School of Computing

FACULTY OF ENGINEERING

UNIVERSITY OF LEEDS

Content based Data Analytics for quality in online reviews

Julia Vasuki Vasudeva

Submitted in accordance with the requirements for the degree of MSc Advanced Computer Science (Data Analytics)

Session 2014/2015

The candidate confirms that the following have been submitted*:*

Items	**Format**	**Recipient(s) and Date**
Deliverables 1,3	*Report*	*SSO (07/09/2015)*
Deliverable 2	*Software codes or URL*	*Supervisor, assessor (07/09/2015)*

Type of Project: _________Empirical Study__________________________

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student) _____________________________

Summary

The importance of online reviews for products and services has significantly increased for customers and manufacturers. Online reviews have become a crucial part of the purchase decision process in particular for customers. Hence, the question arises what determines a good quality review meaning a helpful review for a customer.

The goal of this project is to detect helpful reviews automatically using Natural Language Processing, data-/text mining and machine learning methods. Based on the CRISP-DM methodology an analysis on a corpus of 7,408 reviews about camera and photo products data set and 1,966 reviews about computer and videogames from Amazon.com was undertaken. Feature Engineering as core part resulted in the identification of three main quality criteria: readability, polarisation and informativity for which features have been determined and implemented. Using supervised and unsupervised machine learning algorithms the performance of the algorithms in combination with chosen features are evaluated through experiments. Specific measures and methods, such as cross-validation for classification are used for testing and evaluation.

The results of the experiments show that polarisation features alone do not perform well. Using polarity as control feature by examining positive, neutral and negative reviews in combination with the other features readability or informativity achieve better results. Readability features tend to perform better than informativity features in the present project. Moreover the results depend strongly on the chosen data set (product type) and quality of feature engineering.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Dr Vania Dimitrova for the continuous support of my MSc Project, for her guidance and feedback helped me in all the time of research and writing of this project. Besides my supervisor, I would like to thank my assessor Dr John Stell for his valuable feedback.

A special thanks goes to Alicja Piotrkowicz for her expertise as linguist which were a great benefit for this work.

My sincere thanks also go to my employer Deloitte Consulting GmbH who supported my career plan in doing a Master degree. Without their support it would be not easy to pursue both work and studies simultaneously and gain the precious experiences I made so far.

In addition I would like to thank specially my parents and my sister for supporting me in doing my Master studies in the UK and always believed in me throughout my life. I would not come so far without you. Thank you!

Also, I would like to thank my friends who always supported me and made the time throughout the degree so much more pleasant.

List of Figures

List of Tables

Table of Contents

1 Project Outline

The immense growth of user-generated content (UGC) on e-Commerce platforms (and also on e.g. social media, forums, and blogs) on the World Wide Web has led to a new kind of information source. People tend to share their opinions and experiences on the web, by writing online reviews for example about products and services, such as electronics, books, movies, restaurants, trips or hotels. The importance of online reviews on e-Commerce platforms, such as Amazon, TripAdvisor or Expedia, is increasing significantly, since reviews are predominantly used for evaluating the quality of a product and to support the customer's purchase decisions. The demand for exploiting reviews as knowledge source is immense since they can influence not only the customer's purchase decision but also the product manufacturer's sales performance (Pang & Lee, 2008). Customers use online reviews and ratings to make better decisions. Products or services may have thousands of reviews among customers search for the most helpful to support their purchase decision. *Manufacturers* are able to gain useful insights about the customer's product perception and views on competitors through reviews in order to react proactively to improve sales.

1.1 Problem Statement

Scoping. Key drivers to conduct this project were one the one hand prior experience in the field of visualising social media sentiment analysis as well personal interest in the topic of detecting fraudulent reviews. Fraud in reviews describes manipulation and generation of misleading comments in order to influence the customer's purchase decisions. Research work has been undertaken in this area (Mayzlin, Dover, & Chevalier, 2014). Intentionally manipulated reviews according to (Hu, Bose, Koh, & Liu, 2012) or untruthful reviews (Ott, Choi, Cardie, & Hancock, 2011) are referred to and used as a synonym for fraudulent reviews and discussed in their work. The challenge is to sort fraudulent from genuine customer reviews by focusing on analysing the textual content of the reviews and suspicious reviewer behaviour. Identifying fraudulent reviews is a very difficult task for human readers due to the high number of available reviews and lack of features to detect fraud (Lau, et al., 2011). The problem of detecting fraudulent reviews was introduced by Jindal and Liu in (Jindal & Liu, 2007), (Jindal & Liu, 2008) and will further not be discussed in depth.

The significance of this subject is evident for e-Commerce platforms, such as Amazon. In June 2015, Amazon as one of the leading e-commerce platforms has implemented a new machine learning system to improve the **quality in reviews** for the customer and to combat against fraudulent reviews. The introduced machine learning system will learn over time which product reviews are helpful for the customer and update the top reviews and the star ratings on a product page. Therefore weights will be given to newer reviews, to reviews from verified Amazon purchasers and to reviews from those the customers vote as most helpful product reviews (Marsden, 2015).

Due to the fact that an appropriate data set containing fraudulent reviews is crucial and time and resources were limited to conduct a project in this field, the research shifted towards a different question. Prior to detecting fraud in reviews it is important to understand how quality in review can be identified and automatically detected.

Problem*.* This project focusses on the customer's perspective to automatically detect quality in reviews using machine learning algorithms. It is important that customers can read good quality reviews to make better and faster purchase decisions due to exponential growth of information and products on the web. Consequently, the customer satisfaction increases when customers can read condensed, informative and helpful reviews ranked high on the retailer's web site. For example a current product the "Apple iPhone 6" has on average more than 1000 user-generated reviews on Amazon (Amazon, 2015).The user should to be able for form an opinion after seeing only few reviews, ranked with the most helpful first (Liu, 2012). In contrast the manufacturer has to identify those helpful reviews which are likely to influence the customer and drive sales (Ghose & Ipeirotis, 2007). However in this work the manufacture's perspective is not considered in the scope of this project.

In this work, the assumption is made that a review of "good" quality indicates that the review is helpful for customers. Specific quality criteria support the analysis task to identify if there is a correlation between the level of quality determined by criteria and the helpfulness of a review given the review text. The objective is to identify and extract possible features from the given review text that can be used to algorithmically detect the quality of a review.

This problem involves main challenges:

Feature Engineering. The subjectivity of the term quality describing good (helpful) or poor review makes it difficult to find appropriate criteria to automatically determine the level of helpfulness. A helpful review could have a different meaning for different individuals. For example the exact same review can be classified by two customers differently. For some people a good review contains e.g. many technical details of a product, for others little information, e.g. the functionality and lifetime might be sufficient or focus rather primarily on the product's price. This demonstrates that user may have different expectations and perspectives on the review itself, which makes it difficult to find likely common features which determines quality (helpfulness) of a given the review text. The implementation requires in particular data-/ text mining and Natural Language Processing (NLP) methods.

Machine Learning Algorithms. Supervised and unsupervised methods are investigated. Features are used as input to identify appropriate algorithms to run experiments and to compare and evaluate their performance.

1.2 Project Goal and Objectives

The prime goal of this project is the automatic detection of online reviews whether they are helpful or not helpful for the customers. Therefore the objective is to identify possible features in text by using Natural Language Processing, data mining and text analysis techniques that describe quality in online reviews.

Key objectives

1) Identification of appropriate criteria to determine the quality of online reviews in general and with respect to the domain of online reviews.
2) Identification of an appropriate product review corpus to conduct analysis.
3) Identification of relevant features for the selected quality criteria.
4) Identification of machine learning algorithms to detect the quality of online reviews in e-Commerce based on specified quality criteria.
5) Compare the performance of selected algorithms on the provided dataset.

1.3 Minimum Requirements & Deliverables

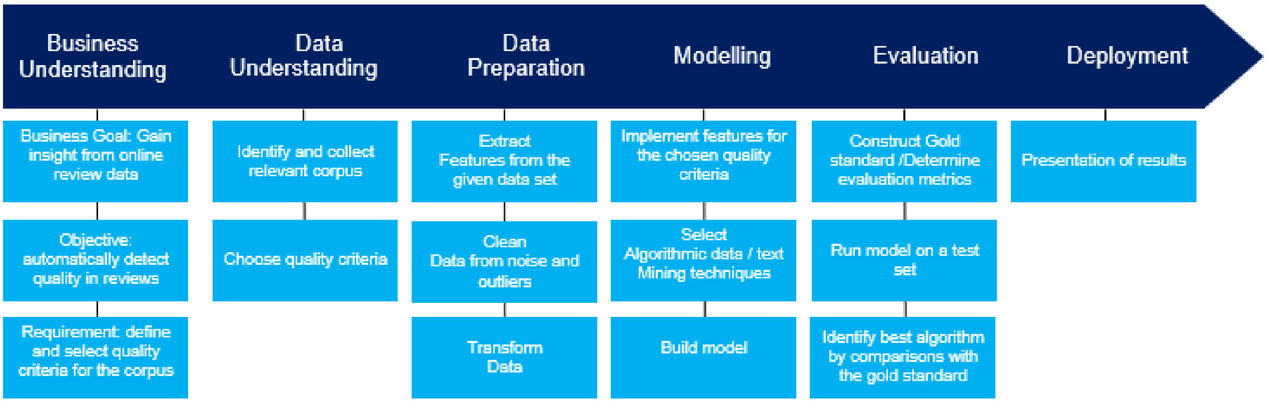
The following minimum requirements and deliverables are set for this project:

1. Identify appropriate quality criteria for online reviews.
2. Identify and implement features to detect quality in online reviews.
3. Find the best machine learning method to identify helpful online reviews.

1.4 Project Methodology

The methodology to conduct the project is based on the CRISP-DM (Cross Industry Standard Process for Data Mining) reference model. It can be considered as a best practice guide to successfully accomplish data mining related projects (Chapman, et.al., 2000). Since this methodology is widely accepted standard in academia and industry developed with the aid of practical and real-word project experience, CRISP-DM is considered as an adequate choice. Based on the chosen methodology, empirical experiments using a review corpus are conducted. By applying data analytics techniques the effectiveness of the proposed approach is evaluated.

The ***CRISP-DM*** model is an *iterative* model used in numerous data- and text mining projects that consists of six phases: business understanding, data understanding, data preparation, modelling, evaluation, and deployment (Chapman, et.al., 2000).



Business Understanding	Data Understanding	Data Preparation	Modelling	Evaluation	Deployment
Business Goal: Gain insight from online review data	Identify and collect relevant corpus	Extract Features from the given data set	Implement features for the chosen quality criteria	Construct Gold standard /Determine evaluation metrics	Presentation of results
Objective: automatically detect quality in reviews	Choose quality criteria	Clean Data from noise and outliers	Select Algorithmic data / text Mining techniques	Run model on a test set	
Requirement: define and select quality criteria for the corpus		Transform Data	Build model	Identify best algorithm by comparisons with the gold standard	

Figure 1: Adapted CRISP- DM model for this project

Figure 1 illustrates the six phases with their detailed steps which will be used as a guideline for the following work.

Business Understanding: This initial phase focuses on understanding the project objectives and requirements from a business perspective. It starts with the overall business goal to mine online review data to get insights for business. The prime objective of this project is to automatically determine quality in reviews by finding the best machine learning algorithm. For this purpose relevant features have to be identified as key step in feature engineering in order to be used as input for the algorithms. By conducting a literature review the quality criteria for the corpus are identified and form the requirements. Hence, this is the basis for the feature identification for this project.

Key Challenge: Feature Engineering is a very crucial part in the overall data-/ text mining process and takes around 80% of the work. Only 20% will be applied for analysis and getting the insights. A clear understanding of the problem and identification of relevant features is required to proceed to the next process step. The success of the outcome and the project is therefore depending on a profound research and identification of the features.

Data Understanding: In the data understanding phase the dataset is to be identified and collected. Key activities are familiarising with the data and connecting the business understanding with the data to discover initial insights from the data.

Key Challenge: The availability of a suitable data set is crucial to conduct the implementation and experiments. The data set has to full fill specific requirements to be eligible to meet the business goal because the following selected methods, algorithms and results for the analysis and identification of quality in online are strongly dependent on the data set. Also, the availability of a representative number of reviews for analysis are important to gain meaningful results.

Data Preparation: The Data Preparation phase contains activities to prepare the dataset to model the data. Feature extraction, ensuring data quality by cleaning data from noise and transformation are conducted in this phase.

Key Challenge: Data Pre-Processing and Data Cleaning are part of the feature engineering process. The preparation of the data takes a core role and is likely to be performed multiple times. Differences in the data format, inconsistency, missing data or duplicates are examples of tasks that have to be performed in this phase.

Modelling: The modelling phase covers the selection of appropriate data- / text mining and machine learning algorithms for the given dataset. The features extracted in the data preparation phase are the input for this phase.

Key Challenge: Several adjustments and parameterisation are possibly required to find a good model. Therefore experiments have to be conducted. Furthermore the identification of an appropriate gold standard is a challenging task itself. Different strategies exist amongst a decisions has to be made for this project. Furthermore training and test have to be constructed for the experiments as these are not supplied.

Evaluation: The evaluation phase covers all activities that enable the evaluation and verification of the model in accordance to the business objectives and demonstrate the quality of the results.

Key Challenge: The selection of appropriate evaluation metricises and applying them on the built models requires an evaluation strategy. Therefore comparisons are required using a gold standard or a test set containing unseen examples to identify the best methods for the given features and the data set.

Deployment: The results and knowledge gained from the previous steps have to be documented and presented in an appropriate way so the end-user can understand and make use of the created models. This project report is considered as documentation. No further system deployment is conducted.

Key Challenge: The appropriate presentation of the project is essential to make readers with no or less background knowledge in data analytics understand the relevance of this project and to share knowledge.

1.5 Project Management

The initial and revised project schedules are provided in Appendix E Project Schedule. The initial project schedule illustrates the specified tasks set at the beginning in order to complete this project. The initial time estimation were set without prior comprehensive experience in the feature identification, feature engineering and data preparation phase of this project. In the revised schedule (see Appendix E Project Schedule), the length of the implementation and experimentation tasks were extended as the familiarizing with new tools took longer than initially estimated and the more challenges were experienced during the implementation of the features. All the tasks set in the project schedule were completed.

1.6 Relation to Degree

Among the number of various modules studied for this degree, the following three are most relevant for conducting this project:

The module Knowledge Representation and Machine Learning (COMP5830M) provided me with an essential understanding of machine learning algorithms, further practical experience which is relevant for this project, and knowledge of the data mining software WEKA to explore various algorithms and analyse the results. In the module Data Mining and Text Analytics (COMP5840M), Text analytics, Natural Language Processing (NLP) techniques and algorithms were presented to obtain a good understanding and knowledge base in the underlying technologies and concepts. The data analytics approach CRISP-DM as best practise approach to conduct analytics projects was introduced. CRISP-DM is followed in this work. Semantic Web Technologies and Applications (COMP5860M) provided me with knowledge of new concepts and technologies in the semantic web. Interesting sources e.g. crowd sourcing platforms like CrowdFlower and Amazon Mechanical Turk were introduced in the module, which were also considered for this project.

1.7 Report Outline

The project report is structured into six parts. In Chapter 2 of this report, background researches of the methods suitable for the implementation of the system are presented. Along with the definition of the quality criteria for online reviews, relevant algorithms and evaluation techniques used in text analytics are demonstrated with respect to existing work in this field. Chapter 3, discusses the experimentation setup. A description of the data set as well as all software tools used to conduct the experiments is presented. In Chapter 4, an overview about the implementations of the quality criteria are provided. Chapter 5, discusses the evaluation results obtained from the analysis in chapter 4. By applying evaluation metricises and techniques the quality of the work is validated. Finally, the report concludes in Chapter 6 with an overall project summary, project evaluation along with indications towards future work.

2 Background Research

This chapter discusses all aspects to obtain a good business understanding as the very initial step in the methodology. Understanding the problem and the associated feature engineering to detect quality in online reviews is crucial as it compromises alone 80% of the work and impacts all proceeding steps. This step is accomplished by a literature review in this project.

This chapter discusses the underlying approaches, methods and tools of this work. Therefore a literature review was undertaken to demonstrate the pre-existing research work in this field and to provide a solid foundation to conduct the required experiments for this project.

2.1 Quality of Online Reviews

The automatic detection of quality in online reviews is a non-trivial task but a very crucial one to determine how helpful a review might be for customers. The question arises on how the customer can identify quality in reviews in the first place. The following section provides a definition along with suggested approaches to tackle the question.

Definition of quality criteria in reviews

Initially a mutual understanding on the term *quality criteria* in reviews and how it is measured has to be obtained. Cheng and Lau developed an Information Theory based methodology for the assessment of quality in online reviews (Cheng & Lau, 2014). The established methodology is based on 16 quality metricises to evaluate the quality of information on the web presented in (Rieh, 2002), (Zhu & Gauch, 2000). However namely the metricises regarding dimensions of subject, breadth, depth, timeliness, source, presentation, accuracy, writing style, credibility, and popularity are considered in their work (Cheng & Lau, 2014).

Based on existing literature mentioned above, an overview about important quality measures for online reviews is given. In the following a distinction is made between *metadata* and *corpus* (review text) related quality criteria. Due to the fact that metadata, such as ratings, user information etc. might not be sufficient to detect helpfulness of a review the content provided through the corpus should be considered simultaneously. Additional criteria are described as well.

Corpus-based quality criteria

Readability refers to how easy a user can comprehend a given text. The reading ease is determined by examining the actual text. The assumption is that the readability correlates with the helpfulness of a review. Reviews that are considered as helpful might have a better readability score than those that were not considered as helpful. To measure readability the review text is decomposed into basic structural elements. Length related measures, readability indices and linguistic measures, such as Part-of-Speech are focussed to measure readability. (Hu, et al., 2012), (Cheng & Lau, 2014).

Polarisation denotes either a positive, neutral or negative position towards a topic. Sentiment analysis in text is used to determine the polarity (positive, neutral or negative language) in reviews for instance towards a specific product.

Sentiment Analysis is conducted in this project to determine polarity. Therefore positive, neutral or negative sentiment polarities are extracted from the review text using sentiment tools (Hu, et al., 2012). At first glance it is not clear that the polarisation of a review provides knowledge about whether a review is considered as helpful or not. Both a positive and a negative review can be considered as helpful depending on the user's intention. Therefore no precise assumption can be made about the polarity of reviews and their level of helpfulness. The challenge is to explore experimentally whether there exists coherence between polarity and helpfulness. Techniques are required to initially count positive, neutral or negative polarisation per sentence and then to determine an overall indication of the review's polarity.

Informativity of a review demonstrates detailed information (facts) about the product such as about single product attributes. If a review is considered as informative, is a very difficult task to detect this because it is highly subjective and dependent on the context and the purpose of the user. Through the measures of "dimension of depth" and "dimension of breadth" (Cheng & Lau, 2014) a review can be classified as informative. If the review mentions information about a lot of e.g. product attributes the review is considered as informative. This is an important criterion to determine the reliability of the review since it demonstrates that the user has knowledge about the product and has likely used it (Cheng & Lau, 2014).

Redundancy describes repeating information (reviews) which do not add value through their existence. It is desired to reduce the amount of redundancy and to increase information richness. Redundant and repeating information can influence the quality score while ranking reviews. The aim is to provide a broad and a small set of high-quality reviews that cover many different viewpoints and attributes of the reviewed product (Liu, 2012). Therefore we assume that redundant content is not considered as helpful when making purchasing decisions where the objective is to reduce the amount of redundancy. Due to the fact that this quality criterion is not sufficiently represented in the given dataset, the further analysis is left as future work.

Metadata-based quality criteria

Timeliness of the review refers to the ratio of the longest elapsed time of all the reviews related to a product over the elapsed time of the online review is generated. An important factor while making a purchase decision are current reviews, to have an indication of how often a product is purchased. The submission time of an online review is available and can be additionally considered for analysis and evaluation. However, this criterion alone provides not enough information. The latest review does not automatically mean the review is of good quality.

Due to outdated reviews in the selected Amazon dataset (see chapter 3.2) from the year 2006 which may not contain current products or newer innovations and as metadata related criteria, timeliness is ignored in this project. (Cheng & Lau, 2014).

Ratings of Online reviews show an *aggregated vote* additionally to textual comments on how helpful the review was overall. This measure can be related to the popularity dimension of (Cheng & Lau, 2014). However since this aggregated view is not been further broken down, to learn how the rating is composed, the true meaning of the helpful vote is not evident. Because a helpful review is subjective depending on the user's preferences, an evaluation of the review text is equally important. This measure considering only the metadata as numerical feature, is therefore not primarily focussed in this project and discarded (Cheng & Lau, 2014).

Source Credibility of a review is related to the trustworthiness of a review providing correct information by the author. Therefore characteristics of the user respective the customer such as age, gender, race, education expert on a specific topic etc. is considered for these criteria. Source credibility is not considered in this project since no information about the user is available except the user name (in the selected Amazon dataset described in chapter 3.2) (Greer, 2009).

In order to summarise, the literature review has directed to following two hypotheses for the quality criteria readability and informativity:

Hypothesis	Description	Quality criteria
H1	The readability correlates with the helpfulness of a review. Reviews that are considered as helpful might have a better readability than those that were not considered as helpful.	**Readability**
H2	The Informativity of a review affects the helpfulness of a review. If a review e.g. contains information about product attributes, it might be considered as more helpful.	**Informativity**

Polarisation. A hypothesis formulation for the quality criterion polarisation is not possible like for readability and informativity. The influence of polarity on helpfulness has to be rather explored via experiments to make conclusions.

2.2 Feature Selection

After discussing relevant quality criteria in the previous section 0 the measures for the selected quality criteria readability, polarisation and informativity are described in this section in order to sharpen the business understanding and specify the requirements. The described measures act as input for the machine learning algorithms within the modelling phase of the CRISP-DM approach.

2.2.1 Readability

Readability refers to the comprehensibility of a text. It can be also stated as how difficult a text is to understand. Users are exposed to huge amount of information every day on the internet. Not only the amount but also the time to process the information is vast. In order to make fast purchase decisions it is important to display comprehensive and compact reviews covering ideally all product features with a good distribution of positive and negative sentiments (Tsaparas, et al., 2011). There exist specific measures to determine readability presented as follows:

Length Measures. The length of a written text can influence the absorption of information. In (Baddley, et al., 1975), Baddley states that memory span is inversely related to word length for various text types. This indicates that in terms of online reviews the length of the review text can influence the readability. Long text can discourage customers to read the review. It can limit the absorption of required information (e.g. product information) to conduct a purchase. Hence the question arises whether the length of a review has an impact on the helpfulness of a review. The defined measures in the table below are based on the definitions provided in (Smith & Senter, 1967) and used in the following calculations to determine the length of a review text.

Table 1: Length related readability measures

Measures	**Description**
#characters	Number of letters or numbers excluding spaces.
#words	Number of alphabetic or alphanumeric tokens enclosed by white space excluding punctuation.
# normalised words	Number of alphabetic or alphanumeric tokens enclosed by white space excluding punctuation. In addition normalisation through removal of stop words , stemming and

	lemmatisation is applied.
Proportion of distinct words	Number of distinct words / Number of words.
#sentences	Number of sentences split by punctuation (e.g. "." , "?", "!").
Average sentence length (in words)	Number of words / Number of sentences.

Readability Indices. Readability indices that are also known as readability tests calculate an index to a given text to determine how difficult the text is. Therefore most of the readability tests take the reviewer's educational background into account. There exist several standard measures however to the author's best knowledge no readability measure exits solely for online reviews. Nonetheless, prominent and standard readability measures are existent and have been applied to online reviews (Korfiatis, et al., 2008), (Ghose & Ipeirotis, 2011). Namely the Automated Readability Index, Flesch Reading Ease, Flesch-Kincaid measure , Coleman-Liau Index, Gunning Fog Index, and Simple Measure of Gobbledygook (SMOG) are few examples of common measures. In the following the selected indices for this project will be explained in detail.

<u>Automated Readability Index</u>

The readability can be calculated using the Automated Readability Index (ARI) by (Smith & Senter, 1967). The index was designed by the U.S. Army for the readability computation on electronic typewriters. The given text is decomposed into basic structural elements and the number of characters per word and the number of words per sentence are considered. The measure gives an indication on how well the text can be understood and what level of education is required for understanding the text. The outcome is an U.S. grade level from 1 (easy) to 12 (hard) as demonstrated in the following table where a lower score indicates a better readability of the text:

Table 2: U.S. grade level based on (The US-UK Fulbright Commission, 2015)

US grade level	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Age	6 - 7	7 - 8	8 - 9	9 - 10	10 - 11	11 - 12	12 - 13	13 - 14	14 - 15	15 - 16	16 - 17	17 - 18

The formula below represent the U.S. grade level based on calculations considering among others textbooks by the Cincinnati School System. For example, if the ARI yields the number

10, this equates to a high school student, ages 15-16 years old; a lower number like 3 means students in 3rd grade (ages 8-9 years old) should be able to understand the text. The ARI has been used in Information Retrieval (IR) (Hu, et al., 2012), (Yan, et al., 2006). Document Ranking as applied for instance in Google is a key field in IR where the readability of documents can help the user to identify relevant documents. Users spend usually only approximately 5 seconds to select relevant documents from their search results. The readability of the text can influence not only the ranking but can also be an indicator for the quality of a document. A better readable text is ranked higher as it is considered as more relevant than a text with low readability. Essential criteria for choosing the ARI is that the document ranking approach can be related to online reviews and NLP processing. Each review can be seen as a document with the readability is to be assessed. Furthermore the ARI was utilized as measure in (Hu, et al., 2012) to detect manipulation in reviews by examining readability respective writing style of reviews. Therefore this measure is considered in this work as it can be a possible indicator to determine quality in reviews. The computation of the ARI is based on a regression model derived from human experiments. The variables (*#characters, #words, #sentences*) are calculated from the given text corpus. The formula is shown below is taken from (Smith & Senter, 1967):

$$\boldsymbol{ARI} = \mathbf{4.71} * \frac{\#characters\,^{1}}{\#words} + \mathbf{0.5} * \frac{\#words}{\#sentences} - \mathbf{21.43} \qquad (\mathbf{1})$$

Equation 1: Automated Readability Index

Flesch-Kincaid measure

The Flesh-Kincaid (F-K) measure introduced by Rudolf Flesch and J. Peter Kincaid in (Kincaid, et al., 1975) outputs a score representing the U.S. grade level from 1 to 12 like the ARI. Unlike the ARI, the Flesch-Kincaid measure considers syllables per word. The drawback of using syllables is that it is less straightforward to determine automatically and relatively unreliable as the number of characters (Smith & Senter, 1967). This measure was initially designed to evaluate readability of technical manuals. As this project focuses on the examination of online reviews for technical products (cameras, computers, videogames, etc.), the Flesh-Kincaid measure is considered as a potential feature. The Flesh-Kincaid formula representing a regression model is shown in equation (2), taken from (Kincaid, et al., 1975). The constant values have been derived based on experimentation with human participants reading passages of technical manuals. The variables (*#words, #sentences, #syllables*) are calculated from the given text corpus.

[1] In the original formula [45] refers to strokes per word. For understanding purposes the word *stroke* is set equivalent to the word *character*.

$$\boldsymbol{Flesch - Kincaid = 0.39 * \frac{\#words}{\#sentences} + 11.8 * \frac{\#syllables}{\#words} - 15.59} \qquad (\mathbf{2})$$

Equation 2: Flesch-Kincaid Index

Coleman-Liau Index

The Coleman-Liau Index (CLI) measure introduced by Coleman and Liau (Coleman & Liau, 1975) outputs a score representing the U.S. grade level from 1 to 12 similar to the ARI and Flesh-Kincaid measure. However unlike ARI and F-K, it measures reading complexity rather than comprehension. It was initially developed for machine-based scoring for organisations such as the US Office of Education to standardise the readability of all textbooks for the public schools in the USA. Along with the ARI the CLI does not consider syllables and is therefore easy to compute. For the domain of online reviews the usage of this index is questionable since it was originally applied for school textbooks. Online reviews are predominately address adults. Texts that are excessively easy to understand might be not considered equally easy to all individuals. Nevertheless this index can be seen as an indicator. Therefore it is considered for the following computations to evaluate in particular the complexity of the text to analyse if the CLI influences the helpfulness of a review. The formula shown below is taken from (llcCallum & Peterson, 1982). It calculates characters per 100 words and sentences per 100 along with the variables (*#characters, #words, #sentences*) from the given text corpus:

$$\boldsymbol{Coleman - Liua = 5.88 * \frac{\#characters}{\#words} - 29.59 * \frac{\#words}{\#\ sentences} - 15.8} \qquad (\mathbf{3})$$

Equation 3: Coleman-Liua Index

Gunning Fog Index

The Gunning Fog Index (FOG) measures the text comprehension by an individual with a high school education (Gunning, 1969). Similar to the CLI the FOG measures outputs a score representing the U.S. grade level from 1 to 12 to measure the reading complexity. FOG considers further the number of complex words in the given text. The reading complexity is particularly context or domain dependent. The usage of domain specific words might be considered as difficult where the knowledge is unknown. The word “*flash brackets*” for instance can be unknown to people who are not familiar with the camera & photography domain and therefore consider this word as difficult. Therefore the collection of complex words specifically from the underlying domain is required. This index is potentially a good measure for the domain of online reviews. However it has to be taken into account that the examined dataset in this work contains a vast amount of products. Even the collection of complex words for one product such as cameras is time-consuming and requires profound domain knowledge. Due to time limitations and no available domain expert this measure was not further pursued. The formula representing a regression model is taken from (Gunning,

1969) is customised for an at least 100 word text passage. The variables (*#words, #complex words, #sentences*) are calculated from the given text corpus:

$$\boldsymbol{Gunning\ Fog\ Index} = \mathbf{0.4} * \left(\frac{\#words}{\#sentences} + \mathbf{100} * \frac{\#complex\ words}{\#\ words}\right) \qquad (\mathbf{4})$$

Equation 4:Gunning Fog Index

Flesch Reading Ease measure

The Flesch Reading Ease (FRE) was developed initially for the US Navy. It is a widely approved measure that evaluates the reading ease where the score ranges from 1 to 100. A lower score like 30 indicates a more “difficult” text and an higher score over 40, such as 70 indicates a more likely “easy” text (O’Mahony & Smyth, 2010). The FRE formula (5) taken from (Kincaid, et al., 1975) demonstrates a regression model with variables (*#words, #sentences, #syllables*) from the given text corpus. It considers the number of personal words (such as pronouns and names) and personal sentences such as quotes, exclamations, and incomplete sentences (DuBay, 2004). An implementation of this index can be also found in Microsoft Office Word (Zamanian & Heydari, 2012). This indicates that the index is applicable as standard readability test for various text types, such as online reviews. Hence, this index will be used in the following:

$$\textbf{Flesch Reading Ease} = \mathbf{206.835} - \mathbf{84.6}\ * \frac{\#syllables}{\#words} - \ \mathbf{1.015} * \frac{\#words}{\#sentences} \qquad (\mathbf{5})$$

Equation 5: Flesch Reading Ease Index

Simple Measure of Gobbledygook

The Simple Measure of Gobbledygook (SMOG) developed by G. Harry McLaughlin (Laughlin, 1969) provides an estimation of the grade level needed to understand a given text. He believed that word length and sentence length should be multiplied rather than added. By counting the number of words of more than two syllables (polysyllable) in 30 sentences, McLaughlin provided this formula:

$$\boldsymbol{SMOG} = \boldsymbol{a} + \sqrt{\#\text{polysyllable}} \qquad (\mathbf{6})$$

Equation 6: SMOG Index

, where *a = 3* is a constant independent from the corpus.

A differentiation is made in the calculation depending on the number of sentences since this measure was designed initially for text passages with greater than 30 sentences. Harold C. McGraw developed a conversion table to translate the SMOG value to the equivalent grade level (Appendix C). Due to the fact that online reviews usually are less than 30 sentences long (see given online review corpus Table 8: Dataset statistics) and the SMOG is preferably used as readability measure for healthcare-related material (Hedman, 2008), (Leya & Florio, 1996), this measure is not considered as adequate for the following analysis of online reviews.

To sum up, the selected readability indices used in this project are:

- Automated Readability Index (ARI)
- Flesch-Kincaid measure (F-K)
- Coleman-Liau Index (CLI)
- Flesch Reading Ease measure (FRE)

Note that apart from the Flesch Reading Ease measure which ranges from a scale from 0-100, all other three indices perform on a different scale 1-12.

Readability Indices Example. In order to illustrate the different indices the selected indices are calculated for a given review text. The text is categorised as "easy" to read due to the high FRE scores. The Automated Readability Index, Flesch-Kincaid measure, Coleman-Liau Index have similar results which requires a grade level of 8 to understand the text. The results can be mapped to an age group of 13-14 years which indicates that the text is easy to understand.

Table 3: Readability Indices Example

Review Text: *I am not impressed with Tiffen filters I got in this set. The finish of the rings are poor. Despite of previously poor reputaion, I recently purchased some 67mm Quantaray filter from the local Ritz Camera store which are of much better quality compared to this set of Tiffen filters. I never got Tiffens before, had good experience with Hoyas. I will probably stay with those from now on. Now come to think of it, Ritz camera indicated the Quantaray they had were made by Hoya. The glass on all the filters are quite a bit more reflective compared to the Hoyas and the Quantarays. Not a good sign. The circular polarizer had some smudges and some lense swirl that after 10 sheets of lense tissue and quite a bit of cleaning solution failed to be fully cleared off. Quite frustrating. All of the filters came with lint and a few little smudges on them, not sure if the store handling or the manufacturer handling was to blame, but none of the other filters I got before had this kind of problem right out of the box. The set might be cheap, but it sure is for a good reason*

Readability Indices	***ARI***	***F-K***	***CLI***	***FRE***
	8.3	*7.9*	*8*	*70.13*

Limitations of readability indices. Due to the fact that online reviews aim at addressing adults purchasing products, it is questionable whether the readability measures are an appropriate measure for online reviews. Typically the measure's outcome is a U.S school grade level therefore it might be difficult to understand what this result means for instance for European or in particular for the UK system. There exist differences in the educational level between both systems that might not be transferred one-to one. An additional mapping has to be conducted to transfer the outcome to a different country's system. Moreover these indices refer to the English language.

Linguistic Measures. The previous sections presented length related readability measures and readability indices which can be categorised as quantitative features as they are based on counting words. These features do not consider qualitative aspects of the text, such as linguistics parameters. Text Mining and Natural Language Processing are techniques to discover knowledge and patterns from unstructured textual documents (Koa & Poteet, 2007). The goal is to enable parsing, generation and extraction of semantics from natural language text using computers.

Part-of-Speech Tagging (POS). A prominent linguistic technique is *Part-of-Speech (POS) Tagging*. POS Tagging is used to identify syntactic or morphological structure of a word. This technique enables to tag the POS in the review text and in particular focus on the analysis of nouns, common nouns, verbs, adjective or adverbs (Feng, et al., 2010). These grammatical items tend to mention opinions through (e.g. adjectives and adverbs) and express information about product functionality and attributes (e.g. nouns, common nouns or verbs). POS tagging can help to disambiguate expressions, such as the word *like* indicating a sentiment (like can be either a verb or a preposition). A translation of the (Penn) Treebank Tag set can be found in (Atwell, 2015). For instance nouns are words with POS tag (NN, NNS, NNP, and NNPS).

Product attributes are frequently represented as nouns or noun phrases in reviews (Hu & Liu, 2004). The POS tagging supports the extraction of features that contain e.g. nouns describing product attributes such as size of a camera or the controller of a video game. This technique helps to identify specifically similar groups of grammatical items in a review. In order to accomplish the POS tagging the given text has to be parsed. The Penn Treebank tag set which is based on the Brown Corpus is applied for this purpose since it is well established and contains approximately 7 million words of part-of-speech tagged text (Penn Engineering, 1999).

Along with POS related features further linguistic based features were calculated as described in the following table:

Table 4: Linguistic readability measures based on (Piotrkowicz, 2015)

Measures		**Description**
POS	*Proportion of nouns*	Nouns are words with POS tag (NN, NNS, NNP, NNPS)
	Proportion of common nouns	Common nouns are words with POS tag (NN, NNS)

	Proportion of verbs	Verbs are words with POS tag (VB, VBD, VBG, VBN, and VBP. VBZ)
	Proportion of adjectives	Adjectives are words with POS tag (JJ, JJR, JJS)
	Proportion of adverbs	Adverbs are words with POS tag (RB, RBR, RBS)
Special characters	*#exclamation marks*	Number of exclamation marks.
	#question marks	Number of question marks
	#quote marks	Number of quote marks
Non English words	*Proportion of non-English words*	Proportion of non-English words. Words that are not existent in Wordnet are considered as not English words.
Amplifiers	*Proportion of intensifier*	Proportion of a word particularly adjectives or adverbs that expresses positive emphasis (e.g. very, really, great, perfect).
	Proportion of downtoners	Proportion of a word particularly adjectives or adverbs that expresses negative emphasis (e.g. only, enough, just, little)

2.2.2 Polarisation

The identification of a ***sentiment and subjectivity information*** (e.g. positive, neutral or negative) has a significant role in analysing the quality of a review. The research in opinion mining started with analysing the sentiment (i.e., positive or negative) of reviews which is also known as *Sentiment Analysis*. It is essentially a machine learning classification task. A common method is that a word lexicon with prior polarities (positive, negative or neutral) exits or is crafted initially for sentiment classification and used to detect polarisation. Profound research work has been conducted in the direction of polarity classification. Few contributions are mentioned as follows. Dave et al. examine the polarity classification on product reviews, they show that bigrams and trigrams perform better than unigram features (Dave, et al., 2003) . Simply described in computational linguistics N-grams predict the next word given N-1 previous words. Pak and Paroubek used Twitter data to develop a classifier to determine positive, negative and neutral sentiments of documents (Pak & Paroubek, 2010). The classifier is based on a multinomial Naive Bayes classifier that uses N-grams and

POS-tags as features. Pang et al. analyse sentiment classification using machine learning algorithms (Naive Bayes, Maximum Entropy classification, and Support Vector Machines (SVM)), They claim these algorithm do not perform well as in other fields such as text categorisation. Naïve Bayes performed the worst whereas SVM yields the best performance among the three (Pang, et al., 2002).

A differentiation has to be made between *document based* and *aspect based* sentiment analysis. Document based polarity classification summarises the polarity for a document (e.g. single online review) as a whole. Single aspects mentioned in the review are not of major relevance. Hu and Liu present in (Hu & Liu, 2004) the aspect based sentiment analysis on customer reviews for electronics which has similarities with the given dataset in this work. For this purpose the sentiment polarity of mentioned product aspects (product attributes e.g. picture quality , size of a camera) of a given product are determined on sentence level and summarised by counting the positive, negative comments for each product attribute. The weakness of this approach is that for each sentence product attributes that are explicitly mentioned are predominately considered, such as *size*. Product attributes can be also implicitly mentioned as in the example sentence “*While light it will not easily fit into pockets*” which also refers to the size of the product. In this case it is more difficult to detect a product attribute with the proposed frequency feature identification using the POS tagging method in (Hu & Liu, 2004).

Techniques exist to determine if the examined text is classified coarse-grained as positive, neutral or negative by looking at typical adjectival words indicating an sentiment, such as amazing, brilliant, awful, poor etc. associated with a product attribute. Furthermore by counting the explicitly mentioned product aspects Hu and Liu the risk of ambiguity can occur. An example is the product attribute *picture quality* which is equivalent to *photo quality* (Hu & Liu, 2004). There is no connection between both terms and therefore counted separately. Considering concepts as to identify features can mitigate this risk. Therefore an ontology based approach to identify domain specific features is presented in (Peñalver-Martinez, et al., 2014). This work concentrates on document based Sentiment Analysis to detect the polarisation of each review, in order to determine whether only the polarisation has an impact on the helpfulness of a review. Therefore, the identified features are grouped according to their semantic distance and linked to the main concept of the ontology. Next, a score for each feature is calculated by considering the position of the feature’s linguistic expression within the text. This score has an influence on the feature’s polarity as well as on the overall review polarity. There is no focus on specific aspects of entities. The quality criteria Informativity presented in the following will independently cover aspect-orientated features.

After reviewing different approaches in the related field, this project uses the sentences level approach similar to (Hu & Liu, 2004) counting the positive, neutral and negative sentences. However there is no focus on product aspects as polarity is analysed independently from informativity. The latter quality criterion includes product aspects.

The polarisation in this project is determined with four measures as shown in the table below:

Table 5: Polarisation measures

Measures	Description
Positivity	Proportion of positive sentences
Neutrality	Proportion of neutral sentences
Negativity	Proportion of negative sentences
Subjectivity	Proportion of only positive and negative sentences. Exclusion of neutral sentences to determine subjectivity.

2.2.3 Informativity

The identification of relevant product features has a significant role for the overall process to determine quality in product reviews. A review is considered as *informative* when it provides information about the product features. In (Hu & Liu, 2004)and (Hu & Liu, 2004), the authors propose architecture for a feature-based opinion summarization system:

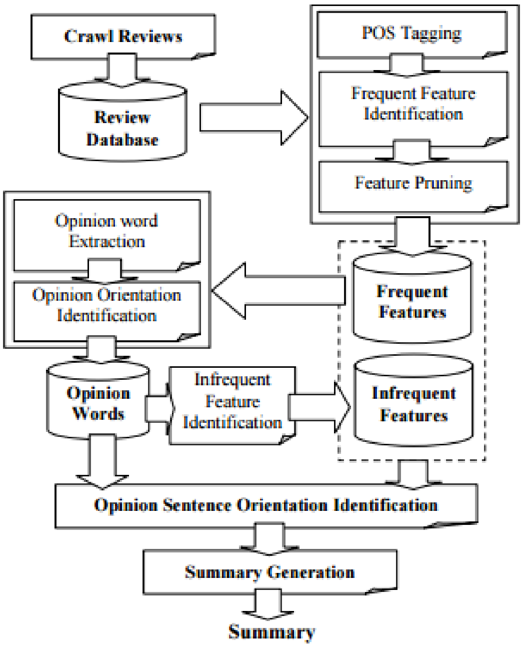


Figure 2: Architecture of an opinion summarization system (Hu & Liu, 2004)

The system first crawls all the reviews and stores them in a review database. After the POS tagging is conducted, the *frequent features* (most popular), are identified and used to extract *opinion words*. Opinion words are adjectives expressing an opinion (e.g. great, excellent, poor etc.) They are used to determine their semantic orientation (e.g. positive or negative*)* and support the identification of *infrequent features*. Finally, the orientation of each sentence is identified and a summary is generated.

Principally two types, frequent and infrequent feature identification have been discussed. An alternative approach is provided in (Peñalver-Martinez, et al., 2014) introducing the *ontology-based feature identification*. All approaches will be explained in more detail as follows:

Frequent and infrequent feature identification. In both, (Hu & Liu, 2004) and (Hu & Liu, 2004) Frequent features describe product attributes which are prominently talked about in the reviews. In the initial POS-tagging phase, the system extracts nouns and noun phrases which are identified as likely product attributes (Hu & Liu, 2004). In the next step, an association rule mining algorithm is used to find all frequent words or phrases (itemsets) that occur together. This technique has shown to be suitable because those features that occur in various opinions are likely to be relevant. These are considered as product attributes unlike those features that are infrequent. The classification is accomplished through an association rule mining which is based on the Apriori algorithm (Agrawal & Srikant, 1994). The association rule mining technique aims at finding patterns between the extracted features by taking confidence as measure.

The algorithm follows two steps:

- Generation of candidate features
- Generation of association rules

After all frequent itemsets have been found association rules can be generated. The association rules are generated by calculating the confidence (support count) for each possible rule from the itemset. Then, it is tested against a user-specified minimum support (e.g. 1 % in (Hu & Liu, 2004)) which acts as threshold. An itemset is said to be frequent if the support count satisfies a minimum support count threshold. In (Hu & Liu, 2004) only the first step of the Apriori algorithm was needed. Finally, feature pruning is required to reduce the number of candidates since associated rule mining may also generate incorrect or redundant features. This technique is important to avoid overfitting of the model. Further research has been conducted in finding frequent features by (Hai, et al., 2012) employing Latent Semantic Analysis (LSA) and Likelihood Ratio Test (LRT) to extract features and detect patterns among them.

Infrequent feature identification

Opinion words (adjectives) can be extracted from a sentence, using frequent features after the pruning phase since they can be found close to the features. If an opinion word modifies a noun / noun phrase then it is extracted. The extraction of infrequent features follows a heuristics described in (Hu & Liu, 2004) to find the nearest noun/ noun phrase that a known opinion word modifies.

The pseudo code for this heuristics is provided below (Hu & Liu, 2004):

```
For each sentence in the review database,
        if it contains no frequent feature but one or more opinion words,
            ▪ find the nearest noun/noun phrase of the opinion word.
            ▪ The noun/noun phrase is then stored in the feature set as an infrequent
              feature
```

A main problem with the described approach is, that also irrelevant noun/ noun phrases can be identified that are not related to the examined product. This is due to the fact, that common adjectives can be used to describe all sorts of objects, including relevant and irrelevant features. An example is given for illustration:

(a) “Red eye is very *easy* to correct.”
(b) “The camera comes with an excellent *easy* to install software”

Both sentences of digital camera reviews contain the adjective *easy* but in sentence (a), it describes an infrequent feature: *Red eye* and in sentence (b) a frequent feature: *software*.

However, this issue is mitigated since the significance and number of frequent features is much higher. Therefore infrequent features which have low significance, will be ranked very low due to their support measure.

This project focuses on the identification of frequent feature identification. Therefore the most common words mentioned in the review text are taken to construct a *bag of words* – representing frequencies of words from a text where the words have no order, as these words could mention relevant aspects of the product.

Ontology-based feature identification. Peñalver-Martinez introduces a feature identification approach using domain ontologies where each product attribute is linked with a class from the ontology. It is claimed, that this approach can be applied to different domains by simply changing the underlying domain-specific ontology. A major advantage is the interoperability and reusability of existing common vocabulary and ontologies for the domain. The work (Peñalver-Martinez, et al., 2014) focusses on the movie domain analysing movie reviews. Given the review corpus after NLP-processing and the domain ontology, the attributes are extracted and mapped by identifying the review sentences that contain classes, individual data types and object properties of the domain ontology.

In this project only data properties of a camera & photography ontology are used to build a bag of words to compare with the given corpus for similarity. If there is a match then the review text mentions specific product attributes and the occurrences are counted. For future improvements enrichments of the bag of words can be obtained by using for instance the Python library RDFlib[2] which supports e.g. parsing of RDF/XML and usage of SPARQL. For the purpose of this project solely the extraction of data properties were sufficient.

Feature identification using Human Computation. In order to identify relevant product features from the review text human computation can be used to build a bag of words. Therefore a sample of the dataset (review text) is given to the participants to annotate relevant words that likely describe the product. The goal is to achieve a similarity match between the created bag of words and the given review text. By counting the occurrence of the words in the review text an indication can be obtained about how informative the review is.

2.2.4 Machine Learning Methods

This section discusses the theoretical understanding of the learning algorithms used for the modelling phase of the CRISP-DM approach. After the feature implementation has been accomplished, appropriate machine learning methods have to be selected in order to build a learning model. Machine learning methods can be classified in two general techniques (Witten, et al., 2011).

Supervised learning. Instances of a dataset are assigned to pre-defined class labels. The dataset is split into a training and test set for evaluation. Given the training set different classifiers can be applied (learned) and evaluated later on the unseen instances from the test set. Mostly, supervised learning is employed to classification problems. Typical text mining algorithms are for example, Naïve Bayes or Perceptron.

Unsupervised learning. There exist no labels on each instance. The labels are found from the data itself. The algorithm divides a set of objects into clusters so that objects in the same cluster are similar to each other, and objects in different clusters are dissimilar. K-Means is an example of a clustering algorithm.

2.2.5 Unsupervised Learning: Clustering

Clustering algorithms group data objects based on information found in the dataset. K-Means is a popular unsupervised learning algorithm which is also used in the domain of text

[2] https://github.com/RDFLib/rdflib. Date of access: 27.08.2015

clustering. Therefore text documents are grouped into clusters according to their similarity of content. This technique is in particular useful in terms of searching a collection of documents (Cutting, et al., 1992) and to organise the search results obtained by a search query (Zamir, et al., 1997). This project regards text documents, such as online reviews to be also clustered according to their helpfulness.

K-Means algorithm clusters given a set of *n* data points in a dimensional space, *n* points into *k* clusters where *k* is a positive number. By using an error - or objective function the goal is to optimise the solution to obtain a good clustering solution. Initially, k points are chosen some heuristics(e.g. calculated distances of centroids are maximal) or at random to form the cluster centre. The choice of the number of k is not a trivial one. In particular in text clustering it can be difficult to set a reasonable number for k depending on the number of documents. The data points which form the instances are assigned to their closest cluster centre by using the Euclidean distance. In the next step the *centroids(means)* of the instances in each cluster are calculated. These centroids are the new cluster centres. The whole process is repeated until the clustering solution does not change anymore and terminates. Essentially the total squared distance from each data point to its cluster has to be minimised (Witten, et al., 2011). Essentially the total squared distance from each data point to its cluster has to be minimised (Witten, et al., 2011). Several iterations might be required until the solution converges to a local optimum. A problem of K-Means is that it is not ensured that the algorithm converges to the global optimum. The initial choice of the k points influence the outcome. Also, it has to be considered that in case of outliers, the calculation of the centroids can be immensely affected. Furthermore K-Means requires numerical attributes as input. Pre-processing is required including normalisation and conversion of nominal attributes to numeric to build the clusters.

Despite the mentioned weaknesses of K-Means, in this project the initial assumption is to apply this algorithm with k=2 clusters for helpful and not helpful reviews. Although information about the relevant features might not be provided. This algorithm can be used as first step to identify e.g. the quantity of reviews clustered as helpful and not helpful. K-Means can be seen as an initiation algorithm in order to use other analytics algorithms, such as classification techniques to deep dive into the impact and choice of relevant of features. In this project the clustering algorithm is used exploratory to find patterns in data

2.2.6 Supervised Learning: Classification

Classification describes the process of assigning documents into a predefined number of categories or classes. Machine learning algorithms can be applied to automatically classify documents to given classes. The goal is to learn a model that can be used to predict the class labels and generalise well on the unseen instances. In this project, we will focus on binary classification – classifying an online reviews either into helpful or not helpful.

The **Naïve Bayes (NB)** algorithm is one of the most prominent and simplest supervised learning techniques applied in many machine learning problems. In particular for document (text) classification a modified version of the algorithm - the *Multinomial Naïve Bayes* is used (McCallum & Kamal, 1988). The occurrences of words in a document are significant where the document is represented as a vector of word counts. The input is a *bag of words* – a set that contains all words in a document (Witten, et al., 2011). In interest of clarity the ordinary Naïve Bayes and the multinomial Naïve Bayes algorithm will be explained.

Ordinary Naïve Bayes classifier applies Bayes' rules to determine the most likely class of an unseen example. All features (attributes) of the example are conditionally independent with respect to the class. This is known as the *naïve Bayes assumption.* The formulation of the Bayes' rule is provided below:

$$p(c_j \mid d) = \frac{p(d|c_j)p(c_j)}{p(d)} \qquad (\mathbf{8})$$

Equation 7: Naive Bayes Formula

- p(cj | d) = Posterior probability of instance d being in class cj
- p(d | cj) = Prior probability of generating instance d given class cj
- p(cj) = Probability of occurrence of class cj.,
- p(d) = Probability of instance d occurring.

Multinomial Naïve Bayes for Text Classification

(McCallum & Kamal, 1988) distinguish in their paper two different types of the algorithm. The multi-variate Bernoulli model and the multinomial model where based on their experiments the latter is proposed as the preferred one for text classification. Compared to the ordinary NB, the multinomial NB denotes that each P[D|C] (9) is distributed multinomial which is appropriate for data that can be counted, such as word counts in text. Experiments are based on a bag of words containing the document words without any order. Typical characteristic is that, n repeated trials are undertaken where each trial has a discrete number of possible outcomes. The multinomial NB formula presented below is based on the Bayes' rule, is taken from (Witten, et al., 2011):

$$\mathrm{P}[D|C] = N! * \prod_{i=1}^{k} \frac{P_i^{n_j}}{n_j\,!} \qquad (\mathbf{9})$$

Equation 8: Multinomial Naive Bayes

where:

- $n_1, n_{2\,,} \dots, n_k$ is the number of times word i occurs in the document
- $P_1, P_{2\,,} \dots, P_k$ is the probability of obtaining word *i* when sampling from all the documents in the class *C*

- P[D|C] = Probability of a document D given its class *C*
- $N = n_1 + n_2 + \cdots + n_k$ is the number of words in the document

The multinomial NB is predominately used in text classification when the input data is represented as a bag of words. In this project the features are pre-calculated for the review text and represented as numeric and nominal features. Therefore the multinomial NB is not considered further for the following experiments.

Strengths of the NB algorithm are that it is useful when many features are equally important as all features are considered independently. Furthermore the algorithm is computational less expensive in terms of CPU and memory (Huang, et al., 2003) and shows robustness against irrelevant features since it cancels out the irrelevant ones without affecting the overall results. Naïve Bayes can be taken also taken as a baseline for comparing the performance with other algorithms. The Naïve Bayes algorithm has been applied for the domain of online reviews. Few examples for its application are demonstrated as follows. (Ye, et al., 2009) used NB for sentiment classification of online travel reviews. In comparison with the other tested algorithms Support Vector Machines and a N-gram based character language model, Naïve Bayes didn’t perform as good as the other two. However all three achieved good accuracy where over 80% examples were classified correctly. In (Wang, et al., 2005), online product reviews were classified on sentence level according to their semantic orientation (recommended or not recommended). Therefore NB is considered as appropriate baseline algorithm for this project.

Random Forest belongs to the decision tree-based supervised learning algorithms. This classifier consists of a collection of decision trees that are let to vote for the most popular class (Breiman, 2001). This classifier is known to perform well on high dimensional data. Advantages of using Random Forest are that little parameter tuning is required. This is in particular beneficial when dealing with large datasets where pre-processing or tree-pruning has a high cost. The algorithm executes an implicit feature selection by using a small subset of relevant features to split the tree. The feature selection on which the tree splits is based on randomisation. Given a set of features, not necessarily the best ones few are chosen randomly. The performance of the algorithm can be influenced by the number of trees chosen as well as on the number of features. Also, the algorithm will likely take longer to run depending the specified parameters.

In the paper (Ghose & Ipeirotis, 2011) Random Forest was applied to detect helpfulness and economic impact of product reviews. The authors claim that Random Forest performs better than SVMs according to their experiments. They built the predictive model by using 20 trees and different classifiers for each product category. Evaluation is conducted by 10-fold cross validation using accuracy and area under the ROC-curve (measures and visualises accuracy) as metricises.

Support Vector Machine (SVM) belongs to the supervised learning algorithms. The general idea of SVMs is to find a hyperplane that separates the feature space into two classes. The *margin* measures the distance of the hyperplane to the nearest point in the dataset, in order to determine how well the data is separated by the hyperplane. A high margin indicates a good separation of the data. The optimal hyperplane lies in the middle of the two nearest data points of the classes and shows the best generalisation (Menon, 2009). The SVM problem can be formulated as an optimisation problem where the goal is to find a maximum margin separating hyperplane.

LibSVM developed by (Chang & Lin, 2014), is a library which provides the ability of using a support vector machine. Originally the library was implemented in C, however in this project the WEKA implementation of this library is used.

2.3 Evaluation of Machine Learning Methods

This section discusses methods and metricises to evaluate and validate the analysis results to in order to ensure quality of the work. Firstly, the gold standard is presented as key evaluation method along with a description of its creation for this work an overview is provided about the gold standard used in related work. Secondly, three prominent metricises: Precision, Recall and F-Measure to evaluate the performance of the used algorithms.

2.3.1 Gold Standard

A gold standard describes the *standard* output of an algorithm. Usually the construction of the gold standard is itself a challenging and resource intensive task which requires domain experts for instance to manually annotate the dataset for a text classification problem. Due to the fact that annotation is a time, cost and data intensive task, crowdsourcing platforms such as Amazon Mechanical Turk (Amazon, 2015) or CrowdFlower (CrowdFlower, Inc, 2015) offering human computation (e.g. human annotation) can support the completion of tasks at large scale that are difficult to solve computationally or by a single person. An alternative approach is to derive the gold standard from the given corpus itself. This approach is pursued in this project where the reviews are automatically classified as helpful or not helpful based on the provided helpful rating. Details of the implementation are explained in chapter 4.6.

2.3.2 Training and Test set

Machine learning methods can be evaluated using training and test sets to obtain an unbiased estimate of accuracy of the learned model. Given examples from a dataset for training the learning method builds a learned model in order to make predictions (or decisions) (Witten, et al., 2011). The training data can be a subset of the data set.

Supervised learning problems use in particular labelled data set that is split into training and test set. The test set contains the unseen examples by the model. The goal of a trained learning model is to generalise well on unseen examples (test set). Therefore the same data which has been used to train the model cannot be applied for evaluation purposes otherwise it would lead to perfect a result which is misleading. This phenomenon is called *overfitting* and has to be avoided. Unsupervised problems do not require a test set evaluation because the labels are found from the data itself.

2.3.3 K-fold Cross-Validation

Cross-Validation is a common approach when evaluating the performance of a machine learning system. A helpful method is k-fold cross-validation where the user can choose in how many k-folds (e.g. k=10) or partitions the data should be split. The data is split into k-1 folds for training the model and the remaining is holdout for testing. In particular when having limited data for training and testing this method is preferred.

As default the Analytics Tool WEKA which will be discussed in the following chapter uses the Stratified Cross-Validation method with 10 folds (Witten, et al., 2011) where 9 folds are used for training and 1 fold for testing. The partitioning is done randomly to make sure the class is represented equally as in the full dataset. Then the error rate is calculated by executing the procedure for 10 times, the 10 error rates are averaged to an overall measure. Different error rates (Mean absolute error, Root mean squared error, Relative absolute error, Root relative squared error) are calculated and rather used for prediction than classification. Therefore the error rate is not considered in this project for evaluation. The number of folds can be also chosen by the user according to the user preferences for testing.

2.3.4 Clustering Evaluation Metrics

The most common measure to evaluate K- Means clustering is through the Sum of Squared Error (SSE). Therefore the error is determined for each data point, representing the distance to the nearest cluster. In order to calculate the SSE, the errors are squared and then summed.

2.3.5 Classification Evaluation Metrics

Three widely used evaluation metricises are relevant for text classification which have their origin in information retrieval: Precision, Recall and F1-Measure which is a combined measure of Precision and Recall.

A 2-by-2 contingency table known as confusion matrix contains four crucial parameters which are essential to calculate these measures. This visualisation form displays the relationship between two categorical variables.

	Predicted Class	
Actual Class	Yes	No
Yes	***True Positive (TP):*** correctly identified instances	***False Positive (FP):*** incorrectly identified instances
No	***False Negative (FN):*** incorrectly rejected instances	***True Negative (TN):*** correctly rejected instances

Figure 3: Definition Confusion Matrix based on (Witten, et al., 2011)

The table below based on (Witten, et al., 2011) provides a description along with formula for the metricises:

Metric	**Description**	**Formula**
Precision	Precision is the fraction of the selected documents which are correct. The quality of the search result is of relevance even if only a small subset of the documents is selected.	$\frac{TP}{TP + FP}$
Recall	Recall is the fraction of the correct documents which has been selected. The quantity of the selected correct documents is of relevance.	$\frac{TP}{TP + FN}$
F1-Measure	F1-Measure (balanced) is a combined measure of Precision and Recall and forms the harmonic mean of both measures with equal weights. It is a measure to express accuracy.	$\frac{2 * Precison * Recall}{Precison * Recall}$

Table 6: Evaluation metricises

It has to be considered that there is a trade-off between Precision and Recall. Generally it is desired to achieve a high number for Precision and Recall however by increasing Precision, Recall tends to decrease since more and vice versa. Depending on the application either a high Precision or a high Recall is desired. For instance when a system should find online reviews on the Nikon D7100 camera in 2015, then Precision is important but if every online review about cameras is required than Recall is the relevant measure.

3 Experimentation Setup

In this chapter all necessary steps to set up the experiments are presented to implement f. With respect to the CRISP-DM approach, this chapter focuses on the data understanding phase to relate the selected quality criteria to an online review dataset. The examined dataset originates from the e-Commerce domain. Therefore product reviews from Amazon are examined and a description dataset is provided. Along with a justification for choosing the dataset, major problems of the dataset are addressed. Further a final selection of the quality criteria for e-Commerce is conducted based on the given dataset. Finally, all software tools for analysis are explained.

3.1 Dataset Selection

To gain a data understanding for this project, an appropriate dataset is required. The following selected methods, algorithms and results for the analysis and identification of quality in online reviews are strongly dependent on the chosen dataset. Therefore the challenge is to select a dataset that meets specific requirements:

Availability. The dataset has to be openly available or directly accessible through an Application Programming Interface (API).

Size. In order to build a model using algorithms for training and testing purposes on unseen examples, the dataset needs to be large enough.

Trustworthiness *of the data source*: The data source where the data is extracted has to be trustworthy and reliable to obtain valuable analysis results. Therefore the reputation of the retailer (author), publicity or levels of turnover are for instance reasonable indicators to consider.

Informative. The extracted data needs to provide sufficient information particularly about the product, customer reviews, and additional sales information, such as ratings. This information is necessary to extract useful features for the following data processing.

3.2 Dataset: Amazon online reviews

For the following analysis, product reviews from Amazon.com containing product data provided by (Dredze & Blitzer, 2009) were used. The dataset containing was preferably chosen because previous work was already conducted based on these datasets (Blitzer, et al., 2007) Furthermore the Amazon dataset consists of product reviews for 25 different product types (for instance Books, DVDs, Music or Electronics etc.). The broadness of the data can be used to leverage generalisation. In contrast to Amazon, an alternative data source, such as Best Buy offers only electronics. Hence, there is a potential to use the analysis results of this project to generalise more and apply the found approach on different e-Commerce platforms. The dataset is given in both forms - unprocessed and pre-processed annotated in negative or positive reviews. However no information was provided in how the data has been pre-processed. Due to the lack of information the pre-processed data cannot be considered and was not utilized as data source.

3.3 Dataset Description

The Amazon reviews are provided as flat files sorted according to product types. For each product type, there exists a folder containing reviews which are again separated into raw reviews in pseudo XML-format and annotated reviews.

For each product the following information are available:

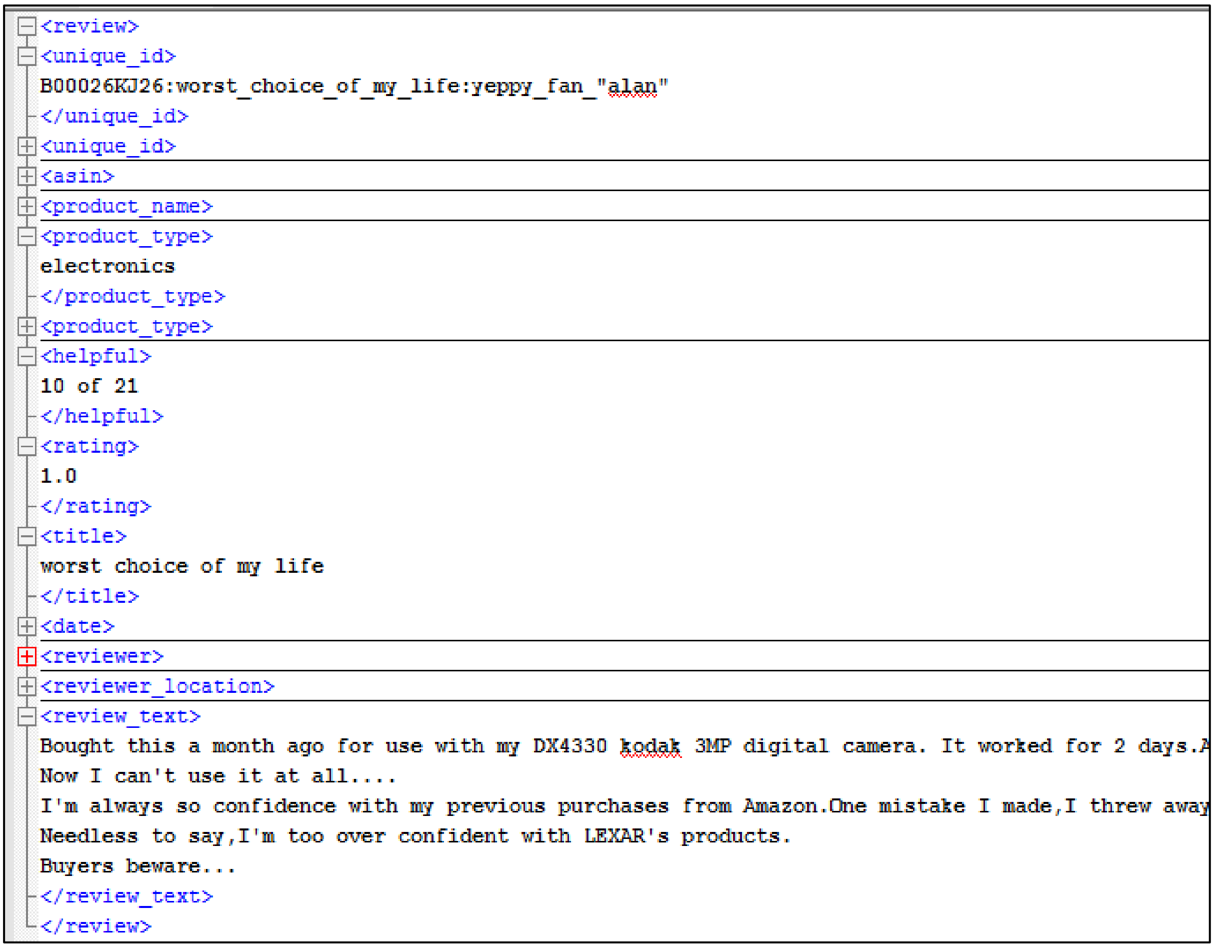
Table 7: Amazon dataset description

Technical field	Description	Used in Project	
unique_id (1)	Not numeric unique identifier concatenated as <asin>:<title>:<reviewer>.	✓	Based on both unique_ids, a new unique_id for a review was generated as identifier
unique_id (2)	Numerical Unique identifier.	✓	
asin	Amazon Standard Identification Number (ASIN) is a unique code to identify items in Amazon.	x	ASIN was not relevant because a unique_id was used as identifier.
product_name	Name of the product.	✓	Product names were extracted for the informativity feature calculations.
product_type	Product type.	x	Not relevant because each product type was processed independently. The product name is sufficient for this project.
helpful	Number of users who found the review helpful (e.g. 3 of 10).	✓	The proportion was used to calculate the class label (helpful and not helpful).
rating	Star-Rating (out of 5 stars)	✓	The rating was only considered in some experiments with machine

			learning methods.
title	Review title.	x	Only the review text was considered as corpus.
date	Review publishing date.	x	Metadata related fields are ignored and are not in the scope of this project.
reviewer	Name of the reviewer.	x	
reviewer_location	Reviewer's location	x	
review_text	Customer's opinion on the product.	✓	Review text represents the corpus for this project.

Data Format

As illustrated in the extract below in Figure 4: Example: Data format of an Amazon , the raw data is provided in pseudo XML-format.



```
<review>
<unique_id>
B00026KJ26:worst_choice_of_my_life:yeppy_fan_"alan"
</unique_id>
<unique_id>
<asin>
<product_name>
<product_type>
electronics
</product_type>
<product_type>
<helpful>
10 of 21
</helpful>
<rating>
1.0
</rating>
<title>
worst choice of my life
</title>
<date>
<reviewer>
<reviewer_location>
<review_text>
Bought this a month ago for use with my DX4330 kodak 3MP digital camera. It worked for 2 days.A
Now I can't use it at all....
I'm always so confidence with my previous purchases from Amazon.One mistake I made,I threw away
Needless to say,I'm too over confident with LEXAR's products.
Buyers beware...
</review_text>
</review>
```

Figure 4: Example: Data format of an Amazon review

Data Sampling. Since the Amazon data is too broad and various product types are available for analysis, a representative sample has to be taken to limit the scope of this project. Due to this fact the following two product types are selected.

- Camera & photo
- Computer & Videogames

Main reason for choosing these two different product types were that for camera and photography equipment the consumer can possibly form an opinion prior to its usage based on its technical details mentioned on e.g. the product description. In contrast for computer & videogames the consumers have to play the game to form an opinion about the product. Therefore different factors may be important to determine the helpfulness of a review. A further reason for choosing these datasets is the size. The processing time and resources needed to conduct text processing are highly dependent on the size of the data. Therefore data files less than 10 MB, such as or the two selected products were regarded as acceptable. In comparison the data file for the product type book has a size of 1,4 GB.

The table below summarises the statistics for the whole camera & photo and computer & Videogames dataset.

Table 8: Dataset statistics

	Camera & Photo	**Computer & Videogames**
Number of reviews before cleaning	7,408	1,966
Number of reviews (after cleaning)[3]	5,704	1,441
Total number of products	937	205
Total number of sentences	42,064	14,799
Average number of sentences per review	7	10
Total number of words	427,128	153,579
Average number of words per review	74	106
Total number of characters	3,361,630	1,235,896
Average number of characters per review	589	857

[3] Reviews which do not contain any review text, rating or a helpful rating are filtered out.

3.4 Software Tools

This chapter describes all important software tools used for conducting the relevant experiments.

3.4.1 Python for Data and Text processing

Python is dynamic object-oriented programming language with broad text processing libraries. Hence Python 3.4.3 has been selected as software tool for data and text processing in this project. The following table summarises the main Python libraries used for text processing.

Table 9: Python Libraries used in this project

Python library	Description	Usage in this Project
Pandas (scikit-learn developers (BSD License), 2015)	• data structures and data analysis (e.g. statistics) tool library • data transformations similar to database operations (e.g. group by, merge, join etc.)	• Data read/ write (csv) • Data storage and manipulation in Pandas DataFrame
Numpy (scikit-learn developers (BSD License), 2015)	• n-dimensional array package	• Data manipulation
BeautifulSoup (Python Software Foundation, 2014)	• Python parser for XML or HTML • Supports the iteration, search and modification of the parse tree	• Used for data pre-processing to remove XML tags
NLTK – Natural Language Toolkit (NLTK Project, 2015)	• Interfaces for over 50 corpora and lexical sources (e.g. Wordnet) • Offers text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning	• Retrieving corpuses • Tokenisation, stemming, lemmatisation of words, sentences etc. • POS, Parsing • Usage of regular expressions
Textstat (Python Software Foundation, 2014)	• Calculates statistics from text to determine readability, complexity and grade level of a given corpus	• Calculation of readability indices
Scikit-learn	• Provides a range of supervised and	• Feature extraction for bag-

(scikit-learn developers (BSD License), 2015)	unsupervised machine learning algorithms	of words • machine learning algorithms

3.4.2 Software tools for Polarity Analysis

There exist different software tools to identify and conduct sentiment and polarity analysis. Three established tools OpinionFinder, LingPipe and the Python library NLTK which supports Sentiment Analysis are presented below.

OpinionFinder was developed by researchers at the University of Pittsburgh, Cornell University, and the University of Utah. It is published under the GNU General Public License as a freely available, known platform-independent command line tool written in Java for automatic identification of subjectivity and sentiment polarity (positive, neutral or negative) in text (Wilson, et al., 2005). It employs multiple NLP techniques on sentence-level where subjectivity is detected and the effected words are marked. Details about the usage are described in chapter 4.4.

LingPipe is a software library for natural language processing, POS tagging, entity extraction, classification or clustering etc. in Java. It is one of the widely and mature Java application programming interface (API) for NLP processing and sentiment analysis in research and industry. The LingPipe API is available under licensing terms that range from free to perpetual server licenses (Alias-i, Inc, 2011).

NLTK – the natural language Toolkit (NLTK Project, 2015) is a software tool available for the Python programming language for text processing and mining. Polarisation or Sentiment Analysis is accomplished for instance through tokenisation techniques, POS tagging and calculation of polarity scores.

For this work the decision was made to experiment with an external and independent command line tool such as LingPipe or OpinionFinder to gain experience with these tools. Therefore OpinionFinder was selected. The main motivation to use OpinionFinder results from the easy installation, good performance and usage process. The usage of OpinionFinder as well as the implementation of polarity features is discussed in chapter 4.4.

3.4.3 Data Analysis tool - WEKA

WEKA (Waikato Environment for Knowledge Analysis) is a data mining software tool developed by the University of Waikato, New Zealand (Witten, et al., 2011). The decision to use this tool originates from prior experience with WEKA and as it is a well-established analytics tool in academia. The software version 3.6.11 of WEKA is evaluated in this work. WEKA is a platform independent tool written in Java. It is mainly used in academia and research purposes. A significant strength of WEKA is its wide range of predefined algorithms for data mining and machine learning tasks from pre-processing till modelling. Essential algorithms cover classification, clustering and regression problems. In addition to the data analysis and visualisation functionality, the efficiency of the algorithm's performance can be evaluated. In order to process data, WEKA needs a special ASCII-Data format (*.arff) for data analysis. Information about the attributes and a Boolean representation which acts as the class variable for prediction are required in the file. WEKA processes single files provided as .arff, or imports csv or Excel files and converts them to the .arff -format. For the purpose of this work, WEKA is considered as an appropriate tool to process the data in form of flat files.

4 Implementation

4.1 Implementation Overview

This section presents an overview about the features calculated for this project. Along with the feature description the formulas for calculation as well as the required implementation steps are demonstrated.

A graphical overview of the process steps is demonstrated by the illustration below:

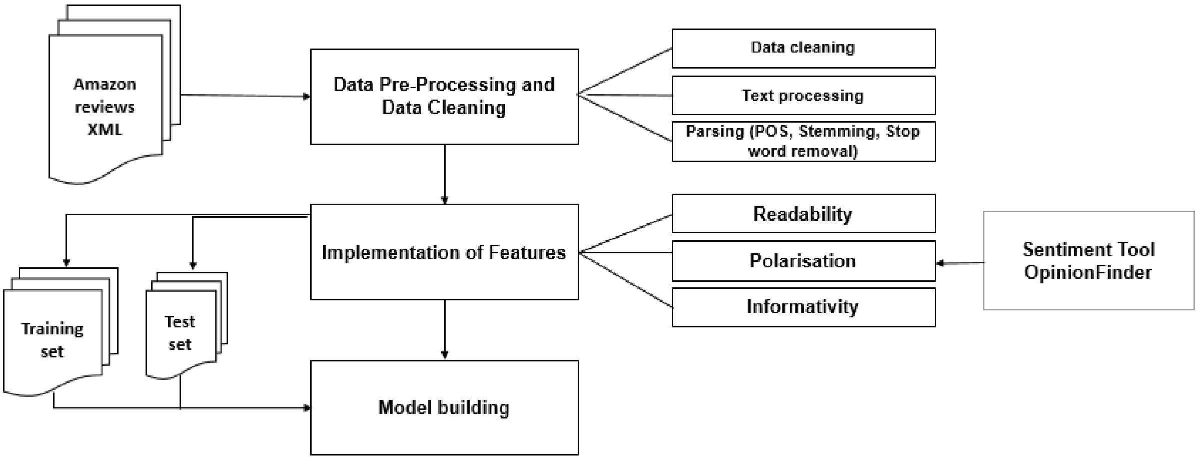

Amazon reviews XML
Data Pre-Processing and Data Cleaning
Data cleaning
Text processing
Parsing (POS, Stemming, Stop word removal)
Implementation of Features
Readability
Polarisation
Informativity
Sentiment Tool OpinionFinder
Training set
Test set
Model building

Figure 5: Overview of implementation steps

4.2 Data Pre-Processing and Data Cleaning

Data format*:* The given Amazon dataset is provided in a pseudo XML-format which requires different processing from traditional XML. Therefore Beautiful Soup – a Python package was utilised. Beautiful Soup is an HTML/XML parser for Python that can process also invalid or customised XML into a parse tree (Behnel, et al., 2015).

Current relevance of data*:* The examined data was collected in 2006 which may lead to the fact that certain products might be not available for purchase anymore or are outdated. In particular the lifetime and developments in electronics are changing rapidly. An example would be the product “Panasonic SC-PM53 180-Watt 5-CD Executive Micro System” which is discontinued by the manufacturer (Amazon, 2015). However the currency of the data or specifically of products is not crucial for this project and is therefore ignored. The focus of the project is to analyse and determine quality criteria of reviews by considering customer or product features. The current market development of the products is not highly relevant. A further reason is, the difficulties in collection of current data through the Amazon API. Amazon’s Product Advertising API (Amazon, 2015) enables users to retrieve product reviews, however a request to the API returns an URL to an HTML iframe, which is used to embed another HTML page into a HTML page and not the actual reviews which expires in 24 hours without periodical updates.

Incompleteness*:* The dataset exhibits partially incompleteness due to missing data when customers had not provided all details, such as location of their origin, helpful-ratings etc. Besides the dataset has an inconsistent provision of a unique id to identify a data record. Since for instance a review text as corpus and a helpful rating for calculating the class label are required for analysis, records with missing values in these fields are ignored.

Inconsistency*:* Technical inconsistency is evident in the dataset. For few products two unique ids with the same field name unique_id are existent. The first id is an alphanumerical id whereas the second one is a solely numerical. This identifier is not available for all product types, e.g. it is maintained for electronics, books or camera and photo but not for automotive. This causes difficulties in pre-processing the data. Therefore an own unique identifier has been generated to identify the reviews.

Duplicates*:* Duplicate reviews of the same user with the exact same review text cause the display of duplicate content on the retailer’s website and do not add any additional information value for the customer. Duplicates can be also an indication of review spam as explained by Lim et al. (2010) in (Lim, et al., 2010), in order to influence the sales either positively or negatively. Examples can be seen in the product type jewellery & watches of the Amazon dataset. Duplicate reviews are listed where every value is identical except from the numerical unique id for the review. Therefore this dataset was not selected.

4.3 Implementation of Readability Features

In order to calculate the length related features the NLTK Python module was utilized. The normalisation of words in the reviews required in particular the removal of stop words. The list of stop words utilized in NLTK contains 2,400 stop words for 11 languages and originates from (Porter et al). Further, pre-defined functions of the NLTK module for stemming (Porter stemmer) (Porter, 2006) and the Wordnet lemmatiser for lemmatisation were applied.

Length Measures

Table 10: Implementation of Length Readability Measures

Feature	Calculation
Number of character excluding blanks	$\#\boldsymbol{character} = \#\boldsymbol{tokens} - \#\boldsymbol{blanks}$
Number of words per review	$\#\boldsymbol{words}$
Number of words per review after	$\#\boldsymbol{normalised\ words}$

normalisation (removal of stop words, stemming and lemmatisation)	
Number of sentences per review	$\#\boldsymbol{sentences}$
Average sentence length per review	$\boldsymbol{avg}\ (\boldsymbol{sentence\ length}) = \frac{\#words}{\#sentence}$

Readability Indices

The calculation of the readability indices were accomplished using the Python textstat module. The implementation of the indices is based on the formulas given in section 2.2.1.

The table below provides an overview with statistics of the readability indices for the given datasets. Therefore the arithmetic mean as well the standard deviation is calculated for the both corpora: Computer & Video and Camera dataset.

Table 11: Readability Indices Statistics for Computer & Videogame dataset

	Helpful		**Not Helpful**		**All**	
	Arithmetic Mean	Standard Deviation	Arithmetic Mean	Standard Deviation	Arithmetic Mean	Standard Deviation
ARI	8.11	3.47	7.91	4.45	7.9	4.01
FRE	73.94	11.28	75.25	13.78	75.4	12.71
F-K	7.25	2.71	7.14	3.62	7.04	3.22
CLI	8.03	2.16	7.71	2.32	7.73	2.26

Table 12: Readability Indices Statistics for Camera dataset

	Helpful		**Not Helpful**		**All**	
	Arithmetic Mean	Standard Deviation	Arithmetic Mean	Standard Deviation	Arithmetic Mean	Standard Deviation
ARI	7.58	3.17	7.16	5.88	7.44	4.27
FRE	73.89	11.41	73.41	17.64	73.83	13.78
F-K	6.99	2.6	7.05	4.79	6.94	3.48
CLI	7.91	2.04	7.79	2.65	7.78	2.27

Linguistic Measures

The Penn Treebank linguistic parser of the NLTK Python library (NLTK Project, 2015) was used for the experiments. Before tokenising each review into words and tagging them with a POS the punctuation was removed using a regular expression, in order to consider only alphanumeric words.

$$Regular\ Expression = ((? <= [\hat{}\backslash w\backslash s])\backslash w(? = [\hat{}\backslash w\backslash s])|(\backslash W)) +$$

In this experiment the focus lies in particular on the following part-of speech tags since these are chosen as most relevant for online reviews: Nouns, common nouns, verbs, adjectives and adverbs. These features tend to mention opinions, product functionality and attributes

(Feng, et al., 2010). Further features are the proportion of Non-English words where the words in the review text are checked for existence in Wordnet. NLTK is used to determine word equivalence in Wordnet. For the calculation of the amplifier features a list of 248 intensifier and 39 downtoners obtained from (Quirk, et al., 1985) and (Biber, 1988) are used to compute the occurrence in the text.

An overview of all implemented linguistic features is provided in the table below.

Table 13: Implementation of Linguistic Measures

Feature Group	**Feature**	**Calculation**
POS	***Proportion of nouns***	$\boldsymbol{nouns} = \frac{\#nouns}{\#normalised\ words}$
	Proportion of common nouns	$\boldsymbol{common\ nouns} = \frac{\#common\ nouns}{\#normalised\ words}$
	Proportion of verbs	$\boldsymbol{verbs} = \frac{\#verbs}{\#normalised\ words}$
	Proportion of adjectives	$\boldsymbol{adjectives} = \frac{\#adjectives}{\#normalised\ words}$
	Proportion of adverbs	$\boldsymbol{adverbs} = \frac{\#adverbs}{\#normalised\ words}$
Special characters	***Number of exclamation marks***	$If\ exclamaion\ mark\ in\ review\ text\ then$ $\#exclamation\ mark, \quad else\ 0$
	Number of question marks	$If\ question\ mark\ in\ review\ text\ then$ $\#question\ mark, \quad else\ 0$
	Number of quote marks	$If\ question\ mark\ in\ review\ text\ then$ $\#quote\ mark, \quad else\ 0$
Non English words	***Proportion of non-English words***	$If\ word\ in\ review\ text\ in\ Wordnet\ then$ $\frac{\#not\ english\ words}{\#normalised\ words}, \quad else\ 0$
Amplifier	***Proportion of intensifier***	$\boldsymbol{intensifier} = \frac{\#intensifier}{\#normalised\ words}$
	Proportion of downtoners	$\boldsymbol{downtoner} = \frac{\#downtoner}{\#normalised\ words}$

The tables below provide an overview of the linguistic measures for readability in the given datasets. Therefore the mean value is calculated for the both corpora: C1 = Computer & Video and C2 = Camera dataset.

Table 14: Arithmetic Mean of Feature Group POS

Arithmetic Mean	**Proportion of nouns**		**Proportion of common nouns**		**Proportion of verbs**		**Proportion of adjectives**		**Proportion of adverbs**	
	C1	***C2***	***C1***	***C2***	***C1***	***C2***	***C1***	***C2***	***C1***	***C2***
Helpful	0.46	0.48	0.38	0.36	0.31	0.33	0.15	0.14	0.12	0.13
Not Helpful	0.47	0.48	0.37	0.35	0.33	0.33	0.14	0.13	0.12	0.14
All	0.46	0.48	0.37	0.36	0.32	0.33	0.15	0.13	0.12	0.13

Table 15: Statistics of Feature Group Special characters

Arithmetic Mean	***# exclamation marks***		***# question marks***		***# quote marks***	
	C1	***C2***	***C1***	***C2***	***C1***	***C2***
Helpful	0.51	0.97	0.12	0.33	0.73	1.39
Not Helpful	0.43	0.99	0.07	0.28	0.27	0.67
All	0.49	0.98	0.1	0.3	0.58	1.02

Table 16: Statistics of Feature Group Non-English words and Amplifiers

Arithmetic Mean	***Proportion of non-English words***		***Proportion of intensifier***		***Proportion of downtoners***	
	C1	***C2***	***C1***	***C2***	***C1***	***C2***
Helpful	46.61	17.52	0	0	0	0
Not Helpful	100.59	23.49	0	0	0	0
All	64.41	20.61	0	0	0	0

4.4 Implementation of Polarisation Features

In this project OpinionFinder was utilized in batch-mode to detect polarisation which is based on the phrase-level polarity classifier described in the work of (Wilson, et al., 2005). Initially a document list is generated during the pre-processing phase in Python. A document with the review text was generated for each review. Then they are passed to OpinionFinder for processing and polarity classification. OpinionFinder determines the polarity by considering at the beginning words with a prior polarity (for example, “love”, “hate”, “think”) extracted from a lexicon of over 8.000 subjectivity clues and then uses a modified version of the polarity classifier to identify the contextual polarity (Wilson, et al., 2005). After executing the software the output is written to files as default located in the path opinionfinderv2.0\database\docs. For each review a folder named *_auto_anns (e.g. reviewfile1_auto_ans) is created containing the automatic annotations. The file *exp_polarity* contains the polarity for single phrases occurring in the reviews. The following table contains example negative, positive and neutral annotated review texts from the computer and Videogames dataset. The table shows the review text with the highlighted clues, the polarity result provided by OpinionFinder and a final calculated polarity score summarising the OpinionFinder result. Based on the results of OpinionFinder, the first step is to calculate the percentage of the polarity categories positive, neutral and negative. Since an overall tendency summary of the polarity is not provided by OpinionFinder, a final polarity summary was calculated for each review.

The chosen approach for this project to determine the overall tendency of the polarisation is implemented according to the following rules:

1) *The maximum value of the polarity result determines the overall score.*
2) *If two polarity categories (positive/ neutral or negative/ neutral) have the same distribution, either the polarity positive or negative is assigned and the polarity neutral is ignored.*
3) *If the polarity categories positive and negative are equally distributed, the polarity neutral is assigned to the review.*

Examples

Review 1

*I bought this mic in December and after only two nights of singing, it completely died! What a complete **disappointment**. Save your money!*

Table 17: Example Review 1 OpinionFinder results

Calculated percentage	**Polarisation tendency summary**	
positive: 0% , neutral: 50%, negative: 50%	-1	**negative**

Review 2

*Yes this game is an easier fighting game, but that's why I like it personally. In most Marvel fighting games, there are so many controls to remember for activating many different types of superpowers that it got hard to keep em straight. With this game having only 2 basic fight options for all characters it is much easier to figure out what you are doing. Of course, each character has it's little extras such as flight for Storm and wall climbing for Spidey, so each character remains unique. But you don't have to memorize 10 different actions for Wolverine and then memorize 10 different controls for Venom. I **like** this game a lot and I **really love** that they have made the fighting controls **easier**.*

Table 18: Example Review 2 OpinionFinder results

Calculated percentage	Polarisation tendency summary	
positive: 75%, neutral: 25%, negative: 0%	1	**positive**

Review 3

A simple DVD player, as long as you or a friend own an X-box console. Simply push in a little plastic knob into the controler socket and their you go oyu know own a standard of the line DVD player.The maginfacation is great the rewind and fastfrward is great.It has all the extras a normal DVD player has and a easy to operate menu and layout.No overloads of pointless buttons just point out the obvious structer.You may be dissapointed in the response time, but that is not entirely the remote or players fault.If you own an old x-box console or a low memory X-box console you may encounter slowness and maxed out freezing.Some mint condition titles may not even work.(I purchased "Little Nicky" sometime ago mint and to this date it still does not work on my X-box console)Batteries may play a roll. With out a ***doubt though*** this is ***the most easy*** and ***portable*** DVD ***player*** around with the ***acknowledgement*** of owning an X-box console first.A ***must*** have for a gamer/movie ***lover*** wit ha ***large*** movie library or a constant renter

Table 19: Example Review 3 OpinionFinder results

Calculated percentage	Polarisation tendency summary	
positive: 80%, neutral: 10%, negative:10%	0	**neutral**

Applying OpinionFinder in this project to detect the polarity in the online reviews has demonstrated strength and weaknesses of the software tool, which are summarised in the table below.

	Strength	Weakness
Automatic Annotation	• automatic annotation is conducted on sentence /phrase-level since this is important when mining reviews and analysing product related information (Morinaga, et al., 2002) [39].	• a summarisation of the polarity per document is not provided and has to be calculated
Algorithm	• OpinionFinder is based only on a prior-polarity subjectivity lexicon containing over 8,000 tagged clues with polarities (positive, negative, both or neutral)	• the already trained polarity classifier identifies and classifies the polarity of new instances based on its prior polarity. This may lead to bias because the polarity is context sensitive for example dependent on the

		domain, topic or the author himself.
Output	• per review a file is generated with an identifier for a phrase that has been annotated	• the output is difficult to read because only the character positions of the expressions with the polarity are given and not the actual words that lead to the result • not all reviews are annotated

Table 20: Strength and Weakness of OpinionFinder

Distribution of polarity in the dataset

The following examples demonstrate the results for the Computer & Videogames and Camera dataset: The graphic illustrate the number of reviews according to their polarity *(0 = neutral, 1 = positive and -1 = negative)* and helpfulness. It becomes obvious that both helpful and not helpful reviews are nearly equally distributed among all polarity categories for the computer & Videogames data set. This indicates that polarity might not be a strong feature for this dataset. In contrast the camera dataset the number of reviews for each polarity category with respect to helpfulness varies slightly. Despite, for both datasets polarisation features tend not to be strong.

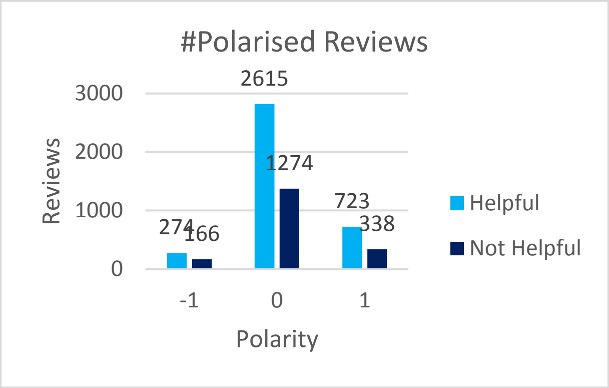


Figure 7: #polarised Reviews for Cameras

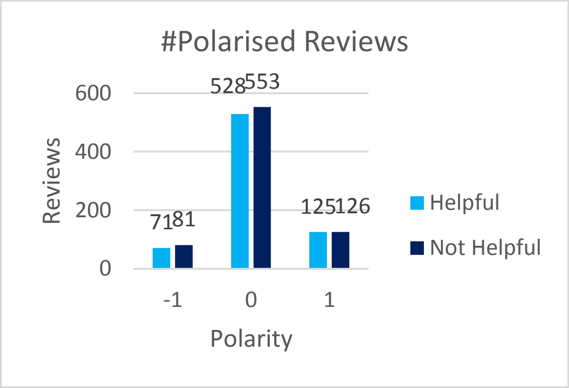


Figure 6: #polarised Reviews for Computer & Videogames

4.5 Implementation of Informativity

The implementation for the informativity features have been conducted using the bag of words approach. Prior to calculating the features data pre-processing was required to convert the words into lowercase and to remove stop words. Relevant features have been calculated by

- counting the most frequent words in the corpus
- counting the number of times a product name was used in a review
- counting the number of words that match the vocabulary in the ontology and
- counting the product attributes that were regarded as important based on Human Computation.

Most frequent words

The 50 most frequent words of the dataset were obtained using the frequency distribution functions of NLTK. A list of the 50 most common words was generated to match against the review text and count their frequencies.

Frequency of Product names

A review is considered as informative if it mentions the product itself. Initially a list of unique products was created by extracting the data from the product name field in the given Amazon dataset. The camera data set contains reviews about 937 products. The computer and videogame dataset contains in total 205 products. Then, the frequency of the product name mentioned in the review text was counted.

Ontology-based feature identification

Ontology provides vocabulary about different domains. Additional to product names product attributes can be retrieved by using data properties from ontology. Therefore ontology for digital cameras created by the E-Business + Web Science Research Group was utilized to extract 96 data properties describing attributes of a camera (E-Business + Web Science Research Group, 2015).

Human Computation based

In this project human computation was used to create a bag of words on computer and videogame product data containing n-gram.

10 participants among MSc Computer Science students and individuals with expertise in computer and videogames were asked to annotate each 10 reviews. The task was to specify maximum 10 words per review that describe the product's attributes that are relevant for a purchase decision.

Example: *This is a very flashy, very fun car chase shooting game. You get over 20 different vehicles and guns. The TV-like gameplay makes for a good story line. There's hardly any dialogue in the cut scenes, but this was one of the first PS2 games. Gran Theft Auto:$50. Driv3r:$30. Starsky and Hutch: priceless*

Annotated words: *car chase; shooting game; vehicles; guns; TV-like; story line; dialogue; cut scenes;PS2; Gran Theft Auto;*

The table below summarises the informativity features implemented in this project. For experimental purposes the camera dataset considers along with number of most frequent words (#most freq. words) and number of product names (#product names) the number of ontology words (#ontology words) whereas the computer and videogames dataset the number of human computed (#human comp. words) words are focussed.

Table 21: Statistics of informativity features for camera

Arithmetic Mean	***#most freq. words***	***#product name***	***#ontology words***
Helpful	*47.35*	*0.002*	*0.34*
Not Helpful	*25.25*	*0.002*	*0.18*
All	*39.99*	*0.002*	*0.29*

Table 22: Statistics of informativity features for computer & videogames

Arithmetic Mean	***#most freq. words***	***#product name***	***#human comp. words***
Helpful	*65.80*	*0.45*	*30.87*
Not Helpful	*41.07*	*0.22*	*18.62*
All	*52.98*	*0.33*	*24.65*

4.6 Model building

After a feature space has been constructed, the next step is to build the prediction model. In the modelling phase of the presented methodology certain steps are required which will be described as follows.

Construct Gold Standard: In this work the gold standard is constructed from the given corpus itself. The prediction class for the dataset is binary - whether a review is helpful (1) or not helpful (0). The challenge is to find data in the corpus where this information can be derived. Based on the given Helpful Rating of each review describing how many customers find a particular review helpful, a ratio was calculated as first step. Records that do not have a Review text or Helpful Rating are ignored during the data pre-processing. Secondly a threshold of 0.5 was set stating if more than 50 percent of the reviewers find a review helpful than it is considered as helpful. Every review is flagged with as helpful or not according to the rule: *If the ratio is greater than 0.5 a helpful flag is set to 1, 0 otherwise.*

For example the Helpful Rating would provide the information *3 of 10* find this review helpful. The ratio of *0.3* is calculated and the helpful flag is set to 0 since it is below the threshold.

Evaluation methods: Training and test set are required to evaluate the performance of the learned model. Therefore two third of the data set is used as training and one third as test set. In this project cross-validation is used and additionally a test set is supplied. The table below gives an overview about the dataset size for training and testing:

Table 23: Training and Test data set size

	Camera	Computer & Videogames
Training set	*3802*	*960*
Test set	*1902*	*481*

Pre-processing in WEKA: To process the data in WEKA as input for the learning methods pre-processing of the data is required. Removal of the unique_id, conversion of numerical attributes to nominal where applicable (e.g. fields: Final_Polarity_score, Rating, Review_Helpful_flag (class)). For clustering with K-Means it is necessary that the input data is numeric since the cluster centroids are calculated using the distance measure, such as Euclidean distance. If a feature is nominal a calculation is not possible without further adjustments. Therefore these features were not converted for the clustering experiments. However this is an important issue to consider when interpreting the results. Moreover, normalisation was applied due to the fact that many different features working on different scales may influence the algorithm's outcome. For certain learning algorithms, such as K-Means or SVM it is particularly important to perform normalisation in order to calculate the distance and hence run correctly.

Baseline Algorithm: To evaluate the different machine learning methods and get the accuracy estimate a baseline is needed for comparison. A baseline is significant for a data analytics project since it tells whether any improvements can be achieved by changing parameters of the learning algorithms. Generally many different algorithms and parameter changes are applied until a near “optimal” solution can be achieved. Different baselines might have to be set for different problem types (clustering, classification and regression. There are two possible approaches to overcome the challenge to find a good baseline. Firstly, a previous existing work which has similarities to the own work can be taken as a baseline. Secondly, a baseline result can be calculated from the own data set. Since in many cases each project is rather unique and might have specific requirements like in this project, the second approach is chosen using Naïve Bayes as baseline algorithm. As explained in section 2.2.6 Naïve Bayes was used prior in many researches in text classification as its strength is its simple independence assumption of features, speed and less computational expensive. For regression experiments the mean value is taken along with k=2 clusters for the clustering experiments.

5 Experimental Results and Evaluation

This chapter summarises the findings of the evaluation step of the project methodology. After a feature space and the model have been constructed, the next step is to compare the performance among the selected features by conducting experiments following the project methodology. These experiments are divided into clustering experiments, classification experiments and regression experiments. By evaluating the comparisons, the most suitable methods are found to detect helpfulness of online reviews. In the following sections, each experiment is given by initially describing the problem that the experiment is to evaluate, a description of the experimental setup, an analysis after performing the experiment and finally an evaluation of the model is given. Finally, this chapter discusses the contributions and findings of this project.

5.1 Feature Selection and Learning Algorithms

Prior to performing the experiments it is crucial to evaluate the features itself and to perform feature selection in order to reduce the feature space and mitigate the risk of overfitting caused by a high number of features. This problem is referred as the *curse of dimensionality* (Bishop, 2006). The number and choice of features as input for the learning algorithms can influence the algorithm's performance. In this project the Information Gain (IG) was calculated using WEKA in order to select relevant features with a high IG score. The Information Gain is a popular feature selection method in machine learning to measure the worth of a feature by measuring the information gain with respect to the class. Features below a threshold (IG equal to zero) are ignored to build the model. A ranked list of features with their calculated IG is provided in Appendix D WEKA.

As previously assumed (chapter Implementation of Polarisation Features4.4), the polarisation features are not very strong and show an IG of zero which affirms the assumption. Nevertheless, the polarisation as criterion to detect quality in reviews was not ignored but rather used as control feature for analysis in combination with the readability and informativity features. Moreover, the classification Experiment 2 using solely polarity features as input emphasises that these features are not strong .This table summarises the selected algorithms for the following experiments in WEKA:

Table 24: WEKA learning algorithms used in the project

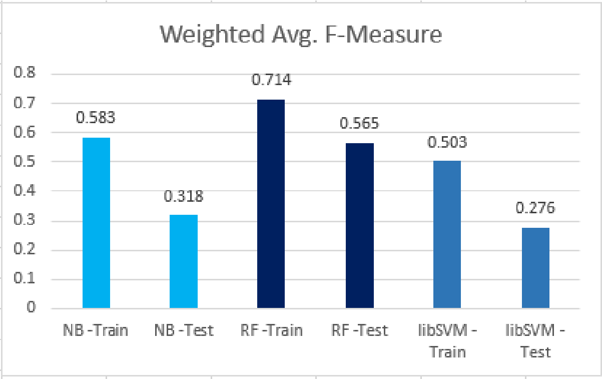
Algorithm	Usage in Project
Naïve Bayes	Baseline algorithm
SimpleKMeans	Clustering
Random Forest	Classification
LibSVM	Classification

5.2 Classification Results

Experiment 1: Readability features ignoring features with IG equals zero

In this experiment all readability features were selected excluding the features with an IG equal to zero for experimentation. The goal is to check the performance by reducing the feature space. A reduction from 31 to 28 features for the computer & videogames dataset and 21 features for the camera dataset was undertaken (see Appendix D WEKA sections on Information Gain).

***Result**:* The prediction model results illustrated in the figures below show that for both datasets the Random Forest (RF) algorithm performs the best for both training and test among the chosen algorithms.



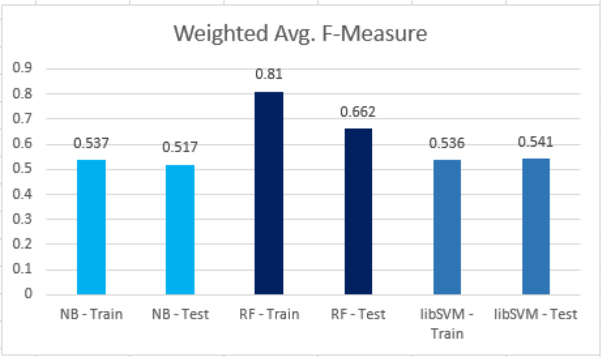


Figure 8: Experiment 1 Computer & Videogames **Figure 8: Experiment 1 Camera**

***Analysis**:* RF outperforms the baseline algorithm Naïve Bayes (NB) and SVM with a weighted average F-Measure (F1-Measure) of 0.714 (training set) using cross-validation and 0.565 (test set) for the computer & videogames dataset. Similarly on the camera dataset RF performs best and achieves a weighted average F-Measure of 0.81 (training set) and 0.662 (test set).

According to the confusion matrix for the RF classifier which was obtained using cross-validation on the camera dataset, it is striking that 752 instances are correctly classified as

“Not Helpful” reviews and 2356 instances are labelled correctly as “Helpful”. 187 instances are incorrectly classified as “Not Helpful”. Also, 507 instances are incorrectly labelled as “Helpful”. 81.75% of the test instances are classified correctly which indicates overall a good result with the given features. By applying a test set for evaluation (Appendix D) only 66.30% instances were classified correctly (F1-Measure = 0.662).

Table 25: Confusion Matrix Experiment 1 - Camera dataset

	Predicted Class		
Actual Class	Not Helpful (0)	Helpful (1)	
Not Helpful (0)	*TP = 752*	*FP = 507*	***1259***
Helpful (1)	*FN = 187*	*TN = 2356*	***2543***

For computer and videogames data set 415 instances are correctly classified as “Not Helpful” reviews and 272 instances are labelled correctly as “Helpful”. 117 instances are incorrectly classified as “Not Helpful” and 156 instances are incorrectly labelled as “Helpful”.

Table 26: Confusion Matrix Experiment 1 - Computer dataset

	Predicted Class		
Actual Class	Not Helpful (0)	Helpful (1)	
Not Helpful (0)	*TP = 415*	*FP = 117*	***532***
Helpful (1)	*FN = 156*	*TN = 272*	***428***

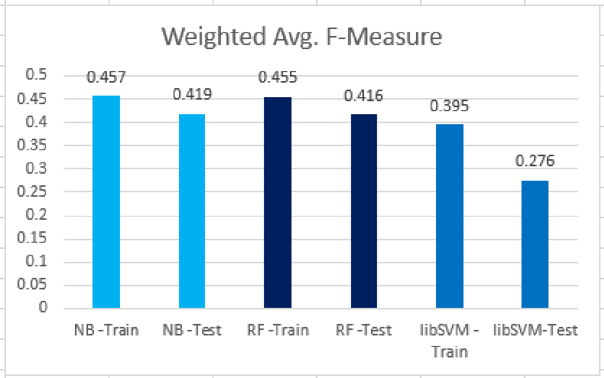
Table 27: Weighted Average Metricises for RF

	TP Rate	**FP Rate**	**Precision**	**Recall**
Computer	0.716	0.3	0.715	0.716
Camera	0.817	0.294	0.816	0.817

Experiment 2: Polarity features

In this experiment solely the polarity features are selected as input. For both datasets the features: Positive ratio, Neutral ratio, Negative ratio, Subjectivity ratio, Final polarity score and the class feature are chosen. The goal is to determine if the polarisation features have an influence on the prediction of the helpfulness of a review.

Result: The prediction model results illustrated in the figures below show that NB performs best for the computer & videogames data set whereas RF is the preferred algorithm for the camera data set when comparing the weighted average F-Measure.



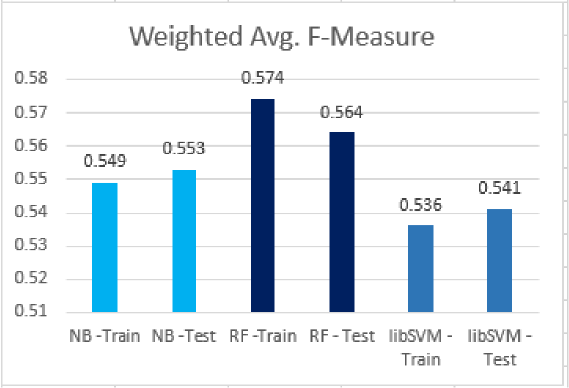


Figure 9: Experiment 2 Computer & Videogames

Figure 10: Experiment 2 Camera

Analysis: The prediction model results for the computer & videogames data set illustrates that the baseline algorithm is not outperformed. NB performs slightly better than RF with a weighted average F-Measure of 0.457 using cross-validation evaluation. For the camera data set RF outperforms the baseline. However for both data sets the difference between the algorithms is not immense. An overview of the confusion matrix values for both data sets is provided in Appendix D. It is striking that for the camera data set the number of false positives is significantly higher than the true positives for the class “Not Helpful”. This is shown by the TP Rate for class “Not Helpful = 0” is below 0.06 (RF) , 0.03 (NB) down to 0 (for libSVM). In contrast for the computer & videogame data set, a similar occurrence can be observed for the true negatives (correctly classified as “Helpful = 1”) where the false negatives are significantly higher. In this case the most instances are correctly classified as not helpful reviews but only very few are correctly classified as helpful.

To conclude, polarisation features show poor results on both training and test sets where the F-Measure is for both data sets approximately 0.5 meaning that only around 50% of the instances are correctly classified. However a tendency towards one class value is visible in both data sets where a correct classification is conducted. Therefore further experiments are considered as necessary. Additionally to an analysis of the whole corpus, the corpus was further split into positive, neutral and negative polarised reviews. Therefore polarity is used as control feature in combination with other features from readability and informativity for the following analysis.

Experiment 3: Readability features with polarity as control feature

The goal of this experiment is to see whether readability has an impact on the helpfulness of a positive, neutral or negative review. For these experiments the data set is split into positive, neutral and negative reviews by using the WEKA filter *RemoveWithValues*. All readability features were taken as input. Cross-validation was applied for evaluation due to the fact that the pool of test instance per polarity was not large enough to conduct experiments on a test set.

Results: This experiment is conducted for the computer and videogames data set. The tables below show the evaluation results for positive, neutral and negative reviews. All readability features as described in chapter 4.3 are considered for experimentation.

Table 28: Experiment 3 positive Reviews computer & videogames

	Precision	**Recall**	**F-Measure**	**Correctly classified instances**
NB	0.627	0.625	0.624	62.55%
RF	**0.825**	**0.825**	**0.825**	**82.47%**
libSVM	0.722	0.717	0.715	71.71%

Table 29: Experiment 3 neutral Reviews computer & videogames

	Precision	**Recall**	**F-Measure**	**Correctly classified instances**
NB	0.56	0.557	0.543	55.69%
RF	**0.825**	**0.823**	**0.823**	**82.33%**
libSVM	0.748	0.748	0.748	74.84%

Table 30: Experiment 3 negative Reviews computer & videogames

	Precision	**Recall**	**F-Measure**	**Correctly classified instances**
NB	0.617	0.599	0.562	59.87%
RF	**0.77**	**0.763**	**0.76**	**76.32%**
libSVM	0.711	0.691	0.677	69.08%

Analysis: The experiments have shown that the chosen algorithms perform generally better when using polarity as a control feature by detecting helpfulness in a pool of positive, neutral or negative reviews. Random Forest yields the best results obtaining F-Measure around 0.82 using readability features for positive and neutral reviews and 0.76 for negative reviews. It outperforms both the baseline and libSVM. The advantage of using RF is that a prior feature selection is not mandatory since the features to be used are chosen at random. The disadvantage is that it not clear which features have been chosen as input as it works like a black box.

Experiment 4: Informativity features with polarity as control feature

Experiment 3 has yielded good performance results for all readability features by using polarity to split the data set and determine helpfulness for positive, neutral and negative reviews separately. This experiment focusses on informativity features (chapter 4.5) as input and examines similarly the performance of the learning algorithms on positive, neutral and negative reviews.

Results for camera data set: The tables below show the evaluation results for positive, neutral and negative reviews considering number of most frequent words, number of ontology words and number of product names (see chapter 4.5) as input features.

Table 31: Experiment 4 positive Reviews camera

	Precision	**Recall**	**F-Measure**	**Correctly classified instances**
NB	0.626	0.498	0.497	49.81%
RF	0.688	0.697	0.691	69.74%
libSVM	**0.694**	**0.71**	**0.692**	**71.03%**

Table 32: Experiment 4 neutral Reviews camera

	Precision	**Recall**	**F-Measure**	**Correctly classified instances**
NB	0.662	0.521	0.518	52.05%
RF	0.668	0.691	0.662	69.15%
libSVM	**0.673**	**0.695**	**0.649**	**69.46%**

Table 33: Experiment 4 negative Reviews camera

	Precision	**Recall**	**F-Measure**	**Correctly classified instances**
NB	0.662	0.521	0.518	52.05%
RF	0.664	0.667	0.665	66.71%
libSVM	**0.678**	**0.686**	**0.676**	**68.59%**

Results for computer and videogames data set: The tables below show the evaluation results for positive, neutral and negative reviews considering number of most frequent words, number of human computed words and number of product names (see chapter 4.5) as input features.

Table 34: Experiment 4 positive Reviews computer & videogames

	Precision	**Recall**	**F-Measure**	**Correctly classified instances**
NB	0.63	0.605	0.564	60.53%
RF	0.798	0.796	0.795	79.61%
libSVM	**0.799**	**0.796**	**0.796**	**79.61%**

Table 35: Experiment 4 neutral Reviews computer & videogames

	Precision	Recall	F-Measure	Correctly classified instances
NB	0.598	0.563	0.498	56.35%
RF	**0.734**	**0.734**	**0.733**	**73.34%**
libSVM	0.7	0.7	0.7	70.02%

Table 36: Experiment 4 negative Reviews computer & videogames

	Precision	Recall	F-Measure	Correctly classified instances
NB	0.625	0.62	0.569	61.95%
RF	0.711	0.712	0.711	71.22%
libSVM	**0.764**	**0.761**	**0.755**	**76.01%**

Analysis: The experiments for the computer & videogames data set show that the libSVM algorithm performs slightly better than RF for positive and negative reviews with an F-Measure of 0.796 for positive reviews where 79.61% reviews have been classified correctly and 0.755 for negative reviews where 76.01% reviews have been classified correctly. For neutral reviews RF shows better results. Both algorithms clearly outperform the baseline.

5.3 Clustering Results

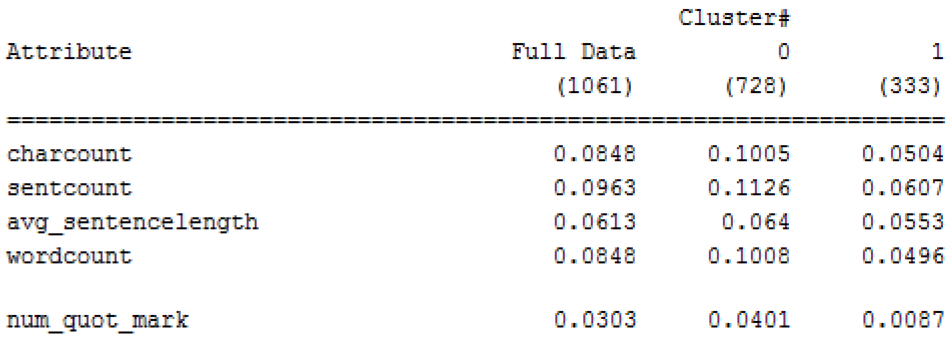
In this project we will not apply clustering to find class labels. Clustering is used to find patterns when forming the clusters. Therefore data set to be examined is spitted into positive and negative reviews as these two polarity categories express subjectivity and hence the most interesting to analyse.

Experiment 5: Clustering on positive reviews

This experiment investigates positive reviews from the camera data set using the SimpleKMeans algorithm. The algorithm uses Euclidean distance as distance function trying to find a minimum for the SSE. The parameters number of cluster: 2 and 10 seeds for clustering are set. The goal is to determine which features are accountable for the cluster formation. The class attribute describing the helpful flag was omitted for the following experiment. Moreover, normalisation was applied.

Results: The results presented in Appendix D WEKA under the section Clustering show that two clusters were produced within 5 iterations. The sum of squared errors (SSE) within clusters is 347.76. An evaluation was undertaken on the training set where 69% of the instances were clustered into cluster 0 (helpful) containing 728 instances and 31% (333 instances) into cluster 1 (not helpful).

Analysis: The results in Appendix D WEKA show that especially the length readability features perform the best by showing significant variation between the two clusters. Also the number of quotation marks (linguistic readability feature) has an influence on the helpfulness. This indicates that positive reviews that consist of many characters, words, sentences or quotation marks are considered as helpful as shorter reviews. Readability indices or readability linguistic features do not show any significant differences between helpful and not helpful positive reviews.



```
                                             Cluster#
Attribute                          Full Data          0          1
                                      (1061)      (728)      (333)
==================================================================
charcount                             0.0848     0.1005     0.0504
sentcount                             0.0963     0.1126     0.0607
avg_sentencelength                    0.0613      0.064     0.0553
wordcount                             0.0848     0.1008     0.0496

num_quot_mark                         0.0303     0.0401     0.0087
```

Figure 11: Clustering Experiment 1 WEKA output

Experiment 6: Clustering on negative reviews

This experiment investigates negative reviews from the camera data set using the SimpleKMeans algorithm. The algorithm uses Euclidean distance as distance function trying to find a minimum for the SSE. The parameters number of cluster: 2 and 10 seeds for clustering are set. The goal is to determine which features are accountable for the cluster formation. The class attribute describing the helpful flag was omitted for the following experiment. Moreover, normalisation was applied.

Results: The results presented in Appendix D WEKA under the section Clustering show that two clusters were produced within 6 iterations. The sum of squared errors (SSE) within clusters is 193.38. An evaluation was undertaken on the training set where 37% of the instances were clustered into cluster 0 (not helpful) containing 163 instances and 63% (277 instances) into cluster 1 (helpful).

Analysis: The results in Appendix D WEKA (Experiment 6: Negative reviews camera dataset) show that helpful reviews within the negative reviews are length wise longer by examining the number of characters, words or sentences. For instance sentence count show a deviation value of 0.0801 for cluster 1 and 0.0413 for cluster 0 which demonstrates that cluster 1 (helpful) contains reviews with more sentences and therefore these reviews tend to be longer. Also, the proportion of downtoners is lower for helpful negative reviews than for not helpful negative reviews. Readability indices indicate no significant differences between the two clusters as well as the proportion of nouns, proper nouns, common nouns or adjectives. Therefore these features do not perform well on the given data set. The proportion of adverbs and the number of not English words that do not exist in Wordnet are slightly higher for not helpful negative reviews than for helpful reviews.

5.4 Summary

The goal of the presented experiments is to show whether the features for readability, polarity and informativity can predict helpfulness in online reviews. The results show that the performance of the chosen learning algorithms is dependent from the selected features and the data set. The experiments have been conducted for different feature combinations and two different data sets containing reviews for computer and videogames and cameras and photo.

Both, supervised for classification and unsupervised clustering methods have been applied. Initially feature selection was applied to reduce the feature space and to avoid overfitting. Information Gain was used as preferred method. However many more feature selection methods exist to experiment with to obtain only relevant features.

Generally Random Forest (RF) performs the best amongst the chosen algorithms followed by libSVM in this project. This RF classifier itself performs feature selection as it selects a certain number of random features as input. However RF resembles a black box where it is not clear from the WEKA output which features are chosen

Polarisations as a feature did not perform well as in the initial feature selection state these features having an Information Gain of zero were excluded to reduce the feature space. By doing experiments solely with polarity features confirmed the initial assumption in Experiment 2 of the classification experiments. Therefore polarity was used as control feature splitting the reviews into positive, neutral and negative. Further analysis on these data sets demonstrated better results in combination with readability and informativity features. Readability features tend to perform better than informativity features. In particular length related and linguistic readability features outperform readability indices which do not appear to have an effect on the helpfulness. The reason why informativity did not outperform readability could result from the fact that the implementation of the features could be improved by state-of-the-art solutions especially in the field of ontology based and human computation feature extraction to achieve better accuracy results.

Further experiments may be done to test different feature combinations and more algorithms. Feature selection in this context becomes also very important when different algorithms, such as tree-based algorithms (e.g. C 4.5 tree) are applied. Numerous features generate a large tree with possible irrelevant features. This may lead to higher error rates and consequently influence the overall accuracy of the classifier.

6 Conclusion

In this chapter the overall project is concluded by providing a summary and evaluating the challenges, achievements. Finally, this chapter describes further enhancements of the implementation and any future work that may be set to continue this project. This chapter concludes with a personal reflection.

6.1 Project Summary

The goal of this project was to automatically detect quality and thereby helpfulness in online reviews using machine learning algorithms. The project was conducted by following the CRISP-DM methodology. Initially the challenge was to scope the project and to obtain a good understanding of the project. Therefore three core quality criteria were identified after conducting a literature review in order to gain a comprehensive business understanding and the requirements of this project. Readability, polarisation and informativity focusing solely on the review text were selected as features to be implemented. The process of feature identification is a very crucial one since all following steps and results are dependent.

Prior to the feature implementation, the challenge was to select a representative data set. A data set from Amazon.com was selected due to its availability and broad product range.

Next pre-processing of the raw data along with the feature calculation has been undertaken in several steps. Then classification (supervised: NB, RF and SVM) and clustering algorithms (unsupervised: K-Means) were used to run experiments with the selected features. A gold standard was constructed as well as evaluation metricises were used to evaluate the performance of the algorithms. The results of the experiments showed that the performance is strongly dependent from the underlying data set and the features chosen.

Overall Random Forest and libSVM performed significantly better than Naïve Bayes which was used as baseline algorithm. Regarding the feature choice the readability features tend to perform better than the informativity features in this project.

6.2 Achievements of goals and objectives

1) *Identification of appropriate criteria to determine the quality of online reviews in general and with respect to the domain of online reviews.*

   Readability, polarisation and informativity were selected as corpus based criteria to detect quality amongst all reviewed criteria in this project by undertaking a literature review in chapter 2.1.

2) *Identification of an appropriate product review corpus to conduct analysis.*

   This objective was met identifying the Amazon data set as suitable data set to conduct experiments. Chapter 3 provides the justification as well as a detailed description about the data set along with some limitations.

3) *Identification of relevant features for the selected quality criteria.*
   An overview about all identified and implemented features for readability, polarisation and informativity in this project are given in chapter 4. Descriptions as well as details about their calculations are specified in detail and thereby this objective was met.

4) *Identification of machine learning algorithms to detect the quality of online reviews in e-Commerce based on specified quality criteria.*

   In chapter 2.3 machine learning algorithms used in related work were presented and identified as suitable for this project. Supervised algorithms, such as Naïve Bayes, Random Forest and Support Vector Machines for classification and the unsupervised clustering method K-Means were used in the experiments.

5) *Compare the performance of selected algorithms on the provided dataset*

   Experimental results and an evaluation are provided in chapter 5 discussing the performance of the chosen algorithms.

6.3 Recommendation for Future Work

Due to limited time and resources this project could not consider further potential extensions. Therefore some indications are given how to develop and enhance this work. Moreover, some potential application areas are presented:

Feature Engineering

- Linguistic Features: The NLTK Stanford POS Tagger could not be used in this project due to limitations of RAM size in order to extract noun phrases, adjective- or adverbial phrases. These features could be relevant in order to see if they influence the helpfulness of a review. Noun phrases could express information about the product itself similar to nouns whereas adjective phrases or adverbial phrases could express a sentiment or describe the product.
- Applying a combination of metadata and corpus related features as this project primarily focussed on the corpus itself. Therefore it is proposed to conduct experiments in this direction since in the numerical rating data might be used as extension to predict helpfulness.
- Extent and implement state of the art solutions for the informativity feature. In particular the usage of ontology and human computation based methods can be

enhanced. Due to time restrictions the implementations for both methods are not completely sound. The selection of a good ontology is a challenging task itself. Moreover, it is recommended to use crowd sourcing platforms in terms of scalability to get possibly a better bag of informative words.

Feature Selection Methods

- Apart from using e.g. the Information Gain to reduce the feature space it is proposed to investigate further feature selection methods.

Modelling

- Further experiments can be conducted by experimenting with different algorithms not presented in this work. Different classifiers, such as decision trees (e.g. C 4.5 tree) could be examined for comparison.
- Regression analysis is a possible extension to classification and clustering where the helpful ratio can be used as the predictive variable.

Applications: There exists the potential of applying the solutions suggested in this project to further review and opinion-based domains such as travel, restaurant, forums or social media platforms. Therefore further domain specific knowledge and adaptions are required.

Business and sales on e-Commerce platforms depend on website visibility. Nowadays, web users mainly use search engines to find content on the web which can be seen as the “single point of entry” to the web content. Therefore, the user wishes to find high quality content to satisfy his search e.g. for products. According to the Forbes article “The Top 7 SEO Trends That Will Dominate 2015”, content marketing is the key driver for search engine rankings. High quality review content is therefore crucial to achieve better visibility. Furthermore, search engine optimisation (SEO) takes on an essential role in this context as well. The manufacturer can optimise their website’s visibility through reviews of good quality. These reviews can be further used for meta- or semantic tagging to link data.

6.4 Personal Reflection

This project enabled me to conduct challenging research work in many different fields of Analytics: Data-/Text Mining, NLP and Machine Learning. In this section I would like to point out my personal experience on the project process, key challenges and lesson learned during the project and conclude by giving recommendations for similar MSc projects.

Project process. In general this project has been very interesting and insightful for me as I learned new methods in text mining and NLP. Moreover, I enjoyed working with new technologies, such as Python and the Sentiment tool OpinionFinder which I might be able to use in my job. Like in many projects there were phases of ups and downs. The challenge is to learn how to overcome an impasse and be persistent in trying to solve these problems. The supervisor meetings were very important to discuss progress but also to address problems. Different perspectives on the topics were provided which were very helpful for the next steps.

Main Challenges and Lessons Learned. Feature Engineering has been a core part of my work. The identification of potential features and the implementation took me more time than expected (Appendix E Project Schedule) as I underestimated the amount of work needed to first understand and select appropriate features. In future I will pay more attention to that fact.

In the middle of the project I encountered few technical problems as I was not familiarised with the Python language and faced performance and memory issues when processing text to calculate the required features. However reading up on unfamiliar topics and testing it by coding the newly learned knowledge helped to overcome the gaps.

Recommendations.

- Planning & Scoping the Project is an essential step to break down the given problem and to determine what research questions can be solved with the given time and resources. Therefore it is important to know exactly what is achievable in this short period of time and to focus on a topic that interest you the most.
- Generally explaining the project to externals (e.g. other MSc Computing students) might also help to see if you have understood the project.
- In cases of problems even technical it is important that problems are raised and communicated proactive to the supervisor as the overall “project manager” at an early stage in order to get help and be on track again. Retrospectively I did this mistake when trying to implement linguistic readability features using the Stanford Tagger. Being fixated on solving this problem I have spent few days without knowing if the features would actually be significant for the project.
- Technical nature. To mitigate performance issue when processing text, it is recommended to process the data files in smaller chunks. Depending on the file size the processing time might take more than one hour. Also, consider of using pickle – Python object serialization concept when dealing with e.g. with pandas data frames in Python. The process of converting an object to a byte stream allows the object to be transmitted and stored and then to be re-built again using the original structure. This is in particular useful when the data has to be processed in several stages from pre-processing to feature implementation which might require several iteration steps. Unfortunately, I found this solution at the very end of my project so that I re-design was not possible anymore which could have optimised the performance.

7 References

CrowdFlower, Inc, 2015. *CrowdFlower.* [Online]
Available at: http://www.crowdflower.com/
[Accessed 17 August 2015].

Agrawal, R. & Srikant, R., 1994. *Fast Algorithms for Mining Association Rules.* [Online]
Available at: https://www.it.uu.se/edu/course/homepage/infoutv/ht08/vldb94_rj.pdf. Date of access: 21.06.2015.

Alias-i, Inc, 2011. *http://alias-i.com/.* [Online]
Available at: http://alias-i.com/lingpipe/
[Accessed 15 August 2015].

Amazon, 2015. [Online]
Available at: http://www.amazon.com/Apple-iPhone-5s-Silver-Unlocked/dp/B00F3J4HCA/ref=pd_sim_107_4?ie=UTF8&refRID=0GHDQKQCGKM68G3J3V0S

Amazon, 2015. [Online]
Available at:
http://docs.aws.amazon.com/AWSECommerceService/latest/DG/RG_Reviews.html

Amazon, 2015. *amazon.com/Panasonic-SC-PM53-Executive-Discontinued-Manufacturer.* [Online]
Available at: http://www.amazon.com/Panasonic-SC-PM53-Executive-Discontinued-Manufacturer/dp/B000F38IQK/ref=cm_cr_pr_product_top?ie=UTF8

Amazon, 2015. *amazonmechanicalturk Artificial Artificial Intelligence.* [Online]
Available at: https://www.mturk.com/mturk/welcome
[Accessed 17 August 2015].

Atwell, E., 2015. *Automatic Mapping Among Lexico-Grammatical Annotation Models (AMALGAM).* [Online]
Available at: http://www.comp.leeds.ac.uk/amalgam/tagsets/upenn.html
[Accessed 15 August 2015].

Baddley, A. D., Thomson, N. & Buchanan, M., 1975. Word Length and the Structure of Short-Term Memory. *JOURNAL OF VERBAL LEARNING AND VERBAL BEHAVIOR ,* Volume 14, pp. 575-589 .

Behnel, S. et al., 2015. *http://lxml.de/elementsoup.html.* [Online]
Available at: http://lxml.de/elementsoup.html. Date of access: 26.06.2015

Biber, D., 1988. *Variation across Speech and Writing.* s.l.: Cambridge University Press.

Bird, S., Klein, E. & Loper, E., 2015. *Natural Language Processing with Python.* [Online]
Available at: http://www.nltk.org/book/ch02.html
[Accessed 15 August 2015].

Bishop, C. M., 2006. *Pattern Recognition and Machine Learning.* Cambridge: Springer.

Blitzer, J., Dredze, M. & Pereira, F., 2007. *Biographies, Bollywood, Boom-boxes and Blenders:Domain adaptation for sentiment classification.* s.l., ACL.

Breiman, L., 2001. Random forests. In: *Machine learning.* s.l.:s.n., pp. 5-32.

Brin, S. & Page, L., 1998. *The Anatomy of a Large-Scale Hypertextual Web Search Engine.* Brisbane, Australia, s.n., pp. 14-18.

Chang, C.-C. & Lin, C.-J., 2014. *LIBSVM -- A Library for Support Vector Machines.* [Online]
Available at: https://www.csie.ntu.edu.tw/~cjlin/libsvm/
[Accessed 27 08 2015].

Chapman, P. et al., 2000. *CRISP-DM 1.0 Step-by-step data mining guide.* [Online]
Available at: http://www.the-modeling-agency.com/crisp-dm.pdf. Date of access: 17.06.2015.

Chen, C. C. & Tseng, Y.-D., 2011. Quality evaluation of product reviews using an information quality framework. *Decision Support Systems,* pp. 755-768.

Cheng, O. K. & Lau, R. Y., 2014. *Towards an Information Theory Based Methodology for the Quality Assessment of Online Comments.* s.l., s.n., pp. 14-19.

Chung, W. & Tseng, T.-L. (., 2012. Discovering business intelligence from online product reviews: A rule-induction framework. *Expert Systems with Applications, Volume 39, Issue 15,* pp. 11870-11879.

Coleman, M. & Liau, T. L., 1975. A computer readability formula designed for machine scoring.. *Journal of Applied Psychology,* 60(2), pp. 283-284.

Cortes, C. & Vapnik, V., 1995. Support-vector networks. In: *Machine Learning Volume 20, Issue 3.* s.l.:Kluwer Academic Publishers, pp. 273-297.

Cutting, D. R., Karger, D. R., Pedersen, J. O. & Tukey, J. W., 1992. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. *SIGIR ‘92,* p. 318 – 329.

Dave, K., Lawrence, S. & Pennock, D. M., 2003 . *Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews.* New York, NY, USA, WWW '03 Proceedings of the 12th international conference on World Wide .

Dredze, M. & Blitzer, J., 2009. *Multi-Domain Sentiment Dataset (version 2.0).* [Online]
Available at: http://www.cs.jhu.edu/~mdredze/datasets/sentiment/
[Accessed 2015 April 2015].

Dredze, M. & Blitzer, J., 2009. *Multi-Domain Sentiment Dataset (version 2.0).* [Online]
Available at: http://www.cs.jhu.edu/~mdredze/datasets/sentiment. Date of access: 16.06.2015.
[Accessed 29 April 2015].

DuBay, W. H., 2004. *The Principles of Readability.* [Online]
Available at: http://files.eric.ed.gov/fulltext/ED490073.pdf
[Accessed 11 August 2015].

E-Business + Web Science Research Group, 2015. *Digital Camera Vocabulary Language Reference.* [Online]
Available at: http://www.ebusiness-unibw.org/ontologies/opdm/digitalcamera.html#DigitalCamera
[Accessed 28 08 2015].

Feng, L., Jansche, M., Huenerfauth, M. & Elhadad, N., 2010. *A comparison of features for automatic readability assessment.* Stroudsburg, PA, USA, COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics: Posters.

Frank-Stromborg, M. & Olsen, S. J., 2004. *Instruments for clinical health-care research.* 3rd ed. s.l.:Sudbury, Mass. : Jones and Bartlett Publishers.

Gartner, Herschel, G., Linden, A. & Kart, L., 2015. *http://www.gartner.com/.* [Online]
Available at:
http://www.gartner.com/document/2987717?ref=QuickSearch&sthkw=Gartner%202015%20Magic%20Quadrant%20for%20Advanced%20Analytics%20Platforms&refval=153816616&qid=218d25e9dbec13bfade6a010a31384e0

Gartner, Randall, L. & Linden, A., 2015. *Why In-DBMS Analytics Deserves a Fresh Look.* [Online]
Available at:
http://www.gartner.com/document/3051917?ref=QuickSearch&sthkw=data%20growth&refval=153786580&qid=4a682eb3981f9420963fed9fade83465

Ghose, A. & Ipeirotis, P. G., 2007. *Designing Novel Review Ranking Systems: Predicting the Usefulness and Impact of Reviews.* s.l., s.n.

Ghose, A. & Ipeirotis, P. G., 2011. Estimating the Helpfulness and Economic Impact of Product Reviews: Mining Text and Reviewer Characteristics. *EEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING,* 23(10), pp. 1498 - 1512.

Greer, J. D., 2009. Evaluating the Credibility of Online Information: A Test of Source and Advertising Influence. *Mass Communication and Society Volume 6, Issue 1,,* pp. 11-28.

Gunning, R., 1969. The fog index after twenty years. *Journal of Business Communication,* 6(2), p. 3–13.

Hai, Z., Chang, K. & Cong, G., 2012. *One Seed to Find Them All: Mining Opinion Features via Association.* s.l., s.n.

Hedman, A. S., 2008. Using the SMOG Formula to Revise a Health-Related Document. *American Journal of Health Education,* 39(1), pp. 61-64.

Huang, J., JingjingLu & Ling, C. X., 2003. *Comparing Naive Bayed, Decision Trees, and SVM with AUC and Accuracy.* s.l., IEEE International Conference on Data Mining.

Hu, M. & Liu, B., 2004 . *Mining Opinion Features in Customer Reviews.* s.l., s.n., pp. 755-760.

Hu, M. & Liu, B., 2004. *Mining and Summarizing Customer Reviews.* New York, USA, s.n., pp. 168-177 .

Hu, N., Bose, I., Koh, N. S. & Liu, L., 2012. Manipulation of online reviews: An analysis of ratings, readability, and sentiments. *Decision Support Systems Volume 52, Issue 3,* p. 674–684.

Jindal, N. & Liu, B., 2007. *Review Spam Detection.* s.l., s.n., pp. 1189-1190.

Jindal, N. & Liu, B., 2008 . *Opinion spam and analysis.* s.l., s.n., pp. 219-230 .

Kincaid, J. P., Rogers, R. L., Liutenant Robert P. Fishburne, J. & Chissom, B. S., 1975. *Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel,* Millington, TN: Research Branch Report 8-75, Naval Technical Training Command.

KNIME.COM AG, 2015. *https://www.knime.org/.* [Online]
Available at: https://www.knime.org/

Koa, A. & Poteet, S. R., 2007. *Natural Language Processing and Text Mining.* London: Springer.

Korfiatis, N., Rodríguez, D. & Sicilia, M.-A., 2008. *The Impact of Readability on the Usefulness of Online Product Reviews: A Case Study on an Online Bookstore.* s.l., Springer-Verlag Berlin, Heidelberg.

Laughlin, G. H. M., 1969. SMOG Grading —– a New Readability Formula. *Journal of Reading,* 12(8), pp. 639-646.

Lau, R. Y. K. et al., 2011 . Text mining and probabilistic language modeling for online review spam detection. *ACM Transactions on Management Information Systems (TMIS) Volume 2 Issue 4.*

Leya, P. & Florio, T., 1996. The use of readability formulas in health care. *Psychology, Health & Medicine,* 1(1), pp. 7-28.

Lim, E. et al., 2010. *Detecting Product Review Spammers using Rating Behaviors.* s.l., s.n., pp. 939-948.

Liu, B., 2012. *Sentiment Analysis and Opinion Mining.* s.l.:Morgan & Claypool Publishers.

Liu, B., n.d. *Opinion Mining.* [Online]
Available at: http://www.cs.uic.edu/~liub/FBS/opinion-mining.pdf. Date of access: 19.06.2016.

llcCallum, D. R. & Peterson, J. L., 1982 . *COMPUTER-BASED READABILITY INDEXES.* New York, NY, USA , ACM '82 Proceedings of the ACM '82 conference.

Malaga, R. A., 2008. Worst practices in search engine optimization. *Communications of the ACM - Surviving the data deluge*, 12 December, pp. 147-150 .

Manning, C. D., Raghavan, P. & Schütze, H., 2008. *Introduction to Information Retrieval.* s.l.:Cambridge University Press.

Marsden, R., 2015. *www.independent.co.uk.* [Online]
Available at: http://www.independent.co.uk/life-style/gadgets-and-tech/features/lets-not-get-too-excited-about-those-fivestar-amazon-reviews-10343401.html

Mayzlin, D., Dover, Y. & Chevalier, J. A., 2014. Promotional Reviews: An Empirical Investigation of Online Review Manipulation. *American Economic Review.*

McCallum, A. & Kamal, N., 1988. *A Comparison of Event Models for Naive Bayes Text Classification.* s.l., AAAI-98 workshop on learning for text categorization.

Menon, A. K., 2009. *Large-scale support vector machines: algorithms and theory. University of California, San Diego.* [Online]
Available at: https://cseweb.ucsd.edu/~akmenon/ResearchExam.pdf. Date of access: 28.06.2015.

Morinaga, S., Yamanish, K., Tateishi, K. & Fukushim, T., 2002. *Mining product reputations on the web.* s.l., s.n., p. 341–349.

mpqa.project@gmail.com, 2015. *OpinionFinder 2.x Release Page.* [Online]
Available at: http://mpqa.cs.pitt.edu/opinionfinder/opinionfinder_2/

NLTK Project, 2015. *NLTK 3.0 documentation.* [Online]
Available at: http://www.nltk.org/
[Accessed 12 August 2015].

O'Mahony, M. P. & Smyth, B., 2010. *Using Readability Tests to Predict Helpful Product Reviews.* Paris, France, s.n., p. 164–167.

Ott, M., Choi, Y., Cardie, C. & Hancock, J., 2011. *Finding Deceptive Opinion Spam by Any Stretch of the Imagination.* Stroudsburg, PA, USA , Association for Computational Linguistics, pp. 309-319 .

Pak, A. & Paroubek, P., 2010. *Twitter as a Corpus for Sentiment Analysis and Opinion Mining.* s.l., In Proceedings of the Seventh Conference on International Language Resources and Evaluation.

Pang, B. & Lee, L., 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval Vol. 2, Nos. 1–2,* p. 1–135.

Pang, B., Lee, L. & Vaithyanathan, S., 2002. *Thumbs up? Sentiment Classification using Machine Learning Techniques.* Stroudsburg, PA, USA, EMNLP '02 Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10.

Peñalver-Martinez, I. et al., 2014. Feature-based opinion mining through ontologies. *Expert Systems with Applications Volume 41, Issue 13,* pp. 5995-6008.

Penn Engineering, 1999. *The Penn Treebank Project.* [Online]
Available at: https://www.cis.upenn.edu/~treebank/
[Accessed 15 August 2015].

Piotrkowicz, A., 2015. *Predicting the social media popularity of news articles from headline text (under review),* s.l.: s.n.

Porter, M., 2006. *The Porter Stemming Algorithm.* [Online]
Available at: http://tartarus.org/martin/PorterStemmer/
[Accessed 15 August 2015].

Python Software Foundation, 2014. *beautifulsoup4 4.4.0.* [Online]
Available at: https://pypi.python.org/pypi/beautifulsoup4
[Accessed 12 August 2015].

Python Software Foundation, 2014. *textstat 0.1.4.* [Online]
Available at: https://pypi.python.org/pypi/textstat/0.1.4
[Accessed 12 August 2015].

Quirk, R., Greenbaum, S., Leech, G. & Svartvik, J., 1985. *A comprehensive grammar of the English language.* New York, USA: Longman.

Rieh, S. Y., 2002 . Judgment of information quality and cognitive authority in the web. *Journal of the American Society for Information Science and Technology Volume 53 Issue 2,* pp. 145-161 .

scikit-learn developers (BSD License), 2015. *scikit-learn Machine Learning in Python.* [Online]
Available at: http://scikit-learn.org/stable/
[Accessed 12 August 2015].

scipy.org, 2015. *scipy.org.* [Online]
Available at: http://scipy.org/
[Accessed 12 August 2015].

Smith, R. & Senter, E., 1967. *AUTOMATED READABILITY INDEX.* s.l.:s.n.

The US-UK Fulbright Commission, 2015. *Fulbright Commission - US School System.* [Online]
Available at: http://www.fulbright.org.uk/study-in-the-usa/school-study/us-school-system
[Accessed 11 August 2015].

Tsaparas, P., Ntoulas, A. & Terzi, E., 2011. *Selecting a Comprehensive Set of Reviews.* s.l., s.n.

Wang, C., Jie, L. & Guangquan, Z., 2005. *A Semantic Classification Approach for Online Product Reviews.* s.l., The 2005 IEEE/WIC/ACM International Conference on. IEEE, 2005..

WEKA The University Waikato, 2015. *weka.wikispaces.com.* [Online]
Available at: https://weka.wikispaces.com/How+do+I+connect+to+a+database%3F

Wilson, T. et al., 2005. *OpinionFinder: A system for subjectivity analysis.* s.l., s.n.

Wilson, T., Wiebe, J. & Hoffmann, P., 2005. *Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis.* Vancouver, Canada, s.n.

Witten, I. H., Frank, E. & Hall, M. A., 2011. *Data Mining: Practical Machine Learning Tools and Techniques.* San Francisco: Morgan Kaufmann Publishers.

Yan, X., Song, D. & Li, X., 2006. *A Study of Concept-based Document Readability in Domain Specific Information Retrieval.* Arlington, VA, USA, s.n.

Ye, Q., Zhang, Z. & Law, R., 2009. Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Systems with Applications,* 36(3), p. 6527–6535.

Zamanian, M. & Heydari, P., 2012. Readability of Texts: State of the Art. *Theory and Practice in Language Studies,* 2(1), pp. 43-53.

Zamir, O., Etzioni, O., Madani, O. & Karp, R. M., 1997. *Fast and Intuitive Clustering of Web Documents*.* s.l., KDD-97 Proceedings.

Zhu, X. & Gauch, S., 2000. *Incorporating quality metrics in centralized/distributed information retrieval on the world wide web.* s.l., s.n., pp. 288-295.

Appendix A
External Materials

For this project the data set was used from Amazon.com containing product data provided by (Dredze & Blitzer, 2009). Moreover, the data set was presented at the Extended Semantic Web Conference (ESWC2015).

Appendix B
Ethical Issues Addressed

No ethical issues apply for this project as no person related data was processed.

Appendix C
Readability

Simple Measure of Gobbledygook (SMOG)

SMOG Conversion Table obtained from (Frank-Stromborg & Olsen, 2004) [54]

# polysyllabic words	Grade Level	# polysyllabic words	Grade Level
0-2	4	73-90	12
3-6	5	91-110	13
7-12	6	111-132	14
13-20	7	133-156	15
21-30	8	157-182	16
31-42	9	183-210	17
43-56	10	211-240	18
57-72	11		

Appendix D
WEKA

Information Gain analysis for Camera data set

```
=== Run information ===
Evaluator:    weka.attributeSelection.InfoGainAttributeEval
Search:weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1
Relation:     final_feature_file-weka.filters.unsupervised.attribute.Remove-R1-
weka.filters.unsupervised.attribute.NumericToNominal-R31-
weka.filters.unsupervised.attribute.Remove-R29-30-
weka.filters.unsupervised.attribute.Remove-R1-5,27-28
Instances:    5704
Attributes:   22
              charcount
              sentcount
              avg_sentencelength
              wordcount
              allword_wordcount
              prop_distinct_words
              Read_Flesch_reading_ease
              Read_Flesch_kincaid_grade
              Read_Coleman_liau_index
              Read_Automated_readability_index
              not_engl_wordcheck
              num_specialchar
              num_questmark
              num_exclmark
              num_quot_mark
              prop_nouns
              prop_adjectives
              prop_adverbs
              Prop_verbs
              prop_propernouns
              prop_commonnouns
              Review_Helpful_flag_x
Evaluation mode:evaluate on all training data
=== Attribute Selection on all input data ===
Search Method:
            Attribute ranking.
Attribute Evaluator (supervised, Class (nominal): 22 Review_Helpful_flag_x):
            Information Gain Ranking Filter
Ranked attributes:
 0.08905    1 charcount
 0.08509    5 allword_wordcount
 0.08509    4 wordcount
 0.06216    2 sentcount
 0.03197    3 avg_sentencelength
 0.03065   20 prop_propernouns
 0.0301    10 Read_Automated_readability_index
 0.02824    6 prop_distinct_words
 0.02498   18 prop_adverbs
 0.02434   19 Prop_verbs
 0.02331    9 Read_Coleman_liau_index
 0.01843    8 Read_Flesch_kincaid_grade
 0.018     17 prop_adjectives
 0.01788   15 num_quot_mark
 0.01713   16 prop_nouns
 0.01647   21 prop_commonnouns
 0.01373   12 num_specialchar
 0.01288    7 Read_Flesch_reading_ease
 0.00369   11 not_engl_wordcheck
 0.00223   13 num_questmark
 0.00217   14 num_exclmark
Selected attributes: 1,5,4,2,3,20,10,6,18,19,9,8,17,15,16,21,12,7,11,13,14 : 21
```

Information Gain analysis for Computer & Videogames data set

```
=== Run information ===
Evaluator:    weka.attributeSelection.InfoGainAttributeEval
Search:weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1
Relation:     final_feature_file-weka.filters.unsupervised.attribute.Remove-R1-
weka.filters.unsupervised.attribute.NumericToNominal-R31-
weka.filters.unsupervised.attribute.Remove-R29-30
Instances:    1484
Attributes:   29
              Positive_ratio
              Negative_ratio
              Neutral_ratio
              Subjectivity_ratio
              Final_Polarity_score
              charcount
              sentcount
              avg_sentencelength
              wordcount
              allword_wordcount
              prop_distinct_words
              Read_Flesch_reading_ease
              Read_Flesch_kincaid_grade
              Read_Coleman_liau_index
              Read_Automated_readability_index
              not_engl_wordcheck
              num_specialchar
              num_questmark
              num_exclmark
              num_quot_mark
              prop_nouns
              prop_adjectives
              prop_adverbs
              Prop_verbs
              prop_propernouns
              prop_commonnouns
              intensifier
              downtoner
              Review_Helpful_flag_x
Evaluation mode:evaluate on all training data
```

```
=== Attribute Selection on all input data ===
Search Method:
        Attribute ranking.
Attribute Evaluator (supervised, Class (nominal): 29 Review_Helpful_flag_x):
        Information Gain Ranking Filter
Ranked attributes:
 0.07564    6 charcount
 0.06636    9 wordcount
 0.06584   10 allword_wordcount
 0.03771    7 sentcount
 0.02085   20 num_quot_mark
 0.02021   23 prop_adverbs
 0.01791   21 prop_nouns
 0.01663   12 Read_Flesch_reading_ease
 0.01643   17 num_specialchar
 0.01423   11 prop_distinct_words
 0.01408   22 prop_adjectives
 0.01407   26 prop_commonnouns
 0.01374   14 Read_Coleman_liau_index
 0.01262    8 avg_sentencelength
 0.01233   25 prop_propernouns
 0.01231   13 Read_Flesch_kincaid_grade
 0.01115   15 Read_Automated_readability_index
 0.00588   24 Prop_verbs
 0.0028    28 downtoner
 0         27 intensifier
 0          4 Subjectivity_ratio
 0          2 Negative_ratio
 0          3 Neutral_ratio
 0          5 Final_Polarity_score
 0         16 not_engl_wordcheck
 0         19 num_exclmark
 0         18 num_questmark
 0          1 Positive_ratio
Selected attributes:
6,9,10,7,20,23,21,12,17,11,22,26,14,8,25,13,15,24,28,27,4,2,3,5,16,19,18,1 : 28
```

Classification Results

Table 37: Experiment 1: Confusion Matrix computer dataset

	Predicted Class		
Actual Class	Not Helpful (0)	Helpful (1)	
Not Helpful (0)	*TP = 153*	*FP = 62*	***215***
Helpful (1)	*FN = 147*	*TN = 119*	***266***

Table 38: Experiment 1: Confusion Matrix camera dataset

	Predicted Class		
Actual Class	Not Helpful (0)	Helpful (1)	
Not Helpful (0)	*TP = 297*	*FP = 325*	***622***
Helpful (1)	*FN = 316*	*TN = 964*	***1280***

Table 39: Experiment 2: Confusion Matrix computer & videogames

	TP	**FP**	**FN**	**TN**
NB -Train	479	53	383	45
NB -Test	165	50	209	57
RF -Train	408	124	354	74
RF -Test	185	30	220	46
libSVM -Train	532	0	428	0
libSVM-Test	215	0	266	0

Table 40: Experiment 2: Confusion Matrix values camera

	TP	**FP**	**FN**	**TN**
NB -Train	41	1218	77	2466
NB -Test	19	603	40	1240
RF -Train	93	1166	97	2446
RF -Test	34	588	55	1225
libSVM -Train	0	1259	0	2543
libSVM-Test	0	622	0	1280

Table 41: Experiment 3 Summary of Confusion Matrix for RF

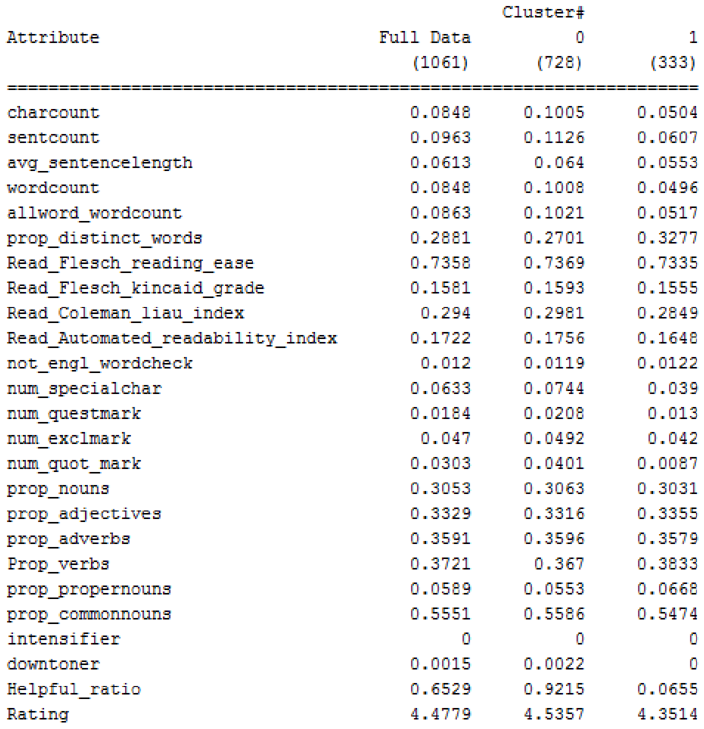
Review	**TP**	**FP**	**FN**	**TN**
positive	103	23	21	104
neutral	476	77	114	414
negative	70	11	25	46

Clustering

Experiment 5: Positive reviews camera dataset

WEKA output for kMeans

```
=== Run information ===

Scheme:weka.clusterers.SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last"
-I 500 -S 10
Relation:     final_feature_file-weka.filters.unsupervised.attribute.Remove-R1-
weka.filters.unsupervised.instance.RemoveWithValues-S1.0-C5-Lfirst-last-
weka.filters.unsupervised.attribute.Remove-R5-
weka.filters.unsupervised.attribute.NumericToNominal-R30-
weka.filters.unsupervised.attribute.Remove-R1-4-
weka.filters.unsupervised.attribute.Remove-R26
Instances:    1061
Attributes:   25
              charcount
              sentcount
              avg_sentencelength
              wordcount
              allword_wordcount
              prop_distinct_words
              Read_Flesch_reading_ease
              Read_Flesch_kincaid_grade
              Read_Coleman_liau_index
              Read_Automated_readability_index
              not_engl_wordcheck
              num_specialchar
              num_questmark
              num_exclmark
              num_quot_mark
              prop_nouns
              prop_adjectives
              prop_adverbs
              Prop_verbs
              prop_propernouns
              prop_commonnouns
              intensifier
              downtoner
              Helpful_ratio
              Rating
Test mode:evaluate on training data
======

Number of iterations: 5

Within cluster sum of squared errors: 347.763766882196

Missing values globally replaced with mean/mode
```



```
                                              Cluster#
Attribute                          Full Data         0          1
                                      (1061)     (728)      (333)
=================================================================
charcount                             0.0848    0.1005     0.0504
sentcount                             0.0963    0.1126     0.0607
avg_sentencelength                    0.0613     0.064     0.0553
wordcount                             0.0848    0.1008     0.0496
allword_wordcount                     0.0863    0.1021     0.0517
prop_distinct_words                   0.2881    0.2701     0.3277
Read_Flesch_reading_ease              0.7358    0.7369     0.7335
Read_Flesch_kincaid_grade             0.1581    0.1593     0.1555
Read_Coleman_liau_index                0.294    0.2981     0.2849
Read_Automated_readability_index      0.1722    0.1756     0.1648
not_engl_wordcheck                     0.012    0.0119     0.0122
num_specialchar                       0.0633    0.0744      0.039
num_questmark                         0.0184    0.0208      0.013
num_exclmark                           0.047    0.0492      0.042
num_quot_mark                         0.0303    0.0401     0.0087
prop_nouns                            0.3053    0.3063     0.3031
prop_adjectives                       0.3329    0.3316     0.3355
prop_adverbs                          0.3591    0.3596     0.3579
Prop_verbs                            0.3721     0.367     0.3833
prop_propernouns                      0.0589    0.0553     0.0668
prop_commonnouns                      0.5551    0.5586     0.5474
intensifier                                0         0          0
downtoner                             0.0015    0.0022          0
Helpful_ratio                         0.6529    0.9215     0.0655
Rating                                4.4779    4.5357     4.3514
```

Experiment 6: Negative reviews camera dataset

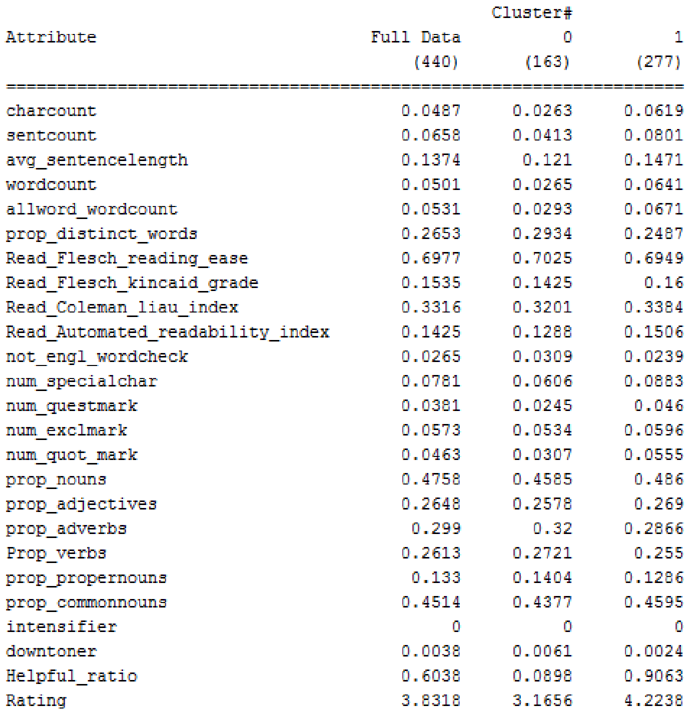
WEKA output for kMeans

```
=== Run information ===

Scheme:weka.clusterers.SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last"
-I 500 -S 10
Relation:     final_feature_file-weka.filters.unsupervised.attribute.Remove-R1-
weka.filters.unsupervised.instance.RemoveWithValues-S0.0-C5-Lfirst-last-V-
weka.filters.unsupervised.attribute.Remove-R1-5-
weka.filters.unsupervised.attribute.NumericToNominal-R26-
weka.filters.unsupervised.attribute.Remove-R26-
weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0

Instances:    440

Attributes:   25
              charcount
              sentcount
              avg_sentencelength
              wordcount
              allword_wordcount
              prop_distinct_words
              Read_Flesch_reading_ease
              Read_Flesch_kincaid_grade
              Read_Coleman_liau_index
              Read_Automated_readability_index
              not_engl_wordcheck
              num_specialchar
              num_questmark
              num_exclmark
              num_quot_mark
              prop_nouns
```

prop_adjectives
prop_adverbs
Prop_verbs
prop_propernouns
prop_commonnouns
intensifier
downtoner
Helpful_ratio
Rating

Test mode:evaluate on training data

Number of iterations: 6
Within cluster sum of squared errors: 193.37514595313002
Missing values globally replaced with mean/mode



```
                                              Cluster#
Attribute                         Full Data          0          1
                                      (440)      (163)      (277)
=================================================================
charcount                            0.0487     0.0263     0.0619
sentcount                            0.0658     0.0413     0.0801
avg_sentencelength                   0.1374      0.121     0.1471
wordcount                            0.0501     0.0265     0.0641
allword_wordcount                    0.0531     0.0293     0.0671
prop_distinct_words                  0.2653     0.2934     0.2487
Read_Flesch_reading_ease             0.6977     0.7025     0.6949
Read_Flesch_kincaid_grade            0.1535     0.1425       0.16
Read_Coleman_liau_index              0.3316     0.3201     0.3384
Read_Automated_readability_index     0.1425     0.1288     0.1506
not_engl_wordcheck                   0.0265     0.0309     0.0239
num_specialchar                      0.0781     0.0606     0.0883
num_questmark                        0.0381     0.0245      0.046
num_exclmark                         0.0573     0.0534     0.0596
num_quot_mark                        0.0463     0.0307     0.0555
prop_nouns                           0.4758     0.4585      0.486
prop_adjectives                      0.2648     0.2578      0.269
prop_adverbs                          0.299       0.32     0.2866
Prop_verbs                           0.2613     0.2721      0.255
prop_propernouns                      0.133     0.1404     0.1286
prop_commonnouns                     0.4514     0.4377     0.4595
intensifier                               0          0          0
downtoner                            0.0038     0.0061     0.0024
Helpful_ratio                        0.6038     0.0898     0.9063
Rating                               3.8318     3.1656     4.2238
```

Appendix E
Project Schedule

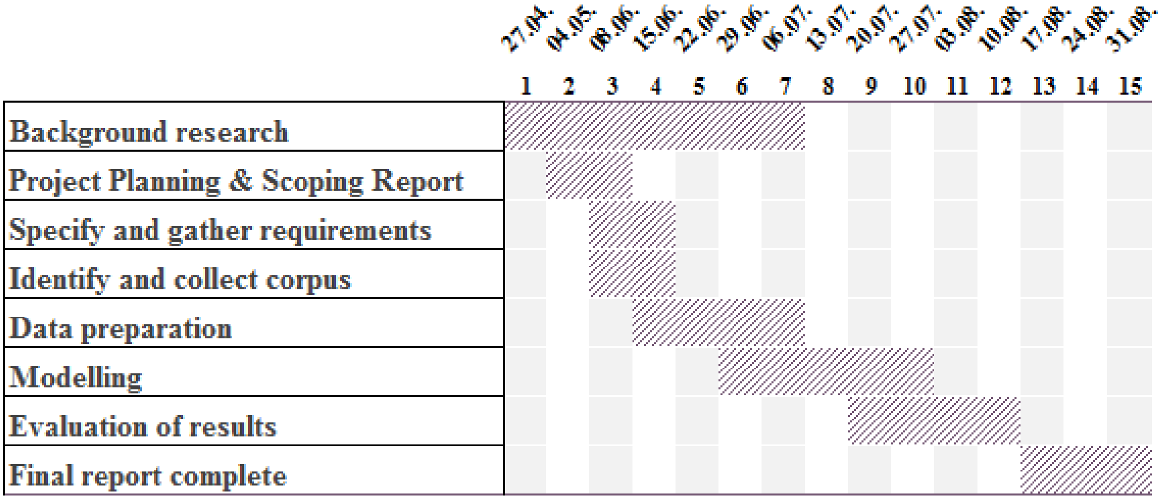


Figure 12: Original Project Schedule

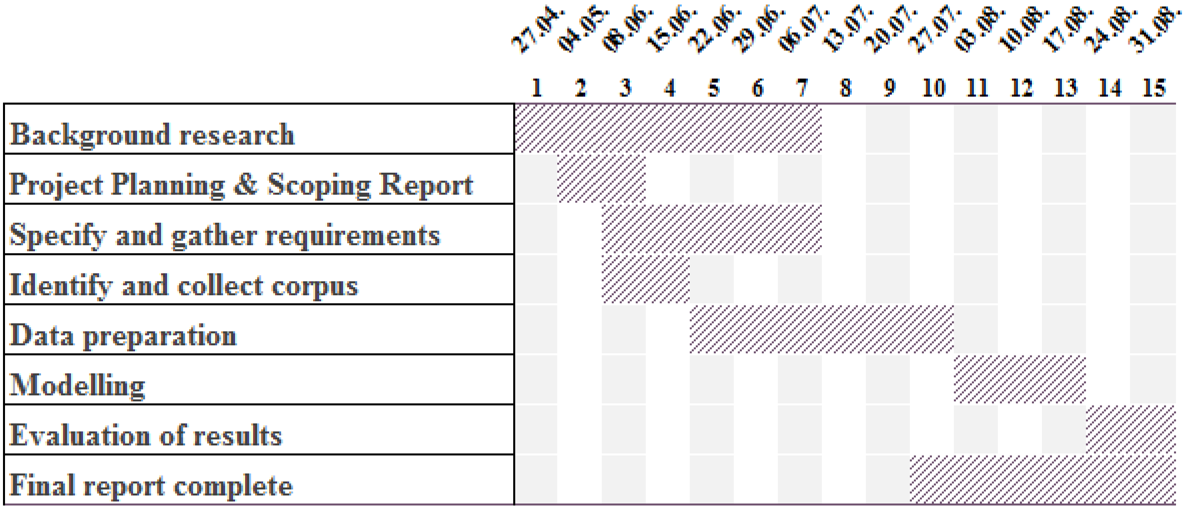


Figure 13: Revised Project Schedule