

CUSTOMER RECOMMENDATION USING MACHINE LEARNIGN TECHNIQUES

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
AWARD OF DEGREE
OF
MASTER OF TECHNOLOGY IN
SOFTWARE ENGINEERING

Submitted by:

ABHIRUP DAS
2K20/SWE/01

Under the supervision of

PROF. DR. RUCHIKA MALHOTRA
(HEAD OF THE DEPARTMENT
, SOFTWARE ENGINEERING)



DEPARTMENT OF SOFTWARE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

MAY, 2022

CANDIDATE'S DECLARATION

I, Abhirup Das, Roll No. 2K20/SWE/01 student of M. Tech (Software Engineering) hereby declare that the project Dissertation titled "Customer recommendation using Machine Learning Technique" which is submitted by me to the Department of Software Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of and Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi

Date

Abhirup Das

ABHIRUP DAS

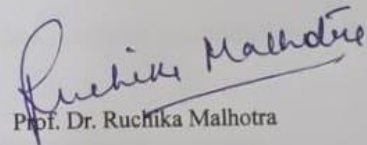
2K20/SWE/01

CERTIFICATE

I hereby certify that the Project Dissertation titled "**Customer recommendation using Machine Learning Technique**" which is submitted by Abhirup Das, 2K20/SWE/01 Department of Software Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge, this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Date:



Prof. Dr. Ruchika Malhotra

Head Of The Department,

Department of Software Engineering

Delhi Technological University

(Formerly Delhi College Of Engineering)

Bawana Road, Delhi - 110042

ACKNOWLEDGMENT

The success of this project requires the assistance and input of numerous people and the organization. I am grateful to everyone who helped in shaping the result of the project.

I express my sincere thanks to **Prof. Dr. Ruchika Malhotra**, my project guide, for providing me with the opportunity to undertake this project under her guidance. Her constant support and encouragement have made me realize that it is the process of learning which weighs more than the end result. I am highly indebted to the panel faculties during all the progress evaluations for their guidance, constant supervision and for motivating me to complete my work. They helped me throughout with new ideas, provided information necessary and pushed me to complete the work.

I also thank all my fellow students and my family for their continued support.

Abhirup Das
ABHIRUP DAS
2K20/SWE/01

ABSTRACT

In the customer relationship management business, currently there is a desire for programmed techniques for anticipating future users utilising recommendation engines. There are currently functions for locating "twins," or potential consumers who are similar to current customers, as well as searching the record of clients divided into class. Machine-learning techniques are commonly used in today's recommendation engines. As a result, it's important to figure out which ML algorithms are optimal for building a recommendation engine that can forecast client behaviour. The conditions for determining appropriateness are investigated in this thesis, as well as an evaluation of various off-the-shelf ML techniques. As a result of this, technique of discovering new potential clients, supervised learner models have showed promise. This study is oriented towards learning the behaviour of our customers, so that we can predict out future customer and also can show them their sets of desired products.

Keywords: Customer Relationship Management, Recommendation Engine, Customer Prediction, Machine Learning, Classification, Clustering, Apriori, k-Nearest Neighbors, C4.5, Decision Tree, k-Means Clustering.

CONTENTS

Candidate Declaration	i
Certificate	ii
Acknowledgement	iii
Abstract	iv
Contents	v
Lists of Figures	vii
Lists of Tables	vii
List of Abbreviations	viii
CHAPTET 1 INTRODUCTION	1
1.1 Customer Relationship Management	1
1.2 Recommendation Engine	2
1.3 Machine Learning	2
1.3.1 Supervised learning	3
1.3.2 Unsupervised learning	3
1.4 Data Requirements	4
1.5 Previous Work	4

CHAPTER 2 METHODOLOGY	5
2.1 Identifying prospective new clients	5
2.2 Analysis of Existing Algorithms	5
2.3 Evaluation Method	6
2.3.1 Confusion matrix	6
2.3.2 Accuracy and error rate	7
2.3.3 Precision and recall	8
2.3.4 Bias-Variance trade-off	9
2.3.5 Visualizing performance trade-offs	9
2.4 Implementation	11
CHAPTER 3 EVALUATION OF ALGORITHMS	12
3.1 Data Preparation	12
3.1.1 Labeling	12
3.1.2 Continuous attributes	12
3.2 Frequent Set Counting using Apriori	15
3.2.1 Implementation	16
3.2.2 Performance evaluation	18
3.3 k-Nearest Neighbors	21
3.3.1 Implementation	22

3.3.2 Performance evaluation	23
3.4 Decision Tree Induction using C4.5	25
3.4.1 Implementation	25
3.4.2 Performance evaluation	25
3.5 k-Means Clustering	26
3.5.1 Implementation	27
3.5.2 Performance evaluation	27
CHAPTER 4 DISCUSSION	28
4.1 Performance Comparison	28
4.1.1 Frequent set counting using Apriori	28
4.1.2 k-Nearest neighbors	28
4.1.3 C4.5 decision tree	28
4.1.4 k-Means clustering	29
4.1.5 ROC graph	29
CHAPTER 5 CONCLUSION	31
5.1 Algorithm for recommendation engine	31
5.2 Future work	31

LISTS OF FIGURES

Figure 1.1: ROC curve and AUC

Figure 2.1: The attribute value distribution among existing customers.

Figure 2.2: The distribution's least squares approximation.

Figure 2.3: Splitting the distribution at the approximation's local minimum results in four parts.

Figure 2.4: The lattice structure depicts the 31 potential itemsets resulting from a data collection with 5 characteristics.

Figure 2.5: Training time for various threshold values

Figure 2.6: As the itemset interval is chosen more conservatively, accuracy improves.

Figure 2.7: When the fixed interval is restricted to only the bigger sizes, recall suffers.

Figure 2.8: Two-dimensional representation of the k-nearest neighbours categorization.

Figure 2.9: Measured running time on the test machine with varying training-set sizes

Figure 3.1: In ROC space, the supervised learning techniques are displayed.

LISTS OF TABLES

Table 1.1: Confusion Matrix.

Table 2.1: Labeled company training set

Table 2.2: List of the most common n-itemsets among organisations designated as clients.

Table 2.3: All of the characteristics are present in the good cases.

Table 2.4: Sparse matrix representation of positive cases in the training set

Table 2.5: Training time for different threshold levels.

Table 2.6: FSC performance using a threshold of 0.75

Table 2.7: Performance outcomes for various training set sizes.

Table 3.1: AUC scores

LIST OF ABBREVIATIONS

CRM : Customer Relationship Management

PM: Predictive Model

TP: True Positive

FP: False Positive

TN: True Negative

FN: False Negative

CHAPTER 1

INTRODUCTION

In the customer relationship management (CRM), an automated techniques for anticipating future clients using recommendation engines. Basic equivalence observations are the only current prediction approaches. More advanced suggestion strategies for discovering new prospects are desired by the rapidly growing of businesses. Complex recommendation algorithms are already in use in other industries, such as music and movie entertainment suppliers. These high-quality recommendations are the result of the widespread use of huge data in conjunction with ML algorithms. As more and more data in the CRM becomes available business, other opportunities for extracting deep insights from the amassed data emerge. Most CRM system users will profit from such learning strategies, as they will be able to make more accurate decisions.

The goal is to create an appropriate algorithm that takes into account the following factors:

- Which criteria should be utilised to assess an algorithm's suitability?
- Which are the appropriate algos for this problem?
- Which applicable algos provides the best results? Lundalogik AB was the site of this master's thesis project.

1.1 Customer Relationship Management (CRM)

Customer Relationship Management (CRM) is a system for tracking and managing interactions between a company's present and potential customers. Businesses need to better connect their products and services with their customers' needs in an increasingly competitive business climate where customer experience is becoming increasingly crucial. To combat this,

companies have increased their emphasis on their users via evaluating the user's point of view and more intelligently managing user's information. CRM can be viewed as a user's response to a more extractable user base, as it collects, refines data on each uses.

1.2 Recommendation Engine

There is a desire for programed techniques which may be employed for user prosecuting till dates, growing CRM demand. There are currently functions for locating "twins," or potential consumers who are similar to current customers, as well as searching via lists of clients divided into classes like areas and markets. In the not-too-distant future, system conclusion will allow systems to automatically recommend prospects with a high likelihood of gracefully profitable users. It is required for such systems to make predictions about the businesses that are important. It can automatically generate fresh consumer suggestions based on data linked with existing clients. Data can be evaluated for similarities that characterise existing clients in order to find fresh suggestions. These parallels can then be exploited.

1.3 Machine Learning

It is a field of study that describes algorithms for making observations and predictions about data. supervised learners, which are usedd to educate PM for grouping tasks, and unsupervised learners, which are usedd to educate circumstantial models for grouping taskss, are the two basic types of ML algorithms.

1.3.1 Supervised learning

Litigation of teaching a prediction model to estimate a likely point from a pool of pre-determined goal values is known as supervised learning. A PM learns to anticipate one value by predicting another modelling the connection amongst the target feature (the feature to be forecasted) and other characteristics in the data set using other values. Because it is given explicit instructions on what and how to educate it, the educating of a PM is known as supervised learning . One of the most typical supervised ML issues is PM which class an exemplar fit to. It can be perceived as classification, the model used to do a particular assignment is known as a classifier. To summarize Supervised learnings that can be in the following steps:

1. Educating - use the labelled training set to train the model.
2. Valuation (optional): fine-tune the model's parameters.
3. Testing - compare the model's performance to the test set.
4. Application - use real-world data to test the model.

1.3.2 Unsupervised learning

Unlike PM, which forecast a certain trait, descriptive models do not place a premium on any particular feature. The litigation of educating a descriptive model is referred to as unsupervised because there is no aim to learn (Lantz, 2013). It aims to group data into same categories in order to summarise it. Clustering or cluster analysis is the term for this. From a pool of pre-determined goal values, which identifies grouse in the data set that are usually related in some way.

1.4 Data Requirements

All ML is dependent on analysing surviving data, and when a learner gets access to huge amounts of data, it frequently performs better. With more samples, a model will be able to draw statistical assumptions about the data's general properties more easily. The information utilised in this project comes from a vast database of Swedish enterprises. This business data comprises information about the location of the company, its lines of operation, the number of workers, and its economic properties. Some of the variables, such as location and line of business, have discrete values, whereas the majority of the economical quirk have consistent array of values.

1.5 Previous Work

Using a content-based filtering technique, demonstrate various ways for suggesting things. The usefulness of ordinary algorithms such as k-nearest neighbours for classification jobs is examined. The findings suggest that if the information is sufficient enough to identify acceptable from unwanted objects, appropriate recommendations may be made. Ungar and Foster (1998) investigate the idea of employing standard algorithms for collaborative filtering, such as k-means clustering. Individual users' movie and music histories are used to do cluster analysis on a data set. Clustering with k-means is said to be difficult since the information is too sparse to create usefulness with distinguishing features, implying a reliance on excellent data to produce effective suggestions.

CHAPTER 2

METHODOLOGY

2.1 Identifying prospective new clients

Here evaluation's major purpose is to find a viable way for automatically discovering new prospects from corporate data. Because existing clients will be used to generate new prospects., the focus should be on developing a predictive model that can recognise typical traits in the data it is trained on and utilise that knowledge to anticipate future consumers with comparable characteristics.

2.2 Examining Existing Algorithms

Because there is already a huge number of well-documented ML algorithms, the assessment will focus on comparing the achievement of a few of them. The selection of algorithms must be based on a thorough examination of their applicability for the task at hand. The algorithms used should also be widely used and obtainable from recognised machine-learning service providers, e.g PredictionIO.

2.3 Evaluation Method

The first step in evaluating achievement amongst learner models is to choose an evaluation technique. This section will go through some typical statistics for evaluating machine learning achievement. When dealing with information retrieval, these numbers will array from ordinary measurements of model correctness to measures of particular model properties, such as the proportion of relevant occurrences. Because computing power is typically not a problem with contemporary technology, achievement in terms of processing usage is deemed second-rate. What matters most is a learner's ability to learn and correctly recognise relevant situations. When applied to enormous volumes of data.

2.3.1 Confusion matrix

It is a table that categorises guesses based on how closely it matches the data's correct value. The potential categories of anticipated values are represented by one of the table's measurements, while the actual values are represented by the other. It could be used to categorise the predictions of a model that predicts many target values, although it is most commonly used to categorise binary predictions, which are represented by the 2X2 confusion matrix. Positive classes refer to the predicted outcomes that are of interest, whereas negative classes refer to the predicted outcomes that are not of interest. The Figure 2.1 illustrating the four classes depicts the link between positive and negative class expectations.

True positive (TP): An outcome of interest that has been correctly categorised.

True negative (TN): Identified as a result of lack interest.

False positive (FP): An result of interest is incorrectly categorised.

False negative (FN): An outcome that was incorrectly rated as uninteresting.

	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Table 1.1: Confusion Matrix.

2.3.2 Accuracy and error rate

Variety of machine learning achievements which can be established for specialised uses. The most basic is accuracy, which describes the success rate of a forecast. It's determined as the percentage of right classifications out of all the classifications made:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

The acronyms TP, TN, FP, and FN relate to the number of times each of these categories was represented in the predictions.

The error rate, which describes the proportion of wrong classifications, is the polar opposite of the success rate. It's calculated by dividing the number of wrong categories by the total number of classifications, or just: 1 precision

$$error\ rate = \frac{FP + FN}{TP + TN + FP + FN} = 1 - accuracy.$$

It is basic indicators of a model's overall performance. When used alone, though, it can be deceiving. If the proportion of achievable target values in the test set is minimal, say 10 percent, a model that, for example, classifies all occurrences as negative would nevertheless have an accuracy of 90 percent since 90 percent of the instances are properly categorised as

negatives.

2.3.3 Precision and recall

Recall, accuracy are the performance metrics which are commonly employed in the field of information retrieval. Both of these metrics are intended to indicate how intriguing and useful a model's outcomes are. The fraction of positive classifications that are actually positive is described as precision . To put it another way, how often is a positive forecast correct? A high level of accuracy might indicate that a model is reliable. A precise model would equate to, for example, a high degree of linked results. A search engine with an erroneous model would return a large number of peripheral results.

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

Instead, "recall" refers to the ability to remember something of how thorough the the outcomes. It calculates the percentage of correctly categorised positive case studies. A model with a high recall will properly categorise a large percentage of positive events. There is no certainty that this will not happen be a significant number of false positives. This is a search engine with a elated recall model.

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

Precision and recall go hand in hand, connected measurements, and there is a rational compromise between a high and a low value for one. It's simple to be exact by just categorising the most evident cases, it's simple for a model to attain a high recall by classifying instances extremely optimistically. However, creating a model with both high accuracy and good recall is challenging. In making decisions, it's common to strike a balance between being cautious and being excessively aggressive. The F-measure is a number that combines accuracy and recall are clubed into a number, using the harmonic mean, the F-measure combines accuracy and recall.

$$F - measure = \frac{2 \times precision \times recall}{recall + precision} = \frac{2 \times TP}{2 \times TP + FP + FN}$$

2.3.4 Bias-Variance trade-off

It compromise is a symptom of the larger bias-variance compromise problem. The bias-variance trade-off impacts all supervise ml work and reflects some references of fault that restrict learners from generalizing outside their training set. There will always be an error if a model is too basic in comparison to the the intricacy of Bayes model1 because it is easy to encrust all occurrences of the trainingg set. Bias refers to the model's lack of complexity (Domingos, 2000). For example, as shown in Figure 2.2, an excessively basic model would eventually fail to notice tiny differences in the training data.

If a model is extremely complicated, on the other hand, it may attain a on the practise set, nearly flawless match. It will, however, the model will typically overfit if it achieves a like on the exercised set. The model will have mistakes even if there is no noise because it is extremely complicated. Variance refers to the exaggerated complexion of the trainingg set in comparison to the complexion of the trainingg set. Figure 2.3 depicts the comparison in the regression problem. A model with a lot of variation shall be highly sensitive to slight changes in the training data, which might lead to problems.

2.3.5 Visualizing performance trade-offs

Visualisations are commonly used to improve human cognition. They also show how to compare ML models . The Receiver Operating Characteristic (ROC) graph may be used to highlight how true positive and false positive detection are traded off. This can be a good predictor of the overall efficacy of a ML model. The ROC graph is a two-dimensional space with the the percentage of genuine positives on one axis

$$TPR = \frac{TP}{TP + FN} = \frac{TP}{T}$$

The False Positive Rate (FPR) is calculated by dividing the total number of negatives by the number of false positives (FPN). FPR is also known as cost or fall-out.

$$FPR = \frac{FP}{FP + TN} = \frac{FP}{N}$$

A distinct classified model which produces a purposed featured label (rather than a probabilistic classifier that produces a possibility) would generate a exclusive magnitude of TPR and FPR that corresponds to a instant on the ROC graph. Some inferences may be formed based on the location of a classifier's corresponding point in the graph. Classifiers on the left-hand side of the ROC graph are conservative because they only produce positive classifications when there is solid evidence, resulting in a low number of false positives. A conventional classifier's disadvantage is that it has a decreased genuine positive rate.

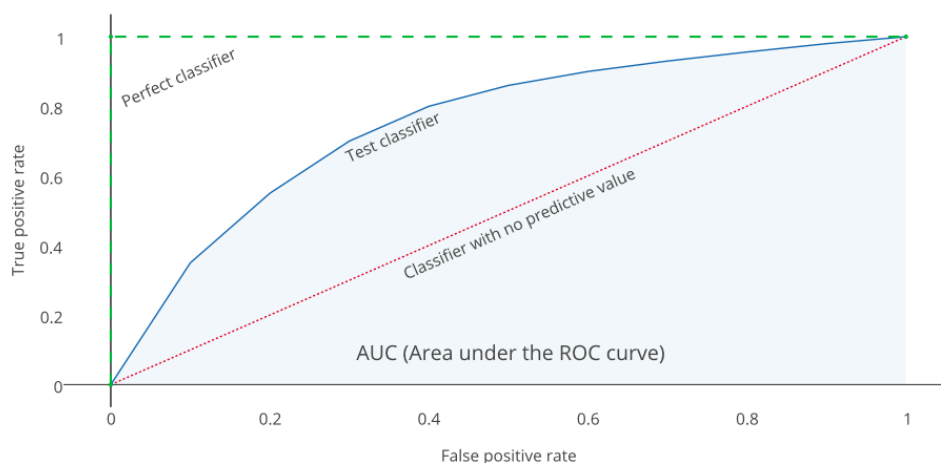


Figure 1.1: ROC curve and AUC

A probabilistic classifier is a classifier that produces a possibility or a score that represents the likelihood that an occurrence is judged positive. By tallying the output to a threshold, such a classifier may simply be converted to a discrete classifier. The instance is considered positive if the output exceeds the threshold. When the threshold

is changed, a different percentage of the events in the test set are categorised as positives. Using different threshold values. By linking the points generated, a ROC curve may be created. By using ensemble learning methods² and probabilistic classifiers, a discrete classifier may be transformed into a probabilistic one.

The AUC is a statistic that may be used to show how well a ROC curve performs in general. It measures the overall area under the ROC curve by treating the ROC graph as a two-dimensional square. The AUC will vary from 0.5 to 1.0 (perfect classifier). There is also a method for interpreting AUC scores that uses classes that are equivalent to academic letter grades (see Lantz, 2013, chap. 10). A range of 0.9–1.0 (outstanding) B (excellent/good) 0.8–0.9 C (acceptable/fair) 0.7–0.8 D = 0.6–0.7 (poor) F = 0.5–0.6

2.4 Implementation

Inevitably, implementing code took up a significant portion of the algorithm evaluation process. Existing software and libraries were used to help speed up the implementation process. Python and C++ statistical, mathematical, and scientific computing libraries, respectively. Multiple implementations were used to test some of the algorithms; however, because all of the algorithms were straight forward to construct, bespoke C++ implementations were created for the sake of parallelization and, last but not least, fun.

CHAPTER 3

EVALUATION OF ALGORITHMS

3.1 Data Preparation

The data had to be prepared before the algorithms could be examined. The preparation's purpose was to provide a broad variety of exclusive training and test sets for the algorithms' assessment. The data had to be prepared before the algorithms could be examined. The preparation's purpose was to provide a broad variety of high-quality training and testt sets for the algorithms' assessment.

3.1.1 Labeling

There must be a means to discriminate between desired and bad outcomes. A difference was drawn between customers and noncustomers in order to construct a concept of positivity and negativity among examples in a set. This difference is necessary in order to train a classifier to recognize all consumers in a set of data. As a result, a positive occurrence was associated with consumers, whereas a negative value was associated with non-customers. Because the firm database holds details about current customers for each records, all occurrences in a training set can be labeled as positive or negative, based on their user statuses.

3.1.2 Continuous attributes

Characteristics have many values ranging from negative infinity to positive infinity. This is problematic because the number of attribute values that the models must account for is theoretically infinite, despite the fact that most ML models only require a small number of

them. For example, it is acceptable to infer that the numbers 5,000 and 5,001 are comparable given a range of integers 1–100,000, but this is not true for a ML model that utilises typical equality as the measure of similarity. The numbers 5,000 and 5,001 are very different from each other.

Entropy-based partitioning separates the range of continuous values into two divisions by splitting the sorted range at a threshold value. The threshold value is found after an exhaustive search for the largest gain score over the attribute values (gain is a measure of gained information, as described in section 3.4.1). To put it another way, the utmost threshold is determined by looping through the sorted values . The final value is chosen as the threshold that optimises the cumulative values of the divisions. The improved discretization technique is used to investigate the distribution.

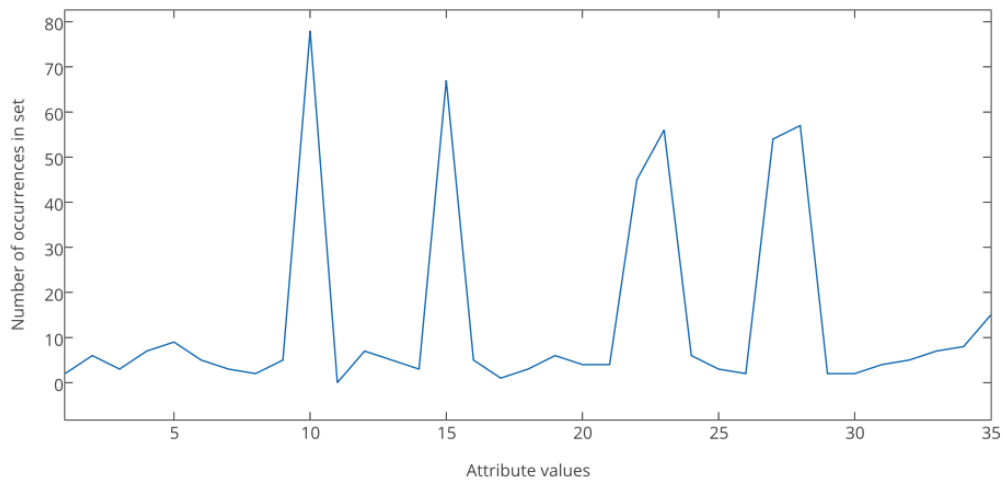


Figure 2.1: The attribute value distribution among existing customers.

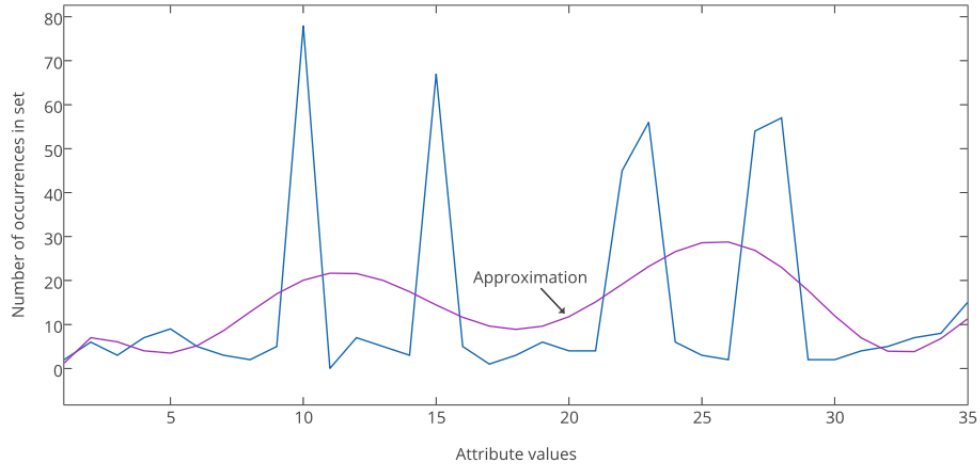


Figure 2.2: The distribution's least squares approximation.

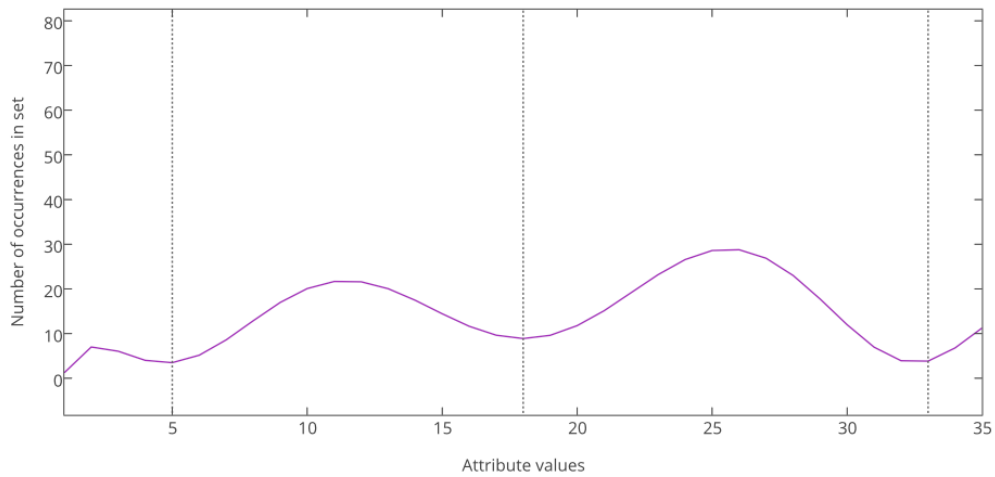


Figure 2.3: Splitting the distribution at the approximation's local minimum results in four parts.

When combined with all the algorithms studied here, the second technique performed slightly better than Entropy-based partitioning. Furthermore, the second technique was far faster, finishing the discretization process in seconds rather than the several minutes required by Entropy-based partitioning, which is computationally costly due to the large number of iterations required for its thorough search for maximum gain.

3.2 Frequent Set Counting using Apriori

Frequent Collection Counting (FSC) is a formula for detecting repeatedly recurring subsets of values in a data set that is typically used as an learning method. The approach is analogous to n-gram removal determines the statistical likelihood of a given order of n words by calculating the n consecutive word frequency in a corpus 1. Every available pair in the data is used to build all conceivable combinations of n attribute values, i.e., the power set², and then all occurrences are tallied. The approach can be improved by using FSC on labelled data.

Company	Location	Revenue (SEK)	# of employees	Label
A	Stockholm	2 mn	300	Customer
B	Copenhagen	2 mn	100	Non-Customer
C	Helsinki	1 mn	100	Non-Customer
D	Stockholm	2 mn	300	Customer
E	Copenhagen	1 mn	200	Non-Customer
F	Stockholm	1 mn	300	Customer

Table 2.1: Labeled company training set

Table 3.1 depicts a training set of firms identified as customers or noncustomers of a fictional corporation attempting to detect new customers using frequent set counting. Firms that are branded as customers are considered good examples, whereas companies that are not labelled as customers are considered bad examples. The most common n-itemsets among the three positive examples are shown in Table 3.2. The brink is set to 0.05, which means that the table only displays the n-itemsets that appear in at least 50 percent of the positive cases. Labeled company training set occurrences.

Using this data, it can be anticipated that firms with 300 workers in Stockholm are more preferably to suit the current consumer profile and, as a result, should be classed as good examples. If the criterion is adjusted to a lower number, companies in Stockholm with 300 workers and a sales of SEK 2 million might also be considered positive examples. It's worth noting that accessing 1-itemsets will produce in a model with a lot of bias, since if a classifier

classified all firms in Stockholm as positive occurrences, there would be a lot of irrelevant positives (underfitting). Using simply the largest, on the other hand,

n-itemset	n	Occurrence
{Stockholm, 300}	2	3 of 3
{Stockholm}	1	3 of 3
{300}	1	3 of 3
{Stockholm, 2 mn, 300}	3	2 of 3
{Stockholm, 2 mn}	2	2 of 3
{2 mn, 300}	2	2 of 3
{2 mn}	1	2 of 3

Table 2.2: List of the most common n-itemsets among organisations designated as clients.

The n-itemset convenient will almost certainly result in a small number of positive classifications, albeit the majority, if not all, will be meaningful. Furthermore, searching the accurate organisations identified as current users in the training set is useless if the goal is to identify new clients. As a result, selecting a n number that is too large will not generalise well to various sets of information and will result in overfitting.

3.2.1 Implementation

The FSC learner model entails finding frequently recurring nitemsets and storing them for future categorization purposes. To develop the technique for finding the frequent itemsets. A feature is a pair of attributes and values that describes a value associated with an attribute. This is necessary because values must be distinct across attributes; for example, the value "3" will have various meanings depending on whether it is detected inside the two separate characteristics workers or revenue. As seen in Table 3.3, the set given in Table 3.1 has eight different properties.

Feature	# Attribute	Value
1	Location	Stockholm
2	Location	Copenhagen
3	Location	Helsinki
4	Revenue	1 mn
5	Revenue	2 mn

Table 2.3: All of the characteristics are present in the good cases.

The goal of adopting a matrix sparse representation is to obtain an organised collection of all features, reduce storage space, and provide a data format that is concise in order to benefit from the locality of reference³. Because all traits may be recorded in a binary matrix rather than the features themselves, the sparse matrix saves storage space. Table 3.4 shows an illustration of a matrix-sparse representation

Company	Feature 1	Feature 4	Feature 5	Feature 8
A	X		X	X
D	X		X	X
F	X	X		X

Table 2.4: Sparse matrix representation of positive cases in the training set

The next step is to generate supersets from all of the combinations of all features in the training set to create the power set of the set of features. The supersets are built from the bottom up, with 1-itemsets being constructed initially being created from all conceivable combinations of 1-itemset combinations, and so on till everything is finished possible n-itemset combinations have been produced. When the number of features is huge, however, recognising all combinations of itemsets of characteristics using a brute force technique becomes quite costly in terms of both time and space. The total amount of supersets that may be constructed from the information set with k distinct characteristics is 2^k .

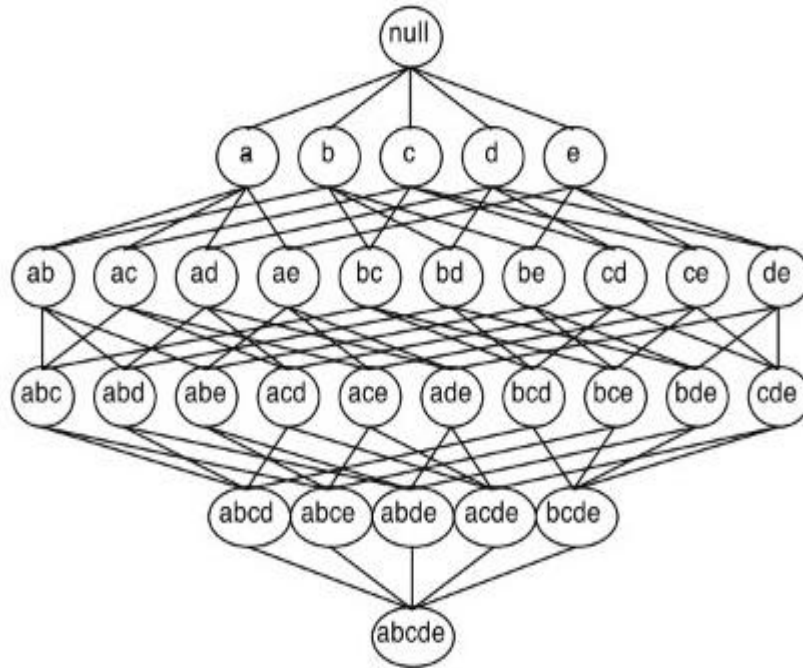


Figure 2.4: The lattice structure depicts the 31 potential itemsets resulting from a data collection with 5 characteristics.

The Apriori principle states:

“If an itemset is frequent, then all of its subsets must also be frequent.”

3.2.2 Performance evaluation

When doing classification with huge itemsets, our testing demonstrate that the model has a high level of accuracy. Identifying bigger itemsets during training, on the other hand, necessitates the use of a low threshold value, which results in higher running time. Table 3.5 shows the measured running time for various threshold settings. The running duration appears to be inverselyexponential in proportion to the cut-off value, as seen in the graphs. Figure 3.5.3 shows the relationship .

Threshold	Running time (s)	Max set size
0.85	1	2
0.8	1	4
0.77	2	6
0.76	7	6
0.75	35	7
0.74	202	7

Table 2.5: Training time for different threshold levels.

As our findings in Table 3.5 reveal, the size of the itemsets discovered during training is also related to the threshold value used. Lower threshold settings allow the model to recognise bigger frequent itemsets, but at the expense of longer run durations. As a result, identifying itemsets with more than about 7 items becomes impracticable.

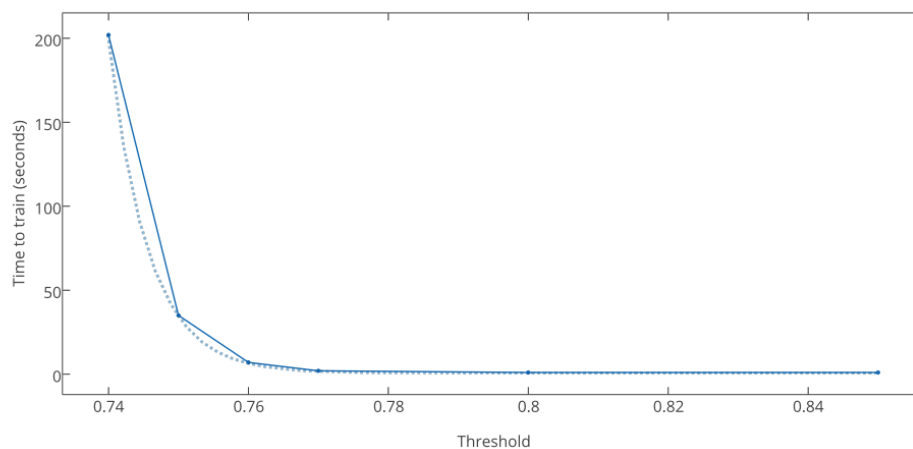


Figure 3.5: Training time for various threshold values

For varied performance results whilst classification, multiple gaps of itemset sizes might be employed. Using a larger array of itemset sizes results in a bigger bias. A

classification interval of 1 to 7 means that all sizes 1 through 7 are used. The performance results are shown in Table 3.6. In the tests, a threshold value of 0.75 was employed. The findings show that just using larger itemset sizes resulted in improved accuracy. Nevertheless comes at the cost of worse ratings. Furthermore, at all defined intervals, the accuracy score is continuously low, never exceeding 0.01. (See Figure 3.7 for further information.)

Set interval	Accuracy	Precision	Recall	F-measure
1-7	0.004	0.002	1.000	0.004
2-7	0.012	0.002	0.999	0.004
3-7	0.105	0.002	0.995	0.004
4-7	0.562	0.004	0.949	0.008
5-7	0.832	0.010	0.867	0.019
6-7	0.838	0.010	0.823	0.019

Table 2.6: FSC performance using a threshold of 0.75

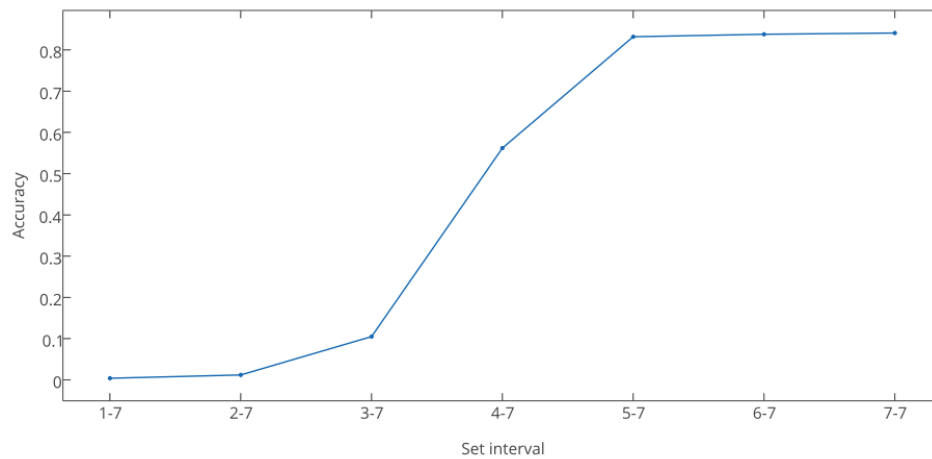


Figure 2.6: As the itemset interval is chosen more conservatively, accuracy improves.

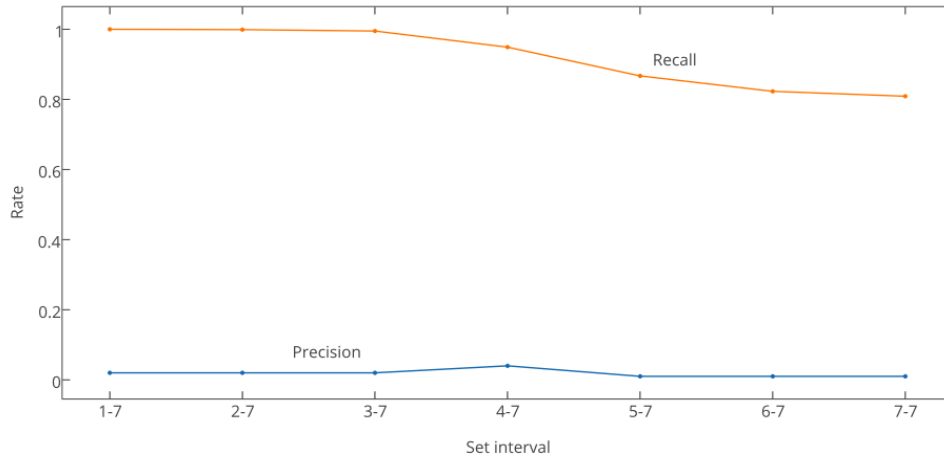


Figure 2.7: When the fixed interval is restricted to only the bigger sizes, recall suffers.

3.3 k-Nearest Neighbors

It is one of the earliest, most basic pattern categorization methods. In spite of this, when compared to other algorithms, it frequently produces good results in a variety of disciplines. Because it waits generalisation on the educating data until the model is employed for categoring, k-NN is regarded as a lazy learner (Lantz, 2013). An enthusiastic learner, on the other hand, does data generalisation throughout the training phase. The approach is typically accurate, although it is computationally costly in terms of both time and space. Unlabeled instances are classified using the majority label of their knearest neighbours in the training set. Instances are identified to be close by using a distance measure that may be calculated.

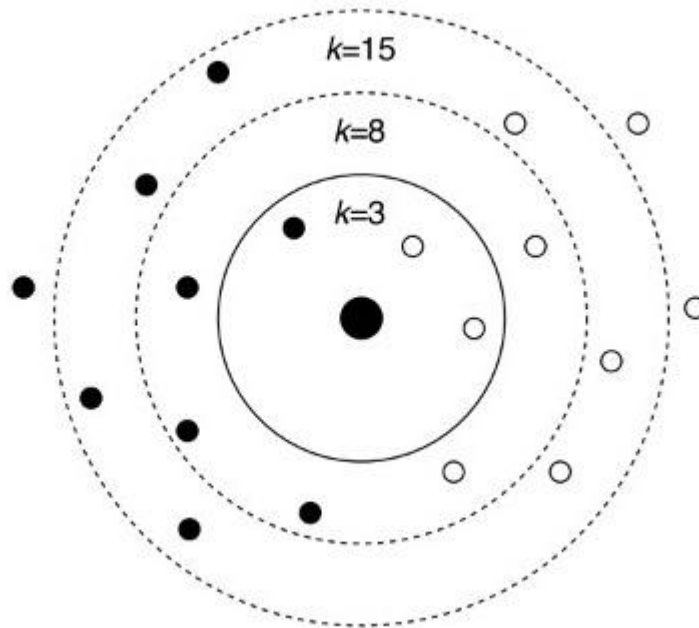


Figure 2.8:Two-dimensional representation of the k-nearest neighbours categorization.

accordingly, searching the k-nearest neighbours and space. The value of k is set by the user. The properties of the data have a role in determining an appropriate value for k. Smaller k values may provide more accurate results since nearby neighbours are more likely to be similar, but they will also make the learner more delicate to disturbance in the information. The amateur will be inferior reliant on noise as k grows higher, but when k appeal to the whole number of neighbours in the training set, the outcome will lose predictive power. The unlabeled instance to categorise is the large black centrepiece. In the training set, black and white dots indicate cases with various labels, as shown in fig 3.8.

3.3.1 Implementation

There are a variety of methods for determining the gap among instances, for continuous characteristics, Euclidean distance is commonly used, while for discrete attributes, Hamming distance is commonly used. These approaches are generally sufficient, although

employing more complex distance measurements can occasionally improve model accuracy dramatically (Weinberger and Saul, 2009)

For two strings of identical length, the Hamming distance may be computed by counting the number of characters that vary between them (Hamming, 1950). The Hamming distance between "paper" , "vapour," for example, is 2. The Levenshtein distance (or edit distance) is an extension of the Hamming distance metric that may be used for strings of different length (Levenshtein, 1966). The smallest number of single-character insertions required for two strings to become identical is known as the Levenshtein distance. To prevent the inefficiencies of recalculating the distances of the same substrings several times, a dynamic programming solution can be utilised.

3.3.2 Performance evaluation

The k-nearest neighbours approach is computationally costly when used to huge volumes of data, according to our tests. Only a tiny portion of the training set data would be used for testing because to the time constraints. Table 3.7 demonstrates the evaluation implementation's measured performance on the experimental system for various training set sizes. The training sets employed were evenly distributed, with 30% positive and 70% negative examples. The categorization was carried out on the entire set of test data, which included 2,327,923 unabled cases. As the shape of instances in the training set grows, running time appears to rise linearly. It is possible to approximate a linear fit on the data by using a linear fit approximation.

#	Running time (s)	Accuracy	Precision	Recall	F-measure
22	72	0.83	0.01	0.86	0.019
83	241	0.88	0.009	0.54	0.017
167	478	0.90	0.012	0.68	0.024
250	712	0.86	0.012	0.75	0.024
333	946	0.88	0.012	0.75	0.023

Table 2.7: Performance outcomes for various training set sizes.

The findings in Table 3.7 demonstrate that even for the smallest training set, accuracy is greater than 80%, while precision is only approximately 1%. The modest training set sizes employed reflect this, since the model has a difficult time identifying the relevance of facts with such a limited quantity of training data.

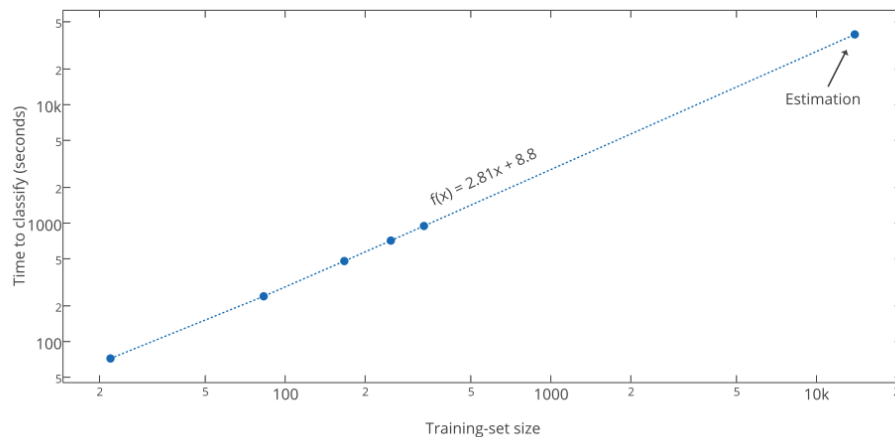


Figure 2.9: Measured running time on the test machine with varying training-set sizes

However, when employing a smaller test set, the entire training set might be utilised to evaluate performance. The results of this method's classification performance are shown in Table 3.8. The test set had roughly 14,000 unlabeled examples, while the reference training set contained 13,933 labelled instances (30 percent positive occurrences). The performance metrics for various k values existing. For small values of k , the report claims that the model is

accurate, with accuracy reaching approximately 0.9

3.4 Decision Tree Induction using C4.5

For supervised learning problems, d-tree induction is a ML approach. The technique focuses on using if-then rules to represent the links between inputs and outcomes. For satisfying the criterion of approach for data extraction, decision tree educating is placed at the highest position of the list of multiple categorization learning methods. It's quick, provides models that people can understand, is impregnable to inappropriate variables, exempt to outliers⁴, and could be quickly adapted to work with a variety of data formats, including distinct classes and array of time data. When employed on real data, decision tree models frequently have low dispersion & high difference and they may endure from poor generalisation.

3.4.1 Implementation

The algorithm C4.5 was utilised to generate a decent decision tree throughout the induction phase (Quinlan, 1993). C4.5 is a modification of Ross Quinlan's original ID3 algorithm. It outperforms ID3 in terms of handling both continuous, and having some built-in pruning capabilities. Some expansion, is available, it is only for blurb usage and was not investigated in this project. This technique has been found to build basic enough decision trees with reasonable achievement in most cases, although it cannot ensure that superior trees have not been neglected (Quinlan, 1986). This was not an issue, however, when it came to identifying organisations as customers or non-customers.

3.4.2 Performance evaluation

It demonstrates that classification can be done with great accuracy on huge datasets. Also feasible to attain more accuracy , but because there is a compromise between the two, achieving a high value for both measures at the same time is difficult. A thorough description

of the performance outcomes may be found in Table 3.10. The reference training set was used to generate these findings, which had a varied proportion of positive cases. When the model was trained with a training set of 0.9 positive cases, the lowest exactness of 76 percent was attained. When the model was educated on the data set of labelled occurrences, the top score of 99.9 percent was reached

As the number of positive examples rises, the accuracy measurements show a downward trend. The link between the frequency of positives and the precision is seen in Figure 3.15. Because More metrics are in the lower range, the balanced scale in the image is logarithmic to better depict the fluctuations.

Our studies demonstrate that the accuracy score rises as the model's complexity rises due to the use of a bigger training set. As the fraction of positives rises, accuracy increases (see Figure 3.16). As can be seen in the same graph, there appears to be a sharp gain in accuracy with positive rates below 10%. As expected by the trade-off effect, recall score appears to follow an inverse trend. The growing trend is seen in Figure 3.17 as the fraction of positive examples grows. As with accuracy, a threshold of roughly 10% appears to exist, producing a rapid drop at rates below this level. The apparent negative link between accuracy and recall suggests that here is a compromise among growing more precise and obtaining more recall in practise.

3.5 k-Means Clustering

It is a simple and fast approach which has been around for a long time (MacQueen, 1967). The approach is used for data cluster analysis and belongs to the unsupervised learning area. There are a few variants on the fundamental method, but the most prevalent one employs a geometric clustering technique to repeatedly enhance the findings (Arthur and Vassilvitskii, 2006). Lloyd's work is the foundation for the common algorithm (1982). It accomplishes this by describing data points (instances) in a feature space as vectors. The algorithm then allocates the value iteratively.

The k means are created at random locations in the feature space to begin the initialization. The means aren't Essentially, the centroids or means of the individual clusters at

this time. Following that, the assignment phase assigns all data points with the shortest determined gap proceeding. Lastly, the modification phase transfers all of the points to the cluster's centroid.

3.5.1 Implementation

Data points were rephrased as array representing the existence of characteristics in the point of implementation that was assessed in this project. To put it another way, each data point is made up of a binary vector of 0s and 1s, with 1 denoting the presence of a feature. When computing the means, this representation allows all attributes to be considered equally. The feature space may be modelled as a d-dimensional hypercube, with all data points clustered in the hypercube's corners.

3.5.2 Performance evaluation

Clustering using k-means is a data clustering technique, and as such, it cannot be utilised as well as the other methods for classification considered in this project. However, we choose to use it may be used to extract information from test data. Because there is no idea of a preset outcome, unsupervised learning algorithms cannot be assessed in the same way that supervised learners can. However, the algorithm was put to the test in a labelled training set to see if it could distinguish between positive and negative cases, with $k = 2$ on training sets with varied proportions of positives was used to carry out the testing process. Clusters emerge as a result.

CHAPTER 4

DISCUSSION

4.1 Performance Comparison

4.1.1 Frequent set counting using Apriori

One disadvantage of employing The model features an FSC for supervised learning. limited potential for providing sophisticated data insight. It cannot make logical assumptions, for example, if two itemsets that are regularly encountered clash, but have disjoint subsets, it cannot make the logic assumption that an The training set contains a lot of objects that are made up of the intersection and one of the disjoint subsets. If the model had been able to grasp the notion of an itemset, it would have been a lot better.

4.1.2 k-Nearest neighbors

When used to tiny pieces of data, k-Nearest Neighbors works admirably. When utilising modest amounts of k, accuracy is somewhat lower than C4.5. In comparison to C4.5, the model suffers less from the bias-variance compromise and retains good accuracy and recall throughout all tests. As a result, the model is predicted to produce accurate and relevant findings. When bigger quantities of data are employed, however, testing demonstrate that the model is computationally expensive in terms of time. When doing classification on the whole data set utilised in the assessment, plan to spend many hours on it. Smaller training sets can be employed as an option to reduce the model's running time, however our testing suggest that this strategy does not work.

4.1.3 C4.5 decision tree

Our findings demonstrate that labelling the full test set and utilising it to train the model can obtain a precision of 98 percent. This will result in a high level of relevance, despite the

small number of people involved. Precision can also be sacrificed for recall by adjusting the fraction of positive cases in the training set, which may be desired in certain scenarios. Because decision trees are prone to reaching large variation, they may not generalise well to diverse types of data (Geurts, 2002). As a result, when the model is too sophisticated, classification on real data may not give enough positive categorization. However, by utilising the observed trade-off effect, the model's complexity may be reduced.

4.1.4 k-Means clustering

This method differs from others in that it is utilised for cluster analysis rather than classification. As a result, comparing it to other algorithms is difficult. k-Means clustering, on the other hand, is a high-performing method in its own right. Our experiments indicate that the approach can successfully build distinct data clusters. It would have been fascinating to test the method with other values of k, however it turns out that defining a desired outcome for such a test is difficult. If the clusters created are meaningful or not without a preset outcome. Internal variance inside a cluster may be determined, although this metric simply tells how homogenous the cluster is..

4.1.5 ROC graph

The ROC curve in Figure 4.1 shows all three classification techniques side by side. The curves on the graph, but rather a sequence of linked points in ROC space, displayed for each method. It's because all three classifiers provide discrete classifications rather than numeric scores or probabilities, which are necessary for a valid ROC curve to be produced. As a result, the curves should not be seen as a comparison, but rather as a representation of the relative performance of distinct measurements of specific algorithms. For example, the k-NN algorithm's curve demonstrates that this model is quite conservative for all measurements

The ideal points can be compared, even if the curves should not be compared. According to the ROC, the ideal data on the graph are the measurements that are closest to the top left corner of the graph, and thus would give better accuracy. The ideal points may be compared amongst the models since they reflect the best performance measurement obtained throughout

the assessments of the various models.

The FSC curve was constructed by altering the itemset size utilised during classification to provide different measurements of the method. Different measurements for differing values of k were created for k-NN. It should be mentioned that the classification results for k-NN were assessed using only a portion of the whole test set because doing classification on the entire test set would have been too time consuming. Because others assessed using the whole test set, the comparison between models may not be totally fair. Training sets with various proportions of positives were used to create the C4.5 curve.

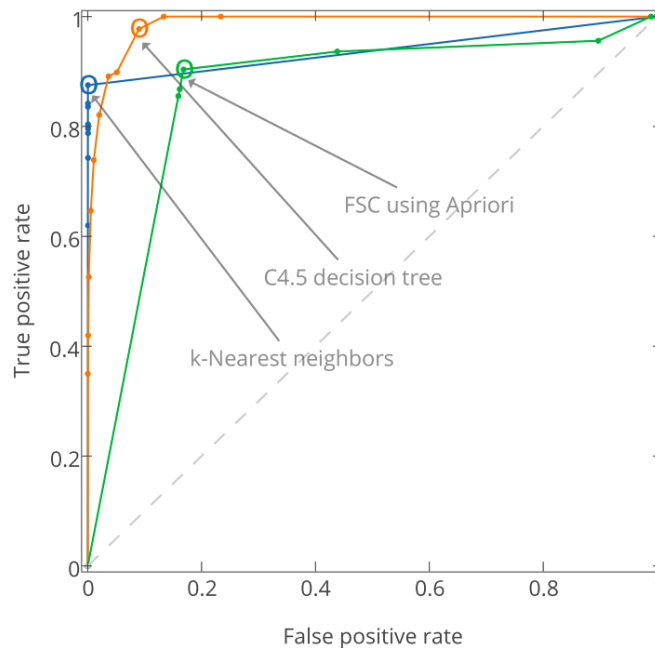


Figure 3.1: In ROC space, the supervised learning techniques are displayed.

C4.5	0.987	A (outstanding)
k- NN	0.938	A (outstanding)
FSC	0.860	B (excellent/good)

Table 4.1: AUC scores

CHAPTER 7

CONCLUSION

The findings of this experiment indicate that useful consumer predictions may be created using standard algorithms, despite the fact that most of the most recent suggestion engines are constructed utilising machine-learning approaches. Four typical off-the-shelf machine-learning algorithms were examined in this thesis. Three of the methods belong to the supervised learning domain, while one belongs to the unsupervised learning area. Using real-world data, the algorithms that belong to the supervised learners were evaluated in terms of accuracy, relevancy of outcomes, and run time. The capacity of the unsupervised learner to produce recognisable clusters was examined.

5.1 Algorithm for recommendation engine

Technique of discovering new potential clients, Models for supervised learning have a lot of promise. To classify appropriate prospects, a classifier algorithm may be trained using a collection of existing customers and then applied to a wide number of different organisations. The FSC method has considerable potential in terms of accuracy, but it suffers from a lack of complexity, resulting in low-relevance matches. The biggest disadvantage of FSC, according to tests, is that it has a strong bias, resulting in an obvious underfit when applied to huge quantities of data. FSC isn't considered a full-fledged machinelearning technique, but its simplicity and obvious usage in data mining make it a good place to start if you're new to the field.

5.2 Future work

The k-NN learner has the advantage of being parallelizable. The method may be beneficial in reality if this trait could be adequately utilised. Current GPU-based approaches for massively parallel processing or distributed cloud computing might be viable options. There is a lot that can be done to improve the performance of decision trees. High variance

can be reduced by applying tree pruning techniques that aim to simplify the tree's structure. Other approaches include ensemble learning, which involves training many decision trees in parallel and using tree majority vote to classify them. Finally, there are a number of different machine-learning algorithms to choose from. more experienced.

REFERENCE

- [1] Forgy, E. W. (1965). Cluster analysis of multivariate data: Efficiency versus interpretability of classification. *Biometrics*, 21(3):768--769.
- [2] Freund, Y. and Schapire, R. E. (1999). A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771--780
- [3] Geurts, P. (2002). Contributions to Decision Tree Induction: Bias/Variance Tradeoff and Time Series Classification. Doctoral thesis, University of Liege Belgium.
- [4] Hamming, R. W. (1950). Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147--160.
- [5] Hastie, T., Tibshirani, R., and Friedman, J. (2013). *The Elements of Statistical Learning*. Springer Science and Business Media.
- [6] Hunt, E. B., Marin, J., and Stone, P. J. (1966). Experiments in induction. Technical report, University of Michigan.
- [7] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer Science & Business Media
- [8] Kostojohn, S., Johnson, M., and Paulen, B. (2011). *CRM Fundamentals*. Apress.
- [9] Lantz, B. (2013). *Machine Learning with R*. Packt Publishing Limited.
- [10] Legendre, A.-M. (1805). Nouvelles méthodes pour la détermination des orbites des comètes [new methods for the determination of the orbits of comets]. Paris: F. Didot.
- [11] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707--710.
- [12] Lloyd, S. P. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129--13
- [13] Lundalogik AB (2015). www.lundalogik.se. <http://www.lundalogik.se>. Accessed: 2015-02-04.
- [14] MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281--297. University of California Press.