Thesis no:
URI: urn:nbn:se:bth-16300

Unsupervised Machine Learning: An Investigation of Clustering Algorithms on a Small Dataset

Alvarez Gonzalez, Pierre
alvarez9549@gmail.com

Forsberg, Fredrik
forsberg.fredrik@hotmail.com

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the bachelor degree in Software Engineering. The thesis is equivalent to 10 weeks of full time studies.

Contact Information:
Alvarez Gonzalez, Pierre
alvarez9549@gmail.com
Forsberg, Fredrik
forsberg.fredrik@hotmail.com

University advisor:
Dr. Huseyin Kusetogullari
Department of Computer Sci. and Eng.

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

Context: With the rising popularity of machine learning, looking at its shortcomings is valuable in seeing how well machine learning is applicable. Is it possible to apply the clustering with a small dataset?
Objectives: This thesis consists of a literature study, a survey and an experiment. It investigates how two different unsupervised machine learning algorithms DBSCAN(Density-Based Spatial Clustering of Applications with Noise) and K-means run on a dataset gathered from a survey.
Methods: Making a survey where we can see statistically what most people chose and apply clustering with the data from the survey to confirm if the clustering has the same patterns as what people have picked statistically.
Results: It was possible to identify patterns with clustering algorithms using a small dataset. The literature studies show examples that both algorithms have been used successfully.
Conclusions: It's possible to see patterns using DBSCAN and K-means on a small dataset. The size of the dataset is not necessarily the only aspect to take into consideration, feature and parameter selection are both important as well since the algorithms need to be tuned and customized to the data.

Contents

1 Introduction

Large amounts of data and information are being collected from different devices such as phones, computers, cars, GPS and all sorts of connected devices. In todays era with large amounts of data, the time taken to compute is increased, and this is where machine learning comes into action, to help process large data in a reasonable amount of time. Machine learning is a subfield of artificial intelligence, which is an area of computer science that emphasizes the creation of intelligent machines or programs that work and react like humans. Goals of machine learning can be to program computers to use example data or past experiences to solve a given problem. Machine learning can help us to understand the structure of data and fit that data into models that can be understood and utilized by humans.

The concept of Machine learning was first defined in 1959 by Arthur Samuel as the field of study that gives computers the ability to learn without being explicitly programmed [11]. With the current processing power and advancements in the field it makes machine learning more realizable and applicable in modern days [1]. With possibilities to spin up dedicated servers from companies such as Amazon and rent high-performance hardware makes it easy for even individuals to work with advanced and computational heavy programs and large datasets. Machine learning can be categorized into three different branches which are supervised learning, unsupervised learning and reinforcement learning.

Supervised machine learning is when the component is observing from input and output data. The input data needs to include labels (defined as correct data) [10]. The goal is to understand the mapping of how they relate to one another. This includes topics such as regression, prediction, and classification.

Unsupervised machine learning has much in common with exploratory data analysis and data mining, it's only an observation of the data. There are only input data and not available output observation. The data is unlabeled so there is no right or wrong in the data. It's restricted to understanding what can be learned from the data.

Reinforcement machine learning includes an agent who takes action depending on the situation and gets rewarded for doing its actions. This learning method doesn't need to specify how the action should be handled, the agent only gets rewarded by performing the correct actions. The goal is to have an agent who does actions correctly from doing trial and error learning in a dynamic environment [8].

This thesis will investigate and be restricted to the study of the unsupervised learning branch. It will investigate how well K-Means and DBSCAN work on small datasets. These are clustering algorithms that use different approaches. DBSCAN is density based and K-means is a centroid based algorithm, this will make it more interesting to compare them with each other. We have defined a small dataset as a dataset with less than 500 observations. The dataset used is collected through a survey and focusing on peoples training habits. It will investigate if it is possible with a dataset with few samples to get interesting results, or if we need more samples of data to create any valuable clusters. To see if it is possible to create any clusters containing clear groupings of peoples, e.g. groups containing people with the same age, gender etc.

Using machine learning for the development of programs make them more reliable and it goes faster then if a human were to develop the program from scratch [1]. The negative aspect of using machine learning is the need for data to train the program, it's also more computer heavy than a regular program. Machine learning can do a lot to improve our world, it can improve various different fields outside of software development as well. An example being *economics*, like [1] mentions. Substitution, price elasticity, income elasticity and more could be improved by using machine learning. Most arguably however the field that machine learning can change the most is automation. Having the component do the work is cheaper and it doesn't need breaks. As [22] mentions the machine learning algorithms understands the concept and use the appropriate manner to any given area. This, in turn, could affect some work areas in the way that the work duties can be more automated or even completely replaced by self-learning components [2].

2 Research Questions

In this chapter, we present the research questions, and motivate why we choose them. We also describe our goals and objectives with our thesis, and then what we thought about the expected outcome.

2.1 Research Questions

RQ1: In which fields can DBSCAN and K-means be used in?

RQ2: What observations can be made by looking at the patterns from the unsupervised clustering algorithm DBSCAN on people's training habits with a small dataset?" ?(max 500 samples)

RQ3: What observations can be made by looking at the patterns from the unsupervised clustering algorithm K-means on people's training habits with a small dataset?" ?(max 500 samples)

2.2 Background

This thesis will investigate where unsupervised clustering algorithms can be applied through a literature study, this will gain the knowledge of where such algorithms can be applied. The investigation is restricted to focus on two clustering algorithms. These algorithms use different clustering approaches, DBSCAN is density based and K-means is a centroid based algorithm, which makes the investigation and the clustering more interesting since it will be possible to get different clusters. The motivation for picking DBSCAN is that it can automatically determine the number of clusters, it can handle data with noise/outliers and it can also detect outliers while identifying clusters [12]. K-means was chosen because of its wide popularity and simplicity. These algorithms will cluster the dataset gathered from the survey. The limit of the dataset is set to 500 samples from our survey, as our definition of a small dataset is under 500. This is done because we want to focus on a dataset with fewer samples. Research question 2 and 3 will provide and present an investigation through an experiment of these clustering algorithms on the dataset. The value of this is motivated by the size of the dataset. The goals with research question 2 and 3 are to see if it is possible to generalize the clusters based on a small dataset. The main objective is to investigate if we can identify patterns in the clusters. Patterns in the clusters can, for example, be if it is possible to identify what types of groups exist or to identify unknown groups with similar habits. When comparing DBSCAN with K-means we want to compare the clusters and analyze if they produce similar results.

2.3 Expectations

From the survey we expected a minimum of 100 answers. We assumed it would take a maximum of 2 weeks to receive that number of answers.

For the literature question (RQ1) we read papers/articles about what fields K-means and DBSCAN can be used in. We expect to find a lot of different fields for both algorithms since both are popular. We also expect to get drastically different results depending on what fields the algorithms are used in because of their difference in finding clusters. The reason being that K-means is centroid based while DBSCAN is density based.

For research question (RQ2) we expected to observe clusters of arbitrary shapes and therefore be able to identify what groups of people have the most similar habits. We also expected to identify outliers in the dataset, i.e. samples of data that don't belong to any cluster. This is interesting because we will then be able to identify patterns and see what makes them differ from the rest.

For the research question 3 (RQ3) we cluster the data with the algorithm K-Means. We expect depending on the predefined number of clusters to get very different results. In this question we expect to observe the way K-means works on a dataset which will put people with similar features in the same clusters but depending of the number of clusters this can be divided up to more filtered clusters.

The expectation from using both algorithms is to find different generalizations. We believe that both algorithms will have no problem to cluster the given input data, what we are unsure of is if we can get anything from looking at the clusters. The expectations we have is that K-means will make very general generalizations, this in turn could make the generalization too unspecific making the result uninteresting. To get around the result being to unspecific we will define a higher number of clusters to get a result that we can learn something from. DBSCAN will be less general and give us anomalies, we expect results with DBSCAN to be more specific but we are unsure if the data is too small to get specific. This could result in DBSCAN clustering almost everything or almost nothing. To get around this we will change *minpoints* and *epsilon* to get clusters that we can see generalizations on.

3 Clustering algorithms

In this chapter, we introduce the two clustering algorithms that were used in the thesis. We give a high level presentation of them and provide four visualizations for each algorithm. The goal here is to give you a quick, high level presentations of the algorithms to more easily follow along.

3.1 DBSCAN

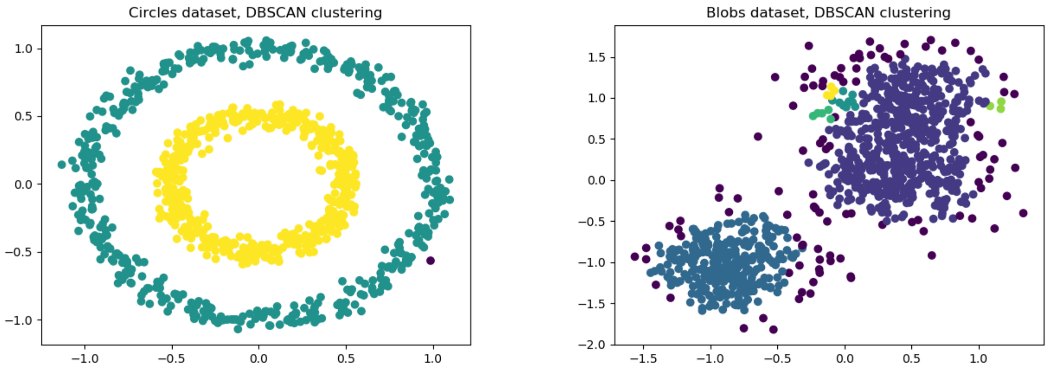
DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996 [6]. The algorithm captures the insight that if a particular point belongs to a cluster, it should also be close to other points in that cluster. One of the advantages of using DBSCAN is that it can find arbitrary shapes of clusters. DBSCAN has been applied in a variety of fields such as network traffic classification[5], computer security, such as malware classification[9], climate studies[5] and anomaly detection[12]. DBSCAN depends on two parameters, a positive number *epsilon* and the minimum number of points called *minPoints*. Initially, all data points in the dataset are unassigned. DBSCAN begins by picking an arbitrary data point from the dataset that has not been visited. If there are more than *minPoints* points, including itself, within the distance of epsilon from point p, then those points form a cluster. *Point p* is said to be a core point and the neighbor points within the distance of epsilon are said to be directly reachable from point p. DBSCAN checks all of the new points in the cluster to see if they too have more than *minPoints* points within a distance of epsilon, if that is the case, DBSCAN expands the cluster by adding them to the cluster, it then repeats this step for the newly added points. When there are no more points to add to the cluster, it picks a new arbitrary, unvisited point from the dataset and repeats the process. If the point has less than *minPoints* points within the distance of epsilon and does not belong to any other cluster, then it's considered a noise point.

3.2 Visualizations - DBSCAN

These visualizations show how DBSCAN clusters on four different datasets, generated by sklearn. The dataset contains 1000 data points and 2 features. *Epsilon=0.5 and minPoints=5*

In figure 1.a, DBSCAN is applied on the circle dataset. This visualization demonstrates the ability of DBSCAN to create arbitrarily shaped clusters. You can also see if you look closely to the bottom right an anomaly outside of the green ring.

In figure 1.b, DBSCAN is applied on the blobs dataset. This visualization demonstrates the ability of DBSACN to identify outliers in a randomly distributed dataset and create clusters where data points tends to be close to each other.



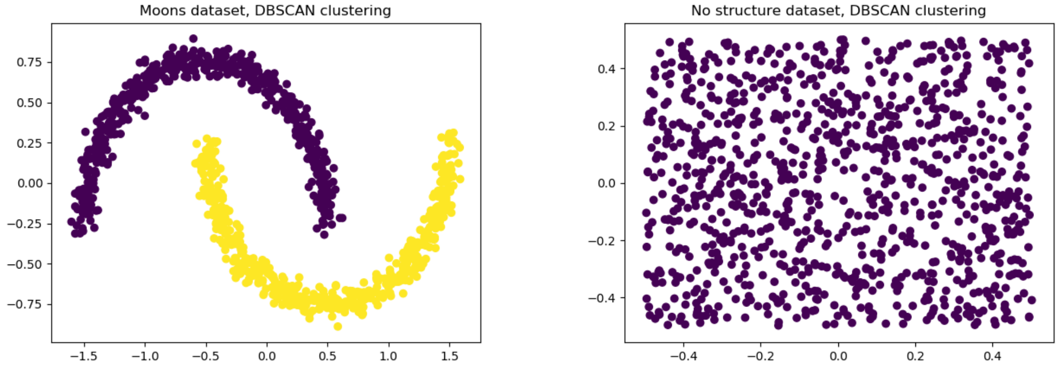
(a) Circles dataset

(b) Blobs dataset

Figure 1: DBSCAN visualization

In figure 2.a, DBSCAN is applied on the moons dataset. This visualization demonstrates the ability of DBSCAN to create arbitrary shaped clusters.

In figure 2.b, DBSCAN is applied on a no structure dataset. This visualization demonstrates the ability of DBSCAN to identify all data points that belongs to a dense group of data points.



(a) Moons dataset

(b) No structure dataset

Figure 2: DBSCAN visualization

3.3 K-means

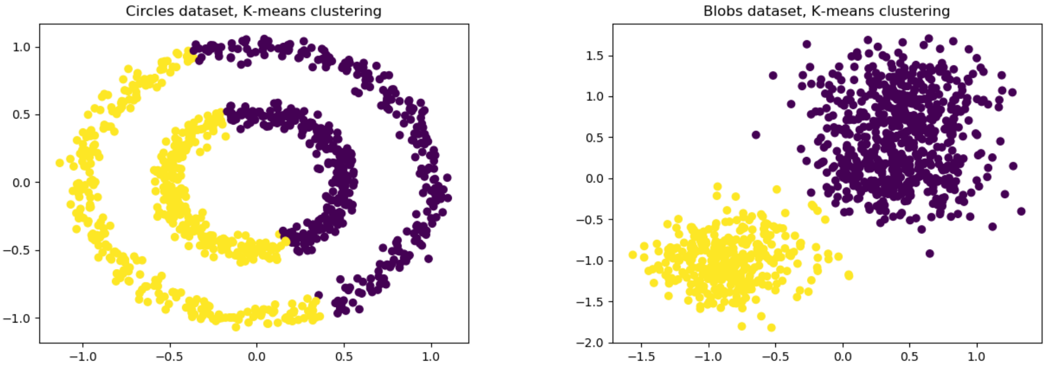
The K-means algorithm was first mentioned by James MacQueen in 1967[14], however the idea started in 1957 by Hugo Steinhaus [23]. The K-means is a centroid-based clustering algorithm. K-means algorithm has been applied in a variety of fields such as image segmentation[3], disease prediction[15], network traffic classification[5]. The algorithm partitions n samples of a dataset into a fixed number of k disjoint subsets/clusters were each sample belongs to one of the k clusters. The value of k must be predefined. The centers of the clusters are called centroids and are initially chosen randomly from within the subspace. K-means algorithm works in 2 steps, in the first step all data points are assigned to the cluster with the nearest centroid. In the second step, all clusters recalculate and updates the centroids location based on the mean of all data points assigned to their clusters. These 2 alternating steps continue until the centroids stop moving.

3.4 Visualizations - K-means

This visualizations show how K-means clusters on 4 different datasets, generated by sklearn. The dataset contains 1000 data points and 2 features each. There are 2 clusters predefined.

In Figure 3.a, K-means algorithm is applied on the circle dataset. This visualization shows how K-means doesn't find the moon shapes and only places the centroid points where most nodes are.

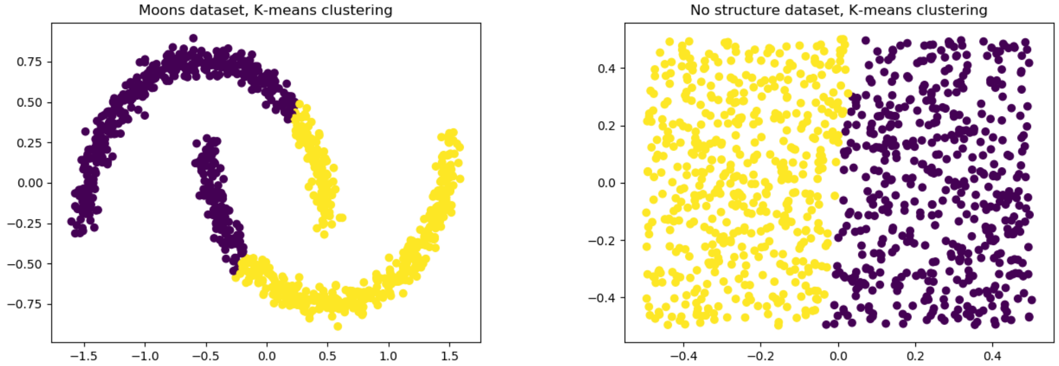
In Figure 3.b, K-means algorithm is applied on the Blobs dataset. This visualization shows how K-means split the data points between the clusters.



(a) Circles dataset

(b) Blobs dataset

Figure 3: K-means visualization



(a) Moons dataset

(b) No structure dataset

Figure 4: K-means visualization

In Figure 4.a, K-means algorithm is on the moons dataset. This visualization shows how K-means doesn't find arbitrary shaped clusters.

In Figure 4.b, K-means algorithm is applied on a no structure dataset. This visualization shows how K-means detects two clusters with the centroid points.

4 Research method

This thesis is carried out in two steps, one empirical part and on literature study. The first part of our thesis is the literature study, this will be done by answering research question number 1 by reading research papers where DBSCAN and K-means have been applied. The empirical part consists of an experiment and a survey. The purpose of the survey is to create a dataset, that can be used during the clustering. The dataset from the survey has been used during the clustering with both K-means and DBSCAN algorithms to cluster the data.

4.1 Literature study

This part describes how the literature study was conducted and its main objectives. Our goal with the literature study is to investigate in what fields and applications DBSCAN and K-means can be applied. It should be noted that this will not cover all fields they can be applied to. The purpose is instead to give some ideas on where it can be used and present some of their applications.

4.2 Search Engines

- Google Scholar
- Microsoft Academics

4.3 Keywords

The keywords used while searching for papers/articles by using the mentioned search engines. The papers/articles were used to back-up our statements with other people's statements. The method used to find papers/articles was too type specific keywords while not making the keywords to long as the wanted outcome is to find many different fields for both algorithms.

- machine learning
- K-means economy
- K-means image segmentation
- K-means usage
- K-means outlier detection
- K-means and DBSCAN
- DBSCAN outlier detection
- DBSCAN fields
- DBSCAN

4.4 Limitations and validity threats

When determining if a source was usable it was drawn by looking through the abstract, introduction and conclusion. Then skimmed through the source to check if it was relevant enough to make a decision on including it in the thesis. The criteria for our sources is that it includes nuanced information as to what fields DBSCAN and K-means can be used on. The literature should say why the field can benefit from using DBSCAN and K-means.

4.5 Empirical study

4.6 Survey

The survey is the foundation for our thesis. It consists of 8 questions about people's training habits. The expectation is between 100 - 500 answers. The survey was sent out by email to all students at BTH and also shared on social media networks such as Facebook and Slack. The survey was created with a Google Form which allowed us to obtain the results/answers as a CSV file. The survey will be up until gaining at least 100 answers.

4.7 Design of survey

The most important part when designing the survey was that it would be possible to use the data in our experiment, i.e. the clustering of the data. The reason for choosing training habits as our survey topic was because it would increase the rate of answers, since fitness is something many people can relate to. Each question represents a feature and decided that 8 would be enough. Both in respect to what could expect people to answer and to get enough number of features to get some values from the clustering. The focus of feature selection for unsupervised machine learning is to find the features that best uncovers clustering [4]. By only having relevant features to cluster upon this can be achieved.

It was limited to chose one option per question because it would be easier to pre-process the survey data. There is always going to be some errors in making the survey[13], this can also be reduced by having a simple question. This simple approach comes with the downside that people can't pick multiple options if they wanted to. To avoid where people would like to have more than one choice there are options such as question 4 "Mix of exercises" instead. Also having many answers for each question leads too more spread in the data, this was avoided by picking questions 1,3 and 4.

The first part of the survey was aimed to focus on personal information, questions 1-3. Question 1 is there to make a big separation by having few answers. As gender is a big definer, it was good to follow up by segmenting the data in smaller chunks by asking for age and employment.

The second part is more aimed at exercising habits. Questions 4-6 and 8 the focus was on giving a stronger identity by further segmenting the data in chunks. Note in question 6 it's 0-1 hours instead of 0 hours to prevent further spread because of our assumption of having many answers with no training habit. Question 7 was made to give each data a greater boost of identity with the training habit by only having two answers. All the questions was mandatory so that the clustering would be consistent. If you would be to cluster on a row and one column doesn't relate to anything it would cluster with nothing, making it harder to see patterns inside the clusters.

4.8 Experiment

This part will describe the clustering performed during our experiment and the libraries, hardware and approach. The experiment was implemented in Python3, with the machine learning library scikit_learn 0.19.1. This library was used because it is one of the most popular machine learning libraries and is it very well documented. The Python data analysis library Pandas was used for manipulating the dataset. The data used for the experiment is the dataset gathered from the survey. The dataset contained 393 samples and 8 features, where 5 of the features are categorical and 3 are continuous/ values. The Euclidean distance metrics was used during the clustering. With Euclidean distance, a small distance between two objects implies a strong similarity whereas a large distance implies a low similarity. In an n-dimensional space of features the distance between two samples p and q can be calculated with:

$$dist(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2 + .. + (p_n - q_n)^2}$$

where n is the number of features.

4.8.1 Testing environment

The experiment was conducted on a computer with an Intel i7-7500U @ CPU 2.70GHz processor with 2 cores (4 threads) and 16GB RAM, running Ubuntu 17.10 distribution. Our metrics are based on running scikit_learn 0.19.1.

4.8.2 Preprocessing

This subsection describes what steps have been conducted to prepare the data for clustering. The dataset was loaded from a CSV file into a Pandas DataFrame [19], which is a two-dimensional size-mutable data structure. All data in the dataset consisted of non-numerical data types. Categorical and non-numeric data is a problem for both K-means and DBSCAN.[20], since they work with numerical values. The first part of the preprocessing mapped non-numerical values to numerical values, e.g feature Gender with the possible answers/categories, Male = 0, Female = 1 and Other = 2. Since the algorithm will interpret Male (0) to be closer to Female (1) than it is to Other (2), the representation needs to let the computer to understand that these things are all actually equally different. The second step of the preprocessing solves this by separating the variable Gender into 3 separate variables[18] "Male", "Female" and "Other", which all can only take a binary value 0 or 1. This though increases the amount of dimension. Encoding categorical integer features using a one-hot (one-of-K) scheme was performed with the scikit_learn "sklearn.preprocessing.OneHotEncoder" [21] module. The dataset contained 3 continuous features, *Age*, *How often do you workout? (days a week)*, *How much time do you spend on each workout?*. These features were mapped to a numerical representation where the lowest numerical value represents the first alternative of the respective feature and the highest numerical value was mapped to the last alternative. The data was then normalized by each feature into values between 0 and 1.

4.8.3 Approach

This section will describe the approaches taken on each algorithm during the experiment. The experiment has been carried out based on different combinations of features. The list below shows the different features that have been used. The parameters were defined by tuning them by hand until it reached desired results.

DBSCAN

1. All 8 features

   Parameters:Epsilon=1.5 and min_samples=5

2. Consider only numerical features (3 features), i.e. *Age, How often do you workout? (days a week)* and *How much time do you spend on each workout?*

   Parameters: Epsilon=0.3 and min_samples=5

3. Consider only categorical features(5 features), i.e. *Gender, Employment, What do you exercise?, Do you workout with a partner?* and *Why do you workout?*

   Parameters: Epsilon=1.2 and min_samples=5

4. The features *Gender, Age* and *Employment*

   Parameters: Epsilon=0.3 and min_samples=5

5. The features *Gender, How often do you workout?* and *How much time do you spend on each workout?*

   Parameters: Epsilon=0.3 and min_samples=5

6. The features *What do you exercise?, How often do you workout? (days a week), How much time do you spend on each workout?,Do you workout with a partner?,Why do you workout?*

   Parameters: Epsilon=1.4 and min_samples=3

7. The features *What do you exercise?, How often do you workout? (days a week), How much time do you spend on each workout?, Why do you workout?*

   Parameters: Epsilon=1.4 and min_samples=5

8. The features *Employment, What do you exercise?, How often do you workout? (days a week), How much time do you spend on each workout?*

   Parameters: Epsilon=1.4 and min_samples=4

K-Means

Every test had the same two parameters

1. Parameter: *k=5*
2. Parameter: *k=15*

5 Literature Review

This is a summary of the research papers that were found during the literature study for research question 1. After the presentation of each relevant reference/paper, a comparison and review is conducted.

Anomaly Detection in Temperature Data Using DBSCAN Algorithm[16]

The paper revolves around finding anomalies in monthly temperature data. The method that was used previously in finding anomalies was a statistical method. When comparing DBSCAN to the statistical method, DBSCAN was a lot more accurate on finding correct anomalies. The only downside is that the data needs preprocessing to get a good anomaly detection. The problem with the data is that the temperature that it considers an anomaly varies from what time of the year it is. To remove the seasonal factor, z-score is used to generalize the data by what time of the year it's. Later the data gets normalized by mean and standard anomaly of the month. The temperature data anomaly detection analysis is used in various different fields such as services, public health and climate studies.

Traffic classification using clustering algorithms[5]

This paper talks about how to accurately identify and categorize network traffic according to application type. The authors describe that it is an important element of many network management tasks such as flow priority, traffic shaping/policing, and diagnostic monitoring. The authors describe 3 different approaches of how to identify and categorize network traffic and talks about the advantages and disadvantages of respective approach. In the first approach, the classification of network traffic is based on the mappings of known ports to applications. In the second approach, packet payload is analyzed to determine whether they contain characteristics of known applications. The paper focusing on a third approach, where they explore how to categorize data based on only transport layer statistics with clustering. They show that cluster analysis has the ability to group and categorize network traffic using only transport layer traffic with DBSCAN and K-Means, using empirical Internet traces. The experimental results show that both K-Means and DBSCAN work very well and much more quickly then AutoClass. The results indicate that although DBSCAN has lower accuracy compared to K-Means and AutoClass, DBSCAN produces better clusters.

A New Approach of Image Segmentation Method Using K-Means and Kernel Based Subtractive Clustering Methods[3]

The paper talks about how image segmentation is the first step in image processing and a proposed machine learning algorithm to make an accurate segmentation. The researchers used clustering because of the high usage in the area, it's simplicity and its efficiency. The proposed algorithm is a combination of K-means and kernel based subtractive methods. The kernel function is there to increase the efficiency by transforming the pixels from the image into other dimensions to more easily separate them. K-means is later used to identify the different types of segments in the image.

Brain Tumor Segmentation Using Fuzzy C-Means and K-Means Clustering and Its Area Calculation and Disease Prediction Using Naive-Bayes Algorithm[15]

The paper talks about how with the help of machine learning it's possible to find brain tumors and risk of disease. Today the findings are done using magnetic or radiation types of scans which take time and are expensive. If image segmentation is done by using K-means and Fuzzy C-means it would both save time and make the process cheaper. This is done by crafting Brain MRI images, preprocessing the image, using both algorithms and lastly predicting the disease by observing the segmentation result. For finding mass tumor using K-means is enough, if however there is noise in the MR image, K-means needs preprocessing by filtering the noise. K-means is not good enough on it's own as it doesn't detect in detail and this is why Fuzzy C-means is later used after K-means to get more accurate tumor shape extraction.

Malware classification based on call graph clustering[9]

In this paper, the researchers explore the potentials of call graph based malware identification and classification. The clustering algorithms used in the experiments include various versions of the k-medoids clustering algorithm, as well as the DBSCAN algorithm. The authors describe that it is desirable to identify groups of malware with strong structural similarities and describes that anti-virus engines can employ generic signatures, targeting the mutual similarities among samples in a malware family. The identification of groups of malware with strong structural similarities is done via clustering algorithms. The conclusion states that k-means clustering is not effective to discover malware families, mainly because it was not possible to determine the optimal number of clusters. The results performed with DBSCAN were successful since it was possible to identify malware families.

Traffic Anomaly Detection Using K-Means Clustering[17]

This paper focuses on network data mining and flow-based anomaly detection by using K-means clustering algorithm. K-means is used to separate time intervals by comparing normal and anomalous clusters. The clustering is done with 3 features, number of packets and number of bytes allows anomaly detection. The third feature helps security by checking detecting network port scans and distributed attacks. This in turn increases the number of source-destination parts. Still just using k-means on these features for the dataset is not enough to find all anomalies. By using two preprocessing tools it's possible to find all outliers as long as the number of outliers is small. The first tool used is doing a distance calculation on nodes that aren't part of a centroid cluster and placing them in the cluster they belong to. The calculation is done by measuring what centroid is closest to the node. The other method is outliers detection, which does a calculation to see if a node is too far from a normal cluster by a threshold it's defined as an anomaly. Both of these methods are combined to make the classification on anomalous and normal behavior.

Towards a Hybrid Approach of K-Means and Density-Based Spatial Clustering of Applications with Noise for Image Segmentation[7]

This paper proposes a hybrid approach to image segmentation using what the authors of the paper call Kmeans-DBSCAN. Because of the high computational complexity of DBSCAN and the large size of image datasets, K-means is applied to reduce the size of image datasets in the proposed approach. The clusters centroids produced by K-means are further clustered by DBSCAN. The image segmentation results are finally provided by merging the results of K-means and DBSCAN. The authors of the paper concludes that the results of the proposed method are more reasonable than either DBSCAN or K-means segmentation results.

Anomaly detection in onboard-recorded flight data using cluster analysis[12]

This paper presents a method that evaluates flight data and identifies anomalous flight without specifying the criteria which defines an anomaly. This is achieved with cluster analysis techniques. DBSCAN is used as the algorithm to cluster the dataset. The dataset used in the experiment consists of a Digital Flight data Recorder (DFDR) dataset from an international airline. The authors of the paper concludes that from the initial evaluation, it indicates that cluster analysis is a promising approach for the identification of anomalous flights from onboard-recorded flight data.

5.1 Literature comparison

In this section we compare the different literature with each other and describe when references are supporting each other, and when they are in conflict with each other.

The researchers in the papers that we have studied have all achieved successful results from the DBSCAN algorithm. In the paper[7], a hybrid version of K-means and DBSCAN is used with the motivation of the high computational complexity of DBSCAN and the large size of image datasets. K-means is applied to reduce the size of image datasets. The other papers that considers DBSCAN, all have small enough dataset to avoid the negative effects from the high computational complexity. The motivation for using DBSCAN in the different papers are mainly because of its ability to find arbitrarily shaped clusters and or/in combinations with its ability to identify outliers. And also the fact that there is no need to specify the number of clusters in advance.

In text [3], [7] and [15] they use k-means for image segmentation. [3] uses kernel function to increase the efficiency, [15] uses fuzzy c after k-means to get a more accurate result as they stated that k-means was not accurate enough on its own because of noise. Another example where K-means is used before of another algorithm is in [7] where k-means is used to segment on the images and then use DBSCAN.

As k-means can't identify anomalies the text [15] has issues with it as it disrupt the quality of the clusters. The solution [15] uses is deleting the noise so k-means doesn't interpret the data wrong. However [17] tries to find noise and anomalies which is something k-means isn't known for. It really depends on the data as [17] anomalies are all very close to each other. Therefore [17] defines normal and anomalous clusters with some help.

Finding and understanding anomalies could be crucial in finding security holes both [5] and [17] try to further secure the security by finding anomalies from network data. Both uses k-means but have different assistant tools, [5] has DBSCAN for comparison [17] uses two preprocessing methods *distance calculation and outliers detection*

DBSCAN is commonly used as a anomaly detection algorithm as both [16] and [12] uses it. Instead of using a anomaly detection algorithm both [16] and [12] uses DBSCAN a clustering algorithm to find anomalies.

Using two algorithms is commonly done to get a more accurate or better understanding from the data. [7],[5] and [9] uses a centroid based algorithm and DBSCAN. [9] and [5] both used K-means separate with DBSCAN for comparisons while [7] used it for segmentation on the data and later run DBSCAN on each segment. [9] and [5] got a better result from DBSCAN than k-means.

6 Result

In this chapter the results from the survey, literature study and the experiment are presented. Each research question is explicitly answered. We first begin by presenting the results from the survey and present the results from each question. Next the result from the literature survey and research question 1 are considered and answered. Finally we present the result from the experiment and show the results of research question 2 and 3.

6.1 Survey

The survey was up between 03/07 - 03/15(9 days) 2018, we decided to close the survey from the decline of answers as seen in Figure 5. We received 393 answers. In the first day we had the survey shared on social media with a response of 127 answers. On the second day the mail was sent to all BTH students with 181 answers. The rest of the days are aftermath answers from the social media and mail therefore the number of survey answers decreased drastically after the first two days.

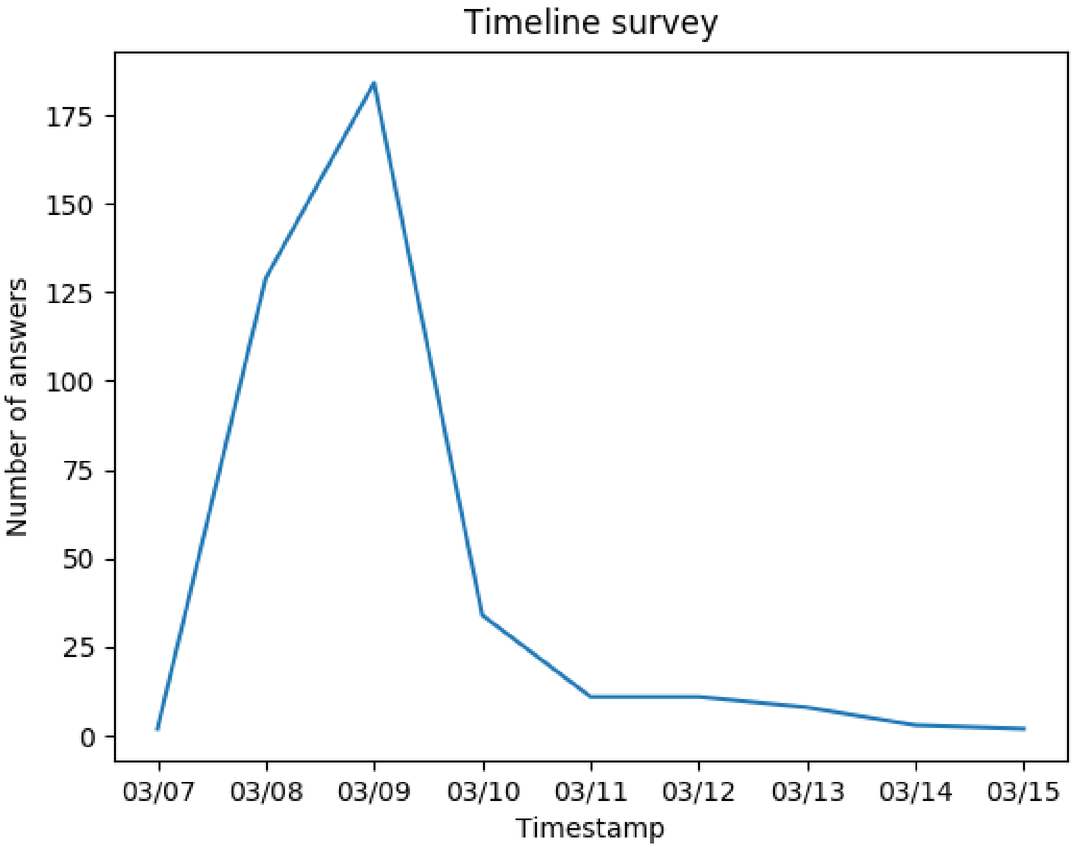
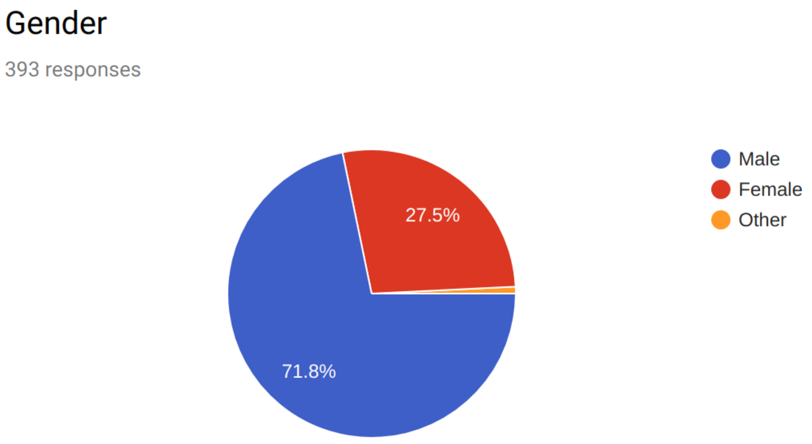


Figure 5: Timeline survey

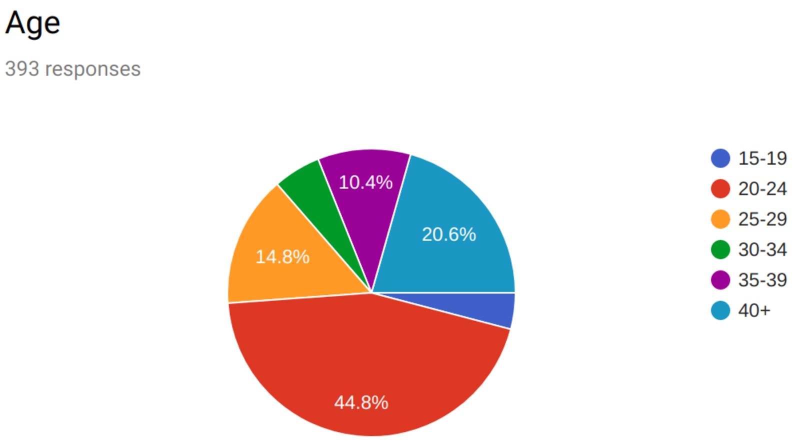
6.2 Distribution of survey answers

This section show the distribution of results from the survey.

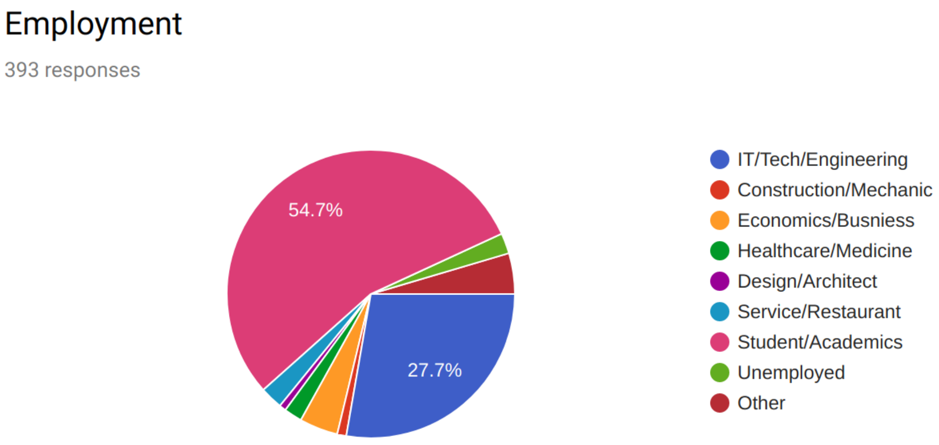
6.3 Survey Questions



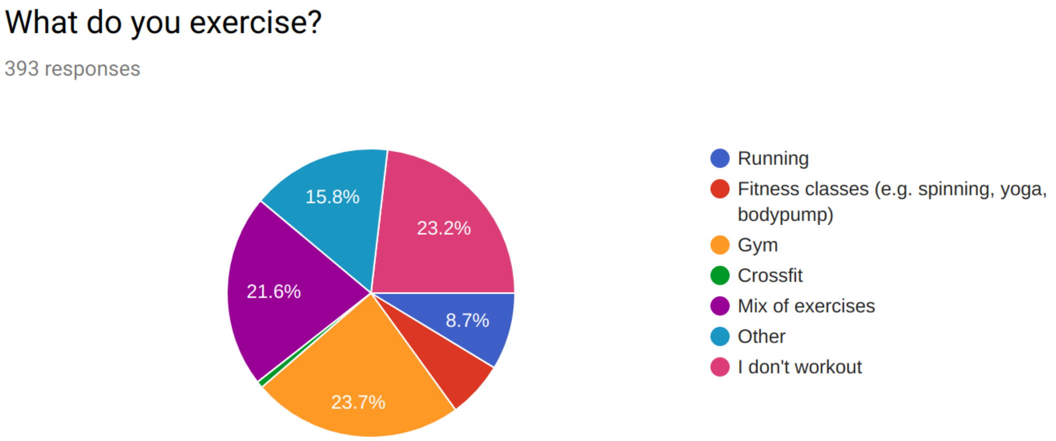
The first question asked about the gender of the responder. There are 3 possible answers, *Male, Female, Other*. This question is a categorical feature. The majority of the survey responder are men.



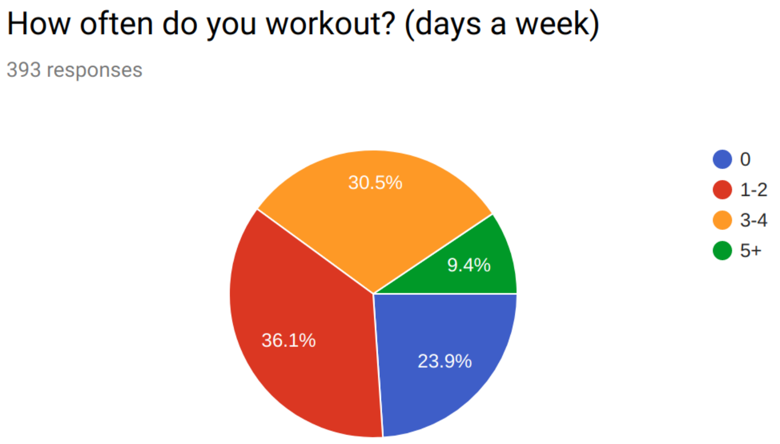
The second question considers the age of the responder. There are 6 possible answers for this question, *15-19, 20-24, 25-29, 30-34, 35-39, 40+*. This question can be used as an numeric feature. Most of the responders are in the age group *20-24*, and the rest of the responders are distributed quite similar over the other age groups.



The third question considers the profession of the responders. There are 9 possible answer, *IT/Tech/Engineering, Construction/Mechanic, Economics/Business, Healthcare/Medicine, Design/Architect, Service/Restaurant, Student/Academics, Unemployed, Other*. This question can be used as an categorical feature. The majority of the responders are *students*. The second largest group works in *IT/Tech/Engineering* and quite few of the responders work in other fields.



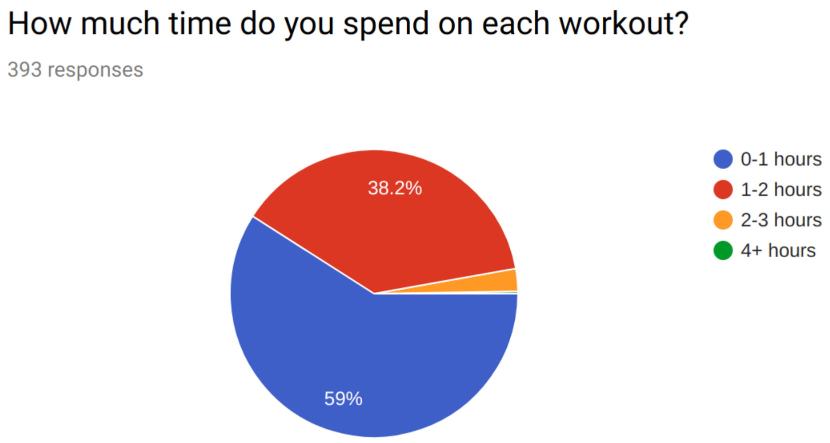
The fourth question considers what kind of exercises the survey responders do. There are 8 possible answer, *Running, Fitness classes (e.g. spinning, yoga, bodypump), Gym, Crossfit, Mix of exercises, Other, I don't workout*. This question can be used as a categorical feature. The distribution of the answers are well distributed over the different options.



How often do you workout? (days a week)

393 responses

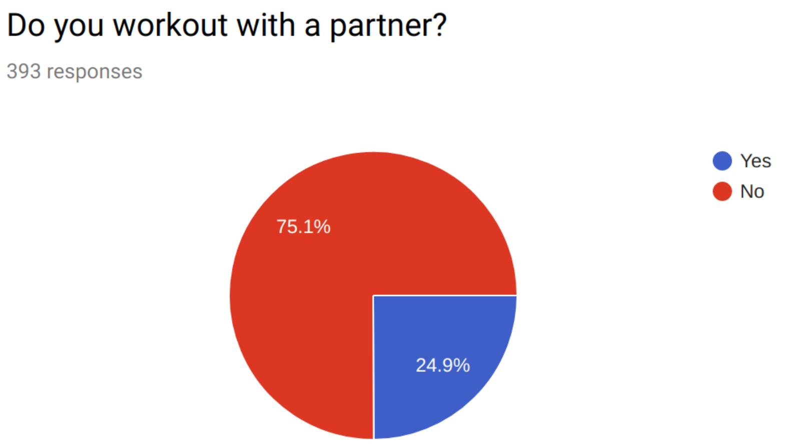
The fifth question considers the frequency of how often the survey responders exercises. There are 4 possible answers *0, 1-2, 3-4, 5+*. This question can be used as an numeric feature.



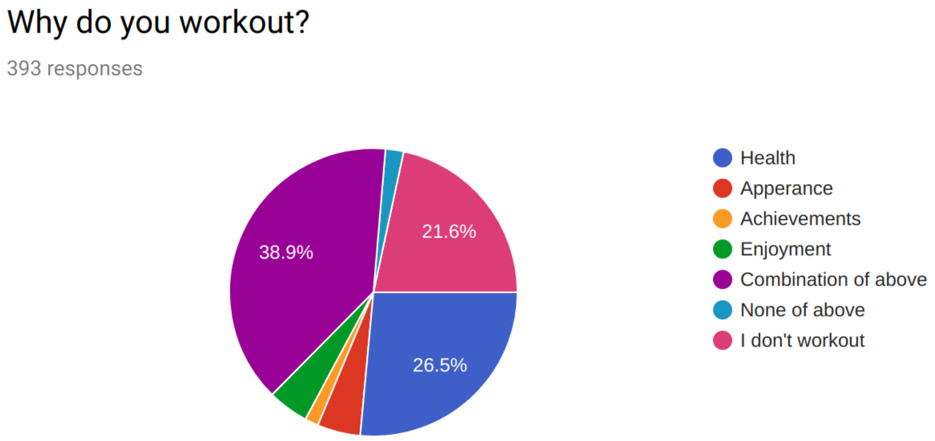
How much time do you spend on each workout?

393 responses

This question considers the amount of time that the survey responders spend on each exercise. There are 4 possible answer, *0-1 hours, 1-2 hours, 2-3 hours, 4+ hours*. This feature can be used as an numeric feature. The majority of the responders exercises 0-1 hour each time.



This question considers whether the responders exercise with an parter or not. There are 2 possible answers on this question, *Yes, No*. This question can be used as a categorical feature. The majority of the responders exercises without a partner.



This question considers why the responders exercise. There are 7 possible answers on this question, *Health, Appearance, Achievements, Enjoyment, Combination of above, None of above, I don't workout*. This question can be used as a categorical feature.

6.4 Literature Study

In this section we present the result of research question 1.

6.5 RQ1: In which fields DBSCAN and K-means can be used in?

The literature studied for this research question is presented and compared to each other in the *Literature Review* chapter and the results from it is presented here.

From the literate study, it shows that there are multiple different fields were the algorithms can and have been applied. Clustering with DBSCAN and K-means are applied in fields such as malware classification and detection[9], traffic classification of network data[5], anomaly detection[12] and traffic anomaly detection of network data[17], image segmentation[3][7], public health[15] and climate studies[16].

6.6 Experiment

This section will present the result from the clustering with both DBSCAN and K-means. First presenting the result from DBSCAN and then K-means.The feature selection was done by testing different combination of features until we got a satisfying result. The we believe the selected combination we have in the results was a good representation on how important feature and parameter selection is to find patterns. There will be no visualization for the result as each feature adds another dimension and it's impossible to visualize more than 3 dimensions in a graph. There will be a case where we have 3 features that are all numerical but we want all results to be consistent with each other.

6.7 RQ2: What observations can we make from the patterns, using the unsupervised clustering algorithm DBSCAN on peoples training habits with a small dataset (max 500 data points)?

All 8 features

From the results of the clustering with all 8 features, there were 18 clusters and 210 outliers. When looking at the results it's hard to determine any patterns since there are a large number of small clusters and a large number of outliers. While observing the small clusters you can see that the rows are very similar and everything that just slightly different is made into an outlier.

Consider only numerical features (3 features), i.e. *Age, How often do you workout?(days a week)* and *How much time do you spend on each workout?*

From the results of the clustering with this set of features, 8 clusters were found and 13 outliers. We observed that the individual clusters contained people with the same training habits, i.e. the same amount of time spent during each workout and the same amount of days per week spent. Besides two clusters who had the same training habits but different age groups. One observation of the outliers was that some responders of the survey had typed an invalid input, such as they workout 1-2 hours each workout but at the same time answered that they don't workout at all.

The features *Gender, Age* and *Employment*

From the results of the clustering with this set of features, 11 clusters were found and 17 outliers. Each individual cluster contains one specific gender and one specific employment, e.g. one of the clusters only contain gender male and unemployed. From the three features used when clustering, only age has different values in the individual clusters. An observation is that so few chose the employment options *Construction/Mechanic* and *Design/Architect* that they were included as outliers. Another observation is that everyone who chose *Other* in the Gender category was seen as an outlier. There was only one female that had picked the unemployed option in the employment category so she was seen as an outlier while 7 males picked unemployed and formed a cluster, which means that there are more men that are unemployed than females in this dataset.

The features *Gender, How often do you workout?* and *How much time do you spend on each workout?*

From the results of the clustering with this set of features, 10 clusters were found and 19 outliers. An observation is that the male students that don't work out are overrepresented in our dataset. The clusters tend to vary a lot on the other features.

The features *What do you exercise?, How often do you workout? (days a week), How much time do you spend on each workout?,Do you workout with a partner?,Why do you workout?*

The result from the clustering is 29 clusters and 27 outliers. The number of clusters is high compared to the previous tests, by observing it's possible to see each cluster has very similar rows. The problem being that even clusters are very similar as there are 3 clusters where people gym and mixed exercise as there form of workout. The difference in the gym clusters if they work with a partner, many times a week and few times a week. As there are so many similarities in the feature selection it's harder to identify difference from cluster to cluster.

The features *What do you exercise?, How often do you workout? (days a week), How much time do you spend on each workout?,Why do you workout?*

The result from the clustering is 16 clusters and 27 outliers. While it's the same number of outliers as the previous they're not the exact same. By simply removing 1 feature and adjusting the parameters for the feature selection the number of clusters decreased by 13. It's a lot easier to see patterns by looking at 16 clusters instead of 29. The result is still very similar however to the previous clustering run with having similar clusters.

It's quite clear how much can change by removing a yes and no feature by doing what we did in this test.

The features *Employment, What do you exercise?, How often do you workout? (days a week), How much time do you spend on each workout?*

The result from clustering is 19 clusters and 41 outliers. The amount of clusters is high but it's highly possible to see what employment does what because the clustering is based on employment and what they workout on. The number of outliers is high but changing the minpoints will make too many clusters that are too similar.

6.8 RQ3: What observations can we make from the patterns, using the unsupervised clustering algorithm K-means on peoples training habits with a small dataset (max 500 data points)?

All 8 features

From the results of the clustering with 5 predefined clusters. There are differences between the samples in the individual clusters. By observing, it's clear there are similarities in the clusters but with these amount of features on 5 clusters, it's hard to see much difference in each cluster. In one cluster the similarities are very clear however all of the rows with the *I don't workout* are all in the same cluster.

From the results of the clustering with 15 predefined clusters. It is possible to identify that similar behavior has clustered together with a few exceptions where anomalies are inside cluster that doesn't relate much to anything else inside the cluster. You can also see that some clusters get bigger than others, for example, *I don't workout* being almost twice as big as the other clusters.

Consider only numerical features (3 features), i.e. *Age, How often do you workout?(days a week)* and *How much time do you spend on each workout?*

From the results of the clustering with 5 predefined clusters. It is possible to identify almost all samples that answered *I don't workout.* You can see the difference in how age groups train by looking at each individual cluster.

From the results of the clustering with 15 predefined clusters. The clustering is a lot more specific with it not finding many similarities in each cluster by looking at the numerical data. Mostly clusters on the identical choices from the numerical. What is positive it that you can see that the other features that are not clustered upon are not related to the clustering.

The features *Gender, Age* and *Employment*

From clustering on the features with 5 predefined clusters, it's hard to see any patterns as all the frequently seen employment are in one cluster while the small represented are mixed together. As students are very overrepresented they have two clusters on their own separated by gender.

From clustering on the features with 15 predefined clusters, it's clear what employment frequently comes up. The clusters contain an employment with an age and gender group. The larger clusters contain frequently seen employments while smaller clusters contain less frequently seen employments.

The features *Gender, How often do you workout?* and *How much time do you spend on each workout?*

From clustering on the features with 5 predefined clusters, it's not possible to clearly see any patterns, there is one cluster where it's possible to see patterns, the clustering being *I don't work out* with the gender *Male.*

From clustering on the features with 15 predefined clusters, it's even more unclear as the training habits don't append to anything similar.

The features *What do you exercise?, How often do you workout? (days a week), How much time do you spend on each workout?,Do you workout with a partner?,Why do you workout?*

From the results of the clustering with 5 predefined clusters. The clusters are very messy besides a cluster where each row contains "I don't work out" as there form of exercise.

From the results of the clustering with 15 predefined clusters. By adding 10 more clusters it's more possible to see what clusters represent. Many clusters are similar clusters where people gym and mixed exercise as there form of workout. The difference in the gym clusters if they work with a partner, many times a week and few times a week. However there was one cluster where each row didn't have much in common, it's like a group of anomalies formed in that cluster.

The features *What do you exercise?, How often do you workout? (days a week), How much time do you spend on each workout?,Why do you workout?*

From the results of the clustering with 5 predefined clusters. The clustering is based on how much time and times a week besides over-represented "I don't work out" and "Gym". The clustering has two different patterns making it contradict itself.

From the results of the clustering with 15 predefined clusters. Adding 10 more clusters will give more similar results as of the previous feature selection clustering run. The clustering is more based on how much time and times a week. An observation to take is that overrepresented forms of exercise gym and mixed exercises are divided into 3 clusters by how much time and times a week. The cluster who looked like each row that didn't have much in common with the previous clustering is gone with this feature selection.

The features *Employment, What do you exercise?, How often do you workout? (days a week), How much time do you spend on each workout?*

From the results of the clustering with 5 predefined clusters. The clustering is mostly based on employment, as over-represented Students/Academics and IT/Tech/Engineering are having their own clusters wherein some mixed clusters they also appear. Even by observing at the clusters where only the over-represented are they have very mixed results so it's hard to see any patterns from this clustering run.

From the results of clustering with 15 predefined clusters. With 10 more added clusters it's more possible to see what employment does what with the exceptions of a small number of anomalies in some clusters.

7 Analysis

7.1 Literature study

During the literature study, we found 8 research papers we decided to use as our main literature to answer research question number 1. The content of the papers was of great value because it presented fields in which DBSCAN and K-means are implemented and used. There were no major problems in finding the literature. The validation of the credibility of the papers was done by only choosing papers written by researchers we found on search engines such as Microsoft Academics and Google Scholar.

7.2 Survey

The survey was open for 9 days and had the expectation of at least 100 answers. The survey was sent by email to all student of BTH and shared on social networks such as Facebook and Slack. The total number of answers was 393, which we consider a success. We are satisfied with the number of questions and the survey overall. Concerning the possibility to use the features in the clustering, we could have done some adjustments in the way we formulated them. The majority of the questions can only be used as categorical features, which led to an increase in the space of dimensions. On the other hand, it would not be possible to only have questions that we could have used as numeric features.

7.3 Experiment

The experiment got very mixed results, it's what we expected but we thought it would be easier to find patterns. When testing on fewer features the clustering had clearer patterns. This is why *I don't work out* comes up in almost every clustering run with different selection feature as most options are the same if you don't work out. It was easier to identify patterns when we used less categorical data, the reason for this can either be because of the way we handle categorical data, where each possible option for each question becomes a dimension/new feature. It can also be because of the number of possible options on each question in combination with the few numbers of data points which result in many variations in the dataset. With all this in mind, it's definitely possible to see patterns using a small dataset.

K-means and DBSCAN both managed to show patterns in their clusters, both got similar results with some few exceptions. The main difference is that k-means worked better with many features while DBSCAN had better clustering results on a small number of features. As we expected we got a better generalization by looking at K-means and DBSCAN identified outliers and more specific clusters. This is similar results to [5], that concludes that *"The experimental results show that both K-Means and DBSCAN work very well and much more quickly then AutoClass. The results indicate that although DBSCAN has lower accuracy compared to K-Means and AutoClass, DBSCAN produces better clusters."* In the paper [9] it's stated that K-means was not available to find malware families while DBSCAN was successfully able to identify malware families.

From reading the literature we expected DBSCAN to over perform against K-means as it did. The parameter selection is harder for DBSCAN however as finding the right epsilon and minpoints is harder than finding a good number of clusters of K-means. This makes K-means easier to work with while DBSCAN has a better clustering quality if you spent the right amount of effort with the parameter selection. While using categorical data there is a fine line for epsilon where if you cross a value very slightly there is a huge difference. This makes DBSCAN a lot more sensitive but it's a lot more customizable for specific scenarios. In the [7] they use K-means to segment the dataset into smaller segments, because of the high computational complexity of DBSCAN on a to large dataset. This show that it could even be beneficial in some cases to reduce the size of a dataset.

While it's easier to see patterns from using less features, finding the correct feature selection is crucial in order to find the best possible patterns. On the last two tests it's clear how much a specific pattern changed the result, e.g. *Do you work out with a partner?*, as it's simply a yes and no feature it grants high identity and loses similarities. To get good results clustering for any dataset it's important that you analyze the data and understand the features. Having more data is always better than having less and you need to have a threshold. Meaning the data can be very minimal and still be able to show patterns with a good feature selection.

8 Conclusion

This thesis investigates 2 unsupervised machine learning clustering algorithms, DBSACN(Density-Based Spatial Clustering of Applications with Noise) and K-means on a dataset gathered through a survey. During this thesis, we were able to answer all our research questions. The thesis has been conducted through a literature study, a survey and an experiment. The literature study conducted during this thesis aimed to answer the first research question, i.e. *In which fields can DBSCAN and K-means be used in?*, and to get further knowledge about the algorithms and in what fields DBSCAN and K-means where used and how they were applied. We found that clustering with DBSCAN and K-means are applied in fields can and have been applied in a variety of areas. For example in malware classification and detection, traffic classification of network data, anomaly detection, image segmentation, public health and climate studies.

During our investigations we wanted unlabeled data where we know statistically what answers that comes up more often so we know that the clustering is accurate. We confronted this problem by making a survey, as we didn't need to make a specific topic we took a general topic which was training habits to get as many answers as possible. Because we wanted as many people to answer as possible we also didn't want people to quit during the survey so we made few questions as we needed all questions mandatory to see patterns of the clustering.

The experiment on finding patterns inside the clusters made by K-means and DBSCAN resulted with mixed results. K-means made broader generalization which resulted in making it possible to see patterns but with a few inconsistency in the clusters because of the lack of anomaly detection and that you need to select the number of clusters beforehand. While using fewer features on our dataset the clusters had a larger ratio between each other with more consistent patterns. While DBSCAN, on the other hand, had a harder time to get a general clustering conclusion, reducing the number of features made DBSCAN more specific than K-means with the power to find anomalies and arbitrary shapes. Having a larger size on the dataset will give a more accurate representation from the questions which in turn gives a more accurate clustering to the real world. The size certainly helps but the biggest factor in having good clustering quality is selecting the correct features for the clustering as it represents the data in the fashion you desire.

9 Future Work

To continue this work, it would be possible to test with more algorithms. It would also be possible to test with other datasets. The experiment can also be extended by adding different combinations of features and tuning the parameters for the algorithms with different values. It would also be possible to use different cluster quality measurements to evaluate individual clusters. The literature study can also be extended by reading more relevant papers.

10 Annexes

10.1 Survey Questions

In this subsection we present our survey questions. All the questions are multiple choice questions and all of them are mandatory. It is only possible to select one answer on each question.

1. Gender
   - Male
   - Female
   - Other
2. Age
   - 15-19
   - 20-24
   - 25-29
   - 30-34
   - 35-39
   - 40+
3. Employment
   - IT/Tech/Engineering
   - Construction/Mechanic
   - Economics/Busniess
   - Healthcare/Medicine
   - Design/Architect
   - Service/Restaurant
   - Student/Academics
   - Unemployed
   - Other
4. What do you exercise?
   - Running
   - Fitness classes (e.g. spinning, yoga, bodypump)
   - Gym
   - Crossfit
   - Mix of exercises
   - Other
   - I don't workout
5. How often do you workout? (days a week)
   - 0
   - 1-2
   - 2-3
   - 3-4

  - 5+

6. How much time do you spend on each workout?
    - 0-1 hours
    - 1-2 hours
    - 2-3 hours
    - 4+ hours
7. Do you workout with a partner?
    - Yes
    - No
8. Why do you workout?
    - Health
    - Apperance
    - Achievements
    - Enjoyment
    - Combination of above
    - None of above
    - I don’t workout

10.2 Code

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import OneHotEncoder, normalize
from sklearn import cluster

# Map all values to numeric data to prepare for one-hot preprocessing
def preprocess_data():
    # Get numeric data
    data = make_all_values_numeric()

    data = map_categorical_data(data)
    data = normalize_data(data)
    return data


def make_all_values_numeric():
    data = pd.read_csv("survey.txt")

    # Remove timestamp columns
    data.drop(labels=["Timestamp"], axis=1, inplace=True)

    possible_answers = [    ["Male", "Female", "Other"],
                    ["15-19", "20-24", "25-29", "30-34", "35-39", "40+"],
                    ["IT/Tech/Engineering", "Construction/Mechanic", "Economics/Busni
                    ["Running", "Fitness classes (e.g. spinning, yoga, bodypump)", "Gy
                    ["0", "1-2", "3-4", "5+"],
                    ["0-1 hours", "1-2 hours", "2-3 hours", "4+ hours"],
                    ["Yes", "No"],
                    ["Health", "Apperance", "Achievements", "Enjoyment", "Combination
                ]

    for i, columns in enumerate(data):

        for index, element in enumerate(possible_answers[i]):
            data[columns].replace(to_replace=element, value=index, inplace=True)

    return data

'''
Encode categorical integer features using a one-hot aka one-of-K scheme.
'''
def map_categorical_data(data):
    mask = [True, False, True, True, False, False, True, True]
    enc = OneHotEncoder(categorical_features=mask, sparse=False)
    data = enc.fit_transform(data)

    return data

def normalize_data(data):
    return normalize(data, norm="max", axis=0)

def cluster_data(data):
    # Initiate the DBSCAN model
    #dbscan = cluster.DBSCAN(eps=1.5)
    dbscan = cluster.KMeans(n_clusters=5)
    dbscan.fit(data)
    return dbscan

def get_full_information_about_outliers(data):
    temp = pd.read_csv("survey.txt")
    temp = temp.values
    for i, e in enumerate(data):
        if(e == -1):
            print(temp[i])

def get_full_information_about_cluster(data):

    temp = pd.read_csv("survey.txt")
    temp = temp.values

    for n in (list(set(data))):
        print("Cluster number: {} = ".format(n))
        for i, e in enumerate(data):
            if(e == n):
                print(temp[i])

        print("--------------")



def main():

    # Get preprocessed data
    data = preprocess_data()

    # Cluster the data
    data_clustered = cluster_data(data)

    print("Number of samples = {}".format(len(data_clustered.labels_)))

    # Get number of outliers for DBSCAN
    outliers = [line for line in data_clustered.labels_ if line == -1]

    print(data_clustered.labels_)

    # Get number of clusters
    number_of_clusters = len(list(set(data_clustered.labels_)))

    # Print out all samples that are outliers for DBSCAN
    #get_full_information_about_outliers(data_clustered.labels_)
    get_full_information_about_cluster(data_clustered.labels_)

    print("Number of clusters: {}".format(number_of_clusters))
    print("Number of outliers = {}".format(len(outliers)))
if __name__== "__main__":
    main()
```

References

[1] Erik Brynjolfsson and Tom Mitchell. What can machine learning do? workforce implications. *Science*, 358(6370):1530–1534, 2017.

[2] JOHN R BUCKLEY. Automation. *Journal of Academic Librarianship*, 20(1):40, 1994.

[3] Nameirakpam Dhanachandra and Yambem Jina Chanu. A new approach of image segmentation method using k-means and kernel based subtractive clustering methods. *International Journal of Applied Engineering Research*, 12(20):10458–10464, 2017.

[4] Jennifer G Dy and Carla E Brodley. Feature selection for unsupervised learning. *Journal of machine learning research*, 5(Aug):845–889, 2004.

[5] Jeffrey Erman, Martin F. Arlitt, and Anirban Mahanti. Traffic classification using clustering algorithms. In *Proceedings of the 2006 SIGCOMM workshop on Mining network data*, pages 281–286, 2006.

[6] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

[7] Chun Guan, Kevin Kam Fung Yuen, and Qi Chen. Towards a hybrid approach of k-means and density-based spatial clustering of applications with noise for image segmentation. In *Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2017 IEEE International Conference on*, pages 396–399. IEEE, 2017.

[8] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[9] Joris Kinable and Orestis Kostakis. Malware classification based on call graph clustering. *Journal in Computer Virology*, 7:233–245, 2011.

[10] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24, 2007.

[11] David Neil Lawrence Levy. *Computer Games I*. Springer, 2011.

[12] L. Li, M. Gariel, R. J. Hansman, and R. Palacios. Anomaly detection in onboard-recorded flight data using cluster analysis. In *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, pages 4A4–1–4A4–11, Oct 2011.

[13] Mark S Litwin and Arlene Fink. *How to measure survey reliability and validity*, volume 7. Sage, 1995.

[14] J. MacQueen. Some methods for classification and analysis of multivariate observations. Proc. 5th Berkeley Symp. Math. Stat. Probab., Univ. Calif. 1965/66, 1, 281-297 (1967)., 1967.

[15] Divyani Sanjay Mane and Balasaheb B Gite. Brain tumor segmentation using fuzzy c-means and k-means clustering and its area calculation and disease prediction using naive-bayes algorithm. *Brain*, 6(11), 2017.

[16] Jenny Matthews and John Trostel. An improved storm cell identification and tracking (scit) algorithm based on dbscan clustering and jpda tracking methods. *American Meteorological Society, Atlanta, GA.[online] Available from: http://ams. confex. com/ams/90annual/techprogram/paper_ 164442. htm*, 2010.

[17] Gerhard Münz, Sa Li, and Georg Carle. Traffic anomaly detection using k-means clustering. In *GI/ITG Workshop MMBnet*, 2007.

[18] Alexander Novikov, Mikhail Trofimov, and Ivan Oseledets. Exponential machines. *arXiv preprint arXiv:1605.03795*, 2016.

[19] Pandas-documentation. pandas.dataframe¶. `https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html`, March 2018.

[20] sklearn documentation. 4.3.5. encoding categorical features¶. `http://scikit-learn.org/stable/modules/preprocessing.html#encoding-categorical-features`, March 2018.

[21] sklearn documentation. sklearn.preprocessing.onehotencoder¶. `http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html`, March 2018.

[22] Nainsi Soni and Manish Dubey. A review of home automation system with speech recognition and machine learning. *International Journal*, 5(4), 2017.

[23] Hugo Steinhaus. Sur la division des corps matériels en parties. *Bull. Acad. Pol. Sci., Cl. III*, 4:801–804, 1957.