

**APPLYING SUPERVISED LEARNING ALGORITHMS ON INFORMATION
DERIVED FROM SOCIAL NETWORKS**

by

Hesaneh Behzadfar

B.E. in Information Technology, Iran University of Science and Technology, Iran, 2008

A thesis

presented to Ryerson University

in partial fulfillment of the
requirements for the degree of

Master of Science

in the Program of

Computer Science

Toronto, Ontario, Canada, 2014

©Hesaneh Behzadfar 2014

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

Applying Supervised Learning Algorithms on Information Derived from Social Network to Enhance Recommender Systems

Hesaneh Behzadfar

Master of Science, Computer Science, 2014

Ryerson University

ABSTRACT

The aim of this research is to show how social networks can be used for marketing purposes. This is implemented with the assistance of learning algorithms. The method proposed in this research is based on the analysis of “Support Vector Machines”, which facilitates analysis of all information gathered from the social websites. It differs from other methods currently being used by social networking websites, which do not take advantage of classification.

By using public information from social networks, a dataset was formed. It comprised of a thousand users and seven features. The examined features were location, age, gender, occupation, relationship status, and average travel time/year. In this research, the dataset will be examined twice: first using a regular SVM; and next by using “Weighted Feature Support Vector Machines”. For the latter, to assign weights, a method called “Pairwise Comparison” will be used to rank the importance of features.

ACKNOWLEDGEMENTS

I am indebted to the kind people around me. Without their help and support, it would not have been possible to write this thesis.

It gives me great pleasure in acknowledging the support of my supervisor Dr. Abdolreza Abhari, for his valuable input and his patience during this period, whose effort made this thesis possible. I would like to thank him for the time he spent in our many weekly meetings and also his effort in reviewing my thesis.

Furthermore, I am grateful to Dr. Isaac Woungang for assuming the role of Defence Chair for my thesis defence. I am equally grateful to the other members of the Thesis Defence Committee, Dr. Jelena Misic and Dr. Vojislav B Misic, for agreeing to review and comment on my work, and also teaching me about research methods and giving me valuable insight about my work in the seminar course.

I also thank my entire family for their support, especially my mother Dr. Ghaemmaghami, for her amazing patience and encouragement throughout my studies, Aneseh for her inputs, and last but not least my grandmother, god rest her soul, who prayed for me every single step of the way.

Finally, I wish to express my gratitude to Jude Amos for his tireless efforts in the proofreading and linguistic editing of the thesis.

TABLE OF CONTENTS

AUTHOR'S DECLARATION	ERROR! BOOKMARK NOT DEFINED.
ABSTRACT	III
ACKNOWLEDGEMENTS	IV
LIST OF TABLES.....	VII
LIST OF FIGURES.....	IX
LIST OF ACRONYMS	IX
CHAPTER 1 INTRODUCTION AND PROBLEM STATEMENT	1
1.1. THESIS STATEMENT	1
1.2. MOTIVATION	1
1.3. GOALS	3
1.4. CONTRIBUTIONS AND METHODOLOGY	3
1.5. SUMMARY.....	5
CHAPTER 2 LITERATURE REVIEW.....	6
2.1. MACHINE LEARNING AND LEARNING ALGORITHMS	6
2.1.1. Unsupervised Learning	7
2.1.2. Supervised Learning.....	7
2.2. SUPPORT VECTOR MACHINES.....	10
2.2.1. Why Weights Improve Accuracy	16
2.2.2. Values of Deviation.....	17
2.2.3. Degrees of Standard Deviation	18
2.2.4. Shannon's Information Theory	19
2.2.5. Scaling.....	19
2.3. PAIRWISE COMPARISON	22
2.4. GRAPH API.....	23
CHAPTER 3 METHODOLOGY.....	24
3.1. DATA COLLECTION.....	24
3.2. THE DATA SET.....	27
3.3. TRAINING THE CLASSIFIERS	27
3.3.1. Pairwise Comparison and Weight Calculation.....	30
3.4. PREDICTION	38
3.5. SUMMARY.....	39

CHAPTER 4 IMPLEMENTATION, RESULTS AND EXPERIMENTS	40
4.1. INTRODUCTION	40
4.2. ASSIGNING WEIGHTS TO DATASETS	40
4.2.1. Assigning Weights Using SVM	40
4.2.2. Assigning Weights using expert judgment.....	53
4.3. ANALYZING THE DATASET OR SETS.....	54
4.4. SUMMARY.....	62
CHAPTER 5 CONCLUSION AND FUTURE STUDIES.....	64
5.1. CONCLUSION	64
5.2. CONTRIBUTION.....	66
5.3.FUTURE STUDIES	66
REFERENCES	67

LIST OF TABLES

TABLE 1.1: PERSONALIZED ADVERTISING	2
TABLE 2.1: EXAMPLE OF DATA (INSTANCES WITH KNOWN LABELS) IN A STANDARD FORMAT.....	8
TABLE 2.2: THE EXPERIMENTAL RESULTS WITH SVM AND FWSVM USING VALUES OF DEVIATION [12]	17
TABLE 2.3: THE AVERAGE PRECISION AND TRAINING TIME COMPARISON OF IMAGE CLASSIFICATION [24].....	18
TABLE 3.1: GRAPH API ROOT NODES AND THEIR REPRESENTATIONS [28].....	26
TABLE 3.2: COMPARISON OF LEARNING ALGORITHMS [7]	28
TABLE 3.3: COMPARISON OF TWO-SOME VARIATIONS OF SVM.....	35
TABLE 3.4: COMPARISON OF THREE-SOME VARIATIONS OF SVM.....	37
TABLE 4.1: RESULTS FROM ONE-DIMENSIONAL SVM	41
TABLE 4.2: PAIRWISE COMPARISON TABLE FOR ONE DIMENSIONAL RENDERING OF SVM	41
TABLE 4.3: THE WEIGHTS OF FEATURES CALCULATED BY GEOMETRIC MEANS.....	42
TABLE 4.4: THE TWO DIMENSIONAL FEATURE COMBINATIONS AND ACCURACY INCLUDING FEATURE 1	43
TABLE 4.5: THE TWO DIMENSIONAL FEATURE COMBINATIONS AND ACCURACY INCLUDING FEATURE 2	43
TABLE 4.6: THE TWO DIMENSIONAL FEATURE COMBINATIONS AND ACCURACY INCLUDING FEATURE 3	44
TABLE 4.7: THE TWO DIMENSIONAL FEATURE COMBINATIONS AND ACCURACY INCLUDING FEATURE 4	44
TABLE 4.8: THE TWO DIMENSIONAL FEATURE COMBINATIONS AND ACCURACY INCLUDING FEATURE 5	44
TABLE 4.9: THE TWO DIMENSIONAL FEATURE COMBINATIONS AND ACCURACY INCLUDING FEATURE 6	45
TABLE 4.10: THE TWO DIMENSIONAL FEATURE COMBINATIONS AND ACCURACY INCLUDING FEATURE 7	45
TABLE 4.11: ACCURACY RATE OF FEATURES BASED ON TWO DIMENSIONAL RENDERING OF SVM.....	45
TABLE 4.12: PAIRWISE COMPARISON TABLE FOR TWO DIMENSIONAL RENDERING OF SVM	46
TABLE 4.13: THE WEIGHTS OF FEATURES CALCULATED BY MEANS OF GEOMETRIC MEANS	46
TABLE 4.14: THE THREE DIMENSIONAL FEATURE COMBINATIONS AND ACCURACY INCLUDING FEATURE 1	47
TABLE 4.15: THE THREE DIMENSIONAL FEATURE COMBINATION AND ACCURACY INCLUDING FEATURE 2	48
TABLE 4.16: THE THREE DIMENSIONAL FEATURE COMBINATION AND ACCURACY INCLUDING FEATURE 3	49
TABLE 4.17: THE THREE DIMENSIONAL FEATURE COMBINATION AND ACCURACY INCLUDING FEATURE 4	49
TABLE 4.18: THE THREE DIMENSIONAL FEATURE COMBINATION AND ACCURACY INCLUDING FEATURE 5	50

TABLE 4.19: THE THREE DIMENSIONAL FEATURE COMBINATION AND ACCURACY INCLUDING FEATURE 6	51
TABLE 4.20: THE THREE DIMENSIONAL FEATURE COMBINATION AND ACCURACY INCLUDING FEATURE 7	51
TABLE 4.21: ACCURACY RATE OF FEATURES BASED ON THIRD DIMENSIONAL RENDERING OF SVM	52
TABLE 4.22: PAIRWISE COMPARISON TABLE FOR THREE DIMENSIONAL RENDERING OF SVM	52
TABLE 4.23: THE WEIGHTS OF FEATURES CALCULATED BY MEANS OF GEOMETRIC MEANS	53
TABLE 4.24: INFLUENCE RATE OF FEATURES BASED ON EXPERT JUDGMENT USING A 1-5 SCALE	53
TABLE 4.25: PAIRWISE COMPARISON TABLE BASED ON EXPERT JUDGMENT	54
TABLE 4.26: THE WEIGHTS OF FEATURES CALCULATED BY GEOMETRIC MEANS	54
TABLE 4.27: THE RESULTS OF DIFFERENT WEIGHTS	60
TABLE 4.28: COMPARISON OF RESULTS BETWEEN UN-WEIGHTED AND WEIGHTED SVM (FWSVM)	61

LIST OF FIGURES

FIGURE 2.1: THE SUPERVISED MACHINE LEARNING PROCESS OF KOTSIANTIS <i>ET AL.</i> (2007)[7].....	9
FIGURE 2.2: LINEARLY SEPARABLE DATA	11
FIGURE 2.3: WHEN NO SEPARATING HYPERPLANE EXISTS SVM USES A SOFT MARGIN HYPERPLANE	12
FIGURE 2.4: SVM MAPPING DATA FROM INPUT SPACE (A) TO A HIGHER DIMENSIONAL FEATURE SPACE (B).....	13
FIGURE 2.5: SVM FEATURE MAPPING THE (2D) INPUT DATA SPACE TO SOME OTHER (3D) DIMENSIONAL SPACE; $\Phi: R^2 \rightarrow R^3, x_1, x_2 \rightarrow (z_1, z_2, z_3)$	14
FIGURE 3.1: HOW WEIGHTS ARE CALCULATED WHEN DATA GOES THROUGH THE PAIRWISE COMPARISON PROCESS.....	32
FIGURE 4.1 : ROC CURVE FOR 1 DIMENSIONAL WEIGHTS.....	57
FIGURE 4.2 : ROC CURVE FOR 2 DIMENSIONAL WEIGHTS.....	58
FIGURE 4.3 : ROC CURVE FOR 3 DIMENSIONAL WEIGHTS.....	60
FIGURE 4.4 : ROC CURVE COMPARISON	61
FIGURE 4.5: THE COMPARISON BETWEEN PREDICTED RESULTS AND ACTUAL RESULTS.....	63

LIST OF ACRONYMS

ACRONYMS

AI

ML

API

ANN

HTTP

IT

SVM

FWSVM

WFSVM

NN

NSV

TA

kNN

2D

3D

DEFINITIONS

Artificial Intelligence

Machine Learning

Application Programming Interface

Artificial Neural Network

Hypertext Transfer Protocol

Information Technology

Support Vector Machine

Feature Weighted Support Vector Machine

Weighted Feature Support Vector Machine

Neural Network

Number of Support Vectors

Total Accuracy

k-Nearest Neighbour

Two Dimensional

Three Dimensional

CHAPTER 1

INTRODUCTION AND PROBLEM STATEMENT

1.1. Thesis Statement

This thesis covers the fields of Artificial Intelligence (AI), Machine Learning (ML), Data Analysis and Social Networking. Specifically, it uses certain AI tools to gather data from social networking websites, in particular Facebook, for the purpose of increasing the accuracy of prediction by improving the performance of the employed machine learning approach. It then applies the proposed method to a dataset gathered by the Facebook Application Programming Interface (API).

1.2. Motivation

Social networking websites have become very popular and are used by an increasing number of people to interact daily, exchange information online and even seek advice or obtain ideas from other people's social networking sites. Due to the open nature of Facebook, Facebook users supply the Facebook database with information ranging from location to explicitly defined interests. Apart from providing a space for people to socialize with old acquaintances and meet new people, social networks have increasingly developed into an arena for teaching, marketing, and even advertising. Social networks such as Facebook, Twitter, LinkedIn and many more have been growing faster with their ever-growing number of members. They each have their own target group, but users may join any of them to become part of a special community, such as business people, computer engineers, fashionistas, etc., or they could just become part of a larger community and interact with friends. Facebook has been one of the largest social networks in the world, with more than one billion monthly and 618 million daily active users as of December 2012 [1]. According to checkfacebook.com, Canada has nearly 18 million users, which amount to a 52.83% penetration into the market [2]. Facebook is not only suitable for advertising and expanding already made businesses, but also for businesses that have not yet been established or have yet to make a name for themselves. It makes most of its money through advertising [3]. Its approach is to show the demographic distribution of users to marketers and researches, which will allow them better to understand the market they are targeting. Table 1.1 shows the current advertising technique used by Facebook.

Table 1.1: Personalized advertising

1	2	3
A business creates an ad A gym opens in a specific neighbourhood. The owner creates an ad to make people come in for a free workout.	Facebook gets paid to deliver the ad The owner sends the ad to Facebook and describes who should see it: people who live in that neighbourhood	The right people see the ad Facebook shows the ad to users who live in that specific town and like to run. That is how advertisers reach targeted users without Facebook sharing their private info.

Following this approach, Facebook decides which ads to show to the user, based on two types of information: (a) the things that users share, such as pages that a user has liked or their location, and (b) information received through third parties that collect non-personally identifiable information about users' activities when they visit the parties' websites [4]. Such information may include the content users have viewed, the date and time they viewed this content, products purchased, or location information associated with users' IP addresses. Seeing the ever-increasing membership of social networks, businesses are increasingly drawn to such a market to advertise. It is environment friendly, cannot be marked as spam and is much less costly.

Accompanying the growth in Information Technology and Social Networks have been the advancements in Artificial Intelligence. Research on Artificial Intelligence has raised many new issues and ventures and given birth to many disciplines, many of which are focused on the ability of machines to mimic human traits such as reasoning, judgement and foresight. Machine Learning is the science dedicated to simulating such human behaviour, with the main goal being to design a machine that can manifest the same level of human intelligence or one much higher. The ability to make predictions is a capability of the human mind. However, if machines were able to do that by themselves, with the help of AI and Machine Learning (ML), which they presently can to some extent, it would play an important role in marketing, businesses and many aspects of life that all depend on possessing knowledge about the future trends.

1.3. Goals

The goal of this thesis is to increase the efficiency of social network marketing and advertising by means of Machine Learning. The fundamental idea behind this research is to have machines do what the human mind can do, but at a much faster rate and in a more accurate manner. Humans are by far the most advanced species with regard to learning new things and making decisions, because when placed in a situation they have never faced before, they draw upon previously gained knowledge to make a decision. They can also assign varying degrees of importance to various issues in making a decision. These capabilities enable them to make decisions and predictions even when faced with a new situation. The method proposed in this research is make predictions based on training sets gathered from social networking sites, in the same manner as would have been made by a human mind. To accomplish such a feat without a ML tool, many humans and many more hours will be needed in order to make decisions for such a large demographic. The accuracy of the proposed system will be analyzed to show its greater effectiveness over currently used methods.

1.4. Contributions and methodology

What is being proposed in this research is to gather the information through Facebook API and to use all of this data to make a prediction about individual users, specifically to predict whether a user will be traveling in the following year. This will enable advertisers to target the groups that are more likely to travel. The next part of this research, which will be enhanced by future research, is to use the analysis from the data gathered to recommend suitable destinations to users. This method of marketing permits targeted advertising and has a wider spectrum. A number of methods have been applied by others in the use of social networking for business purposes, and each method has had some success. We consider the method proposed in this research to be successful even if its outcome is an improvement of only 1% over previous approaches. By this is meant that even if the proposed method has the slightest advantage over not using it, this will be a significant advantage. Although there are many other social network platforms to choose from, this study focuses on only one: Facebook. This is partly because other researchers have used similar methods in a study of the same network, and our focus therefore permits readers to see the significance of the improvement of our proposed method compared to previous ones.

We explained earlier the significance of the fact that the human mind can assign varying degrees of importance to certain aspects of an issue when making a decision. Thus, it is evident that such an attribute should be incorporated into the design of a more reliable Machine Learning environment, thus requiring the assignment of weights to specific features. There are several ways in which such an assignment can be accomplished. In this thesis, we will use pairwise comparison, which is the preferred method used with the goal of integrating human subjectivity [5]. It was chosen because it makes it easier to rank the importance of objects in pairs, rather than several at the same time.

In this thesis, we start with the information gathered through the Facebook API, and then use pairwise comparison to rank the different features in the data set. After calculating the weights, we use Weighted Support Vector Machines to classify the data. The data set will be tested once with the assigned weights and once without, to demonstrate that the results of the first test are considerably better.

There are many classification algorithms, each of which possesses certain trademarks that set it apart from other classifiers. While each of these attributes serves a certain purpose, not all of them are useful for our purposes. As advantageous as each of these may be, not all of them take into account the importance of certain features over others. In this thesis, the term “features” is used to refer to the different attributes (age, gender, location, relationship status, and occupation) of anonymous users.

Following are a list of contributions made in this thesis:

- 1- Using a method based on Support Vector Machine to predict travelling behaviour of Facebook users, this will be the first time that this method has been used to predict the behaviour of Facebook users.
- 2- Comparison of Support Vector Machines and Feature Weighted Support Vector Machines
- 3- Comparison between having experts fill out the pairwise comparison table to having Support Vector Machine fill out the table.

1.5. Summary

This research consists of three parts:

1-Collecting data from Facebook API

2-Training the classifier(s) (SVM)

3-Prediction

Each of these parts will be covered in detail in the different sections of this thesis. Chapter 2 provides background information about the methods used in the thesis. Chapter 3 explains the methodology of this research. Chapter 4 then shows the implementation of the proposed method.

In conclusion, following an extensive literature review on studies implemented in this area, this study enables the display of the benefits, novelty and advantages of its proposed method, thereby showing the importance of analysing data gained through social networks with pairwise comparison and weighted support vector machines.

CHAPTER 2

LITERATURE REVIEW

The first section of this chapter will begin by a literature review of Machine learning and learning algorithms, it will then move onto unsupervised learning, supervised learning, Support Vector Machines and Pairwise comparison.

2.1. Machine Learning and Learning Algorithms

With the growth of the Internet and Information Technology, many new areas have opened in the research of Artificial Intelligence, the goal being to simulate human thought. The ability to learn is what makes the human mind intelligent and therefore superior to machines. Machine Learning is the method used to make machines intelligent. It is a science that studies how to simulate the human brain and learning abilities, in order to develop new knowledge and skills as well as continuously improve on previous achievements. As the amount of data continues to increase by the second, the absence of sufficient human resources to analyze them and make a decision means that they have to be stored, and the decision making postponed to another date. This has necessitated data mining and the advantages that it brings. Un-analyzed data is worthless, while data that has been analyzed might be worth billions. Knowledge is and has always been a valuable resource, but it is the manner in which it is used that makes the difference. Machine learning looks for patterns in data, and learns from these patterns.

The terms Machine Learning (ML) and Data Mining are often synonymously used. Machine Learning deals with the study, design, and development of the algorithms that give computers the ability to learn, while Data Mining may be defined as a process that starts from seemingly unstructured data and proceeds to extract knowledge and patterns from that data. During this process, Machine Learning algorithms are used.

Learning algorithms are divided into two categories: supervised and un-supervised. Unsupervised learning algorithms do not need training sets. The most popular method in this area is clustering. However, in this thesis we focus on supervised learning algorithms, the most notable [7] among them being Neural Networks and Support Vector Machines.

2.1.1. Unsupervised Learning

A major part of the information that is absorbed by our brains on a daily basis comes without any forms of instruction, or any obvious relationships to what we have absorbed or learnt before.

According to Dayan [6] “ Unsupervised learning studies how systems can learn to represent particular input patterns in a way that reflects the statistical structure of the overall collection of input patterns. By contrast with SUPERVISED LEARNING or REINFORCEMENT LEARNING, there are no explicit target outputs or environmental evaluations associated with each input; rather the unsupervised learner brings to bear prior biases as to what aspects of the structure of the input should be captured in the output.”

An example would be looking at something and identifying it. A person looks at objects and knows from prior knowledge what those objects are; he/she recognizes them from such prior knowledge.

Two methods are used for unsupervised learning: density estimation techniques and feature extraction techniques. The first technique is based on building statistical models (such as Bayesian Networks), while the second technique is based on an effort to extract statistical regularities or irregularities directly from inputs.

2.1.2. Supervised Learning

Kotsiantis, explains supervised learning as: “the search for algorithms that reason from externally supplied instances to produce general hypotheses, which then make predictions about future instances”[7].

The human mind is mistake prone when it comes to decision making, analysis or trying to establish relationships between entities or features, thus making solution finding more difficult. This is when Machine Learning can be applied, which assists the decision making process and adds efficiency and accuracy. There are several machine learning applications involving tasks that can be set up as supervised.

Machine learning has many applications, one of them being data mining. In any dataset, there are many instances, but in machine learning, all instances in a data set are represented by the same set of features. These features could be anything; they could be binary, continuous or categorical.

If the instances are given known labels (the correct outputs) then the learning is considered supervised, as opposed to unsupervised learning, where the instances are unlabeled.

Table 2.1: Example of data (instances with known labels) in a standard format

Instance (users)	Feature 1 (age)	Feature 2 (gender)	...	Feature n (location)	Class (Did they travel in 2012?)
1	User 1 age	User 1 gender		User 1 location	Yes
2	User 2 age	User 2 gender		User 2 location	No
3	User 3 age	User 3 gender		User 3 location	Yes
...	User n age	User n gender		User n location	...

Supervised machine learning is about algorithms that process instances from a training set and derive a set of rules from them. Therefore, they actually create a classifier that generalizes new instances, thus making predictions about instances and training sets in which their labels are unknown. Figure 2.1 shows how supervised learning is applied to real world problems.

For the Purpose of this research, the “Instance” column will be filled by the Facebook users, and the “Features” being used are, in order: Age, Location, Gender, Occupation, Relationship status, and Average Travel Time per year. The last column in Table 2.1 is “Class”, which in SVM is also called the “label”, which in the case of this research it will be whether or not the specific Facebook user went on vacation in the year 2012.

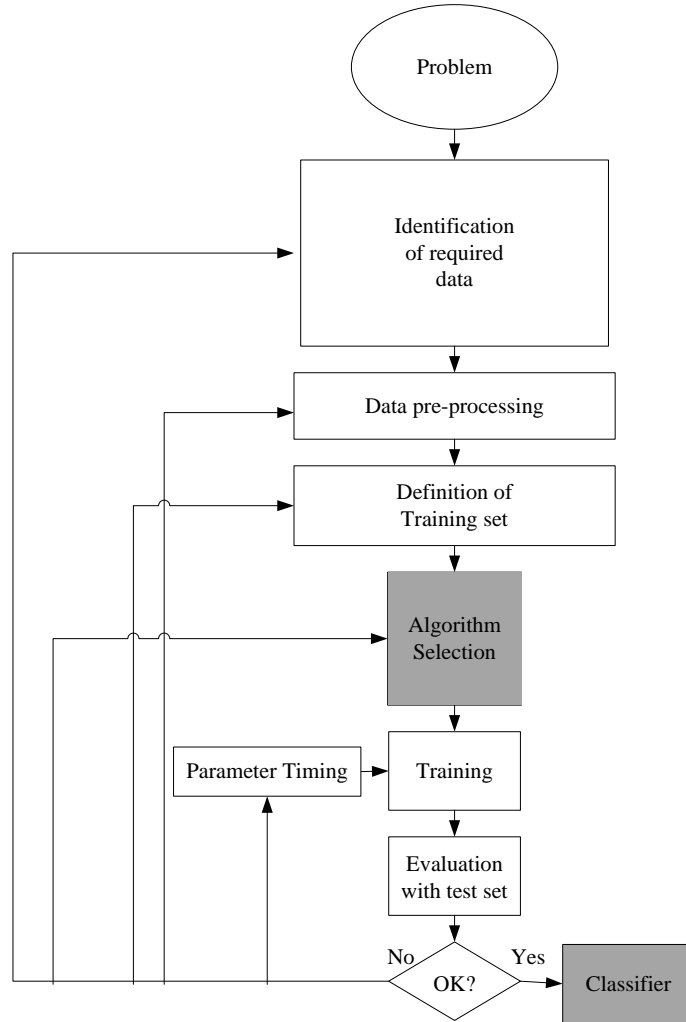


Figure 2.1: The supervised machine learning process of Kotsiantis *et al.* (2007)[7]

As is shown in figure 2.1, the process starts with data collection. There are two ways to do this: either by having an expert choose the most informative ones or through a brute-force method. After completing this step, it is time to choose which features are relevant and informative. This is called data pre-processing and preparation. This step is necessary because although there is a large amount of information available, a lot of it is low-quality data. In order to profit from the data, this information needs to go through a cleaning process. Zhang *et al.* [8] argue that the importance of data preparation is based on three facts: “(1) real-world data is impure; (2) high-performance mining systems require quality data; and (3) quality data yields high-quality patterns.”

1. Real-world data may be incomplete, noisy, and inconsistent, which can disguise useful patterns. This is due to:

- Incomplete data: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data.
- Noisy data: containing errors or outliers.
- Inconsistent data: containing discrepancies in codes or names.

2. Data preparation generates a dataset smaller than the original one, which can significantly improve the efficiency of data mining. This task includes:

- Selecting relevant data: attribute selection (filtering and wrapper methods), removing anomalies, or eliminating duplicate records.
- Reducing data: sampling or instance selection.

3. Data preparation generates quality data, which leads to quality patterns.

For example, we can:

- Recover incomplete data: filling the values missed, or reducing ambiguity.
- Purify data: correcting errors, or removing outliers (unusual or exceptional values).
- Resolve data conflicts: using domain knowledge or expert decision to settle discrepancy.

From the above presentation, it can be seen that even the first steps play a crucial role in supervised learning. Many studies have been undertaken on data preparation; this is not part of the focus of this thesis.

It must also be borne in mind that data preparation methods differ from one learning algorithm to the other. That is why choosing a specific learning algorithm is a critical step in this procedure. This thesis mainly focuses on the learning algorithm of Support Vector Machines introduced in 1995 by Vapnik [9]. It is a well-known and popular supervised learning algorithm, along with its by-product called “feature weight support vector machines”, which will be discussed next.

2.2. Support Vector Machines

Support Vector Machines (SVM's) are the newest supervised machine learning technique [9]. The main idea behind SVM's is that there is a hyperplane that separates two data classes.

Maximizing the margin and creating the largest possible distance between the hyperplane and the instances improves accuracy and reduces error. In contrast to other classical statistical methods where they decrease the feature dimensionality in order to control the outcome, SVM actually relies on the large margin factor that comes with an increased feature dimensionality [23].

When considering linearly separable data, once the optimum separating hyperplane is found, the data points that lie on the margin are known as the support vectors or support vector points and the solution is represented by the linear combination of only these points. Figure 2.2 [7] illustrates data belonging to two different classes using circles and squares. The classes are separated via an optimal hyperplane, and the data points that lie on its margin are known as the support vector points.

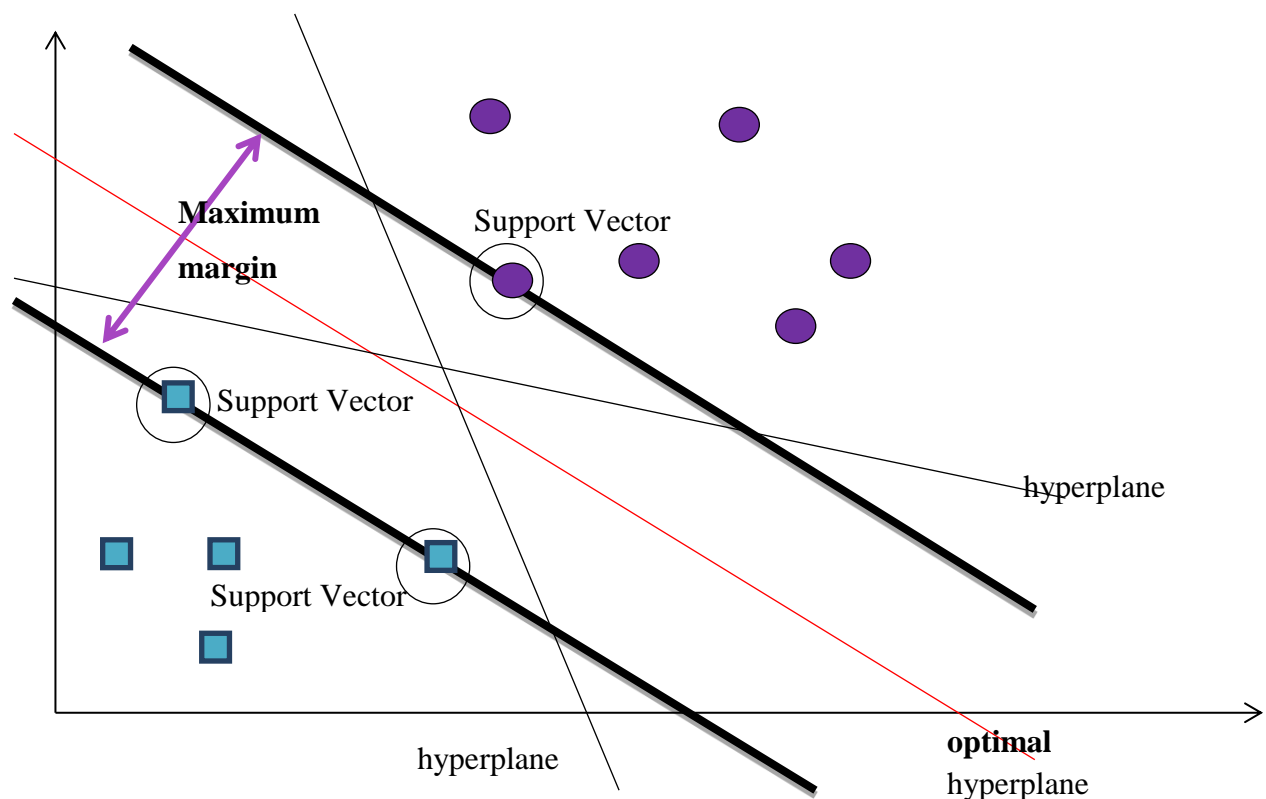


Figure 2.2: Linearly Separable data [7]

As can be seen in the image, all other data points are ignored, and the number of features in the training data does not affect the complexity of the Support Vector Machine. The number of support vectors made by the SVM learning algorithms is usually quite small. This is also one of

the reasons why SVM's are the preferred algorithm when dealing with learning tasks in which the number of features is much larger in comparison with the training instances.

The original SVM algorithm was developed by Vapnik, and the current soft margin was subsequently proposed by Corina Cortez and Vapnik in 1993 and published in 1995[8]. In the case of very high dimensional problems, only occasionally are the data linearly separable. In practice, not all training sets are linearly separable, and even if they are, it is often preferable to find a solution that better separates the bulk of the data while ignoring the few unrelated noises. This is the concept of the soft margin. In those cases where no hyperplane can fully separate the test set, the soft margin method creates a hyperplane that separates the data in the best possible manner, while allowing some data points belonging to one class to be classified as a different class. See Cristianini [26] for a precise formulation of soft margin approaches.

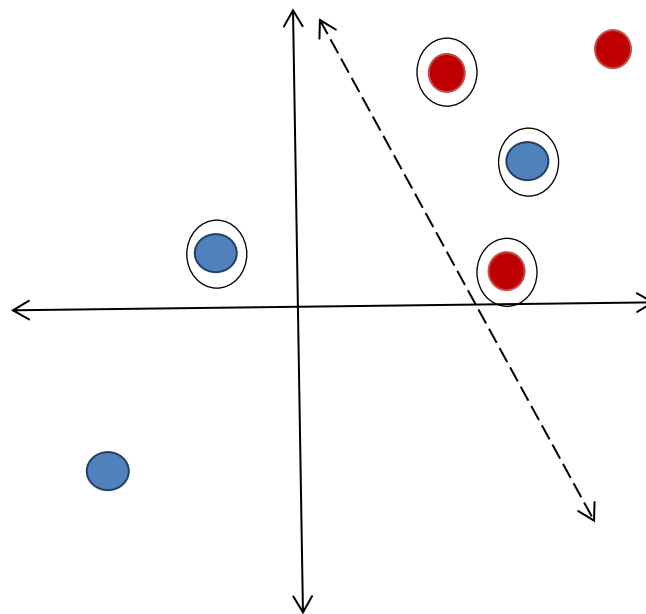


Figure 2.3: When no separating hyperplane exists, SVM uses a soft margin hyperplane

Nevertheless, while the soft margin does add a bonus value to SVM, it is not the singular most important feature in this method. When data is not linearly separable, SVM resolves this issue by first mapping the input data space to a higher dimensional space and then defining a separating hyperplane there.

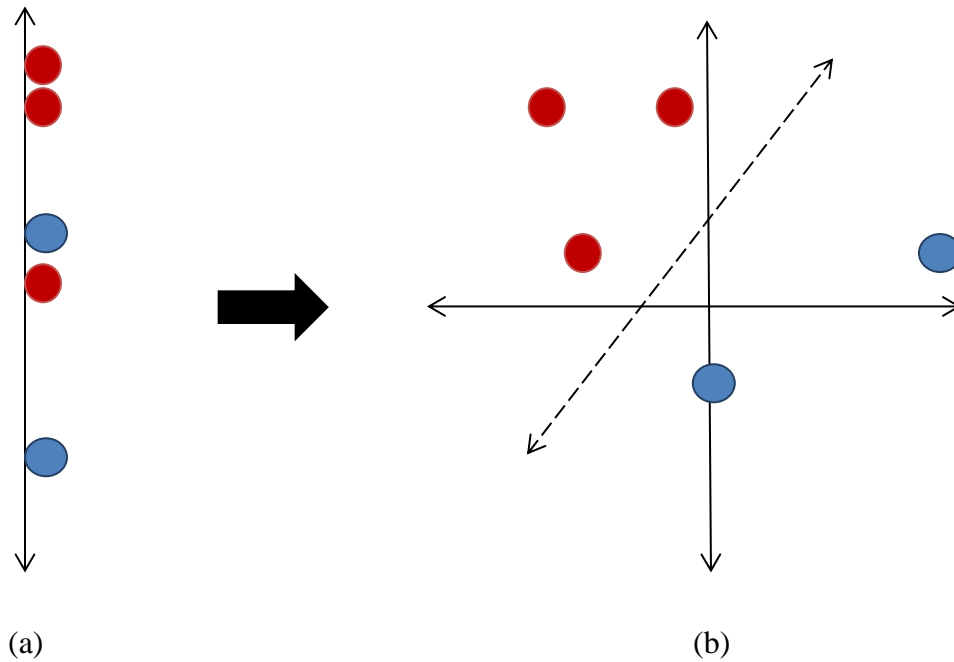


Figure 2.4: SVM mapping data from input space (a) to a higher dimensional feature space (b)

The mapping is achieved by SVM's use of kernel functions. Figure 2.4 is a simple portrayal of how SVM maps data from an input space to a higher dimensional feature space. As it can be seen separating the data into two groups in the linear space would have been more complex, while when mapped to a higher dimension it gained simplicity. According to Weston [27] SVM maps every point of the data input space into a higher dimensional feature space through such a transformation as $\Phi: x \mapsto \Phi(x)$. With an appropriately transformed feature space of sufficient dimensionality, any training set can be suitably separated. Figure 2.5[27] portrays the complexity of lower and the simplicity of higher dimension separation through (a) a non-linearly separable data in a 2D input space and (b) SVM mapping the data into a 3D feature space where data is separable via a hyperplane.

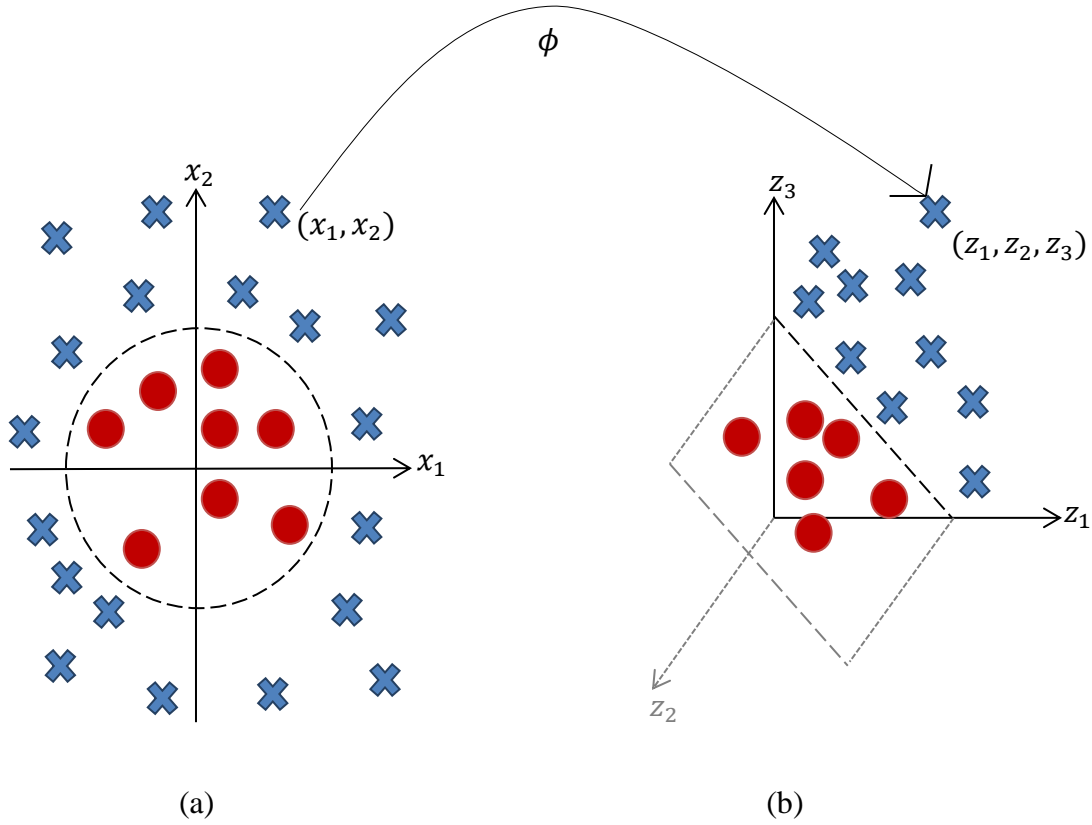


Figure 2.5: SVM feature mapping the (2D) input data space to some other (3D) dimensional space; $\Phi: R^2 \rightarrow R^3, (x_1, x_2) \rightarrow (z_1, z_2, z_3)$ [27]

In this section, we undertake a brief review of some basic work on SVM's for classification problems. For further details, the reader is referred to (Vapnik, 1995; Vapnik, 1998).

Given a training set of N data points $\{y_k, x_k\}$ (from N to k) $= 1, \dots$. In this instance, we start with a two-class training set $T = \{(x_1, y_1), \dots, (x_k, y_k)\}$, in which $x_i \in R^n$ (in which R represents real numbers), $y_i \in \{-1, +1\}$, and each $x_i = (x_{i1}, \dots, x_{in}) \in R^n$ is a vector which has the values of n different *features*. The goal is to find the maximum-margin hyperplane, whether linear or non-linear, that divides the points that have $y_i = -1$, from those having $y_i = +1$.

As stated previously the choice of a specific learning algorithm over other learning algorithms is a critical one. One of the methods used to evaluate a classifier is prediction accuracy, which can be implemented in different ways. One is to split the training set, use two-thirds for training purposes, and the remaining one-third for testing. Another method is cross validation in which the training set is split into equally sized subsets and for each of those cases the classifier is trained on the union of all the other subsets. This second method is more computationally expensive.

Now if the error rate of a classifier is unsatisfactory, we will have to go back a step and reconsider a few aspects (as seen in figure 2.1). There is a variety of factors to use in such re-examination, such as reassessing the choice of the learning algorithm, the relevancy of the features selected, the size of the training set, etc.

In general terms, in support vector machines, all features in a certain training set have the same level of importance. While this might work for some problems, this is not the case for real-life problems. In a real-life problem, not all features in a data set carry the same degree of importance; some features are more relevant and some less relevant to the problem being solved or the research being done, thus also less relevant to our classification goals. For the purpose of this research, data was gathered from Facebook users, now not all of the data that users supplied on their profiles was relevant to this research. For example among the information that Facebook users provide on their profiles are music, books, TV shows, sports, apps, and games most of which would have no bearing on whether or not that particular user is going to travel the following year. Among the information users supplied on their profile are features that are considered more relevant for the purposes of this research, such as age, gender, location, average travel time per year, occupation and relationship ship status. However not all of these features hold the same amount of importance to predict the outcome of the research being followed, the age of a person might play a more significant role on a person traveling habits than the persons occupation. This issue can be efficiently solved through Weighted Feature Support Vector Machines (WFSVM), which assigns weights to features. In this thesis, we will show how the application of weights to features permits a more accurate and faster SVM and therefore more accurate and faster predictions. This is shown by first applying (unweighted) SVM to our data set and, at the second stage, applying WFSVM to our data set, in which weights will be assigned to relevant features. Thereafter, by comparing the error rate and accuracy of each method

conclusions may be drawn about the effectiveness of the methods for this particular problem, thus validating our choice. For further studies on WFSVM, the reader is referred to Suykens *et al.* [10], Amutha *et al.* [11] and Sun *et al.* [12].

If $W = (w_1, \dots, w_n)$ is the feature weight vector, then w_i is the indicator of the relative importance of the feature i , ($i = 1, \dots, n$), for the classification. In this instance the fundamental difference between SVM and WFSVM is that in the latter, we use the afore-mentioned vector:

$$\mathbf{y}_i = (w_1 x_{i1}, \dots, w_n x_{in}) = \text{diag}(W) \cdot X_i$$

This is used instead of the X_i vector alone.

In actuality, the assigned weights are determined by the principle of maximizing deviations between categories. Several methods may be used in calculating the weights, the most well-known of which are: Shannon's Information Theory Xing *et al.* [46], Degrees of Standard Deviation Wang *et al.* [25], Values of deviations Sun *et al.* [24] and Pairwise Comparison.

2.2.1. Why Weights Improve Accuracy

The classical SVM only weights instances, not the features. While properly and sufficiently normalizing the features is a significant aspect of an accurate classification, the main reason behind the concept of adding weights is to take into account the level of importance of each feature. In solving classification problems with SVM, if the sample data contain features that are somewhat or even completely irrelevant to the main problem, this will affect the classification results. This is why in this thesis we are proposing to add weights. Sun *et al.* [12] describe other SVM-based methods that have previously been proposed to improve classification results, methods such as Fuzzy Support Vector Machine (FSVM) and the Weighted Support Vector Machine (WSVM). Although the importance of weight is acknowledged in these methods, the significance of the weightiness of features on classification is neglected. Sun *et al.* [12] acknowledge this issue and address it.

As mentioned before, there are many ways to calculate weights, the most popular among which include:

1. Values of deviation
2. Degree of standard deviation

3. Shannon's Information Theory

4. Pairwise Comparison

Each of these methods will be explained in turn.

2.2.2. Values of Deviation

In mathematics and statistics, deviation is considered the measure of difference between the value of a variable and another value. The sign of this new value (whether negative or positive), shows the importance of the values. In this method, the weight of a vector is estimated through the value of deviation between two variables.

In the paper by Sun *et al* [12], the shortcomings of known weighted SVMS were acknowledged. It was the fact that SVM took the importance of the sample into account but failed to take into account the relative importance of each feature in respect to the entire classification results. In the paper by Sun *et al*. [12], the weighted feature classification was used to add weights, thereby improving the accuracy of classification. In the first step, the deviation between two random variables was estimated and then the weights of features were determined based on the concept of maximizing the deviations between the different categories. Experimental results proved that their proposed method improved accuracy and decreased support vectors (number of support vectors). In Table 2.2, the results of Sun *et al* [12] are shown.

Table 2.2: The experimental results with SVM and FWSVM using values of deviation [12]

Table of data	SVM	FWSVM
---------------	-----	-------

	Accuracy	Number of Support Vectors	Accuracy	Number of Support Vectors
1	82.36	198	88.66	172
2	76.18	206	84.82	198
3	79.36	222	82.16	202
4	76.67	221	80.00	196
5	66.67	298	72.18	282
Mean of results	76.25	229	81.56	210

2.2.3. Degrees of Standard Deviation

In reality, standard deviation or variance is principally the measure of how close a value is to the average. If all values are close, the standard deviation will be small and the average can be considered an accurate estimation of the distribution.

Wang *et al.* [13] proposed to use WFSVM for the classification of semantic images in order to overcome the fact that the regular SVM algorithms on image classification did not distinguish between different features, but assigned the same weight to all low-level features. However, it was obvious that in high dimensional image data, some features were either more or less relative to the classification process. In semantic image classification, the more concentrated a feature is in a cluster, the greater the dominance of the feature in that cluster and more relevance in the outcome. Conversely, the more discrete a feature, the less dominant it is and therefore irrelevant to the outcome of classification.

Standard deviation can estimate the degree of discreteness. The denser a feature's distribution is in regards to the cluster, the smaller the standard deviation for that feature. It was through this method that Wang *et al.* [12] assigned the weights to each feature. In Table 2.3, the experimental results that Wang *et al.* [12] received using WFSVM and standard deviation are shown.

Table 2.3: The average precision and training time comparison of image classification [24]

	Average Precision (%)	Training Time (s)
SVM	77.00	~3
WFSVM	93.86	~0.6

2.2.4. Shannon’s Information Theory

According to Jaynes (1957)[47], “Information theory provides a constructive criterion for setting up probability distributions on the basis of partial knowledge, and leads to a type of statistical inference which is called the maximum-entropy estimate. It is the least biased estimate possible on the given information.” [27]

Information theory, as developed by C. E. Shannon[28], is a branch of mathematics and computer science, which involves the quantification of information. It was developed to work with compressing communication data. Since then, the horizon of its application has been broadened to many other aspects of science.

In order to make the classic SVM take into account and use the importance of certain samples, weighted samples, Xing et al [46] proposed the Linear Feature weighted Support Vector Machine. In it, they had two stages for constructing their proposed model. In the first stage, they determine the weights for each feature in the data set using Shannon’s Information Theory. These weights then displayed the contribution each feature had in the construction of the optimal hyperplane. In principle, the weight of a certain feature was assigned based on the amount of classification information that such a feature possessed. In the second stage, the retrieved feature weighted vector was applied to the original SVM using “dual quadratic programming formulae through detailed theoretical deduction.”[26] The experimental results gained through their study showed an improvement in performance in comparison to the traditional SVM.

2.2.5. Scaling

Generally, in most data sets the range of values in the raw data sets varies. Most machine learning algorithms cannot function objectively in such cases. For example, in a data set that has many features to factor from, if one of the features has a wide range of values ranging from 10 to

90, and another feature has the limited range of either 1 or 2, the outcome of the classification will be governed by the feature with the broader range.

In Hastie *et al* [37], it is explained how scaling improves the outcome in neural networks. The concept can easily be applied to Support Vector Machines as well.

In this part of the research, readers will come across various terms, such as rescaling, normalizing or standardizing, each of which has its own implementation and application.

According to Sarle's Neural Networks [38], "Rescaling" a vector means to add or subtract a constant and then multiply or divide by a constant, as would be done to change the units of measurement of the data, for example, to convert a temperature from Celsius to Fahrenheit.

"Normalizing" a vector most often means dividing by a norm of the vector, for example, to make the Euclidean length of the vector equal to one. In Neural Network literature, "normalizing" also often refers to rescaling by the minimum and range of the vector, to make all the elements lie between 0 and 1.

"Standardizing" a vector most often means subtracting a measure of location and dividing by a measure of scale. For example, if the vector contains random values with a Gaussian distribution, you might subtract the mean and divide by the standard deviation, thereby obtaining a "standard normal" random variable with mean 0 and standard deviation 1.

However, all of the above terms are used more or less interchangeably depending on the custom within various fields. Now the question is, should any of these things be done to a dataset? The answer is, it depends. In Elements of statistical learning [37], the scaling of the input data, determines the effectiveness of the weights. Therefore it can have a large effect on the final quality of the classification. By scaling, it ensures that all inputs are treated equally in the training and classification process. Figure 2.6, clearly shows the benefits of scaling the input data.

Figure 2.6 consists of two figures; the one on top is the data set without any scaling, standardizing or added weights, the figure in the bottom is the same data set only the features have been weighted. As it can clearly be seen, the classification results of the figure on top are more complicated and not as clear as the classification results achieved by figure in the bottom.

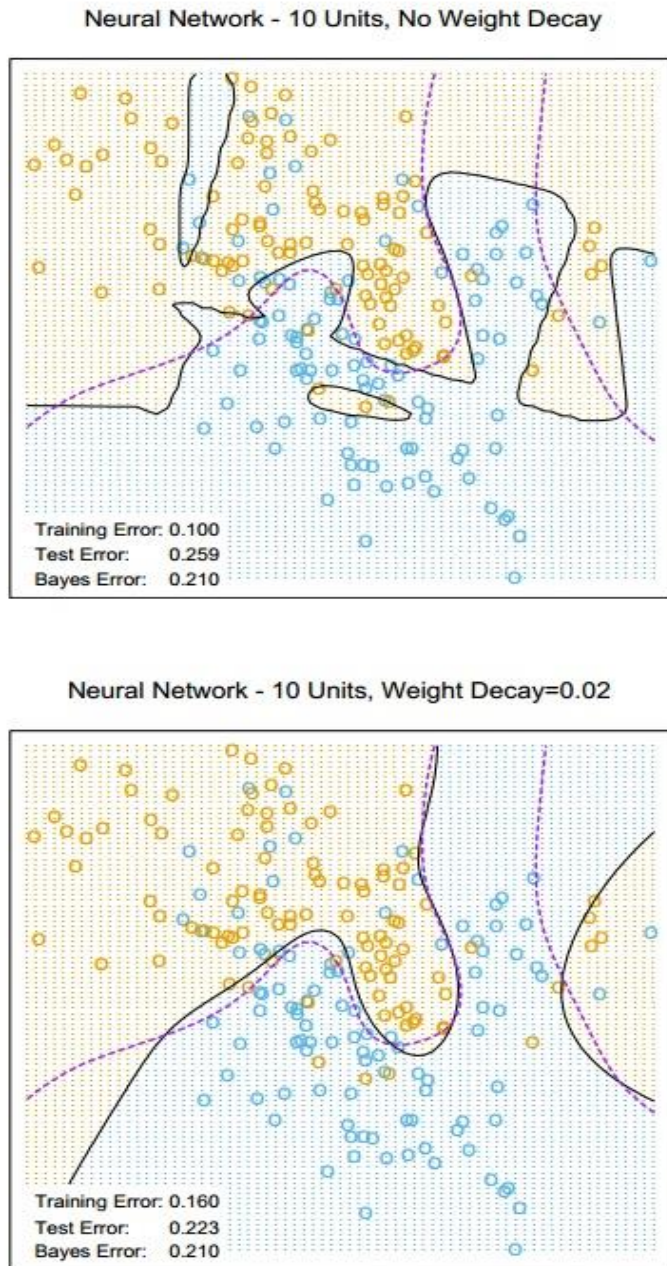


Figure 2.6: Different classification results, based on weights and no weights from Elements of statistical learning [37]

There is a common misconception that the inputs to a multilayer perceptron, an artificial intelligence model widely used for artificial neural networks to map out sets of input data into appropriate sets of output data, must be in the interval $[0,1]$. There is in fact no such requirement, although there often are benefits to standardizing the inputs as discussed below. However it is

better to have the input values centered around zero, so scaling the inputs to the interval $[0, 1]$ is usually a bad choice. If one's output activation function has a range of $[0, 1]$, then obviously one must ensure that the target values lie within that range.

2.3. Pairwise Comparison

Pairwise Comparison is a method that has been widely used to recreate human subjectivity. It is a new approach for deriving priorities when comparison judgments are needed. What it does is that it divides the judgment into a series of interval comparisons. In this case, the features are being judged and evaluated. The assessment of the priorities derived in this fashion is formulated as an optimization problem, which in fact maximizes the decision makers with a specific crisp priority vector [14].

For features $E_1 \dots E_n$, in Pairwise comparison it is assumed that the decision maker can compare between any two elements such as E_i and E_j , and then give a numerical value such as a_{ij} as a ratio for their importance. In this scenario if E_i is more important than E_j , then $a_{ij} > 1$ and consequently $a_{ji} = 1/a_{ij}$. For a sample with n features, $n(n-1)/2$ judgments are required.

The following is the relation between the features:

A: $(E_i, E_j) \in X \times X \rightarrow a_{ij} = A(E_i, E_j) \in G \subseteq R$ is represented by the matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

This is the *Pairwise Comparison Matrix*: a_{ij} expressing how much E_i is more important than E_j , $a_{ii} = 1$ for all i 's, a_{ij} is basically the measure of the relationship between two features, the bigger a_{ij} is the bigger the difference between them, if $a_{ij} = 1$ then the two features are indifferent and they have the same values. Moreover, there is also another relationship between the entries in the pairwise comparison matrix of $A = [a_{ij}]_{n \times n}$, for i, j , and $k = 1, \dots, n$: $a_{ik} = a_{ij} \cdot a_{jk}$.

Pairwise Comparison Matrices play a very important role in the analytical hierarchy process, a procedure that Saaty created in the 80's [15]. It is even widely used by companies and governments to adjust their strategies [16].

2.4. Graph API

The Graph API is the primary method to obtain data from and post data on Facebook's social graph. It is also the method used to collect data for the purpose of this thesis. The data regarding a thousand users age, gender, location, relationship status, occupation, and average travel time per year was accessed using this API. The Graph API is basically a low level http based API that can be used to query public data[48][49], upload images, post new information and a variety of other similar tasks that an app might need to do in Facebook.

API's are composed of nodes, edges and fields. Nodes consist of users, a photo, a page or a comment. Edges consist of the comments on a photo or the photos on a page. Finally yet importantly, the field option consists of the birthday of a user or the name of a page.

In general, information can be read from API's by making simple HTTP GET requests to the relevant end point. For example, if it is needed to retrieve all the photos by someone, it could be done by making an HTTP GET request as follows:

```
GET graph.facebook.com
/{node-id}/photos
```

Publishing to API's is made possible by making a HTTP POST request to the node in question:

```
POST graph.facebook.com
/{node-id}
```

Deleting is also accomplished through the same method by using the HTTP DELETE request.

One of the advantages of working with the graph API is that it allows the customization of requests. For example If it needed to to find the places a user has been tagged in or were created by that user, such parameters as `center` and `distance` or a `placeID` can be added to specify where the objects should be near to:

```
GET graph.facebook.com
/search?
type=location&
center=37.76,-122.427&
distance=1000
```

CHAPTER 3

Methodology

In this chapter, an approach is presented for implementing a system that applies supervised learning algorithms to information derived from Facebook to enhance recommender systems. There are three steps to this application: the first is gathering the data from Facebook, the second is training the classifiers, and the third is to use what we have gathered to predict the future outcome. The system takes advantage of public[48][49] and non-identifiable information that users put in their social networking websites (in this case Facebook) and uses it to make predictions about future actions of that specific user. This prediction could later on be used to enhance recommender systems.

3.1. Data Collection

There are many methods to elicit personal data from people, including online polls and surveys, in person surveys, observation, interviews, questionnaires, or simply gathering information from people's social networking websites. However, many of these methods have their own limitations. For example, when put on the spot, people find it hard to describe the knowledge they regularly use, or not have the presence of mind to answer certain questions fully. Their answers might be biased as they might not feel comfortable telling a surveyor what they need to know, or they just do not want to share certain information on a random questionnaire. Therefore none of the above methods were used. As explained in Chapter 2, certain challenges were faced during the data collection phase. Attaining personal details for 1000 Facebook users was a challenging task. Furthermore, all the information used in this research, was data that had passed through the Facebook privacy filter, meaning it was information that was publicly available [48][49]. In addition, to avoid confusion, it is emphasised that the dataset is comprised of anonymous data, and it does not generate user identifiable information. Also due nature of the information being used in this research, which was publicly and legally accessible [48][49], no human participation was needed, and no one else other than the researcher was involved this process.

In this section, in order to gather the information that is on users' Facebook pages, the Graph API was taken advantage of. This is more of a direct method, since the information is taken from

people's social networking sites. It is done through a simple HTTP GET request. A lot of information can be gathered using this process, but not all of it is useful or relevant, and by default not all the fields in a node or edge are returned when you make a query. One can choose the fields (or edges) they want returned with the "fields" query parameter. This is useful for making one's API calls more efficient and fast.

For example, the following Graph API call will only return the birthday in bgolub's profile:

```
GET graph.facebook.com
  /bgolub?
    fields=birthday
```

Our model should get the information about our user's Name, Age, Location (by continent), Sex, Occupation, Relationship Status, and Average Travel Time per Year.

To access certain data, the developer must, before executing the GET request, initially Get Access Token and acquire certain permission in order to be able to access user information. These select permissions fall into two categories: user data permissions and extended permissions. They, the afore mentioned permissions, consist of information regarding users' friends, birthdays, location and other similar attributes. They can even be used to publish content to Facebook on behalf of a person who has granted an open graph publishing permission application to the publisher. One can determine which permissions are needed in the access token with the help of the Graph API reference.

In Table 3.1 [8] a list of some roots nodes (or Endpoints) of the Graph API, along with their prospective representations is produced. A root node is an endpoint, which can be used to access certain information. For example by having the root node /friendlist as part of the GET request, the friend's lists published by user will be displayed.

Table 3.1: Graph API root nodes and their representations [28]

Endpoint	Represents
/album	A photo album
/app	A Facebook app
/comment	A comment published by another node.
/event	An event
/friendlist	A grouping of friends created by someone on Facebook.
/group	A Facebook group
/link	A link shared on Facebook
/object/likes	A set of likes on a particular object.
/place-tag	An instance of a person being tagged at a place in a photo, video, post, status or link

3.1.1. Challenges faced with data collection

As with most social networking sites, finding users with public info was difficult. Users place different privacy settings for different aspects on their profile. One of the challenges that arose when gathering information for this research was finding users with publicly accessible profiles. It should be noted and emphasised that the Facebook Graph API has only access to public information [48][49].

Another challenge was incomplete information. Not everyone fills out all the aspects of their profiles. For example some leave out their age or gender, or don't update their relationship status. Therefore it was a very time consuming task for the researcher to obtain that information, in some cases it had to be done manually.

The last issue in the data collection phase was the great amount of data needed for this research, because it was decided that the method should be implemented on a thousand users.

3.2. The Data Set

In order to validate the methods and finding in this research, the system was initially tested on a smaller data set. After the initial testing stages accomplished in Behzadfar and Abhari [40] were achieved, the proposed method was implemented on a dataset of 1000 Facebook users. The features extracted from the Facebook API were, age, gender, location, occupation, average travel time per year, relationship status and whether or not did each specific user travel in the year 2012. Table 3.2 shows is an example of how the data set will look like. However this is the dataset with only 5 users, the complete version has 1000 users. As it can be seen the Facebook users are defined by User 1 to User N, because no identifiable traits were used or saved (or even needed) during the course of this research. It must also be added that the actual dataset has one difference to this as the data had been scaled (as explained in section 2.3.5), so all the number would not be any higher than 10. In order for it to be understandable to the reader, the Table 3.2 has been made to look like real values, so as to not confuse the reader.

Table 3.2: Sample of training dataset for 5 users

Facebook Users	Age	Location	Gender	Occupation	Relationship Status	Income Level	Average Travel/year	Traveled In 2012
User 1	21	1	2	4	1	1	5	0
User 2	82	4	1	6	1	7	6	1
User 3	24	1	1	7	1	3	9	0
User n	23	2	2	10	1	7	3	0

3.3. Training the Classifiers

After the data has been gathered, the algorithm must be trained using that data. With the growth of the internet and IT, the need for AI has also arisen. Gathering the data was one part of the work: the other part is making use out of all this data. One of the many applications of AI is simulating the human thought. In 1959, Arthur Samuel defined ML as a “Field of study that

gives computers the ability to learn without being explicitly programmed”[29]. The aim of Machine Learning is to obtain new knowledge. Machine Learning has two major types of algorithms: Supervised and Unsupervised. While supervised learning, such as “Classification”, focuses on prediction based on already known aspects of data, unsupervised learning such as “Clustering” focuses on the discovery of the unknown aspects of data. As mentioned earlier the human mind cannot be the best judge in every case and instance. ML scholars want the machine not only to simulate the human mind, but also to go over and beyond human capacity. Since the labels, whether the user has gone on vacation or not, are available from the beginning, in this research the focus is on using the methods of classification as opposed to clustering. [40]

The training of the data is the starting point in filling a gap in the literature. This approach, along with its clarifications, sets this project apart from other work undertaken in this field. In machine learning, classification means identifying to which subsets a group on instances belongs [40]. There are many methods of classification but not all of these methods have the ability to learn. Among those that can be trained are Decision Trees, Support Vector Machines, and Neural Networks. The difficulty is to find a suitable method to realize the goals set for this research.

In “Supervised Machine Learning: a review of classification Techniques” [7], the author goes into detail about the various supervised learning classification methods. He does not specify which method is the best, but carefully illustrates the different lines of research done in this area, giving the reader a comprehensive review of various learning algorithms, thereby letting readers decide which method is the best for their application. Table 3.2 displays the comparison between different types of learning algorithms. Each of these methods has certain attributes about it, and those specific attributes are what sets that particular method apart from others. In order to pick a suitable method for classification, the reader must find the one that fits their purpose. For further reading on this matter, please refer to [35][36].

Table 3.3: Comparison of learning algorithms [7]

	Decision Trees	Neural Networks	Naïve Bayes	kNN	SVM	Rule-learners
Accuracy in general	**	***	*	**	****	**
Speed of learning with respect to no. of attributes & instances	***	*	*****	*****	*	**
Speed of classification	****	*****	*****	*	*****	*****
Tolerance to missing values	***	*	*****	*	**	**
Tolerance to irrelevant attributes	***	*	**	**	*****	**
Tolerance to redundant attributes	**	**	*	**	***	**
Tolerance to highly interdependent attributes	**	***	*	*	***	**
Dealing with discrete/binary/continuous attributes	*****	***	***	***	**	***
Tolerance to noise	**	**	***	*	**	*
Dealing with the danger of overfitting	**	*	***	***	**	**
Attempts for incremental learning	**	***	*****	*****	**	*
Transparency of knowledge	*****	*	*****	**	*	*****

The table shows, many similarities between Neural Networks and Support Vector Machines. However, in this research Support Vector Machines were chosen due to certain qualities it possesses in comparison to Neural Networks. Due to the nature of the data set being used and its source (social networking websites), a number of important factors determined the choice of Support Vector Machines over Neural Networks. These include the classifier's tolerance to redundant, irrelevant, and missing values, and the fact that the training optimization problem of the SVM necessarily reaches a global minimum and avoids ending in a local minimum [45], which may happen in NN's.

After conducting several analyses on the data being used in this research, it was realized that not all features had the same influence on the outcome of the classification. This was done by analysing the results based on training the algorithms with only one feature. Therefore, it can be concluded that certain features were more important in the outcome of the research than were others. In some instances, when different features were tested together, the outcome was better than when the feature was being tested separately [40]. Therefore, it was concluded that using the Feature Weighted Support Vector Machine (FWSVM) was a suitable choice for this data set and the objectives of this research. Thus, although SVM had rather desirable outputs, the breakdown point was further improved by adding weights (WFSVM)

3.3.1. Pairwise Comparison and Weight Calculation

This section is devoted to explaining, how pairwise comparison and weight calculation work for our collected data. Assuming that we have a dataset with F features, N samples and C classes, and the goal is to create a classifier based on these assumptions, the first step is to assign weights to the F features. To do this, we need an expert to rank the importance of each feature and fill the pairwise comparison table $F \times F$.

In this part of the research, another method to fill out the pairwise comparison table is proposed (as opposed to experts filling out the table): to let SVM take the place of the experts and help create the pairwise comparison table. Afterwards, by comparing the results of each of the features against the results of the whole the table can be fully filled out.

To investigate fully the advantage of these two methods, in this thesis the weights are calculated using both methods and then the results are compared to see which one results in a better classifier.

In either method, it is our responsibility to ensure that the pairwise comparison table is consistent. This is done by analyzing all of the a_{ij} , a_{jk} and a_{ik} triads from $A = [a_{ij}]_{F \times F}$. The triads must be modified until the table is fully consistent. This part can be repeated as many times as needed. Figure 3.1 illustrates the process in a manner that is very easy to follow.

In some cases, adding weights not only does not increase the performance of the classifiers, but also decreases it as well. Since an average user is not in the position to judge such a fact, in this

thesis two different approaches to the same weight calculation method (pairwise comparison table):

1-weight calculation reached with experts filling out the pairwise comparison table and

2-weight calculation reached with filling out the pairwise comparison table with the assistance of support vector machine itself

are observed and then the performance of the preferred outcome is compared to the performance of the non-weighted classifiers [43], to judge which method has better results.

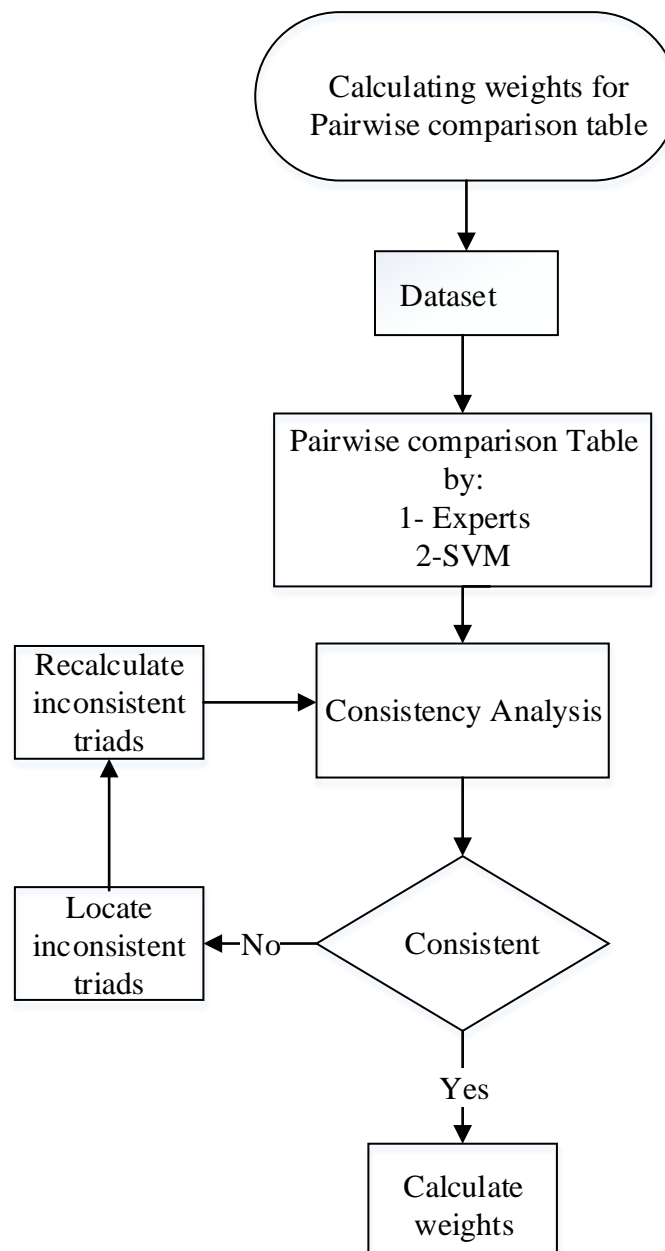


Figure 3.1: How weights are calculated when data goes through the pairwise comparison process.

After the weights have been calculated by means of geometric weights or mean weights, the numbers can then be normalized [31-34]. However, this is not necessary in this case because whether or not the weights are normalized, they will give the same results. For example, if all the numbers in the data set were to be normalized and rounded up to the next even number it will make any difference in the outcome, as the difference between each of those instances in the dataset will still have stayed the same. In this project, it is the relative difference between the features that shows the preferential value of each feature. The aim is to ascertain the importance of certain features over others. Because it did not add to the efficiency of the system, it was decided not to normalize the weights and save time. By efficiency of the system, the accuracy of the outcome is meant. In conjunction with these explanations, readers can also deduce that whether normalized or not, the difference will be the same.

The main advantage to scaling is that it avoids the risk of being faced with the issue of features with greater numeric ranges dominating those with smaller numeric ranges. For example if one feature, i.e. age, ranges from 5-100 and another feature, i.e. gender, has the value of 0 or 1, the models outcome falls the risk to be dominated by the first feature. Furthermore, because kernel values usually depend on the inner products of feature vectors, e.g. the linear kernel and the polynomial kernel, large attribute values might cause numerical problems [39]. However, the point to remember is that the same method of scaling must be used on both the training and testing data. In this thesis both the training and testing data was scaled.

After the weights have been calculated, there is no need to normalize them. The values retrieved from the pairwise comparison table can be used directly.

3.3.1.1. Weight Calculation Using Experts

The weight calculation as explained before is done with the assistance of the pairwise comparison table. Therefore, it can be judged that the results of the weights are greatly influenced by how the table is filled out. There are many methods for assigning weights to features. However, from the very start, one of the chief concerns of this research was realizing an

unbiased result. That is the reason for using expert's opinions and not simply the general population to fill out the table. Furthermore, by using the pairwise comparison table in this process, it was hoped also to eliminate certain errors of inconsistency, the main concern being that the human mind with its biases and limitations cannot be free of error.

To fill out the pairwise comparison table, the researcher did extensive research in the area to be able to sufficiently give opinions as an expert. After much research in this area and the traveling habits of people, the expert (or in this case the researcher) filled out the pairwise comparison table. Travel agents work and earn money by booking flights and getting a certain commission off those travels. Although many trips are either impromptu or last minute or the traveler just found the agency online and by accident, many agents work through their existing clients. They know when a specific client of theirs usually takes a trip and what tours or destinations they are interested in. The agent keeps track of clients in the lookout for good deals and potential travel times. An ideal travel agent not only keeps track of all their clients, but also makes notes about what customers could potentially be prospective clients. They keep an eye out for certain features in each customer. These were among the points that the researcher had to bear in mind in order to fill out the comparison table. Such detail is not exclusive to traveling, in any retail setting; such issues are important and are practised daily by sales people. In order to be the best at what they do, and to keep their customer base, sales people keep note of such details and make predictions about what purchases will their client might be interested in.

The objective of this part of the research was to gain this information from the experts in this area, in order to fill out the pairwise comparison table aptly as possible. This is the process of developing a unique weighted system through eliciting expert judgments.

This is what any expert travel agent or salesperson does on a daily basis: making judgments about their customers and taking the information they receive daily and applying it to other situations. However, in the case of the human mind, this process is more time consuming and not always as accurate. The approach chosen in this project was to take advantage of the evaluations of multiple experts.

3.3.1.2. Weight Calculation Using SVM

While filling out the pairwise comparison table and later on using it to improve the results of the basic SVM, another course of action came to mind. Every time the SVM was run on matlab certain attributes were returned, such as accuracy, number of support vectors and so on. Through further research in this area, a new course of direction presented itself. Why not let SVM decide the importance of each feature, that is, the relative dominance of certain features over others?

It is this weightiness that assists in filling up the pairwise comparison table. However, in this instance, the amount of weightiness is judged by the outcome of SVM.

When running SVM in matlab, certain attributes will be returned after the SVM has been trained and classified. In line with what is being proposed in this section, SVM was first trained and classified for each feature separately and then the results were compared to examine the level of accuracy. In this step, SVM determines which feature is more important by itself. However, because a feature can seem important in a stand-alone situation but might give a different result when combined with other features, this process was repeated, but this time the features were paired in two's.

Pairing the features in two's resulted in SVM being trained using two features at a time. This gave 21 different results. Just obtaining these results was not the end of the exercise; each of these results had to be compared and evaluated. To evaluate the results involving the two pairs, a specific method was followed. Once the training and classification was completed, and the accuracy rate for the combination of two features was returned, this number was compared with the maximum accuracy that the individual feature SVM returned. The results of that comparison were then divided into two, giving each feature the same amount of responsibility. At the end of all the comparisons, the end results for each feature would be the sum of all their related comparison results. For example, the end result for feature 1 would be the sum of all the results of the comparisons for R_{12} , R_{13} , R_{14} , R_{15} , R_{16} and R_{17} . In Table 3.3, the different combinations and their respective comparisons are shown.

Table 3.4: Comparison of two-some variations of SVM

								Sum of Comparisons
Feature 1	$R_{11}with$ R_1	$R_{12}with$ $\max(R_1, R_2)$	$R_{13}with$ $\max(R_1, R_3)$	$R_{14}with$ $\max(R_1, R_4)$	$R_{15}with$ $\max(R_1, R_5)$	$R_{16}with$ $\max(R_1, R_6)$	$R_{17}with$ $\max(R_1, R_7)$	68.06
Feature 2	$R_{21}with$ $\max(R_1, R_2)$	$R_{22}with$ R_2	$R_{23}with$ $\max(R_2, R_3)$	$R_{24}with$ $\max(R_2, R_4)$	$R_{25}with$ $\max(R_2, R_5)$	$R_{26}with$ $\max(R_2, R_6)$	$R_{27}with$ $\max(R_2, R_7)$	75.53
Feature 3	$R_{31}with$ $\max(R_3, R_1)$	$R_{32}with$ $\max(R_3, R_2)$	$R_{33}with$ R_3	$R_{34}with$ $\max(R_3, R_4)$	$R_{35}with$ $\max(R_3, R_5)$	$R_{36}with$ $\max(R_3, R_6)$	$R_{37}with$ $\max(R_3, R_7)$	67.78
Feature 4	$R_{41}with$ $\max(R_4, R_1)$	$R_{42}with$ $\max(R_4, R_2)$	$R_{43}with$ $\max(R_4, R_3)$	$R_{44}with$ R_4	$R_{45}with$ $\max(R_4, R_5)$	$R_{46}with$ $\max(R_4, R_6)$	$R_{47}with$ $\max(R_4, R_7)$	72.68
Feature 5	$R_{51}with$ $\max(R_5, R_1)$	$R_{52}with$ $\max(R_5, R_2)$	$R_{53}with$ $\max(R_5, R_3)$	$R_{54}with$ $\max(R_5, R_4)$	$R_{55}with$ R_5	$R_{56}with$ $\max(R_5, R_6)$	$R_{57}with$ $\max(R_5, R_7)$	68.16
Feature 6	$R_{61}with$ $\max(R_6, R_1)$	$R_{62}with$ $\max(R_6, R_2)$	$R_{63}with$ $\max(R_6, R_3)$	$R_{64}with$ $\max(R_6, R_4)$	$R_{65}with$ $\max(R_6, R_5)$	$R_{66}with$ R_6	$R_{67}with$ $\max(R_6, R_7)$	68.28
Feature 7	$R_{71}with$ $\max(R_7, R_1)$	$R_{72}with$ $\max(R_7, R_2)$	$R_{73}with$ $\max(R_7, R_3)$	$R_{74}with$ $\max(R_7, R_4)$	$R_{75}with$ $\max(R_7, R_5)$	$R_{76}with$ $\max(R_7, R_6)$	$R_{77}with$ R_7	68.15

Once the sum of comparison for each feature was established, those values were then used to fill up the pairwise comparison table. From there, the specific calculations needed for the pairwise comparison table were performed and, together with the weights, were calculated using geometric means.

The same method was followed for comparison in three's. This will bring the plane to the third dimension. In this section, the different variations will include R_{ijk} (for $i = 1:7, j = 1:7$ and $k = 1:7$).

This process can go on for a very long time, but for the sake of this research and weight calculation, the separate comparisons between the dimensions were only carried out up to the third dimension, resulting in 124 different possible combinations.

The SVM method for calculating the values for the different dimensions in one's then in two's and three's is explained below. The training and classification of SVM for one dimension is displayed in this pseudo-code displayed below:

Run the following for each dimension (feature 1 to 7) and store result:

Train the algorithm using one out of all of dimensions of first 800 samples (a dimension of Training set)

Get the predicted labels for one out of all of dimensions of the last 200 samples (a dimension of test set)

Get the accuracy by dividing the (correct-predictions)/(total number of samples in test set)

As it is clearly displayed in the coding above, the SVM is trained and classified with the scaled training dataset. The result of the classification is saved in another object and that is used to calculate the following:

Run the following for each possible combination of 2 dimensions out of all dimensions (2 of 7) and store result:

Train the algorithm using two out of all of dimensions of first 800 samples(two dimensions of Training set)

Get the predicted labels for two out of all of dimensions of the last 200 samples (two dimensions of test set)

Get the accuracy by dividing the (correct-predictions)/(total number of samples in test set)

After the results from running SVM have been gathered, the next step is to analyze these results. This is done by comparing the result of each combination with the maximum result for the stand-alone results, that is, by comparing R_{ijk} (for $i = 0:7, j = 0:7$ and $k = 0:7$) with maximum of R_i, R_j and R_k (for $i = 0:7, j = 0:7$ and $k = 0:7$).

In Table 3.4, the different possible combinations are displayed. However, as certain combinations are simply a repetition of an earlier combination, we refrained from reposting those certain combinations. For example, for the features 1, 4 and 7, the combinations 1:4:7, 1:7:4, 4:1:7, 4:7:1, 7:1:4 and 7:4:1 will all give the same result. Therefore, by figuring out the answer to one of them, the answer to all possible combinations would have been resolved.

Table 3.5: Comparison of three-dimensional variations of feature one in SVM

Feature1				
$R_{123} - \max(R_{12}, R_{23}, R_{13})$	$R_{134} - \max(R_{13}, R_{34}, R_{14})$	$R_{125} - \max(R_{12}, R_{25}, R_{15})$	$R_{126} - \max(R_{12}, R_{26}, R_{16})$	$R_{127} - \max(R_{12}, R_{27}, R_{17})$
$R_{134} - \max(R_{13}, R_{34}, R_{14})$	$R_{135} - \max(R_{13}, R_{35}, R_{15})$	$R_{163} - \max(R_{16}, R_{63}, R_{13})$	$R_{137} - \max(R_{13}, R_{37}, R_{17})$	$R_{145} - \max(R_{14}, R_{45}, R_{15})$
$R_{146} - \max(R_{14}, R_{46}, R_{16})$	$R_{147} - \max(R_{14}, R_{47}, R_{17})$	$R_{156} - \max(R_{15}, R_{56}, R_{16})$	$R_{157} - \max(R_{15}, R_{57}, R_{17})$	$R_{167} - \max(R_{16}, R_{67}, R_{17})$
Feature one's contribution in discrimination on 3 dimensional space will be equal to the average of above				

The discriminatory power of other features (2-7) in third dimension can be calculated by following the same pattern demonstrated in Table 3.4.

With the results gained from the different combinations of the features, each feature gained new importance. When training a SVM the number of support vectors in each training is supplied by *SVMstruct*. When running the application with only one feature, the number of support vectors was greater than when 2 were combined. As stated in the literature above, apart from the accuracy of SVM, the number of support vectors is also important. The lower the number of the Support Vectors, the less conflict there is between the features. This extra step served two purposes: first, it revealed the significance of having dimensions, and secondly, it gave a new face value to each feature. The pairwise comparison table was then filled out using the insights gathered from this step.

In the following pseudo-code, the above mentioned step will be repeated for 3-dimensional combinations:

Run the following for each possible combination of 3 dimensions out of all dimensions (3 of 7) and store result:

Train the algorithm using three out of all dimensions of first 800 samples(three dimensions of Training set)

Get the predicted labels for three out of all of dimensions of the last 200 samples (three dimensions of test set)

Get the accuracy by dividing the (correct-predictions)/(total number of samples in test set)

3.4. Prediction

In the previous section, the methodology behind how the datasets were trained and different weight judging techniques were experimented, was explained. In this section the methodology behind the prediction of the SVM algorithm is explained. The outcome of the research is judged by the results gained in this section.

In most of supervised learning methods, such as SVM, the first step is to train the algorithm and generate the classifier model. The second step is to test the model by evaluating the performance and accuracy of classification. There are multiple methods to evaluate the accuracy of a SVM classifier, one of the reliable methods is cross validation, which is also one of the options that can be added on to the classifier in libsvm. The classification process is achieved through the classifier function. When classifier returns the results of the classification, in order to be able to fully evaluate these results and then compare them with other results, the result of each classification is put into a separate value called “dimension-x accuracy”. Then the result of that specific dimension is compared to the other dimensions results using pairwise comparison. This process was followed for all the one-dimensional, two-dimensional and three-dimensional results.

During the prediction part of this research, the results from the classification process are realized. These are the results that assist in proving the importance of weights. The results from both the classification of the un-weighted SVM and the weighted SVM (using the calculated weights) are prepared and finally compared.

3.5. Summary

In this Chapter, the processes taken to fill out the pairwise comparison table and the reason for choosing them, the methods behind data collection, data training, and finally prediction were all explained. Most importantly, this part of the research has established how it has been proposed to improve the system by adding weights.

CHAPTER 4

Implementation, Results and Experiments

4.1. Introduction

In this chapter, 2 different sets of weights will be achieved by implementing pairwise comparison tables; once with the assistance of experts and once with the assistance of SVM. Next both sets of weights will be applied to the data set. The results gathered from each of the implementations will be compared with each other, to assess the more effective method of assigning weights. Next the algorithm will be trained without the use of weights and the result of this non-weighted training and the previous weighted training will be compared with each other. The result of that comparison will shed light on the effectiveness of the suggested methods.

4.2. Assigning Weights to Datasets

In order to assign the weight to the datasets, the pairwise comparison table must be filled out. Before doing so, the appropriate weights must first be calculated. As explained in Chapter 3, other than filling out the table based on the judgment of experts, another course of action is filling it out based on the results gathered from SVM.

4.2.1. Calculating Weights Using SVM

This part of the project is accomplished in a couple of steps. The first step is training and classifying our SVM based on a single feature at a time. This means that the training and classification is run once for each feature. Once the SVM has been trained, it returns a value and an accuracy measure. This is needed to fill out the pairwise comparison table. The training and classification of the data set based on a single feature is displayed below.

```
dimTrain(1:800,1) = dimension1(1:800);  
svmstruct = svmtrain(labels(1:800),dimTrain);  
dimPred(1:200,1) = dimension1(801:1000);  
  
Answer =svmpredict(labels(801:1000),dimPred,svmstruct)
```

Each time this command is executed for each dimension (i.e. feature), the result below is displayed.

```

optimization finished, #iter = 862

nu = 0.682500

obj = -545.999787, rho = -1.000510

nSV = 603, nBSV = 460

Total nSV = 603

Accuracy = 65.5% (131/200) (classification)

```

As is displayed above, the accuracy of the SVM using only the first feature for training is 65.5%. This process has to be executed once for each feature. Table 4.1 shows the results gained from the one-dimensional training of SVM.

Table 4.1: Results from one-dimensional SVM

Feature	Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7
Accuracy (%)	65.87	75.87	65.87	72.12	65.87	65.87	65.87

From looking at the results in Table 4.1, it can be deduced that feature 2 (location), with an accuracy rate of 75.87%, has the most influence on the results. Using the results gathered from Table 4.1, the related pairwise comparison table can be calculated. This process is shown in Table 4.2. The value of each cell in table 4.2 is corresponding row divided by corresponding column. For example cell in row one column two is (feature 1 accuracy / feature 2 accuracy) which is :

$$\frac{65.87}{75.87} = 0.868$$

Table 4.2: Pairwise Comparison Table for one-dimensional rendering of SVM

F/F	Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7
Feature1	1	0.86	1	0.91	1	1	1
Feature2	1.15	1	1.15	1.05	1.15	1.15	1.15
Feature3	1	0.86	1	0.91	1	1	1
Feature4	1.09	0.95	1.09	1	1.09	1.09	1.09
Feature5	1	0.86	1	0.91	1	1	1
Feature6	1	0.86	1	0.91	1	1	1
Feature7	1	0.86	1	0.91	1	1	1

As explained earlier, after the pairwise comparison table is assigned and validated, the weights can then be calculated. There are many ways to calculate weights from a pairwise comparison table, but for the purpose of this research, the geometric means method was used. In geometric means the weight of each feature is calculated by the seventh root of the result gained by multiplying each of the seven values in the same row as the feature being analysed.

Table 4.3: The weights of features calculated by geometric means

Feature	Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7
Accuracy (%)	0.96	1.11	0.96	1.05	0.96	0.96	0.96

These results, Table 4.3, will be used in the next phase of the research by insertion into the SVM.

However, as explained in previous sections, the appeal of SVM is its multi-dimensional feature.

The next step is to obtain the two dimensional results from SVM. Below is the Matlab code used to calculate the accuracy of 2-dimentional classifications.

```

i(1:200,1)=scaled(801:1000,1)
i(1:200,2)=scaled(801:1000,2)
model=svmtrain(scaled(1:800,1:2),scaled(1:800,8))
comboln2=svmpredict(scaled(801,1000,8),i(1:200,1:2),model)
countln2 = 0
    for j=1:200
    if (comboln2(j)==scaled(j+800,8))
countln2=countln2+1
    end
end
end

```

This means obtaining the results from all of the possible 2-combinations of the seven features. In Tables 4.4 - 4.10, different combinations will be displayed. In Table 4.4, the results of combining one other feature with feature number 1 are displayed. This, means that in the training phase of SVM, feature number one and another feature were only included in the training process.

Table 4.4: The two dimensional feature combinations and accuracy including feature 1

Feature Combination	F1F2	F1F3	F1F4	F1F5	F1F6	F1F7
Accuracy (%)	74.9	65.2	71.9	64.8	65.8	65.8

As displayed in Table 4.4, the combination of features 1 and 2 has given the best results, that is, in terms of the accuracy of the SVM.

Table 4.5: The two-dimensional feature combinations and accuracy including feature 2

Feature	F1F2	F2F3	F2F4	F2F5	F2F6	F2F7
Accuracy (%)	74.9	74.9	78.1	74.8	75.5	75

In Table 4.5, the combination of feature 2 with other features for the purpose of training the SVM is displayed. It can be seen from the table that the result of combining feature 2 and 4 has so far given the best results.

Table 4.6, displays the various combinations that can be made with feature number 3. As in the case of the previous combinations, the accuracy rate is important for the next step.

Table 4.6: The two-dimensional feature combinations and accuracy including feature 3

Feature Combination	F3F1	F3F2	F3F4	F3F5	F3F6	F3F7
Accuracy (%)	65.2	74.9	70.5	64.5	65.8	65.8

Table 4.7 reveals the effect of the combination of feature number 4 with other features. From the results in the table, it is clear that the highest accuracy rate is when feature number 4 is combined with feature number 2.

Table 4.7: The two-dimensional feature combinations and accuracy including feature 4

Combination	F4F1	F4F2	F4F3	F4F5	F4F6	F4F7
Accuracy (%)	71.9	78.1	70.5	73.6	71	71

In Table 4.8, the two dimensional combination of feature number 5 with other alternative features is displayed. This all follows the idea that, although one feature by itself has certain discriminatory powers, it might have a higher distinction rate when combined with other features or one other specific feature in this case.

Table 4.8: The two-dimensional feature combinations and accuracy including feature 5

Combination	F5F1	F5F2	F5F3	F5F4	F5F6	F5F7
Accuracy (%)	64.8	74.8	64.5	73.6	65.8	65.5

The combination between feature number 6 and the other features in its set is displayed in Table 4.9.

Table 4.9: The two-dimensional feature combinations and accuracy including feature 6

Combination	F6F1	F6F2	F6F3	F6F4	F6F5	F6F7
Accuracy (%)	65.8	75.5	65.8	71	65.8	65.8

The final combination is that of feature number 7 with others in its dataset, which is displayed in Table 4.10.

Table 4.10: The two-dimensional feature combinations and accuracy including feature 7

Combination	F7F1	F7F2	F7F3	F7F4	F7F5	F7F6
Accuracy (%)	65.8	75	65.8	71	65.5	65.8

It is clear from the results in Tables 4.4 to 4.10 that feature 2 and 4 have a greater influence on the results when they are combined with other features. However, because guesswork is not permitted in research, Table 4.11 below shows the product of the evaluation of the two dimensional combination results.

Table 4.11: Accuracy rate of features based on two-dimensional rendering of SVM

Feature	Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7
Weight	68.06	75.53	67.78	72.68	68.16	68.28	68.15

These results are needed to fill out the pairwise comparison table. To reiterate what was explained before, the method being proposed involves letting SVM fulfil the role of the experts in filling out the pairwise comparison table. Instead of experts identifying which feature is more important and how much more important it is, the outcome from utilizing SVM will reveal the influence of the features in classification results. Table 4.12 displays the pairwise comparison table based on the two dimensional accuracy of the features in the dataset.

Table 4.12: Pairwise Comparison Table for two-dimensional rendering of SVM

F/F	Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7
Feature1	1	0.90	1.00	0.93	0.99	0.99	0.99
Feature2	1.10	1	1.11	1.03	1.10	1.10	1.10
Feature3	0.99	0.89	1	0.93	0.99	0.99	0.99
Feature4	1.06	0.96	1.07	1	1.06	1.06	1.06
Feature5	1.00	0.90	1.00	0.93	1	0.99	1.00
Feature6	1.00	0.90	1.00	0.93	1.00	1	1.00
Feature7	1.00	0.90	1.00	0.93	0.99	0.99	1

With the help of our pairwise comparison table the weights based on the two dimensional amalgamation of the features can be calculated. The final feature and their respective weights based on two dimensional combinations and pairwise comparison are presented in Table 4.13.

Table 4.13: The weights of features calculated by means of Geometric means

Feature	Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7
Weight	0.97	1.08	0.97	1.04	0.97	0.97	0.97

These weights will be used in the training and classification of the SVM of the dataset for this thesis.

This process can go on for many more variations, but for the purpose of this thesis, only combinations up to three levels are being considered. In the tables below, the results of the different combinations can be seen. Table 4.14 displays the different possible combinations with feature number one and the accuracy they each achieve.

Table 4.14: The three-dimensional feature combinations and accuracy including feature 1

Feature Combination	Accuracy (%)	Feature Combination	Accuracy (%)	Feature Combination	Accuracy (%)
F1F2F3	75.9	F1F4F2	76	F1F6F2	75.2
F1F2F4	76	F1F4F3	71.5	F1F6F3	65.8
F1F2F5	75.3	F1F4F5	70	F1F6F4	69.8
F1F2F6	75.2	F1F4F6	69.8	F1F6F5	65.4
F1F2F7	75.2	F1F4F7	71.2	F1F6F7	65.8
F1F3F2	75.9	F1F5F2	75.3	F1F7F2	75.2
F1F3F4	71.5	F1F5F3	64.2	F1F7F3	65.8
F1F3F5	64.2	F1F5F4	70	F1F7F4	71.2
F1F3F6	65.8	F1F5F6	65.4	F1F7F5	65
F1F3F7	65.8	F1F5F7	65	F1F7F6	65.8

In Table 4.14, the different possible combinations including the first feature are displayed. The first column displays the features being combined while the second column displays the accuracy of the combination.

In Table 4.15, the different possible combinations involving the second feature are displayed.

Table 4.15: The three-dimensional feature combination and accuracy including feature 2

Feature Combination	Accuracy (%)	Feature Combination	Accuracy (%)	Feature Combination	Accuracy (%)
F2F1F3	75.9	F2F4F1	76	F2F6F1	75.2
F2F1F4	76	F2F4F3	78.1	F2F6F3	74.8
F2F1F5	75.3	F2F4F5	78	F2F6F4	78.2
F2F1F6	75.2	F2F4F6	78.2	F2F6F5	74.2
F2F1F7	75.2	F2F4F7	78.3	F2F6F7	75.1
F2F3F1	75.9	F2F5F1	75.3	F2F7F1	75.2
F2F3F4	78.1	F2F5F3	76.5	F2F7F3	75.3
F2F3F5	76.5	F2F5F4	78	F2F7F4	78.3
F2F3F6	74.8	F2F5F6	74.2	F2F7F5	74.5
F2F3F7	75.3	F2F5F7	74.5	F2F7F6	75.1

Table 4.16 displays the results of combining feature number 3 with feature 6 in this data set. As has probably been observed up to this point, some of the features are repeated in different combinations. For example, the combination between features 1, 2 and 3 on one hand and features 3, 1 and 2 on the other are very similar; in fact, they are the same. However, carrying out three different assessments, once for feature 1 and then for feature 2 and 3, is vital. Because the result of the outcome we are interested in is the influence of all three features, the contribution of each of these features to the result of this combination should be assessed separately in order to reach an unbiased and fair conclusion.

Table 4.16: The three-dimensional feature combination and accuracy including feature 3

Feature Combination	Accuracy (%)	Feature Combination	Accuracy (%)	Feature Combination	Accuracy (%)
F3F1F2	75.9	F3F4F1	71.5	F3F6F1	65.8
F3F1F4	71.5	F3F4F2	78.1	F3F6F2	74.8
F3F1F5	64.2	F3F4F5	71.8	F3F6F4	69.5
F3F1F6	65.8	F3F4F6	69.5	F3F6F5	66.1
F3F1F7	65.8	F3F4F7	70	F3F6F7	65.8
F3F2F1	75.9	F3F5F1	64.2	F3F7F1	65.8
F3F2F4	78.1	F3F5F2	76.5	F3F7F2	75.3
F3F2F5	76.5	F3F5F4	71.8	F3F7F4	70
F3F2F6	74.8	F3F5F6	66.1	F3F7F5	65.4
F3F2F7	75.3	F3F5F7	65.4	F3F7F6	65.8

Table 4.17 shows the various different combinations involving feature number 4 and the additional features in the dataset.

Table 4.17: The three-dimensional feature combination and accuracy including feature 4

Feature Combination	Accuracy (%)	Feature Combination	Accuracy (%)	Feature Combination	Accuracy (%)
F4F1F2	76	F4F3F1	71.5	F4F6F1	69.8
F4F1F3	71.5	F4F3F2	78.1	F4F6F2	78.2
F4F1F5	70	F4F3F5	71.8	F4F6F3	69.5
F4F1F6	69.8	F4F3F6	69.5	F4F6F5	71.8
F4F1F7	71.2	F4F3F7	70	F4F6F7	69.7
F4F2F1	76	F4F5F1	70	F4F7F1	71.2
F4F2F3	78.1	F4F5F2	78	F4F7F2	78.3
F4F2F5	78	F4F5F3	71.8	F4F7F3	70
F4F2F6	78.2	F4F5F6	71.8	F4F7F5	73.4
F4F2F7	78.3	F4F5F7	73.4	F4F7F6	69.7

Although in some cases the best combination can be guessed from a comparison of only a few features, this is not the proper procedure. Because of the error of the human mind, it is neither safe nor correct to guess the best method by just comparing a few different combinations. As explained before, some features do not show their importance alone but reveal their influence only when they are combined with others. In Table 4.18, the different combinations containing feature number 5 are displayed.

Table 4.18: The three-dimensional feature combination and accuracy including feature 5

Feature Combination	Accuracy (%)	Feature Combination	Accuracy (%)	Feature Combination	Accuracy (%)
F5F1F2	75.3	F5F3F1	64.2	F5F6F1	65.4
F5F1F3	64.2	F5F3F2	76.5	F5F6F2	74.2
F5F1F4	70	F5F3F4	71.8	F5F6F3	66.1
F5F1F6	65.4	F5F3F6	66.1	F5F6F4	71.8
F5F1F7	65	F5F3F7	65.4	F5F6F7	65.8
F5F2F1	75.3	F5F4F1	70	F5F7F1	65
F5F2F3	76.5	F5F4F2	78	F5F7F2	74.5
F5F2F4	78	F5F4F3	71.8	F5F7F3	65.4
F5F2F6	74.2	F5F4F6	71.8	F5F7F4	73.4
F5F2F7	74.5	F5F4F7	73.4	F5F7F6	65.8

In Table 4.19 presents the combination of feature 6 with the other features in the dataset for the purpose of this thesis.

Table 4.19: The three-dimensional feature combination and accuracy including feature 6

Feature Combination	Accuracy (%)	Feature Combination	Accuracy (%)	Feature Combination	Accuracy (%)
F6F1F2	75.2	F6F3F1	65.8	F6F5F1	65.4
F6F1F3	65.8	F6F3F2	74.8	F6F5F2	74.2
F6F1F4	69.8	F6F3F4	69.5	F6F5F3	66.1
F6F1F5	65.4	F6F3F5	66.1	F6F5F4	71.8
F6F1F7	65.8	F6F3F7	65.8	F6F5F7	65.8
F6F2F1	75.2	F6F4F1	69.8	F6F7F1	65.8
F6F2F3	74.8	F6F4F2	78.2	F6F7F2	75.1
F6F2F4	78.2	F6F4F3	69.5	F6F7F3	65.8
F6F2F5	74.2	F6F4F5	71.8	F6F7F4	69.7
F6F2F7	75.1	F6F4F7	69.7	F6F7F5	65.8

The final group of combinations is the one comprising feature number 7. Feature number 7 goes through the various different combinations (thirty to be exact). This is displayed in Table 4.20.

Table 4.20: The three-dimensional feature combination and accuracy including feature 7

Feature Combination	Accuracy (%)	Feature Combination	Accuracy (%)	Feature Combination	Accuracy (%)
F7F1F2	75.2	F7F3F1	65.8	F7F5F1	65
F7F1F3	65.8	F7F3F2	75.3	F7F5F2	74.5
F7F1F4	71.2	F7F3F4	70	F7F5F3	65.4
F7F1F5	65	F7F3F5	65.4	F7F5F4	73.4
F7F1F6	65.8	F7F3F6	65.8	F7F5F6	65.8
F7F2F1	75.2	F7F4F1	71.2	F7F6F1	65.8
F7F2F3	75.3	F7F4F2	78.3	F7F6F2	75.1
F7F2F4	78.3	F7F4F3	70	F7F6F3	65.8
F7F2F5	74.5	F7F4F5	73.4	F7F6F4	69.7
F7F2F6	75.1	F7F4F6	69.7	F7F6F5	65.8

A comparison of the results gathered from tables 4.14 - 4.20 is presented in Table 4.21, which displays the results of this comparison by giving each feature a new accuracy rate (%).

Table 4.21: Accuracy rate of features based on third dimensional rendering of SVM

Feature	Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7
Weight	70.14	76.04	70.43	73.15	70.49	70.2	70.42

Once again, the influence of feature number 2 and 4 is evident. This is not to say that the other features have had less influence on the outcome of our SVM; it is only the fact that when considering our data set in three dimensions, two of the features have a higher influence rate on the outcome than the others.

Having analyzed the results from SVM, these results can assist in filling up the pairwise comparison table as discussed in Chapter 3 section 3.3.1.2 . Table 4.22 serves this purpose, for the 3D SVM results.

Table 4.22: Pairwise comparison table for three-dimensional rendering of SVM

F/F	Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7
Feature1	1	0.92	0.99	0.95	0.99	0.99	0.99
Feature2	1.08	1	1.07	1.03	1.07	1.08	1.07
Feature3	1.00	0.92	1	0.96	0.99	1.00	1.00
Feature4	1.04	0.96	1.03	1	1.03	1.04	1.03
Feature5	1.00	0.92	1.00	0.96	1	1.00	1.00
Feature6	1.00	0.92	0.99	0.95	0.99	1	0.99
Feature7	1.00	0.92	0.99	0.96	0.99	1.00	1

After the calculations for the pairwise comparison table have been completed, the weights can finally be estimated. These weights, estimated with the assistance of three-dimensional SVM rendering, are displayed in Table 4.23.

Table 4.23: The weights of features calculated by means of Geometric means

Feature	Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7
Weight	0.98	1.06	0.98	1.02	0.98	0.98	0.98

Up to this point, two different methods for weight calculation have been revealed. The first is the method proposed in this research, that is, making the determination based on SVM (i.e. letting SVM decide), and the second is the traditional method of using experts in the field (Table 4.24). One set of weights was gathered based on the judgment of the experts, and three different weights from the training of SVM, one for each of the dimensions tested. In the next section, these weights will be applied and their results will be compared.

4.2.2. Assigning Weights using expert judgment

One of the ways in which a pairwise comparison table can be filled is with the assistance of experts. The expert in the case of the project proposed in this research is one of the highest rated travel agents in a well-known travel agency.

This expert was asked to assign weights to the features, based on their influence, on a scale of 1-5. The level of this influence is measured by what aspects this agent deemed more important in whether or not a client will be traveling in the following year. Table 4.24 displays the result of that judgement.

Table 4.24: Influence rate of features based on expert judgment using a 1-5 scale

Feature	Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7
Weight	4.5	3	1.5	2	1.5	2	2.5

The next step was to construct a pairwise comparison table with each of those listed objectives. This is displayed in Table 4.25.

Table 4.25: Pairwise Comparison Table based on expert judgment

F/F	Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7
Feature1	1	1.5	3	2.25	3	2.25	1.8
Feature2	0.66	1	2	1.5	2	1.5	1.2
Feature3	0.33	0.5	1	0.75	1	0.75	0.6
Feature4	0.44	0.66	1.33	1	1.33	1	0.8
Feature5	0.33	0.5	1	0.75	1	0.75	0.6
Feature6	0.44	0.66	1.33	1	1.33	1	0.8
Feature7	0.55	0.83	1.66	1.25	1.66	1.25	1

From the analysis of the above pairwise comparison table, a table of weights was obtained, which is displayed as follows in Table 4.26.

Table 4.26: The weights of features calculated by geometric means

Feature	Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7
Weight	1.98	1.32	0.66	0.88	0.66	0.88	1.10

In the next section these weights will be used to assign weights to the SVM, the results of which will be compared to those obtained when the weights were calculated using only SVM.

4.3. Analyzing the Dataset or Sets

Having calculated, the respective weights in the previous section, this section is devoted to adding the weights to the training and classification process. The previous section provided this research with three sets of weights. Each of these weights will be applied to the system and afterwards the results will be compared to each other and then finally compared to the results of the SVM without weights.

The process for the one dimension is as follows:

```
dimTrain(1:800,1) = dimension1(1:800)
svmstruct = svmtrain(labels(1:800),dimTrain)
(1:200,1) = dimension1(801:1000)
svmpredict(labels(801:1000),dimPred,svmstruct)
```

Alternatively, we can simply use SVM train with the cross validation option activated as well:

```
svmtrain(scaled(:,8),scaled(:,1:7)*oneDimWeights,'-v 3')
```

In using libsvm, there is a basic model that is followed and also certain libsvm options that users can add on, if needed. In the model above, the first instance “scaled(:,8)” represents the training label. The next instance is the training matrix, which has the one-dimensional weights multiplied to it. The final instance contains the libsvm options, in which, for the purpose of this research, it is stated that a 3-fold cross validation mode is needed. This means that the dataset will be divided into three parts, the first two parts of which will be used for training, and the remaining part used for prediction.

```
Optimization finished, #iter = 327
Nu = 0.562775
Obj = -350.586842, rho = -2.001969
nSV = 385, nBSV = 367
Total nSV = 385
Cross Validation Accuracy = 77.1%
```

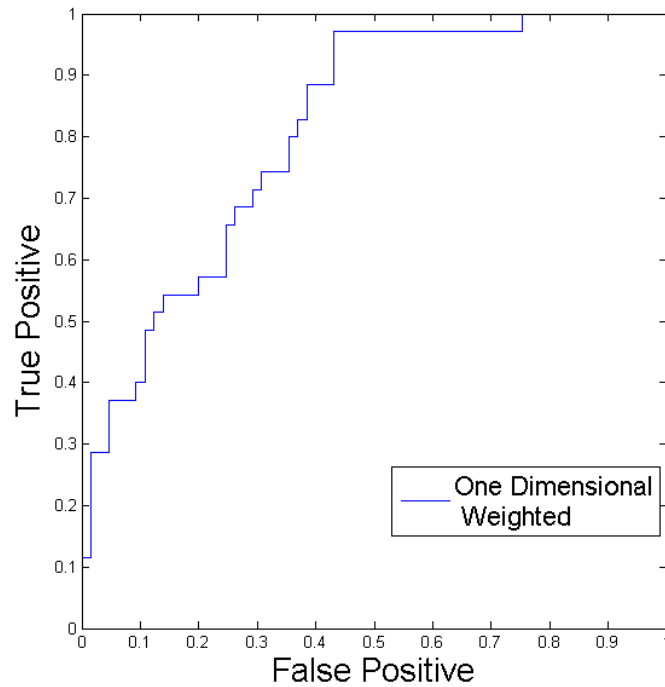


Figure 4.1 : ROC Curve for one dimensional weights

In Figure 4.1 the ROC curve for one dimensional weights is displayed. The space below the line shows the accuracy of the model for that dimension.

The next step is to apply the weights gathered from the two dimensional SVM and PWC.

```
svmtrain(scaled(:,8),scaled(:,1:7)*twoDimWeights,'-v 3')
```

This is the same process as followed for the one-dimensional weights. In the command above, the first instance is the label, the second instance the training matrix, the third instance is a matrix of the two dimensional weights achieved in the previous section and the third section is the 3-fold cross validation command. The results of this command are as follows:

```
optimization finished, #iter = 357
```

```
nu = 0.583225
```

```
obj = -362.393274, rho = -2.612750
```

```
nSV = 395, nBSV = 380
```

```
Total nSV = 395
```

```
Cross Validation Accuracy = 77.1%
```

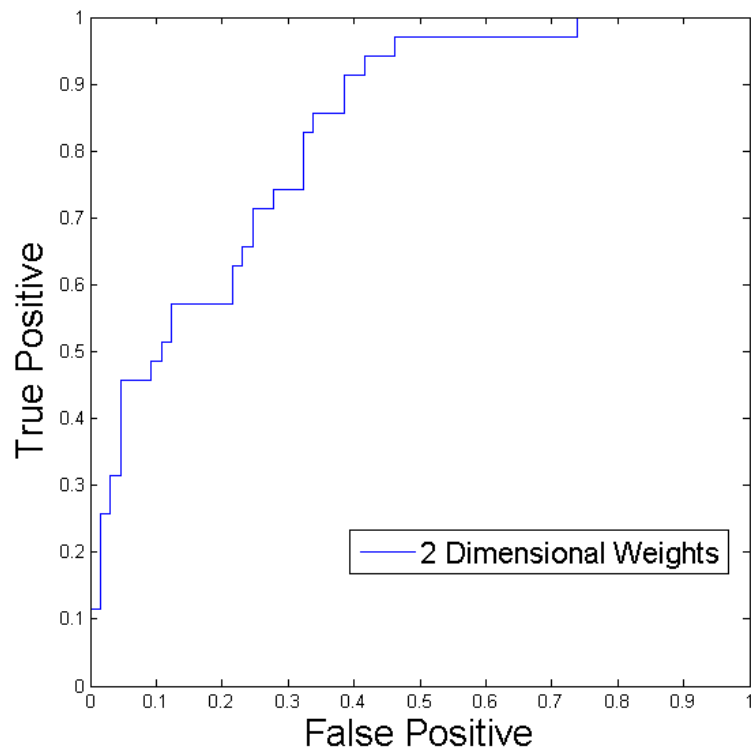


Figure 4.2 : ROC Curve for two dimensional weights

In Figure 4.2 the ROC curve for two dimensional weights is displayed. The space below the line shows the accuracy of the model for that dimension.

The next part is applying the weights from the third dimension and obtaining their results. Below is the specific command needed for this part. The first two instances are the same, while the third instance is the weighted matrix of the results gathered from the three dimensional SVM and PWC, which is being multiplied with the training instance matrix.

```
svmtrain(scaled(:,8),scaled(:,1:7)*threeDimWeights,'-v 3')
```

These are the results from executing the above command.

```
optimization finished, #iter = 345  
nu = 0.569758  
obj = -354.534925, rho = -2.161477  
nSV = 389, nBSV = 371  
Total nSV = 389  
Cross Validation Accuracy = 77.2%
```

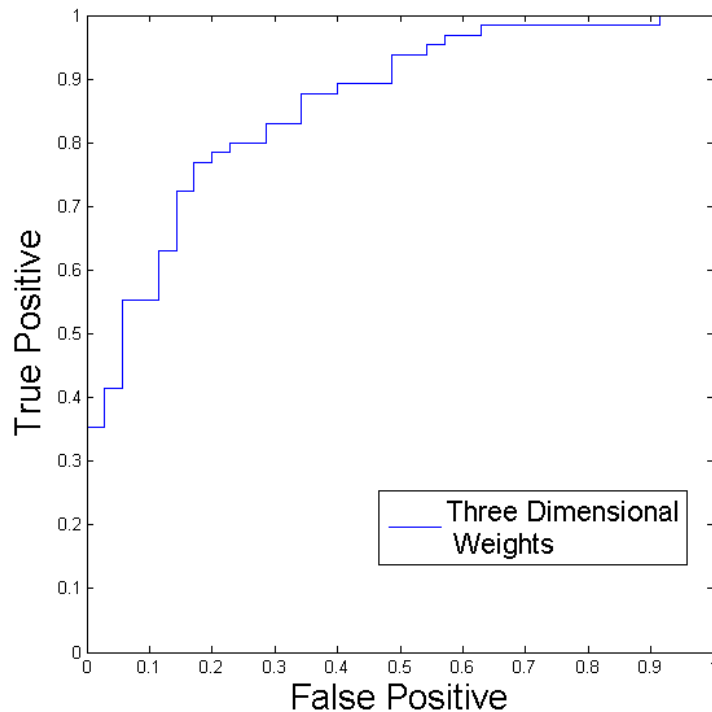


Figure 4.3 : ROC Curve for three dimensional weights

In Figure 4.3 the ROC curve for three dimensional weights is displayed. The space below the line shows the accuracy of the model for that dimension.

With the results for weighted-by-SVM calculated, the next step in this procedure is to add the results of the weights obtained with the help of experts in the area.

```
svmtrain(scaled(:,8),scaled(:,1:7)*pwcWeights,'-v 3')
```

Below are the results from that command:

```
optimization finished, #iter = 1009
nu = 0.588133
obj = -239.667414, rho = -0.469234
nSV = 649, nBSV = 193
Total nSV = 649
Cross Validation Accuracy = 73.6%
```

With all the weights added to the SVM, this thesis will proceed first with comparing these results with each other and then compare the best of such results with those of a simple SVM. Table 4.27 fully displays these results.

Table 4.27: the results of different weights

Weight achievement method	SVM accuracy (%)
One dimensional SVM weights	77.1
Two dimensional SVM weights	77.1
Three dimensional SVM weights	77.2
Expert judgement weights	73.6

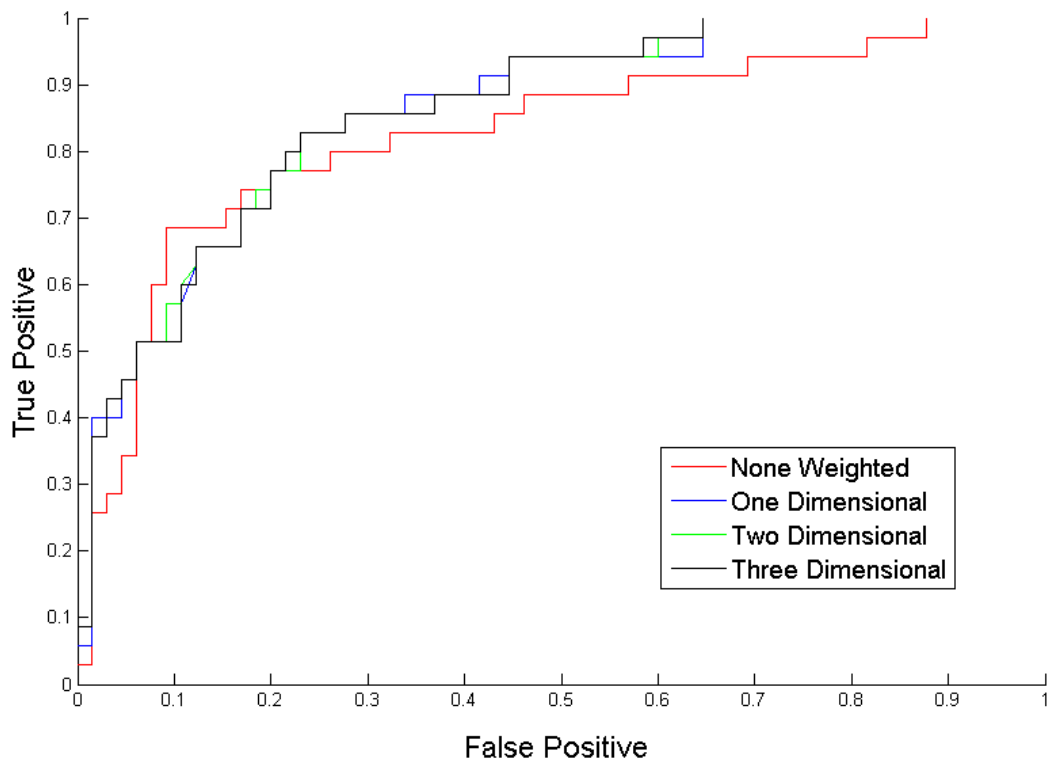


Figure 4.4 : ROC Curve Comparison

In Figure 4.4 the ROC curve comparing all models is displayed. The space below the line shows the accuracy of the model for that dimension. As it is clearly shown in Figure 4.3, the space below the black line, which signifies the accuracy of third dimensional weights, is significantly larger than the rest. Also the space below the red line which signifies the non-weighted model has the lowest amount of space.

The best result, or the most accurate rate of prediction, belongs to the results of the weights that were achieved through the third dimensional rendering of SVM. In Table 4.28, these results will be compared to the results of an un-weighted SVM. The SVM being used in this section will be scaled as well. As explained earlier in Chapter 2, scaling is different from weights.

```
svmtrain(scaled(:,8),scaled(:,7)*0.921,'-v 3')
```

These are the results from that command:

```
optimization finished, #iter = 310
nu = 0.684685
obj = -455.999551, rho = -1.010194
nSV = 476, nBSV = 434
Total nSV = 476
Cross Validation Accuracy = 65.8%
```

Table 4.28: Comparison of results between un-weighted and weighted SVM (FWSVM)

	Accuracy (%)	Number of Support Vectors
Weighted SVM (WFSVM)	77.2	389
Un-weighted SVM (SVM)	65.8	476

The results displayed in Table 4.28 reveal the final outcome of the proposed method in this research. It enables us to reach a verdict on the issue of whether weights were a helpful factor in this exercise, since it is obvious that when the weights were added, the accuracy increased and the number of support vectors decreased, both of which are desirable. In Table 4.27, the preferred method for adding weights was decided upon based on the accuracy level.

With the results that are visible in Table 4.28 , in order to get a perfect percentage of improvement of the preferred weighted method over the non-weighted method, the Table 4.28 results were divided by each other. And the following is the outcome of that:

$$\frac{\text{Weighted SVM Accuracy}}{\text{Non - Weighted SVM Accuracy}} = 1.17$$

$$\frac{\text{Weighted SVM Number of Support Vectors}}{\text{Non - Weighted SVM Number of Support Vectors}} = 0.81$$

As in can be seen by applying weights, the overall accuracy has improved by 17% and the number of Support Vectors has decreased (which is also an improvement) by 8%.

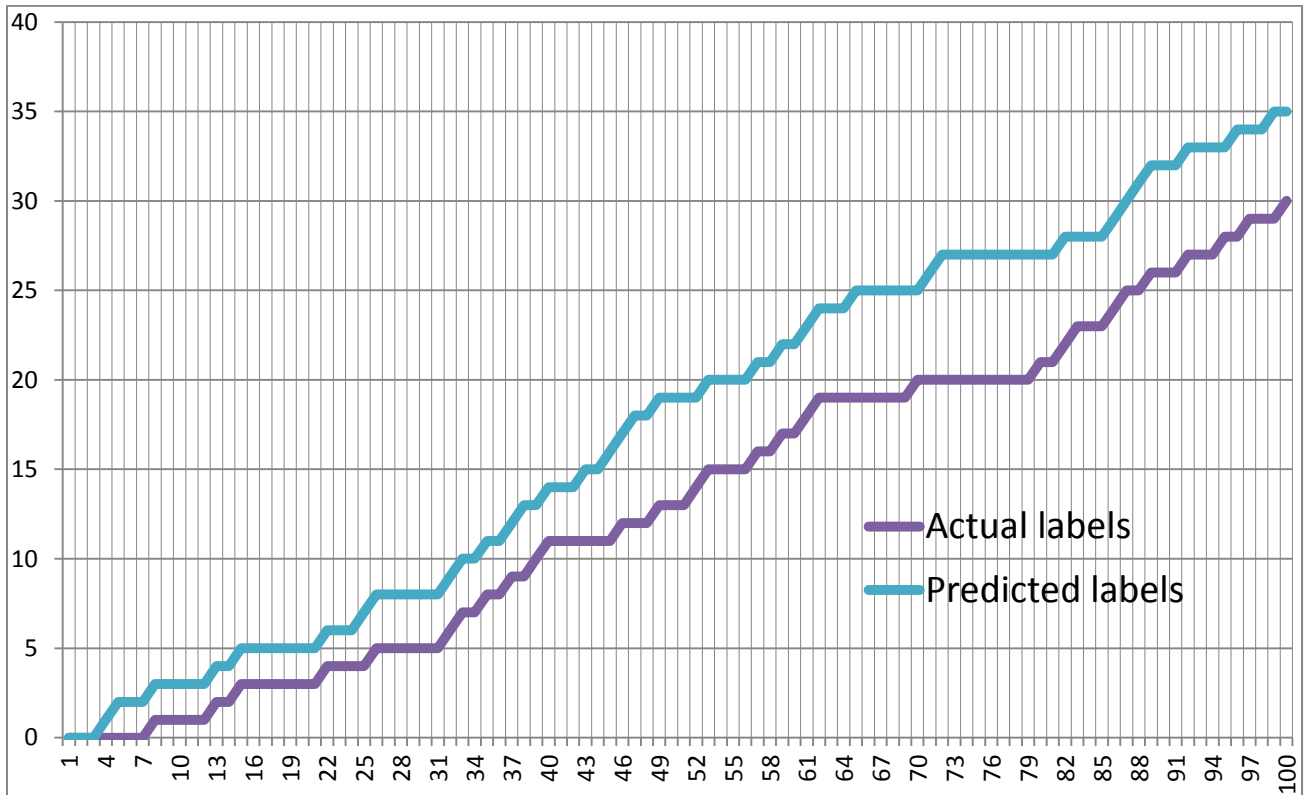


Figure 4.5 : The comparison between predicted results and actual results

As a final explanation of the results and experiments in this thesis, Figure 4.5 displays the comparison between the actual outcome and predicted outcome in an ROC Curve. In this type of curve every time the result is positive, the previous results gets incremented. If the result is negative, the curve stays the same. As it can be seen in the figure above, the two lines standing for the predicted and actual labels, are steadily growing together, other than a few slight differences.

4.4. Summary

In this chapter, two different methods for achieving weights were proposed, each of these methods was implemented and the weights achieved. Those weights were then incorporated into SVM and based on those results, a preferred method of attaining weights was chosen. The result of that method was then compared with a regular SVM, which had only been scaled. The outcome was that adding weights to SVM for the purposes of this research was successful, and the method proposed in the thesis was effective.

CHAPTER 5

Conclusion and Future Studies

5.1. Conclusion

The goal of this research was to make use of two already known practices. The application of Support Vector Machines to predict and classify information, along with the use of social networks for advertisement, is a widely used practice. It is by combining the two together and adding weights to certain features (WFSVM) that this research shines a light on a potentially different level of advertising: “Smart Advertising”[40].

During the course of this thesis, it was proposed that by the addition of proper weights to SVM, the results of the classification will show improvement. It increases the accuracy and decreases the support vectors. Two different methods were proposed for the purposes of this research.

The first method used the knowledge of experts to assist in weight calculation. It fell upon the experts to judge the importance of features by providing the measures for the mutual relationship between the features.

The second method proposed was to let the SVM assist in this judgment by having the SVM train and classify the data in one, two and three-dimensional cases. The results gathered from this approach were significantly superior to those of the first mentioned approach.

When compared to each other, the method in which SVM was allowed to assist in the weight calculation process achieved a higher accuracy and lower support vectors, both of which are desirable outcomes of an SVM classification method.

The results realized in this thesis, were a positive step in the direction of designing an effective system for “Smart Advertising”. However, it should be stated that the proposed method would have yielded more positive results if it had involved the use of “natural data”, data gathered from nature or medical resources. This was not the case for the data set used for the purpose of this thesis, which was derived from the Facebook depository.

The type of data that was used for the purpose of this research had certain attributes, one of them being that it was often irregular and unpredictable. As stated before, more natural data would

have yielded significantly better results. However, this research did have many benefits, a number of which are listed below:

- Having SVM determine the classification ability of each feature.
- Proposing a method for taking advantage of Facebook data, and finding this method useful.
- It proved the advantage of having weights over not having weights.
- Using a simple method of classification (i.e. SVM), which created hyperplanes that solved classification problems.
- Using Simpler models: rather than having a very complicated classification, the datasets were taken into multi-dimensions to make classification simpler.
- Faster classification
- Quick learning.
- Human mind simulation: being able to surpass the judgment of experts.

And most importantly,

- A contribution to the knowledge currently being utilized in social networks.

5.2. Contributions

Following are a list of contributions made in this thesis:

- 1- Using a method based on Support Vector Machine to predict travelling behaviour of Facebook users, this will be the first time that this method has been used to predict the behaviour of Facebook users.
- 2- Comparison of Support Vector Machines and Feature Weighted Support Vector Machines
- 3- Comparison between having experts fill out the pairwise comparison table to having Support Vector Machine fill out the table.

5.3. Future Studies

In this thesis, data were gathered and classified using the most appropriate method conceivable. However, this should not be and is not the end of this endeavor.

During this research the approach of having SVM determine the importance of features was proposed, which was undertaken until the third dimension. This can be continued in multi-dimensional planes, and there is no reason why a particular feature cannot play a different role in multi-class classifications. This is only one aspect of the future development of this research.

Completing this work and adding recommender systems to it could lead to the achievement of, smart recommendation. The current practice is for people to find data. A person can go and search the web to find what he/she is looking for. Cookies from the websites previously visited are stored and next time he/she goes onto a social networking site, he/she is shown ads for a specific thing. The success of such an approach is based on the random fact that the user has not yet found what they were looking for. In this age, the time has passed for a user to go looking for data; instead, it is time for data to find users, and that is what this research is aimed at. Thus, data will be trained to predict future trends, so that in the future, at the opportune time, the data will go to find the users. There is no failure in this new approach, compared to the older system of doing things, which just collected cookies and then sent relative ads (randomly), something we are still doing but less randomly. The new approach still does that, but less randomly, and more on principle. This is not a very difficult task; it simply uses a lot of the information gathered in this study for a different purpose. However, that is not within the scope of this thesis. Here, the focus was only to prove the ineffectiveness of previous studies, how they can be improved by machine learning, and how a more optimal result can be achieved by adding weights to the features.

REFERENCES

- [1] Facebook Statistics. Retrieved April 19, 2013, from Facebook Key Facts: <http://newsroom.fb.com/key-facts>
- [2] Check Facebook. Retrieved April 19, 2013, From CheckFacebook: <http://www.checkfacebook.com/>
- [3] Official Facebook website, Retrieved April 19, 2013, From Facebook about: <http://www.facebook.com/about/ads/#change>
- [4] Official Facebook website, Retrieved April 19, 2013, From Facebook help: <http://www.facebook.com/help/364957366911074>
- [5] Taira,H.; Fan, Y.; Yoshiya, K.; Miyagi, H. Oct 1996, “A method of constructing pairwise comparison matrix into decision making”, *IEEE International Conference on Systems, Man, and Cybernetics*,vol.4, p-p 2511- 2516.
- [6] Dayan, Peter. 1996, “Unsupervised learning.” *The MIT Encyclopedia of the Cognitive Sciences*.
- [7] S. B. Kotsiantis, 2007, “Supervised Machine Learning: A Review of Classification Techniques”, *Informatica* vol.31, p-p 249–268.
- [8] Shichao Zhang, Chengqi Zhang and Qiang Yang, 2003, “Data preparation for data mining”, *Applied Artificial Intelligence*, vol. 17, p-p 375—381.
- [9] Cortes, Corinna, and Vladimir Vapnik. 1995, "Support-vector networks." *Machine learning* vol. 20, no. 3 p-p 273-297.
- [10] Suykens, J.A.K., De Brabanter, J., Lukas, L. and Vandewalle, J., 2002, “Weighted least squares support vector machines: robustness and sparse approximation”, *Neurocomputing*, Volume 48, Issues 1–4, p-p 85-105, ISSN 0925-2312, [http://dx.doi.org/10.1016/S0925-2312\(01\)00644-0](http://dx.doi.org/10.1016/S0925-2312(01)00644-0). (<http://www.sciencedirect.com/science/article/pii/S0925231201006440>)
- [11] Amutha A.L. and Kavitha S., 2011, “ Features based Classification of Images using Weighted Feature Support Vector Machines”, *International Journal of Computer Applications* vol. 26 no. 10 p-p 23-29. Published by Foundation of Computer Science, New York, USA.

- [12] Bo Sun, Shi-Ji Song and Cheng Wu, 2009, “A new algorithm of support vector machine based on weighted feature”, *Machine Learning and Cybernetics, 2009 International Conference*, vol.3, p-p 1616-1620. doi: 10.1109/ICMLC. 2009.5212256, URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5212256&isnumber=5212225>
- [13] Lei Wang, Khan, L., Liu, L. and Weili Wu, 2004, “Automatic image annotation and retrieval using weighted feature selection,” *Multimedia Software Engineering, 2004. Proceedings. IEEE Sixth International Symposium.* p-p 435-442. doi: 10.1109/MMSE.2004.30
- [14] Mikhailov, L., 2003, “Deriving priorities from fuzzy pairwise comparison judgements”, *Fuzzy Sets and Systems*, vol. 134, Issue 3, p-p 365-385, ISSN 0165-0114, [http://dx.doi.org/10.1016/S0165-0114\(02\)00383-4](http://dx.doi.org/10.1016/S0165-0114(02)00383-4).
(<http://www.sciencedirect.com/science/article/pii/S0165011402003834>)
- [15] Saaty T.L., 1980, *The analytic hierarchy process*, New York: McGraw-Hill.
- [16] Saaty T.L. 1982, *Decision Making for Leaders; the Analytical Hierarchy Process for Decisions in a Complex World*, Wadsworth, Belmont, Calif.
- [17] Färber, Ines, Günnemann, Stephan, Kriegel, Hans-Peter, Kröger, Peer, Müller, Emmanuel, Schubert, Erich, Seidl, Thomas and Zimek, Arthur, 2010. “On Using Class-Labels in Evaluation of Clusterings”. In Xiaoli Z. Fern, Ian Davidson, Jennifer Dy. *MultiClust: Discovering, Summarizing, and Using Multiple Clusterings*. ACM SIGKDD.
- [18] Duda, R.O. and Hart, P.E. 1973, *Pattern Classification and Scene Analysis*. (Q327.D83) John Wiley & Sons. P. 218 ISBN 0-471-22361-1.
- [19] Fisher, R.A., 1936, "The use of multiple measurements in taxonomic problems" *Annual Eugenics*, vol. 7, Part II, p-p 179-188.
- [20] <http://archive.ics.uci.edu/ml/datasets/Iris>
- [21] <http://archive.ics.uci.edu/ml/datasets/Vertebral+Column>

- [22] da Rocha Neto, A., Sousa, R., Barreto, G. and Cardoso, J., 2011, “Diagnostic of pathology on the vertebral column with embedded reject option” in *Proceedings of the 5th Iberian Conference on Pattern Recognition and Image Analysis*, p-p 588–595.
- [23] Vapnik, Vladimir, 2000, “The nature of statistical learning theory”, in *Springer*.
- [24] Bo Sun; Shi-Ji Song; Cheng Wu, 2009, “A new algorithm of support vector machine based on weighted feature”, *Machine Learning and Cybernetics, 2009 International Conference*, vol.3, no., p-p.1616-1620.
- [25] Keping Wang, Xiaojie Wang and Yixin Zhong, 2010, “A Weighted Feature Support Vector Machines Method for Semantic Image Classification”, *Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference*, vol.1, p-p.377-380.
(<http://www.sciencedirect.com/science/article/pii/S0305054807001086>)
- [26] Cristianini N. and Shawe-Taylor J., 2000, *An Introduction to Support Vector Machines*. Cambridge , Cambridge Press.
- [27] Weston, Jason, 2013, *Support Vector Machine Tutorial*, NEC Labs America, http://www.cs.columbia.edu/~kathy/cs4701/documents/jason_svm_tutorial.pdf, Last visited: July 3, 2013.
- [28] <https://developers.facebook.com/docs/graph-api/reference/v2.0>
- [29] SAMUEL, A. L. 1959, “Some studies in machine learning using the game of checkers”, *IBM Journal of Research and Development* vol. 3 no. 3, p-p 211–229. Reprinted in 1963 by Feigenbaum and Feldman with an additional appendix and in 1990 by Shavlik and Dietterich.
- [30] Smola, Alexander J. 2000. *Advances in large margin classifiers*. Cambridge, Mass. [u.a.]: MIT Press.
- [31] Chang, P.T. and Lee, E.S. 1995, “The estimation of normalized fuzzy weights”, *Computers & Mathematics with Applications*, vol. 29, Issue 5, p-p 21-42, ISSN 0898-1221, [http://dx.doi.org/10.1016/0898-1221\(94\)00246-H](http://dx.doi.org/10.1016/0898-1221(94)00246-H).
- [32] Da-Yong Chang, 1996, “Applications of the extent analysis method on fuzzy AHP”, *European Journal of Operational Research*, vol. 95, Issue 3, p-p 649-655, ISSN 0377-2217, [http://dx.doi.org/10.1016/0377-2217\(95\)00300-2](http://dx.doi.org/10.1016/0377-2217(95)00300-2).

- [33] Boender, C.G.E., de Graan, J.G. and Lootsma, F.A., 1989, "Multi-criteria decision analysis with fuzzy pairwise comparisons", *Fuzzy Sets and Systems*, vol. 29, Issue 2, p-p 133-143, ISSN 0165-0114, [http://dx.doi.org/10.1016/0165-0114\(89\)90187-5](http://dx.doi.org/10.1016/0165-0114(89)90187-5).
(<http://www.sciencedirect.com/science/article/pii/0165011489901875>)
- [34] Ying-Ming Wang, Celik Parkan and Ying Luo, 2008, "A linear programming method for generating the most favorable weights from a pairwise comparison matrix", *Computers & Operations Research*, vol. 35, Issue 12, p-p 3918-3930, ISSN 0305-0548, <http://dx.doi.org/10.1016/j.cor.2007.05.002>.
- [35] Joachim, Thorston, 1998, "Text categorization with Support Vector Machines: Learning with many relevant features". *Machine Learning: ECML-98*. Lecture Notes in Computer Science. Springer Berlin Heidelberg. Vol. 1398, p-p 137-142.
- [36] Niculescu-Mizil, A. and Caruana, R., 2005, "Predicting Good probabilities with supervised learning", *proceedings of 22nd International Conference on Machine Learning (ICML'05)*
- [37] Hastie, Trevor, Robert Tibshirani, Jerome Friedman, and James Franklin. 2005, "The elements of statistical learning: data mining, inference and prediction." *The Mathematical Intelligencer* vol. 27, no. 2, p-p 398-400.
- [38] Sarle, W.S., 1997, *Sarle's Neural Networks FAQ*, <http://www.faqs.org/faqs/ai-faq/neural-nets/part2/section-16.html>
- [39] Hsu, C.-W., Chang, C.-C., and Lin, C.-J., 2003. "A practical guide to support vector classification" Tech. rep., Department of Computer Science, National Taiwan University
- [40] Behzadfar, H, Abhari, A, 2014, "Applying supervised learning algorithms on information derived from social network to enhance recommender systems", *Spring Simulation Conference, Tampa*.
- [41] Merrick, Jason RW, van Dorp, J. Rene and Singh, Amita, 2005, "Analysis of correlated expert judgments from extended pairwise comparisons" *Decision Analysis* vol. 2, no. 1, p-p 17-29.
- [42] Devorshak, Christina, ed., 2012, *Plant pest risk analysis: concepts and application*. P. 114.

- [43] Waldemar, W. Koczkodaj, Herman, Michael W. and Orłowski, Marian 1997, "Using consistency-driven pairwise comparisons in knowledge-based systems", *Proceedings of the sixth international conference on Information and knowledge management. ACM*.
- [44] Barlow, Horace B. 1989, "Unsupervised learning", *Neural computation* vol. 1 no.3 p-p 295-311.
- [45] Montana, David J., and Lawrence Davis. "Training Feedforward Neural Networks Using Genetic Algorithms." *IJCAI*. Vol. 89. 1989.
- [46] Xing, H.-j., Ha, M.-h, Hu, B.-g., and Tian, D.-z. (2009). Linear Feature-Weighted Support Vector Machine. *Fuzzy Information and Engineering*, 1(3), 289-305.
- [47] Jaynes, Edwin T. "Information theory and statistical mechanics." *Physical review* 106.4 (1957): 620
- [48] Official Facebook website, Retrieved November 15, 2013, From Facebook Data Use Policy: https://www.facebook.com/full_data_use_policy
- [49] Official Facebook website, Retrieved November 15, 2013, From Facebook Statement of Rights and Responsibilities: <https://www.facebook.com/legal/terms>