powerful for many kinds of different classification tasks. What might not be as good for human understanding in terms of the classification is that it's harder to interpret the structure of them from the outside. Decision trees are, thanks to their structure, very easy for computers to visualize and for humans to interpret. It would also be interesting to make the data discrete and use the ID3 decision tree algorithm instead of the C4.5. This would mean that the tree wouldn't repeat attributes as splitting points and, maybe, a more detailed view would emerge.

It is in the interest of Lundalogik AB to integrate this program with their product Lime Go as decision making support for their users.

Bibliography

Arthur, D. and Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Bratel, J. (2015). The business insight index (evaluating customer insights through hybrid models). Master's thesis, Lund Institute of Technology, Sweden.

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87.

Geurts, P. (2002). *Contributions to Decision Tree Induction: Bias/Variance Tradeoff and Time Series Classification*. PhD thesis, University of Liege, Belgium.

Ghahramani, Z. (2004). Unsupervised learning.

Glas, F. (2015). Machine-learning techniques for customer recommendations: A practical study in data-driven customer prediction for customer relationship management. Master's thesis, Lund Institute of Technology.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer Science & Business Media.

Kostojohn, S., Johnson, M., , and Paulen, B. (2011). Crm fundamentals.

Lloyd, S. (1982). Least squares quantization in pcm.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations.

Marbán, O., Gonzalo, M., and Segovia, J. (2009). A data mining & knowledge discovery process model. In *Data Mining and Knowledge Discovery in Real Life Applications*. I-Tech, Vienna.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers.

Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423.

Appendices

Appendix A

A.1 List of attributes

Feature	Type	Description
FinancialInfoNumberOfSubsidaries	discrete	The number of subsidiaries
FinancialInfoNumberOfEmployees	continuous	The number of employees
FinancialInfoResultBeforeTax	continuous	The net income (kilo SEK)
BasicInfoTotalTurnover	continuous	The total turnover (kilo SEK)
FinancialInfoTurnoverGrowth	continuous	The turnover growth (%)
EmployeeGrowth	continuous	The growth of the employee count (%)
FinancialInfoQuickRatio	continuous	The quick ratio (%)
FinancialInfoEquityRatio	continuous	The equity ratio (%)
TurnoverPerEmployee	continuous	The turnover per employee (kilo SEK)
Profit_Margin	continuous	The profit margin (%)
Result_Before_Tax_Growth	continuous	The growth of the net income (%)
Nbr_Offices	discrete	The total number of offices

Table A.1: List of all original attributes.

INSTITUTIONEN FÖR DATAVETENSKAP | LUNDS TEKNISKA HÖGSKOLA | PRESENTATIONSDAG 2016-03-18

EXAMENSARBETE Machine Learning for Categorizing Companies in Sweden
STUDENT Christian Frid
HANDLEDARE Pierre Nugues (LTH), Anders Pålsson (Lundalogik AB)
EXAMINATOR Mathias Haage

Categorizing Companies for Customer Relationship Management Systems

POPULÄRVETENSKAPLIG SAMMANFATTNING **Christian Frid**

Results show that large amounts of company data can be categorized into groups with descriptive labels using k-means clustering and C4.5 decision tree induction. Clients can now search for companies more efficiently.

The company Lundalogik AB supplies a service where users can look up data about companies in Sweden and search for customers. As companies rise and fall, brands and names will change. This means that a person (or company) searching for new customers will have a constantly changing pool of new alternatives. Thankfully, there are recommendation services to assist the user when searching for customers. However, it is not necessarily easy for the user to grasp the overall picture of the companies that were recommended. Are they big? How are they doing economically? It would also be useful to have an idea of why they were recommended. It is clear that there is a need for a system that can supply a helping tool that gives fast insights about companies in these situations. A systemized division of companies into different groups with regards to interesting attributes, such as economy and size, is a sought after feature. This way, a user can pinpoint a particular target group more easily. To address this need, machine learning techniques were used and are proving to be a successful tool when searching for groups and trends in company data. Using the right key figures as attributes brings greater understanding of different groups and helps the user alleviate some of the confusion.

Customer Relationship Management
Customer relationship management (CRM) is a business technique that is about building the best relations with the customers. This is because it benefits both ends as the customer satisfaction becomes greater and the chances of them remaining customers will increase. This is done, to a great extent, by collecting data about customers and analysing it. Then, clients will have a better understanding of the customer's behaviours and needs. The communication with the customers are adapted thereafter.

Lundalogik AB
Lundalogik AB is a company that provides CRM services to their customers. With these, a user can look for new customers and manage relations with them. The way a user searches for companies is a lot like searching for music in Spotify or Soundcloud. Instead of bands or songs, the user searches for company names or anything resembling a target company. It is also possible to book meetings or look up contact information. The backbone of these services is a database consisting of names, locations, codes, economic figures etc.

Categorizing companies
Lundalogik AB suspected that within the large quantity of data, there were groups that were more alike than others. The results show that, given a large data set, groups can be created and labeled with k-means clustering. Then, predictive models are inducted (using the C4.5 decision trees) that describe each company label. This yields a set of company groups with common intra-group characteristics. Instead of looking at every company's data figures, a user can now save time by looking at what group (or tag) a company falls under and decide whether to contact them or not.