Subjective machine learning models for restaurants quality evaluation based on user opinions

Bernardo Martínez Parras

UNDERGRADUATE DISSERTATION

DEGREE IN COMPUTER SCIENCE

MAJORING IN Computing

School of Computer Science and Engineering

University of Castilla-La Mancha

Albacete, July 10, 2019
To my family
Authorship Statement

I, the undersigned, BERNARDO MARTINEZ PARRAS with National Identification Number 49214774H declare that I am the sole author of the Undergraduate Dissertation titled Subjective machine learning models for restaurants quality evaluation based on user opinions and that this work does not break the current intellectual property law and that all the non-original material contained in this work is properly attributed to their legitimate authors.

Albacete, July 10, 2019

Signed by: BERNARDO MARTÍNEZ PARRAS
Abstract

Internet has become almost omnipresent in every aspect of our lives. In fact, not only is an unlimited source of information but also influences our behaviour, desires and opinions. Fundamentally, user opinions provoke an enormous effect in consumers and their actions, however, these opinions are extremely subjective. Therefore, the assessment of services and products by users considering only this information is a complex task.

In this undergraduate dissertation, we have developed a data mining project related to restaurants and their respective user opinions using Yelp dataset. The primary purpose is to analyse and measure the quality of restaurants through the use of machine learning, considering notably a probabilistic perspective. First, we have performed a comprehensive data analysis and cleaning of our particular dataset which will improve our understanding about the problem to be solved. Second, we have implemented a quality predictive model based on the Expectation Maximisation algorithm to evaluate the adequacy of the results and use them in the next steps of the project. Finally, we have summarised the user opinions by means of topic modelling in order to analyse and compare intrinsic and latent features from that information in terms of the distinct quality levels.
Acknowledgements

First of all, I would like to express my gratitude to my family for the effort and investment put in my education during these four years. Furthermore, I must mention the satisfactory experience with the colleagues and teachers because I have acquired lots of new knowledge from each other in this stage. To conclude, I specially can highlight the contribution of my supervisors because otherwise this project had not been so interesting and rewarding from the research viewpoint.
Contents

Index of Figures ...................................................... xi
   List of Figures ................................................ xiii

Index of Tables/Charts ........................................... xiii
   List of Tables/Charts ....................................... xv

Index of Acronyms ................................................ xv
   Acronyms ..................................................... xv

1 Introduction ....................................................... 1
   1.1 Motivation .................................................. 2
   1.2 Objectives ................................................. 3
   1.3 Structure of the document .............................. 4

2 Background and current status ............................... 7
   2.1 Bayesian Networks ...................................... 7
   2.2 Clustering techniques ................................. 13
   2.3 Topic Modelling ......................................... 17
   2.4 Natural Language Processing .......................... 20

3 Methodology and development ............................... 25
   3.1 Methodology .............................................. 25
3.2 Development .............................................................. 26

4 Exploratory data analysis ................................................. 31
4.1 Business ................................................................. 32
4.2 Tips ................................................................. 40
4.3 Checkin ............................................................ 42
4.4 Review ............................................................ 42
4.5 User ............................................................... 45

5 Expectation Maximisation results ................................. 51
5.1 Preliminary steps: data cleaning ................................ 51
5.2 Generative model .................................................... 52
5.3 Proof of concept of EM Algorithm .............................. 53
5.4 Classification for the whole dataset ............................. 55
5.5 Remove neutral restaurants ........................................ 56
5.6 Remove neutral reviews ............................................ 57
5.7 Transform data into a balanced classification problem ..... 57
5.8 Selection of users with variety of opinions .................... 58
5.9 Conclusions ........................................................ 58

6 Latent Dirichlet Allocation results ................................. 61
6.1 LDA considering term frequency .................................. 61
6.2 LDA considering Term frequency-Inverse document frequency .......... 70
6.3 LDA filtering most common terms ................................ 79
6.4 Conclusions ........................................................ 81

7 Conclusions and future work ........................................ 83
7.1 Conclusions ........................................................ 83
7.2 Future work ........................................................ 84

Bibliography ................................................................. 88
CD content ................................................................. 89
List of Figures

2.1 Naive Bayes conditional independence .............................. 8
2.2 Steps of k-means algorithm .......................................... 14
2.3 LDA generative model .................................................. 18
2.4 NLP pipeline ............................................................. 20

3.1 CRISP-DM process diagram ............................................ 27

4.1 Relationship between datasets ....................................... 31
4.2 Distribution of business categories ................................. 32
4.3 Heatmap of locations .................................................. 33
4.4 Cluster of restaurants ................................................ 33
4.5 Proportion open/closed restaurants ............................... 34
4.6 Distribution of star rating ............................................ 34
4.7 Average stars per region ............................................. 35
4.8 Distribution of reviews ................................................ 35
4.9 WiFi availability ....................................................... 36
4.10 Age restrictions ........................................................ 37
4.11 Dress code ............................................................. 37
4.12 Alcohol restrictions or availability ............................... 38
4.13 Noise level ............................................................. 39
4.14 Available smoking areas ....................................... 39
4.15 Wordcloud of business names .................................... 39
4.16 Wordcloud of text tips ............................................. 40
4.17 Time series of number of tips/day ............................... 41
4.18 Compliment count ................................................... 41
4.19 Tendency of checkins during the week ....................... 42
<table>
<thead>
<tr>
<th>Table/Chart Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Example of tokenisation</td>
<td>21</td>
</tr>
<tr>
<td>2.2 Example of stop word removal</td>
<td>21</td>
</tr>
<tr>
<td>2.3 Example of stemming</td>
<td>22</td>
</tr>
<tr>
<td>2.4 Example of lemmatisation</td>
<td>22</td>
</tr>
<tr>
<td>3.1 Project schedule</td>
<td>27</td>
</tr>
<tr>
<td>4.1 Statistics of business reviews</td>
<td>36</td>
</tr>
<tr>
<td>5.1 Matrix representation of input data</td>
<td>54</td>
</tr>
<tr>
<td>5.2 Results small dataset</td>
<td>55</td>
</tr>
<tr>
<td>5.3 Analysis of results in the synthetic dataset</td>
<td>55</td>
</tr>
<tr>
<td>5.4 Analysis of results for the whole dataset</td>
<td>56</td>
</tr>
<tr>
<td>5.5 Analysis of results from dataset without neutral businesses</td>
<td>56</td>
</tr>
<tr>
<td>5.6 Analysis of results without neutral reviews</td>
<td>57</td>
</tr>
<tr>
<td>5.7 Analysis of results for balanced dataset</td>
<td>58</td>
</tr>
<tr>
<td>5.8 Analysis of results for variety of opinions</td>
<td>58</td>
</tr>
<tr>
<td>6.1 Results 20 topics using Term frequency</td>
<td>63</td>
</tr>
<tr>
<td>6.2 Results 25 topics using Term frequency</td>
<td>69</td>
</tr>
<tr>
<td>6.3 Results 20 topics using Tf-idf</td>
<td>72</td>
</tr>
<tr>
<td>6.4 Results 25 topics using Tf-idf</td>
<td>77</td>
</tr>
<tr>
<td>6.5 Results 20 topics omitting most frequent terms</td>
<td>80</td>
</tr>
</tbody>
</table>
Acronyms

**CRISP-DM**  Cross Industry Standard Process for Data Mining. 25, 84

**EM**  Expectation Maximisation. 14–16, 51, 52, 58, 81, 84

**KDD**  Knowledge Discovery in Databases. 25

**LDA**  Latent Dirichlet Allocation. 17, 19, 28, 61, 79

**LSI**  Latent Semantic Indexing. 17

**MAP**  Maximum A Posteriori. 8, 10, 11

**MLE**  Maximum Likelihood Estimation. 12, 15

**NLP**  Natural Language Processing. 1, 2, 20, 23, 28, 52, 84

**SEMMA**  Sample, Explore, Modify, Model, Assess. 25

**TF-IDF**  Term frequency – Inverse document frequency. 17, 24, 70, 78, 79
1. Introduction

Needless to say, the expansion of the digital world has caused an enormous impact in the population decisions. In fact, consumers apart from looking for products or services that satisfy their needs, it is increasingly more typical the search of experts or other consumers opinions in order to validate our perception about a purchase. More specifically, this revolution has contributed to the arrival of businesses that provide users with information in order to guide them in the right choice of products or services. From the point of view of restaurant and hotel industry, there are applications such as Yelp, TripAdvisor or Google Places that allow users to explore and discover the opinion of previous customers. These new changes are beneficial for both stakeholders. Regarding business owners, it is possible to analyse customers opinions, trends to adapt towards new changes and improve quality. Moreover, it encourages business competition. On the other hand, these tools are beneficial for users because more information is accessible and the capacity to explore new services increases.

One of the pioneers in this sector was Yelp, which founded in 2004, is currently the most widely used restaurant and merchant information tool across United States. Undoubtedly, these businesses require a well-defined procedure to analyse, organise and summarise such quantities of data. Considering this premise, the business Yelp annually launches a set of challenges aimed at students and researchers willing to conduct research or analysis on Yelp’s data, although there is a special interest in applying machine learning to solve particular problems such as image classification, Natural Language Processing (NLP) or Graph Mining. Intending to accomplish that goal, it has been released a public dataset containing data about businesses, users, reviews and images. Therefore, the ultimate purpose of this challenge will consist of the improvement of current services as well as the development of new features in a way that users and businesses owners satisfy their needs.
1.1. Motivation

Currently, steps forward have be done regarding user opinions but there are still some crucial concerns which should be solved. Some of the following issues were the reasons why we decided to choose this project.

Firstly, the cited recommendation services are based on their user community which provide feedback about the businesses they have visited. Usually, in order to estimate the quality of a service, prospective users can analyse the average star rating and the textual reviews of the users. However, this procedure involves two main problems. First of all, the star rating although it is useful and may act as a straightforward filter, only provides a general overview of a business. As a consequence, should users would like to explore further details, in the worst case they must read all these textual reviews or select a representative group of these reviews. As we can imagine, a user usually compares a set of businesses, hence, the task of finding the most suitable business could require a huge effort and time. Currently, some solutions to this problematic consist of dividing the overall stars into some groups such as Location, Room, Service or Cleanliness and assess these features independently. Until now, this approach is acceptable but does not reflect the key concepts of the textual reviews such as advantages, drawbacks or significant situations in a business.

In the second place, a business assessment is extremely subjective because it implies the participation of different people with different criteria. For that reason, we will require to process huge volumes of data in order to ensure a certain level of confidence in our predictions. Related to this, the measurement of the quality of services and products is not accurate because of two main factors. A high number of evaluators usually provides a metric about how crowded the business is and a full range of opinions, but few opinions do not offer reliability. In addition, business quality is not fixed during its life cycle but there is some variability overtime due to internal and external factors. Therefore, predictions must be continuously updated with new data.

Thirdly and finally, user opinions are still a research area of interest with an enormous quantity of issues to be solved from the computer science viewpoint. Some of these problems include from methods for NLP to detection of fake reviews, which can affect business reputation. In particular, this last fact is very difficult to control although there are some implementations to verify that at least users have visited the businesses they are commenting about.
1.2. Objectives

In this section, we will enumerate and explain some of the objectives we will have accomplished after completion of the project.

1.2.1. Main objective

In general, the main objective of this project is to perform a complete data mining project. Particularly, we will obtain and use the source of information provided by Yelp to analyse and predict the quality of restaurants according to their own intrinsic features. Over the course of the project, we will determine which technique best fits our problem, but in general, we will consider the problem is defined inside weakly supervised learning.

1.2.2. Secondary objectives

From the obtained results from classification process defined in the main goals, we will obtain as a result a trained model which will be used to achieve the following objectives:

- Infer the corresponding values for our class variable using our data, guaranteeing a certain level of accuracy.

- Analyse, summarise and compare textual reviews by means of machine learning techniques to finally extract those significant terms which represent features that determine the distinct levels of quality available in our data.

1.2.3. Computing specific competences

In terms of educational purposes, during this project will guarantee the acquisition of certain competences. Some of the specific competences related to the majoring in computing are enumerated below.

- Ability to evaluate the computational complexity of a problem, discover those algorithmic strategies that may lead to its resolution and recommend, develop and implement those which guarantee the best performance according to the requirements established.

- Ability to know the inherent fundamentals, paradigms and techniques of the intelligent systems and analyse, design and build systems, services and computer applications based on those techniques in any field of application.
• Ability to acquire, obtain, formalise and represent human knowledge in a computable way to solve problems through a computer system in any field of application, particularly those related to aspects of computing, perception and intervention in intelligent environments.

• Ability to know and develop computational learning techniques and implement systems and applications that use them, including those dedicated to automatic extraction of information and knowledge from large volumes of data.

1.3. Structure of the document

In this section, it is shown the structure of document along with a brief description to give the reader a general overview of this document.

• **Chapter 1. Introduction:** In the current chapter, we have introduced this project as well as the motivations to develop it. Afterwards, we have included a description of the objectives to be accomplished and the corresponding competences.

• **Chapter 2. Background and current status:** In this chapter, we will provide a detailed explanation of the machine learning fundamentals that are usually implemented to solve problems similar to our project.

• **Chapter 3. Methodology and development:** It is described the data mining process as well as the software and hardware resources that will be used for this project. Moreover, we will mention the schedule established for the different tasks.

• **Chapter 4. Exploratory data analysis:** In this chapter, we will analyse the datasets in order to get familiar with the data and extract relevant information, which can be used in the following steps.

• **Chapter 5. Expectation Maximisation results:** Throughout this section, we will describe the experiments we have carried out to predict the quality of the restaurants and to finally analyse the results that have been obtained.

• **Chapter 6. Latent Dirichlet Allocation results:** Throughout this section, we will also describe the experiments we have carried out to summarise the textual reviews and to the finally compare the results for different levels of quality.
Chapter 7. Conclusions and future work: In this chapter, we will summarise in general terms which objectives have been reached or not in this project. Furthermore, we will discuss some improvements or additional steps which may be applied in future projects.
2. Background and current status

In this chapter, we will discuss some of the techniques that will help us during the development of our project.

2.1. Bayesian Networks

Bayesian Networks provide a classification system based on a probabilistic approach. They are able to predict the probability of a certain class considering the probability that a given instance belongs to a class. Researchers have carried out distinct studies regarding this kind of classification algorithms. One of them has proved that Bayesian classifiers such as Naive Bayes are able to achieve similar results compared to other machine learning algorithms such as decision trees [1]. Moreover, Bayesian classifiers stand out for the high accuracy and speed.

In this section, we will focus on Naive Bayes classifiers, which assume that the effect of an attribute value on a given class is independent of the values of the other attributes (see Figure 2.1). In order to understand the basic functionality of this algorithm, we need to explain the Bayes’ Theorem.

2.1.1. Bayes’ Theorem

We can define $D = \langle D_0, D_1, ..., D_i \rangle$ as an instance of our data or, from the point of view of Bayesian terminology, an evidence. Moreover, we will need a hypothesis $h \in H$ such that $D$ belongs to a class $C$.

In order to understand Bayes’ Theorem, it is required some concepts. For instance, $p(h)$ is called the prior probability or a priori probability and denotes the probability of $h$ being correct considering our previous knowledge about the problem domain. Similar to this, $p(D)$ represents the probability of observing our data, whereas $P(h|D)$ is the
posterior probability, or a posteriori probability, of h conditioned on D and reflects the probability that the hypothesis h holds after observing D. Finally, $P(D|h)$ is also the posterior probability but denotes the probability of observing our data D given that the hypothesis h holds.

Most of the time, machine learning is interested in determining $P(h|D)$. From this moment, Bayes’ theorem plays a significant role because provides a way of computing the posterior probability, $P(h|D)$, from the prior probabilities $P(h)$ and $P(D)$, and the posterior probability $P(D|h)$. Bayes’ theorem is defined as follows in Equation 2.1.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$ \hspace{1cm} (2.1)

2.1.2. Naive Bayes Classification

Naive Bayes classifiers are based on the Bayes’ theorem and is considered the simplest of the Bayesian classifiers [2]. In general, the classification process would be as follows:

1. We may consider the training set $X$, composed by vectors of attributes, each one assigned to a class label.

2. Suppose we have $n$ hypothesis or classes. Thus, a classifier will find the hypothesis having the highest posterior probability conditioned on D. That hypothesis or class, for which $P(h|D)$ is called Maximum A Posteriori (MAP) hypothesis. The MAP hypothesis can be computed using the Bayes’ Theorem by calculating the a posteriori probability for each hypothesis, whose mathematical representation
2. Background and current status

is established in Equation 2.9.

\[ h_{\text{MAP}} = \arg\max_{h \in H} p(h|D) \]
\[ = \arg\max_{h \in H} \frac{p(D|h) \ p(h)}{p(D)} \]
\[ = \arg\max_{h \in H} p(D|h) \ p(h) \] (2.2)

3. Due to the large number of attributes, computing the a posteriori probabilities may be computationally expensive. However, the Naive Bayes classifier considers the class-conditional independence. In particular, this assumes that there is no dependence between the different attributes of our data. Formally, this property is defined as follows:

**Definition 2.1** X is conditionally independent of Y given Z, if the probability distribution governing X is independent of the value of Y, given the value of Z. We often represent this

\[ P(X|Z, Y) = P(X|Z) \]

As a consequence, \( p(D|h) \) could be computed as it is referred in Equation 2.3.

\[ p(D|h) = \prod_{i=0}^{\left| D \right|} p(D_i|h) \]
\[ = p(D_0|h) \ p(D_1|h) \ldots p(D_i|h) \] (2.3)

Once we have discovered how to classify an instance, we must determine the attribute types in order to compute the respective a posteriori probabilities. Regarding the attributes, we will have to distinguish between categorical and continuous attributes.

- If the attribute is categorical, the a posteriori probability \( P(D_i|h) \) is computed as the number of instances labelled with h whose attribute i equals \( D_i \) divided by number of instances labelled with h.
- On the other hand, if the attribute is continuous, the a posteriori probability \( P(D_i|h) \) is computed considering that \( D_i \) describes a Gaussian or Normal distribution defined by its mean \( \mu \) and standard deviation \( \sigma \). Therefore, the
a posteriori probability is computed using Equation 2.4:

\[
P(D_i|h) = N(D_i, \mu_h, \sigma_h) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(D_i - \mu)^2}{2\sigma^2}}
\]  

Theoretically, the inference process should work perfectly. Nevertheless, when put into practice, we may find that multiplications of any conditional probabilities may lead to floating point underflow due to the software and hardware constraints. The issue could be addressed by two different approaches according to our goals.

The first approach is intended only when we need the MAP hypothesis. In that case, we could avoid floating point underflow performing the computation by adding logarithms of probabilities instead of multiplying probabilities, achieving that the highest log-probability class will be still the most probable. This technique is described in Equation 2.5:

\[
h_{MAP} = \arg\max_{h \in H} \log (p(h|D)) = \arg\max_{h \in H} \log \left( p(h) \prod_{i=0}^{\mid D \mid} p(D_i|h) \right) = \arg\max_{h \in H} \log (p(h)) + \sum_{i=0}^{\mid D \mid} \log (p(D_i|h))
\]  

The last option is known as the log-sum-exp trick and it is used when it is required the class probability. The objective of this method, based on logarithm properties, is to avoid floating point underflow during the normalisation process.

\[
\log (p(h|D)) = \log \left( \frac{p(h) p(D|h)}{\sum_{i=0}^{\mid H \mid} p(h_i) p(D|h_i)} \right) = \log (p(h)) - \log \left( \frac{\sum_{i=0}^{\mid H \mid} p(h_i) p(D|h_i)}{\text{numerator}} \right)
\]  

As we can deduce from the formula, the numerator cannot lead to underflow if we apply the same technique as in the first approach but it may happens in the denominator (see Equation 2.6). Thus, we should apply the log-sum-exp trick (see Equation 2.6).
2. Background and current status

2.7.}

\[
\log (p(h|D)) = \log (p(h) p(D|h)) - \log \left( \sum_{i=0}^{|H|} e^{a_i} \right) \\
= \log (p(h) p(D|h)) - \log \left( \sum_{i=0}^{|H|} e^{a_i} e^{A-A} \right) \\
= \log (p(h) p(D|h)) - \left( A + \log \left( \sum_{i=0}^{|H|} e^{a_i-A} \right) \right)
\]

(2.7)

where

\[
a_i = \log (p(h_i) p(D|h_i)) \\
A = \max_{i \in |H|} a_i
\]

2.1.3. Bayesian Networks applied to Text Classification

One the most applications of Naive Bayes is Text Classification. For instance, a common application of this technique is spam detection. In general, we are given a set of documents \(X\) and a set of classes \(C = \{c_1, c_2, ..., c_n\}\). Traditionally, each document has been manually labelled with a representative class or topic after reviewing the entire document by an expert. Thus, our objective is to create a classification method that predicts the class label for every document.

The idea behind this approach is to compute the probability of a document belonging to a class \(c\), which is computed as indicated in Equation 2.8

\[
P(c|D) = P(c) \prod_{i=0}^{|D|} p(t_i|c)
\]

(2.8)

where \(P(t_i|c)\) is the conditional probability of term \(t_i\) occurring in a given class \(c\) and \(p(c)\) is the a priori probability of a document \(D\) belonging to class \(c\). The elements \(t_i\) are the terms that compose the document \(D\) and which are used for the classification process.

Finally, we must estimate the MAP hypothesis or class, considering the stated in
Equation 2.9

\[ c_{MAP} = \arg\max_{c \in C} p(c|D) \]

\[ = \arg\max_{c \in C} \frac{p(D|c) \cdot p(c)}{p(D)} \]

\[ = \arg\max_{c \in C} p(D|c) \cdot p(c) \]

Fundamentally, there are two alternatives to implement our Naive Bayes algorithm for text classification: multinomial and Bernoulli models [3]. The first generates one term for each position of the document whereas in the Bernoulli model is generated 0 or 1 depending or not on the presence of the term in the document. These features imply the existence of different estimation and classification rules. Specifically, when computing the a posteriori probability \( p(t_i|c) \), Bernoulli model uses binary term occurrence whereas multinomial model keeps track of the number of occurrences. This fact implies that Bernoulli model is very prone to errors respect to the other model when dealing with long documents. Particularly, the inference in Bernoulli models have some slight differences respect to the usual way it is computed the a posterior probabilities, as we could observe in Equation 2.10.

\[ p(c|d) = P(c) \prod_{i=0}^{|T|} b_{t_i} \cdot p(t_i|c) + (1 - b_{t_i}) \cdot (1 - p(t_i|c)) \]  

(2.10)

where \( b_{t_i} \in 0, 1 \) and indicates the presence or not of the term \( t_i \) and \(|T|\) is the number of elements of the vocabulary.

The remaining task to do in every Naive Bayes model is to estimate the corresponding parameters through the MLE, which corresponds with the relative frequency [4]. Concretely for text classification, we have to define the estimation for the probabilities of categorical data. For instance, the a priori probability \( p(c) \) is computed using Equation 2.11.

\[ p(c) = \frac{N_c}{N} \]  

(2.11)

where \( N \) is total number of documents and \( N_c \) is the number of documents assigned to the class \( c \).

We also need to estimate the conditional probability \( p(t_i|c) \) as the relative frequency of the term \( t_i \) in documents belonging to class \( c \). This relationship is defined in Equation 2.12.
2. Background and current status

\[
p(t_i|c) = \frac{T_{t_i,c}}{\sum_{t' \in T} T_{t',c}} \tag{2.12}
\]

where \(T_{t_i,c}\) is the number of occurrences of the training documents corresponding to class \(c\), including the possible multiple occurrences of a term \(t_i\) in our document.

Finally, in order to avoid the problem of zero probabilities as a consequence of the limitations of the train data, we can use Laplace smoothing (see Equation 2.13) in our multinomial Bayes model, which basically adds one to every count.

\[
p(t_i|c) = \frac{T_{t_i,c} + 1}{\sum_{t' \in T} (T_{t',c} + 1)} = \frac{T_{t_i,c} + 1}{(\sum_{t' \in T} T_{t',c}) + |T|} \tag{2.13}
\]

2.2. Clustering techniques

A classification outcome could be represented as a set of clusters. This means that instances with similar features will belong to the same cluster or class. From the point of view of clustering literature, the principal clustering methods are Partitioning, Hierarchical, Density-based and Grid-based. One of the most fundamental clustering techniques is partitioning, which divides a set of instances into several exclusive groups. More specifically, this kind of algorithms requires a set of elements \(n\), the number of clusters \(k\) and an optimisation function or criterion.

A classic partition clustering algorithm is \(k\)-means, which performs two main steps iteratively. Prior to execute the algorithm, it is necessary to define the number of clusters \(k\) and choose randomly \(k\) points denoting the clusters centers. First of all, each object or instance is assigned to the nearest cluster center according to a metric such as the Euclidean distance. Secondly, the algorithm adjusts or re-estimates cluster centers through the mean of the objects of each cluster. It is said in this step that the level of similarity among the objects of each cluster is maximised. Finally, the process is repeated with the new changes until the cluster centers remain the same overtime. For instance, Figure 2.2 shows how clusters are updated along with their centers for each iteration until a convergence stage is reached.

This clustering method is intended to minimise the sum of the distance from all points to their cluster centers. However, there is no guarantee that the algorithm converges to global optimum due to the initial cluster centers. In fact, slight variations in the choice of the initial parameters have a huge impact in the final outcomes. As a
result, a method to increase the likelihood of improving the results consists of executing the algorithm several times, each one with a distinct initial cluster centers and select the best one.

![Steps of k-means algorithm](image)

**Figure 2.2: Steps of k-means algorithm**

### 2.2.1. Probabilistic clustering

Contrary to algorithms such as $k$-means, probabilistic clustering gives us the flexibility of assigning instances to several clusters. This approach allows us to detect clusters or categories that cannot be observed directly and obtain the probability of belonging to a cluster. These clusters are inferred from the training set and try to resemble the hidden features or categories of the data [2]. Statistically, probabilistic clustering is based on mixtures, which is a set of probability distributions. It is assumed that a hidden category or cluster is one of these probability distributions denoted by the corresponding probability density function.

For example, considering a univariate mixture, the goal of probabilistic clustering is to find the mean $\mu$ and the variance $\sigma$ for each cluster.

### 2.2.2. Expectation Maximisation Algorithm

In the section, we are going to describe an approach that came to light when the researchers Demper, Laird and Rubin published a report in 1977 [6]. This method is known as Expectation Maximisation (EM) algorithm, whose main task is to obtain an approximation of the maximum likelihood or maximum a posteriori parameter $\theta$ of
the statistical models. The most useful situation in which the EM algorithm is used are incomplete-data-problems where there exist missing data or truncated probability distributions [7].

To use the EM algorithm is required some observed data \( y \), a parametric density \( p(y|\theta) \), a description of some complete data \( x \) that you require, and the parametric density \( p(x|\theta) \). Considering that you only know \( y \), our objective is to find the MLE of \( \theta \), taking into account Equation 2.14.

\[
\theta_{\text{MLE}} = \arg\max_{\theta \in \Omega} p(y|\theta)
\]  

(2.14)

Now, we are going to carefully detail the steps followed by this algorithm [8]:

1. Start the first iteration \( m = 0 \) with an initial random configuration \( \theta^m \).

2. We wrongly assume that the estimation of the parameter \( \theta^m \) is correct to compute the conditional probability distribution \( p(x|y, \theta^m) \) for the complete data \( x \).

3. Using the conditional probability distribution \( p(x|y, \theta^m) \), we can obtain the conditional expected log-likelihood, \( Q(\theta|\theta^m) \). We will not prove the calculations behind this concept because it is out of the scope of this project.

4. Find \( \theta \) that maximises \( Q(\theta|\theta^m) \), assign that value to \( \theta^{m+1} \) and update \( m \) to \( m + 1 \).

5. Repeat this process from the second step until a criterion holds. This criterion could be a certain number of iterations in the algorithm or the parameter \( \theta \) stops changing because it is negligible.

Generally, the algorithm only consists of two steps which are a combination of the previous ones [9].

- **Expectation**: Given the estimate from the previous iteration \( \theta^m \), compute the conditional expectation \( Q(\theta|\theta^m) \).

- **Maximisation**: The estimation of \( \theta \) in the iteration \( m + 1 \) is defined by Equation 2.15.

\[
\theta(m + 1) = \arg\max_{\theta \in \Omega} Q(\theta|\theta^m)
\]  

(2.15)
Alternative variant of EM algorithm

One of the problems that may arise using probabilistic clustering is that we do not know neither the mixture model parameters nor distribution that each training instance belongs to. In the context of fuzzy or probabilistic model-based clustering, the variant of the EM algorithm is strongly related to $k$-means algorithm. In particular, $k$-means requires a set of instances which will correspond to the observed data in the case of EM algorithm. On the contrary, missing data determines which cluster each observed data or instance belongs to. Therefore, the goal would be the estimation of the $k$ cluster centers defined by the parameter $\theta$.

Each iteration also consists of two steps:

- **Expectation**: This step computes the cluster probabilities and assigns instances to clusters according to the current or probabilistic clusters parameters $\theta$.

- **Maximisation**: The maximisation step finds the new clustering or parameters that maximise the sum of squared errors in fuzzy clustering or the expected likelihood in probabilistic model-based clustering.

Note that, in general, the EM algorithm may not converge to the optimal solution but it may instead converge to a local optimal. For instance, some of the techniques to avoid this includes running the EM process multiple times using different random initial values and choosing the best local solution. However, it is guaranteed that EM estimate never gets worse. We must mention that the convergence could be slow specially is the number of visible observations is less than missing observations.

Some of the advantages of this algorithm includes:

- The algorithm is stable because after each iteration the likelihood increases.

- The algorithm may be used to estimate values for the missing data.

- It does not require much storage space.

- The cost per iteration is low which balances the necessary cost or number of iterations to reach the convergence stage.

- The EM algorithm is usually easy to program and implement.
2.3. Topic Modelling

Topic modelling consists of an unsupervised classification method for documents, whose objective is to extract those relevant and latent topics inside a dataset and classify a text into a particular topic. A topic depend on the appearance of several terms in a document and their repetition along the texts of our dataset [10].

One of the first steps that contributed to the advent of Topic Modelling as currently it is known was the representation of each document in the corpus in terms of frequency of occurrence their words. However, this approach was problematic due to the fact that the most common terms in any language scarcely have semantic meaning. Subsequently, the Term frequency – Inverse document frequency (TF-IDF) scheme was conceived to be a solution to the mentioned issues. Basically, TF-IDF establishes a relationship between the number of occurrences of each word in each document with the number of occurrences of the word in the entire corpus.

Over the years, researchers proposed new alternatives, although the most remarkable was Latent Semantic Indexing (LSI), which applies the TF-IDF computations in a matrix of documents by terms. This means that each document is represented by a row and each term by a column. Finally, LSI transforms the new matrix through a process known as Singular Value Decomposition. Overtime, one of the most widespread algorithms for topic modelling is Latent Dirichlet Allocation (LDA) which is a generative probabilistic model published in 2003, which is based on two foundations.

- **Each document represents a mixture of topics**: A document is composed of several words, each one belonging to one or several topics according to a determine proportion.

- **Each topic is a mixture of words**: A topic is represented as a set of words according to a probability.

2.3.1. Generative model

The goal of topic modelling is to automatically discover the topics from a collection of documents [11]. The documents themselves are observed, while the topic structure, the per-document topic distributions, and the per-document per-word topic assignments are a hidden structure.

Before starting to explain how LDA works it is necessary to establish some notations for some basic structures of data.

- A word is the basic unit of the system, being an element from the vocabulary $V$. 

• A document is a sequence of $N$ words $w = \langle w_1, w_2, ..., w_N \rangle$, where $w_n$ is the $n$th word in the sequence.

• A corpus $D$ is a set of $M$ documents.

This is a statistical model, which tries to imitate or acquire the intuition that humans possess to identify and classify documents. The core of the model is outlined by its generative process, which is a random and imaginary process that assumes the document generation procedure [12]. This process for each document consist of the following steps:

- Choose randomly a distribution over topics
- For each word in the document
  - Choose a topic from the topic distribution
  - Choose a word from the corresponding distribution.

Figure 2.3: LDA generative model

In Figure 2.3, we can see the graphical model of LDA, where we can deduce a set of dependencies. Each node is a random variable and is labelled according to its role in the generative process. The hidden nodes (i.e the topic proportions, assignments, and topics) are in white colour. The observed nodes, the words of the documents, are shaded. The rectangles are “plate” notation, which denotes replication. The $N$ plate denotes the collection words within documents whereas the $D$ plate denotes the collection of documents within the collection. The rest of the parameters define the control the following properties of the algorithm:

- $\alpha$ is the parameter that defines the distribution of topics for all the documents in the corpus.
- $n$ is the parameter that defines how distribution of words in each topic looks like.
2. Background and current status

- The topics $\beta_1:K$ where each $\beta_k$ is a distribution over the vocabulary.

- The topic distributions for the $d$th document are $\theta_d$, where $\theta_{d,k}$ is the topic proportion for topic $k$ in document $d$.

- The topic assignments for the document in the $d$th positions are $z_d$, where $z_{d,n}$ is the topic assignment for the $n$th word in document $d$.

Formally, this process defines a joint probability distribution for observed and latent random variables (see Equation 2.16). According to this probability, we can estimate the a posteriori probability of some hidden variables given the observed variables. Ideally, the search of new topics consist of reversing this process.

$$p(\beta_1:K, \theta_1:D, z_1:D, W_1:D) =$$

$$= \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d) \prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\beta_1:K, z_{d,n})$$

This distribution specifies a set of dependencies [13], which basically defines the LDA algorithm and which are encoded in the generative process in the form of joint probability distribution.

Using the previous information we can explain the inference process for this algorithm. In particular, we can define the posterior probability as follows in Equation 2.17.

$$p(\beta_1:K, \theta_1:D, z_1:D|W_1:D) = \frac{p(\beta_1:K, \theta_1:D, z_1:D, W_1:D)}{p(W_1:D)}$$

In this case, the numerator is easily to compute and corresponds with the joint distribution previously defined, whereas the denominator is the probability of observing the corpus under any topic model, however, the number of possible topic structures is very large. Therefore, this probability cannot be computed because is intractable. Topic modelling algorithms computes approximations for the a posteriori probability according to sampling-based algorithms and variational algorithms. Therefore, researchers are defining new methods such as Gibbs sampling.
2.4. Natural Language Processing

Natural Language Processing is a research field of computer science whose main objective is the ability to design computer programs to analyse and process natural language data.

In this section, we will focus on documents and texts, hence, we must explore the standard procedures to prepare our data. The main drawback of this kind of data is that not only may contain useful terms but also meaningless and strange words which will not improve our classification process. Therefore, in order to clean these texts we are going to use SpaCy, which is a state-of-art python library for NLP. To prepare this kind of data, we have defined pipeline similar to Figure 2.4, although we will describe a more advanced set of actions.

2.4.1. Tokenisation

Tokenisation is process of breaking down a text corpus into a set of token or terms representing words, numbers or symbols [14]. Apart from that, this procedure usually is performed along with other tasks such as removal of non-alphabetic terms or punctuation symbols. In addition, we must convert all these terms into lowercase in order to avoid that, terms lexically equivalent are considered different terms. In Table 2.1 we provide an example of the process we have described. Tokenisation basically converts texts in a suitable structure to be processed by machine learning algorithms.

2.4.2. Stop words

During this process, it is advisable to use stop word removal. Stop words are considered the most common words of a language such as articles, pronouns or conjunctions. The intention of removing stop words is that they do not provide useful or relevant semantic
information and should be removed. Regarding to this task, there are two different approaches. The first one consists of obtaining a dictionary of stop words for a specific language. The second alternative is to create a dictionary containing the words of our corpus sorted by their frequency. The terms to be removed will be the top $k$ words more frequent in the dictionary.

Moreover, we will not consider those terms whose length is less than three characters because they probably are meaningless terms. Certainly, we may notice that these filters reduce drastically our dataset. In Table 2.2 we provide an example of the process we have described.

<table>
<thead>
<tr>
<th>Best Indian food I have ever eaten, the staff is very friendly</th>
</tr>
</thead>
<tbody>
<tr>
<td>best indian food i have ever eaten the staff is very friendly</td>
</tr>
</tbody>
</table>

Table 2.2: Example of stop word removal

### 2.4.3. Stemming and Lemmatisation

We could still reduce the number of unique words by means of figuring out the issue of inflection. Formally, we can define inflection as the modification in the form of a word such that expresses a change in the way it is used [15]. The main dilemma about inflected words is to be considered distinct terms while expressing a similar concept. Currently, there are two text normalisation techniques for this sort of words: stemming and lemmatisation. In general terms, both strategies reduce terms to accomplish a decrease in the number of word and preserving the semantics. For instance, after applying some of these procedures in a document containing the terms run, runs and running, the evident transformation or reduction would be the term run.

On the one hand, stemming is the simplest approach and reduces a word to its base form according to predefined set of rules, i.e removing common prefixes and suffixes. Currently, the most significant stemming algorithms are Porter, Snowball and Lancaster. Until the date, Porter Stemmer is the most accurate algorithm, which includes approximately 1200 rules. This algorithm dates back to 1980 but some enhancements
have been proposed. However, some of the drawbacks of the method are understemming and overstemming, which provoke that words are meaningless, not reduced properly or grammatically incorrect. In Table 2.3, we can observe the result of applying a stemmer in a sample text.

<table>
<thead>
<tr>
<th>A swimmer likes swimming, thus he swims.</th>
</tr>
</thead>
<tbody>
<tr>
<td>↓</td>
</tr>
<tr>
<td>a swimmer like swim , thu he swim .</td>
</tr>
</tbody>
</table>

Table 2.3: Example of stemming

On the other hand, lemmatisation is the option we have chosen. In general, it involves computationally a more advanced process because converts words into their dictionary form through the calculation of the corresponding part of speech belonging to each term. It is said that these normalisation techniques have a little impact on the performance of the machine learning algorithms [16]. Finally, we can check as an example the results provided by a lemmatiser in Table 2.4

<table>
<thead>
<tr>
<th>A swimmer likes swimming, thus he swims.</th>
</tr>
</thead>
<tbody>
<tr>
<td>↓</td>
</tr>
<tr>
<td>a swimmer like swimming , thus he swim .</td>
</tr>
</tbody>
</table>

Table 2.4: Example of lemmatisation

2.4.4. Translation

Previously, we assumed that the reviews were written entirely in English. However, if after an in-depth analysis we discover that there are a small percentage of documents in a different language such as Chinese or French, we will have a problem of consistency in our dataset. An straightforward approach could consist of using the libraries such as TextBlob to detect the corresponding document language and translate it if the language is not English. For instance, this library relies on Google Translate API, ensuring a high level of confidence in the given results and the same language in the entire corpus.

2.4.5. Spelling correction

Another issue we have assumed is that the words are correctly spelled. Nevertheless, this fact cannot be true when documents are written by several users in informal
situations. There are some alternatives to deal with this issue. These include using built-in libraries such as TextBlob or implementing custom algorithms based on NLTK modules like edit_distance.

2.4.6. N-grams

The n-gram model is defined by means of a sequence of n terms. The simplest case is called unigram, where each token is composed of exactly one word, whereas we denote bigrams and trigrams through two and three word sequences respectively. The main objective of n-grams is to consider as a single entity terms that usually appear together. Moreover, this model could be used to perform spelling error corrections. The functionality of this model consist of assigning probabilities to each possible next word. In particular, the model must compute \( p(w|h) \), which is the probability of a word \( w \) given some previous set of words \( h \) \([17]\).

For instance, if we want to compute the probability of the word given the previous sentence, we need to use Bayes’s rule and the chain rule of probability to obtain the joint probability distribution (see Equation \([2.18]\)).

\[
p(\text{the } \mid \text{its water is so transparent that}) = \frac{p(\text{the its water is so transparent that})}{p(\text{its water is so transparent that})} \tag{2.18}
\]

2.4.7. Data representation

Finally, in order to train a model from text documents, we need to represent this data in a two-dimensional matrix. A commonly used model in NLP is the so-called bag of words. In this approach, each document is represented by means of a set of words which occur in the document. Beforehand, we must create a vocabulary or dictionary which contains all the terms in the corpus. Using this dictionary, we need to create a matrix where each row is a documents and each column is a term.

In the basic approach, the matrix position \((d, t)\) contains the metric Term frequency, \(tf_{d,t}\), reflecting the number of times term \(t\) occurs in document \(d\). There are some variants of this metric such as:

- **Binary approach**: Indicates whether the term occurs or not in the document.

- **Logarithmic approach**: Computes the matrix in terms of logarithms such as \(\log(1 + tf_{d,t})\).
- **Normalisation**: In this case the metric is normalised according to the most frequent term in the document.

The main drawback of this metric is that there exists common terms which are both meaningless and very frequent. A possible strategy would be the metric Inverse Document Frequency, which penalises the terms respect to their frequency in the corpus. This metric can be computed using Equation 2.19

\[
idf_t = \log \left( \frac{\text{corpus length}}{\text{documents including term } t} \right)
\]  

(2.19)

However, it is usually used a combination of both metrics known as Term frequency – Inverse document frequency (TF-IDF), whose formula is described in Equation 2.20.

\[
tf \cdot idf_{d,t} = tf_{d,t} \cdot idf_t
\]  

(2.20)
3. Methodology and development

3.1. Methodology

Similar to other kinds of projects, it is strongly advisable to implement a process or methodology due to the ease of managing and replication of the project. Some of the most well-known methodologies intended to data mining projects like this are Knowledge Discovery in Databases (KDD), Sample, Explore, Modify, Model, Assess (SEMMA) or Cross Industry Standard Process for Data Mining (CRISP-DM). Despite the fact that we will not strictly follow the process guidelines because they are out of our scope, we will take as a starting point CRISP-DM, which was conceived as an European Commission project in 1996 [18]. CRISP-DM is a non-proprietary model intended to be an industry tool, being developed by some of the most important businesses of the data mining sector such as DaimlerChrysler, SPSS and NCR. This methodology establishes six data mining phases [19], which are described below.

- **Business understanding**: This first phase focuses on the understanding of the problem to be solved and its requirements in order to select the data and evaluate the outcomes properly. In addition, an action plan must be designed to achieve the data mining and business goals. In general, this step will allow us to convert this problem into a data mining project taking into account all constraints and requirements.

- **Data understanding**: This step consists of gathering the initial data and getting familiar with this information through a description process of the data properties. The tasks also include data exploration using visualisation tools and statistical analysis. Finally, the stage should be concluded by assessing data quality in terms of missing values or errors.
• **Data preparation**: The data preparation implies a set of techniques whose goal is to adjust and build a dataset which meets certain criteria so that it can be processed by the distinct machine learning algorithms of the upcoming phases. Some of the tasks to be done in this phase may include data selection, data cleaning, feature engineering, or integrate data. Particularly, we must perform the selection of significant attributes/instances, combine data from multiple and different sources.

• **Modelling**: This stage deals with the identification of the data mining algorithms which best fits with the problem considering their constraints such as adequacy for the problem, requirements, and data availability. During this stage, the experts must select the predictive models, generate measurements of goodness of fit, generate the models along with the best parameters, and assess the models.

• **Evaluation**: Assessment of the obtained results not in terms of accuracy but respect to the objective that we must reach. Apart from that, we must review the whole process to detect fails and solve them. Depending on the results, we need to decide to either move to the next phase or repeat another iteration.

• **Deployment**: In the previous phases, the project was implemented and validated. First of all, it is necessary to gather the evaluation reports and set the required actions to deploy the system. Afterwards, we must define a maintenance and monitoring in order to avoid incorrect usage of the data mining results.

In Figure 3.1, it is depicted the iterative process along with the required steps as well as the flow of execution. Regarding the schedule of the project, we have agreed to arrange meetings every two weeks with the supervisors to keep track of the progress and determine the next steps. Approximately, the development of the different phases have been done in the time interval defined in Table 3.1. During each one of the tasks, we have documented all the performed steps in order to keep track of the progress of the current project.

### 3.2. Development

In this section, we are going to mention the software and hardware tools that we will use during the project.
3. Methodology and development

Figure 3.1: CRISP-DM process diagram

<table>
<thead>
<tr>
<th>Task</th>
<th>Start date</th>
<th>End date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business understanding</td>
<td>15/11/2018</td>
<td>11/12/2019</td>
</tr>
<tr>
<td>Data understanding</td>
<td>12/12/2018</td>
<td>31/01/2019</td>
</tr>
<tr>
<td>Data preparation</td>
<td>01/02/2019</td>
<td>28/02/2019</td>
</tr>
<tr>
<td>Modelling</td>
<td>01/03/2019</td>
<td>15/05/2019</td>
</tr>
<tr>
<td>Evaluation</td>
<td>15/05/2019</td>
<td>31/05/2019</td>
</tr>
</tbody>
</table>

Table 3.1: Project schedule

3.2.1. Software

All the software tools that we are going to use across the project will be listed below.

General purpose tools

- **Python 3.6**: It is an interpreted programming language, which along with R are the most widespread options for machine learning projects.

- **Jupyter Notebook**: It is an interactive workspace that allows to develop Python code at the same time that we can integrate code blocks, images, graphs and equations. Some of its purposes are data cleaning, data visualisation ...

- **Pandas**: A Python BSD-licensed library for efficient data manipulation and analysis through data structures such as dataframes or series.
Visualisation tools

- **Matplotlib**: A Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

- **Seaborn**: A Python library, based on Matplotlib, intended to draw graphics for statistical data visualisation.

- **Folium**: Folium is a powerful data visualisation library in Python that was built primarily to help people visualise geospatial data. With Folium, one can create a map of any location in the world as long as its latitude and longitude values are known. Also, the maps created by Folium are interactive in nature, so one can zoom in and out after the map is rendered.

- **pyLDAvis**: pyLDAvis is designed to help users interpret the topics in a topic model that has been fit to a corpus of text data. The package extracts information from a fitted LDA topic model to inform an interactive web-based visualisation. This tool has been developed by through a research project [20] by researchers from Iowa State University and AT&T Labs Research.

Natural Language Processing tools

- **SpaCy**: Written in Python and Cython, Spacy is an open-source library for advanced Natural Processing Language. One of the main features is the availability of pre-trained models for several languages such as Spanish, French and German.

- **TextBlob**: TextBlob is another Python3 library intended for NLP tasks such as classification, part-of-speech tagging or sentiment analysis.

Machine learning classification tools

- **Scikit-learn**: It is an open source machine learning library which offers a full range of tools and algorithms for data mining and data analysis. More specifically, includes methods for preprocessing, classification, dimensionality reduction and model selection.

- **Gensim**: Gensim is a Python library for topic modelling, document indexing and similarity retrieval with large corpora. Target audience is NLP and information retrieval community.
3. Methodology and development

3.2.2. Hardware

In order to develop our algorithms, we will use Kaggle which is an online platform or community of data scientists and machine learners. In general terms, its services include public datasets and cloud-based workbenches. According to Kaggle, all the users have at their disposal the following technical specifications, which are one of the best advantages of the platform:

- 9 hours execution time
- 5 Gigabytes of auto-saved disk space (/kaggle/working)
- 16 Gigabytes of temporary, scratchpad disk space (outside /kaggle/working)
- CPU Specifications
  - 4 CPU cores
  - 17 Gigabytes of RAM
- GPU Specifications
  - 2 GPU cores
  - 14 Gigabytes of RAM
4. Exploratory data analysis

In this chapter, we will examine the provided dataset in order to extract the most relevant information and consequently to get a comprehensive perspective of the problem we are dealing with. The Yelp dataset is not a single file but the information is splitted into five distinct datasets. Concretely, we will explore the datasets business, checkin, review, user and tip whose relationships and primary attributes are depicted in the Figure 4.1.

In general terms, we will have to opportunity to analyse and evaluate up to 8GB of data distributed mainly on 5,200,000 user reviews and information on 174,000 businesses. Therefore, there is enough information to discover solid patterns in the users and design reliable predictive models.

Figure 4.1: Relationship between datasets
4.1. Business

The business dataset contains specific information about themselves such as location, star rating or main features. Although, we can analyse up to 192602 establishments with their respective 106 features, we will only try to focus on those businesses whose category belongs to restaurants. In our data, each company can be labelled with one or more of the 1300 categories available. In Figure 4.2 it is shown a ranking with the ten most frequent categories, where Restaurants take first place representing approximately 33% of the instances. In the second and third place, the frequency for Shopping and Food is basically the same but the most significant aspect is that there is an enormous difference in the number of instances respect to the restaurants category, reaching a 50% reduction. The rest of the categories belong to approximately 10000 and 20000 establishments. From now on, only the 59371 instances of the Restaurant category will be subject to upcoming analysis.

![Figure 4.2: Distribution of business categories](image)

Regarding the location of our restaurants, we could use their corresponding latitude and longitude attributes to plot a world heatmap (Figure 4.3). In general, USA and, to a lesser extend, Canada are the countries in which restaurants are located. In the case of USA, most of the restaurants are placed in the North-East and South-West areas whereas Canadian restaurants are closed to the South frontier with USA.

In addition, we might go deeper into the location in such a way that quantitative data, in terms of regions or areas, is obtained. In Figure 4.4, these regions denote clusters of restaurants considering the location or distance among them as a metric. Each cluster is labelled with its number of members. When it comes to American restaurants, we can conclude that they are mostly located near Arizona, Ohio and
4. Exploratory data analysis

Pennsylvania. Respect to Canada, the concentration are on the provinces of Ontario and Alberta with 18363 and 2733 restaurants respectively.

Figure 4.3: Heatmap of locations

Figure 4.4: Cluster of restaurants

Among some features of this dataset, it would be important to identify the relationship between open and closed restaurants because that information could influence, in
the long term, in the performance of the predictions. Particularly, we could identify the factors that caused the closing down. In this case, a closed restaurants does not refer to timetables but if there is current activity or not. Therefore, depicting a pie plot similar to Figure 4.5 we observe that a 71.1% of the restaurants are still open.

![Pie chart showing open and closed restaurants](image)

**Figure 4.5: Proportion open/closed restaurants**

For each restaurant, there is an associated star rating which is computed as the average of all the star ratings assigned for each user to a restaurant. As usual, the star rating is in the range $[1 - 5]$ where 5 is the best possible value. Moreover, we can find out how star rating is distributed in our dataset. More specifically, a bell-shaped curve skewed to the left is created in Figure 4.6 whose average is close to 3.5, which states that most of our restaurants, in general, have a positive feedback from their customers.

![Histogram showing distribution of star ratings](image)

**Figure 4.6: Distribution of star rating**

As a result of the generality of the results above, we are going to compute the average star rate per region, that is in our case the states composing the United States of America. As a consequent, Figure 4.7 shows the result of that approach.

In general, those territories marked in yellow correspond to those in which there is no available data. On the one hand, Vermont is the most outstanding state, being in
the interval $[4.2 - 5.0]$, followed by other states such as Washington, Wisconsin or Ohio belonging to the range $[3.3 - 4.2]$. On the other hand, for instance, the worst restaurants can be found Florida or Arkansas. Not only can we perform a similar analysis to the star review but also to the number of reviews associated to each restaurant. First of all, in Figure 4.8 we are plotting the distribution of reviews of our restaurants. In particular, the vast majority of the companies have been reviewed by less than two thousand users although there exist a very small proportion that reaches up to eight thousand reviews. More precisely, Table 4.1 indicates that at least and at most a business has 3 and 8348 opinions respectively.
<table>
<thead>
<tr>
<th>Statistical metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>33</td>
</tr>
<tr>
<td>std</td>
<td>110</td>
</tr>
<tr>
<td>min</td>
<td>3</td>
</tr>
<tr>
<td>max</td>
<td>8348</td>
</tr>
</tbody>
</table>

Table 4.1: Statistics of business reviews

Now, we are going to extract or analyse the most interesting intrinsic properties for a restaurant. Nevertheless, we have discovered a huge amount of missing values so that we will impute those values using the string *Unknown*.

Increasingly, people choose a restaurant taking into consideration a set of factors rather than just the meal. One of these factors could be the WiFi availability, which recently is not a luxury but a must that should be guarantee by the business owners.

![Figure 4.9: WiFi availability](image)

In Figure 4.9 we observe that approximately one third of the instances are assigned to the options *No WiFi*, *Unknown* and *Free WiFi*. Moreover, there is an insignificant number of restaurants with the categories *Paid WiFi* and *None*. Respect the last option, we presume that the meaning will be similar to the *Unknown* tag despite being considered as distinct entities.

Another criterion to be considered for families with children and teenagers are the age restrictions. This is especially interesting because our dataset is mainly focused on USA and thus, each state has its own regulations. However, as shown in Figure 4.10, almost 100% of our restaurants does not provide this kind of information, whereas a small percentage explicitly sets restrictions to people younger than 21, 19 and 18 years old. By and large, these few restaurants are also nightclubs and casinos.
More aspects regarding restaurants could be their policy. Principally, when going to specific restaurants, features such as the dress code may arise a problem to the customers because there was no previous knowledge of it. In our case, there are four dress codes: *None, Dressy, Formal* and *Casual*. In this moment, *Casual* is by far, as shown in Figure 4.11 the most frequent way of attire. On the contrary, a negligible proportion comprises the *Dressy* and *Formal* attire.

Other curious attribute of the dataset is *Alcohol*. Beforehand, we do not exactly know to which context refers to. However, after exploring this data (Figure 4.12), we can deduce that this attribute refers to the kind of drinks included in the restaurant’s menu. In this case, nearly one third of the restaurants will not include any kind of
drink (or there is no available information). Similarly, customers can opt for full bar in 17500 restaurants while in a lower percentage it is offered beer and wine. There exists also a third of Unknown values.

![Figure 4.12: Alcohol restrictions or availability](image)

Depending on the kind of activity we want to experience on a restaurant, we can labelled a restaurant according to the noise. At this moment, we do not know if the noise level is subject to customers perception or to the different kinds of events that can be performed. In fact, we should not infer a relationship between a noise level and the quality of a restaurant. In Figure 4.13, we can observe that one half of the data have an Average noise level. In contrast, less than 5000 restaurants are categorised as Loud o Very loud and Quiet places represent almost 10000 restaurants. Finally, the rest of the data (Unknown or None) does not imply a certain level of noise.

Regarding smokers, we can gain insight of this particularity through Figure 4.14. Although, we know this graph is not really informative because of the high number of missing values, we can extract the alternatives that are applied to some restaurants. In this case, there is no a huge difference between No smoking, Smoking inside or Smoking outdoor.

To conclude this section, we will analyse the name of the restaurants using the technique of wordclouds (Figure 4.15), which basically relates the use of a certain font size of a word with its frequency. Obviously, the word Restaurant is quite frequent as we have focused on this specific category. This word is followed by others such as Pizza, Cafe or Grill, which are supposed to be common terms in this sector.
4. Exploratory data analysis

Figure 4.13: Noise level

Figure 4.14: Available smoking areas

Figure 4.15: Wordcloud of business names
4.2. Tips

Usually, customers can assess a business selecting a star rate along with a review explaining some facts related to the customer experience. Apart from that, Yelp allows the users to write what it is called tips. These tips can be defined as concise and very short reviews, which are intended to work in a similar way as Twitter does. Our dataset is composed by a total of 1223000 tips which can be reduced up to 800000 restaurant tips, whose attributes are a text, the date of the tip and a compliment count.

First, we could analyse the text tips in the same way we did with the restaurant names, that is, using wordclouds (see Figure 4.16). In general, we would highlight the positivity of the wordcloud because of the appearance of words such as Good, Food, Great, Service or Best. Nonetheless, this reasoning could not be accurate due to the existence of modifiers such as not, never, few ...

![Figure 4.16: Wordcloud of text tips](image)

Previously, we mentioned that tips were assigned to date they were created. Therefore, we could plot the behaviour of the users overtime. In Figure 4.17 we could observe that our instances date back 2009, when start to grow exponentially until 2012 reaching a peak of 600 tips/day. From that movement, it is created a periodic tendency where local maxima occur by the middle of year. On the other hand, local minima occur at the beginning of the year. One explanation to this behaviour is that May, June and July are in general months when the restaurant sector is in continuous growth as a consequence of warm temperatures and summer holidays. Finally, from 2017 to 2019 there is notably a decrease in the number of tips.
Eventually, we need to explore the attribute `compliment_count`, which provides us with a measurement of how good or useful a tip is. For our data (see Figure 4.18), this does not give us further information as the number of compliments is concentrated in the range $[0 - 2]$ although there is a small proportion that certainly exceeds this interval.
4.3. Checkin

The next dataset is called checkin, which stores for each establishment a series of dates in which a customer visited it. After a thorough exploration, we have discovered that on average there are 190 checkins in restaurants spread over more than 10 year. Hence, we have not sufficient data uniformly distributed in order to gather valuable information in terms of years or months. Consequently, we have decided to organise the data according to the days of the week, as could be observed in Figure 4.19. Following our expectations, during the weekdays the number of checkings remain stable until Thursday, when this quantity increases slightly reaching a peak during the weekend, more specifically on Saturday.

![Figure 4.19: Tendency of checkins during the week](image)

4.4. Review

This dataset contains two different kinds of information. The first one is related to the assessment of a business, including star rate, the date and the text review. The second kind includes information about an assessment of the revision itself. In our particular case, other users can vote if a specific review has been funny, cool or useful. According to the star rate distribution of Figure 4.20, we realise that the vast majority of reviews are positive and surprisingly, these mostly come from five stars reviews (160000 user opinions) and to lesser extend from four stars. Negative reviews are defined by one
or two stars but the proportion of each one does not exceed 50000 reviews. Finally, neutral reviews tend to follow the same tendency as the negative ones.

The distributions of the attributes funny, cool and useful (see Figure 4.21) are quite similar, being concentrated on the range \([0 – 100]\), although in some special cases the number of compliments reaches more than 500. As a consequence, the greater is the number of these attributes the more reliable will be a user review.

Taking into account the previous data, if we summarise the textual reviews and users are totally coherent when writing their reviews, the most frequent terms should denote positive feelings. In general, some of those words described in Figure 4.22 are Great, Good, Chicken, Authentic.

The last feature that we will discuss for the users dataset is the number of signups in the platform. We can realise that the time series defined in Figure 4.23 has an upward trend.
trend. Moreover, there exists an evident periodicity. For instance, local minima are reached at the beginning of the year whereas local maxima always occur in the middle of the year. Finally, we can observe that there is an increasing difference between maximum and minimum values overtime.
4.5. User

The last dataset included in Yelp that will be analysed is users. Apart from common features such as name, each instance basically contains statistics about the user interaction within the platform such as the average rating of all reviews or the number of compliments received by the user.

When it comes to the compliment attribute, Yelp categorises the compliments according to the actions that can be done inside the platform. Some of them are compliment about photos, writer or profile and denotes the users perspective for a certain user.

In the following set of pictures, we observe that the attributes `compliment_cool`, `compliment_funny` and `compliment_hot` are extremely similar since most of the distribution is in the range $[0 - 5000]$ and the maximum number of compliment is approximately 35000. On the contrary, the quantity of the `compliment_more` is more reduced, being in the interval $[0 - 500]$.

Every user instance is also assigned the average star rate of all his reviews. This feature allows us to determine whether a user is impartial or not in the assessments. As we may deduce from Figure 4.24, most of the users have a tendency to positive reviews due to the fact that his average star rate is predominantly five and four. This fact could have two reasons: the vast majority of restaurants are fairly good or our users do not analyse negative aspects in the reviews.
To conclude this chapter, we will analyse the historical data about new users in the platform. Our time series starts in the year 2006 and finishes about 2018. This data has an increasing tendency until 2016 when it is reached the global maximum. From that moment, there is a rough decline. Similar to the previous time series, there exists the pattern where local minima occurs at the beginning of a year and local maxima in the middle of the year. The reason why this happens is due to that the use of the platform is subject to seasonality.

Inside the Yelp community, there exist a distinguished group of members known as Elite Squad. These members characterise by some features such as detailed personal profile, well-written reviews, active participation or high quality tips. This status is not for life but a annual program that determines who users deserve to hold this rank. This program aims for encouraging users to follow best practices guidelines and provide
valuable information in the platform. To get an idea about the difficulty of being a member, Figure 4.26 describe that only 5.7% of all the users of the platform were or are currently members of the Elite Squad.

Moreover, we can analyse the described tendency of the elite group overtime. In general, the number of members has increased exponentially. In fact, in 2018 there are 35000 elite members which is nearly 10 times more people respect to the beginning of the sample in 2006. As we can observed, every year increases the numbers of members respect to the previous one. However, we can not assert whether the origin of the increment is due to the number of new numbers or because Yelp has softened the requirements to become a member of the elite group.

The membership to the elite group one year does not imply membership in the upcoming years. In any case, there are many users that hold the status during some
years both consecutive or not. What stands out from this information is that there
exists more users belonging to the Elite group twice than only once. From that point,
the period of permanence decreases. The most curious aspect is that there are even
members that belongs as many years as information we have at our disposal.

Figure 4.28: Elite members in several years

Finally, we are going to discover whether there is a relationship between the years
of membership with the corresponding assessments. According to the data reflected in
Figure 4.29, we deduce that younger members have a more diversity in their opinions
i.e they review restaurants as neutral, good o bad. When a user is a elite member
for six or more year, we observe that there are no reviews for excellent restaurants,
that currently they could deserve 5 stars in a review. Furthermore, the behaviour
also occurs in bad reviews. In that case, reviewers have a tendency to assign to a restaurant at least 3 stars. One of the most convincing explanations of this pattern is that older members have a better reputation and influence in the Yelp community than other members. This implies that some business owners attract the attention of these experienced reviewers to increase the visibility and status respect the competitors. Evidently, these businesses must ensure that the reviewer satisfaction reaches a certain level.
5. Expectation Maximisation results

Once we have carried out an accurate analysis of the data, we can build and train our models. In this chapter, we will focus on building a model to predict the restaurant quality and describe the outcomes obtained for distinct trials. In this case, we will use an implementation of the EM algorithm to solve this task.

5.1. Preliminary steps: data cleaning

Before beginning with the modelling stage, it is important to consider the data preprocessing stage, which will be mainly aimed at improving data quality and adjusting our data to the constraints of the data mining algorithms. Particularly, we will perform operations such as data cleaning, reduction and transformation. Needless to say, the guidelines mentioned below represent a baseline for our data mining project and there may exist variations through the project iterations until we reach the expected results.

Noisy, incomplete or inconsistent data is strongly related with accuracy and quality of our data. In this way, we need some techniques to handle such issues that may affect the performance of our predictive models. In our project, we will only extract information from the datasets review and business.

Regarding the business dataset, we are going to select instances taking into account two criteria. First, we will leave out those instances which do not belong to the restaurants category. Secondly, each restaurant has a valuable feature indicating the number of user reviews. Thus, in order to prevent noisy data and improve quality, we will filter out instances whose maximum number of associated reviews is less than 20.

Respect to review dataset, we will only select the attributes stars and review (textual review). Due to the fact that our first step is to classify restaurants depending on the good or bad quality, we may need to perform some modifications. If we observe
closely the star rating of the reviews, we can determine that 1-2 stars refers to bad reviews, 3 stars to neutral reviews and 4-5 stars to excellent reviews. Therefore, to improve the discriminative power of the algorithm, we should consider some approaches for our project. For instance, some techniques include choosing most active users i.e those users whose number of reviews is greater than a threshold or consider only extreme reviews/restaurants. In addition, when working with this raw data it occurs that some users have assessed a restaurant more than once. This issue could be addressed by deleting duplicate reviews and maintaining only the newest ones, because we assume that they reflect the most accurate user perspective. Finally, we will apply some basic NLP techniques to these reviews.

5.2. Generative model

First of all, we must mention the underlying features of our generative or predictive model. Basically, we are going to combine a Naive Bayes classifier with the EM algorithm, due to the fact that we must solve an incomplete data problem by estimating some missing parameters. In particular, the predictor variables are the star rate and the textual reviews of each user, whereas the class variable will be the level of quality, describing a binary classification problem. Therefore, our model could be represented as it is shown in Figure 5.1. As we can deduce, this model has been defined in such a way that the information given for each user is independent from each other, hence, the conditional independence assumption holds.

![Figure 5.1: Naive Bayes model to predict restaurant quality](image)

For our specific problem, the required steps for the EM algorithm can be described considering several stages.

- **Preliminary action**: It involves the random generation of unknown a priori and a posteriori probabilities.

- **Expectation step**: For each instance or restaurant, we need to infer the probability of belonging to each one of the clusters or classes (good or bad quality). This process is expressed mathematically in Equation 5.1, where $P(S_t \mid y)$ and
$P(R_i \mid y)$ denote, for $i$th user, the a priori probability of the star rate and textual review given the type of quality. Finally, due to the fact that the obtained values are not strictly probabilities, it is necessary to normalise each one of the probabilities to be in the range $[0 - 1]$. This normalisation is computed following the pattern described in Equation (5.2).

\[
P(y = c \mid b_i) = P(y) \prod_{i=1}^{\mid U \mid} P(S_i \mid y) \prod_{i=1}^{\mid U \mid} P(R_i \mid y)
\]

\[
P(y = c \mid b_i) = \frac{P(y \mid b_i)}{\sum_{i=0}^{\mid C \mid} P(y \mid b_i)}
\]

**Maximisation step**: In this last step, we need to re-estimate the unknown parameters according to the new information extracted from the previous step. In particular, both kind of probabilities i.e a posteriori and a priori are not obtained in the usual way by counting number of occurrences of the instance but these occurrences are weighted by the corresponding probability computed in the E step. For instance, the probability of the class variable is computed using the mean of the probability of the instances belonging to a specific cluster.

\[
P(y = c) = \frac{\sum_{i=0}^{N} y_i^j \cdot c}{N}
\]

where $N$ is the number of instances

### 5.3. Proof of concept of EM Algorithm

Once we have implemented our customised Expectation Maximisation algorithm to satisfy our needs, we are going to test it against a synthetic dataset and analyse the results. In general, our input data will consist of a matrix where rows and columns contain information about businesses and users respectively. Simultaneously, each user assigns the corresponding restaurant with a review and the number of stars. In the case shown by Table 5.1, we have created a balanced dataset where businesses $b_1$, $b_2$ and $b_2$ could be labelled as good quality whereas $b_4$, $b_5$ and $b_6$ would denote bad quality. Respect to the review column, we will assume that each review is a list of key features. For our purposes, terms $t_1$, $t_2$ and $t_3$ refer to negative terms. On the contrary, terms $t_6$, $t_7$ and $t_8$ denote positive attributes. In addition, we have added some terms which
are neither good or bad but neutral terms that can be used in all kind of instances. Evidently, these text and stars distribution have been carefully selected to obtained the expected results.

<table>
<thead>
<tr>
<th>business id</th>
<th>User 1</th>
<th>User 2</th>
<th>User 3</th>
<th>User 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Stars</td>
<td>Review</td>
<td>Stars</td>
<td>Review</td>
</tr>
<tr>
<td>$b_1$</td>
<td>1</td>
<td>$t_1t_2$</td>
<td>3</td>
<td>$t_1t_3t_4$</td>
</tr>
<tr>
<td>$b_2$</td>
<td>2</td>
<td>$t_3t_4$</td>
<td>1</td>
<td>$t_1t_2t_5$</td>
</tr>
<tr>
<td>$b_3$</td>
<td>3</td>
<td>$t_1t_4$</td>
<td>2</td>
<td>$t_1t_2t_3$</td>
</tr>
<tr>
<td>$b_4$</td>
<td>4</td>
<td>$t_5t_6$</td>
<td>4</td>
<td>$t_5t_7t_8$</td>
</tr>
<tr>
<td>$b_5$</td>
<td>5</td>
<td>$t_6t_7$</td>
<td>5</td>
<td>$t_1t_7t_8$</td>
</tr>
<tr>
<td>$b_6$</td>
<td>4</td>
<td>$t_6t_8$</td>
<td>5</td>
<td>$t_6t_7t_8$</td>
</tr>
</tbody>
</table>

Table 5.1: Matrix representation of input data

Having explained the dataset, we are going to execute the EM algorithm three times to choose the best alternative and avoid local optima solutions. In particular, this algorithm will compute the probability of belonging to each one of our two clusters. In Table 5.2 we can observe the results for each execution of the algorithm. Analysing the results, we can conclude that the best configuration is obtained in the Test 3 because it assigns bad restaurants in Cluster 0 and good restaurants in Cluster 1. In this example, it is straightforward to choose the best execution of the algorithm. However, if it is necessary to process an immense quantity of data the visual approach is impracticable. Thus, we should select a criterion in order to get the global optimum solution. This issue could be addressed by defining two parameters: **centroid** and **inter-cluster distance**.

The centroid will represent the center of each cluster and is computed by means of average stars of the cluster members. Respect to the inter-cluster distance, this is calculated as the absolute difference among the cluster centroids. As a consequence, our optimum solution will be the one which maximises the inter-cluster distance.

Considering the mentioned method to detect the most suitable solution, Table 5.3 provides a more clear perspective of the results. As a whole, both executions seem to provide a certain level of discriminative power although this especially stands out in the Test 2 and 3 where each cluster play the role of a sort of quality. For example, in Test 2, Cluster 0 could represent good restaurants and Cluster 1 bad restaurants. Finally, according to the results, we might verify two distinct hypothesis: our implementation algorithm works properly and this approach should be fulfil our problem requirements on large scale.
5. Expectation Maximisation results

<table>
<thead>
<tr>
<th>business id</th>
<th>Test 1 Cluster 0</th>
<th>Test 2 Cluster 0</th>
<th>Test 3 Cluster 0</th>
<th>Cluster 1</th>
<th>Cluster 1</th>
<th>Cluster 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_1$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$b_2$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$b_3$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$b_4$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$b_5$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$b_6$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.2: Results small dataset

<table>
<thead>
<tr>
<th>Cluster 0</th>
<th>Cluster 1</th>
<th>Inter-cluster distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>3.0625</td>
<td>2.875</td>
</tr>
<tr>
<td>Test 2</td>
<td>3.6875</td>
<td>1.625</td>
</tr>
<tr>
<td>Test 3</td>
<td>1.75</td>
<td>4.25</td>
</tr>
</tbody>
</table>

Table 5.3: Analysis of results in the synthetic dataset

5.4. Classification for the whole dataset

Using the EM algorithm in a real dataset differs to some extend respect to our synthetic data. First of all, we will need to use a sparse matrix to store efficiently our data due to the fact that every business is not reviewed by each user contrary to the representation shown in Table 5.1. After this step, we have applied several times the Expectation Maximisation algorithm against the matrix in order to get the best possible configuration.

In general, the algorithm reaches a convergence stage in three iterations. In Table 5.4, we have a summary of the most significant results. As we can deduce, there is no a significant difference among the clusters to guarantee with a high level of confidence what each cluster represents. For these results, the clusters centroids is around 3.6 stars which is a positive tendency while the maximum inter-cluster distance is 0.012. One of problems which may occur is that the distribution of businesses stars that we described in Chapter 4 (see Figure 4.6) is mainly positive affecting negatively to the results. In addition, we have a wide set of businesses whose quality would be neutral, hence, there would not be a clear cluster to assign these instances. Thus, we may conclude that both clusters are composed by a mixture of all kind of reviews.
Cluster 0  Cluster 1  Inter-cluster distance
Test 1  3.666  3.660  0.006
Test 2  3.659  3.667  0.008
Test 3  3.664  3.662  0.002
Test 4  3.657  3.669  0.012
Test 5  3.667  3.660  0.007
Test 6  3.663  3.664  0.001

Table 5.4: Analysis of results for the whole dataset

5.5. Remove neutral restaurants

In this part, our next objective will consist of trying to improve the results by obtaining a more clear dataset. As a consequence, we are going to examine the behaviour of the algorithm once we have removed some noise from our data. Should we recall the star rate distribution (see Figure 4.6), we could state that businesses with three stars may difficult our classification model, hence, omitting these instances should improve the performance because the frontier among good and bad quality will be clearer.

Cluster 0  Cluster 1  Inter-cluster distance
Test 1  3.787  3.782  0.005
Test 2  3.781  3.788  0.007
Test 3  3.782  3.787  0.005
Test 4  3.784  3.786  0.002
Test 5  3.779  3.789  0.01
Test 6  3.786  3.788  0.003
Test 7  3.781  3.788  0.007

Table 5.5: Analysis of results from dataset without neutral businesses

The results for this approach are shown in Figure 5.5. Nevertheless, analogous to previous outcomes, the clusters are almost exactly if we compare the inter-cluster distance, whose maximum value is 0.01. The notable difference is that as a consequence of leaving out neutral restaurant, the cluster centroids have increased slightly. More specifically, the centroids have changed from 3.66 to 3.78. In addition, the results from all the performed tests does not differ at all.
5.6. Remove neutral reviews

Considering again the premise of the existence of noisy data, we are going to evaluate the performance of the algorithm when only valuable or useful reviews appear. For instance, users who award a restaurant with 3 stars will not provide the algorithm with relevant information during the classification process. In particular, our assumption is that textual reviews will contain either too many general terms or a combination of terms which include words describing drawbacks or advantages of the business.

<table>
<thead>
<tr>
<th>Test</th>
<th>Cluster 0</th>
<th>Cluster 1</th>
<th>Inter-cluster distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>3.670</td>
<td>3.657</td>
<td>0.013</td>
</tr>
<tr>
<td>Test 2</td>
<td>3.665</td>
<td>3.661</td>
<td>0.004</td>
</tr>
<tr>
<td>Test 3</td>
<td>3.662</td>
<td>3.664</td>
<td>0.002</td>
</tr>
<tr>
<td>Test 4</td>
<td>3.673</td>
<td>3.653</td>
<td>0.02</td>
</tr>
<tr>
<td>Test 5</td>
<td>3.650</td>
<td>3.676</td>
<td>0.026</td>
</tr>
<tr>
<td>Test 6</td>
<td>3.661</td>
<td>3.666</td>
<td>0.005</td>
</tr>
<tr>
<td>Test 7</td>
<td>3.781</td>
<td>3.788</td>
<td>0.007</td>
</tr>
</tbody>
</table>

Table 5.6: Analysis of results without neutral reviews

In this alternative represented by Figure 5.6 both tests obtain similar results although it is true that Test 5 is slightly better.

5.7. Transform data into a balanced classification problem

The last strategies did not obtain significant results which may be caused because the predominance of good restaurants over the bad ones. This disadvantage could be solved if the dataset is transformed into a balanced classification problem. This means that we need to perform a selection of instances in such a way that the number of bad and good instances are quite close.

After performing this and according to Figure 5.7 we can conclude that average stars of the clusters have decreased. However, the most significant aspect is that inter-cluster distance has improved slightly but the values are not enough relevant. In this option, Test 6 would be the best solution whose inter-cluster distance is 0.065.
Cluster 0  Cluster 1  Inter-cluster distance
Test 1  3.463  3.464  0.001
Test 2  3.456  3.470  0.014
Test 3  3.450  3.476  0.026
Test 4  3.486  3.441  0.045
Test 5  3.466  3.460  0.006
Test 6  3.495  3.430  0.065
Test 7  3.781  3.788  0.007

Table 5.7: Analysis of results for balanced dataset

5.8. Selection of users with variety of opinions

Finally, we will try select our users following another criteria. This approach will consist of choosing those users who have a variety of opinions (positive and negative) but balanced at the same time. This set of actions avoids that we take into account users whose tendency is to vote always positively or negatively. In this case, the results (see Table 5.8) describe the same tendency as in the previous experiments. In general, clusters do not seem to describe clearly that they are able to capture different levels of quality. Moreover, there is no a special difference between them but all reviews are mixed equally.

Cluster 0  Cluster 1  Inter-cluster distance
Test 1  3.787  3.782  0.005
Test 2  3.781  3.788  0.007
Test 3  3.782  3.787  0.005
Test 4  3.784  3.786  0.002
Test 5  3.779  3.789  0.01
Test 6  3.786  3.783  0.003
Test 7  3.781  3.788  0.007

Table 5.8: Analysis of results for variety of opinions

5.9. Conclusions

To sum up this section, we claim that unexpected outcomes were obtained despite the fact that the EM algorithm should be suitable to estimate the parameters of our mixture of distributions, considering the proof of concept. There is no doubt that
textual reviews is part of this controversial problem. Specifically, one of the main causes is that the algorithm is not able to deal with semantics, which combined with some equally probable terms for positive and negative opinions such as food, good or place provoke these unexpected results. Therefore, we could admit that the current data is not sufficiently suitable for the classification objectives we want to achieve. This fact triggers a setback for the development of the project and implies reconsider the following steps. As a consequence, the next steps will no longer consist of predicting the quality of restaurants but summarising and analysing the aspects or topics that users mostly describe in their reviews.
6. Latent Dirichlet Allocation results

Continuing with the chapters specialised in the obtained outcomes, we are going to apply Topic Modelling techniques to analyse and summarise the distinct issues that users include in their textual reviews. For our project purposes, we will consider the built-in LDA implementation of the python libraries such as Scikit-learn and Gensim. In addition, we will compare the results obtained by different numbers of topics. In our case, we will organise our corpus in 20 and 25 topics.

For the basic functionality of the LDA algorithm is required a dictionary of terms and convert our input data into a review-term matrix indicating for each of the cells the corresponding metric. In these experiments, we will limit the maximum number of terms to 5000. The final outcome of this process will consist of a probability distribution for each topic, where a term has a certain probability of belonging to each topic. Therefore, a topic will be characterised by the $k$ most frequent words and as a result, we will need to extract the general idea that defines each topic.

6.1. LDA considering term frequency

For this set of experiments, the metric used to convert and represent our corpus into the review-term matrix will be the absolute term frequency. This means that each word will be associated to the number of times that a term appears in a document/review.

6.1.1. Results for 20 topics

After adjusting the algorithm with the described features, we obtain the topics represented in Table [6.1], where we can appreciate the results obtained according to the algorithm along with the most repeated words sorted in descending order by their
membership probability in the topic. Furthermore, every topic has been assigned manually a representative word which describes the general idea of the topic. In general, there are topics which can be more easily identifiable than others. Not only, can we analyse inner aspects of restaurants but also external features. External factors occur less frequently than internal but some users even explain some features such as the Location or the Environment. On the contrary, the vast majority of users focus on internal factors such as Cuisine, Menu or Waiting time.

Analysing the topics, we may find the following patterns. For instance, Topic 7 and 11 seem to represent a general idea because they only refer to aspects such as the Place, Service or Food and do not enter into further details. Conversely, other topics are very specific as they describe features of restaurants. In fact, we are able to identify a variety of cuisines in Topic 2, 3, 10 and 14 which can describe Mexican, Japanese, Chinese and Thai food respectively. Moreover, users usually review certain meals such as Breakfast and Lunch, by commenting on food variety of this specific meals. A similar behaviour occurs in Topic 16 and 19 which mainly highlight the time that customers must wait either to order, serve food or find an available seat. Respect to external restaurant features, we could describe some topics such as Topic 17 and 4. The first one includes the terms Location, Drive and Sub (possibly subway). On the other hand, Topic 4 seems to talk about The Vegas Strip, which is a distinguished zone known for its concentration of resort hotels and casinos.

Once we have discovered the topics and describe them with a relevant term, we are going analysed in detail. To do that, we are going to visualise the LDA topics using the tool pyLDAvis. This visualisation tool is composed by two parts, whose results for our data are shown in Figure 6.1. The left part of the visualisation presents an overview of the topic model, allowing us to detect the prevalence of each topic and the relationship between the topics. In this representation, topics are plotted as circles in a two dimensional place whose centers are computed by means of inter-topic distance. The area of the circles stand for the overall topic prevalence. The second section of the visualisation shows a barchart which represents the individual terms that are the most relevant for understanding the currently selected topic. Each term consist of two bars which denotes both the corpus frequency of a given term as well as its topic frequency.

For example, the most frequent topics in our dataset are Topics 7, 9, 11 and 12 on the grounds that they describe general facts. However, there exists a minority topics such as Topics 1, 4 and 5. Another issue of special interest is that some topics are closer than others. In this visualisation, the distance metric is correlated with level of similarity so that near topics tend to denote common features among topics. In some cases, the distance inter-topic is so close that topics overlap which means that they share some common terms. In particular, if we analyse the group composed by
<table>
<thead>
<tr>
<th>Topic 1</th>
<th>Topic 2</th>
<th>Topic 3</th>
<th>Topic 4</th>
<th>Topic 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffet</td>
<td>Mexican</td>
<td>Japanese</td>
<td>Location</td>
<td>Ambient</td>
</tr>
<tr>
<td>buffet</td>
<td>pizza</td>
<td>roll</td>
<td>vegas</td>
<td>happy</td>
</tr>
<tr>
<td>station</td>
<td>taco</td>
<td>sushi</td>
<td>parking</td>
<td>hour</td>
</tr>
<tr>
<td>step</td>
<td>chip</td>
<td>thai</td>
<td>strip</td>
<td>wine</td>
</tr>
<tr>
<td>prime</td>
<td>salsa</td>
<td>fish</td>
<td>room</td>
<td>month</td>
</tr>
<tr>
<td>pour</td>
<td>mexican</td>
<td>order</td>
<td>las</td>
<td>birthday</td>
</tr>
<tr>
<td>rib</td>
<td>good</td>
<td>fresh</td>
<td>park</td>
<td>pita</td>
</tr>
<tr>
<td>leg</td>
<td>order</td>
<td>salmon</td>
<td>order</td>
<td>special</td>
</tr>
<tr>
<td>les</td>
<td>bean</td>
<td>tuna</td>
<td>mall</td>
<td>drink</td>
</tr>
<tr>
<td>est</td>
<td>crust</td>
<td>japanese</td>
<td>question</td>
<td>hummus</td>
</tr>
<tr>
<td>que</td>
<td>sauce</td>
<td>rice</td>
<td>stay</td>
<td>greek</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topic 6</th>
<th>Topic 7</th>
<th>Topic 8</th>
<th>Topic 9</th>
<th>Topic 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu</td>
<td>General</td>
<td>Breakfast</td>
<td>Menu</td>
<td>Chinese</td>
</tr>
<tr>
<td>restaurant</td>
<td>great</td>
<td>breakfast</td>
<td>good</td>
<td>rice</td>
</tr>
<tr>
<td>menu</td>
<td>food</td>
<td>coffee</td>
<td>dish</td>
<td>noodle</td>
</tr>
<tr>
<td>look</td>
<td>place</td>
<td>egg</td>
<td>order</td>
<td>dish</td>
</tr>
<tr>
<td>variety</td>
<td>good</td>
<td>cream</td>
<td>steak</td>
<td>soup</td>
</tr>
<tr>
<td>wall</td>
<td>love</td>
<td>ice</td>
<td>dessert</td>
<td>beef</td>
</tr>
<tr>
<td>feel</td>
<td>service</td>
<td>tea</td>
<td>sauce</td>
<td>bowl</td>
</tr>
<tr>
<td>dining</td>
<td>time</td>
<td>brunch</td>
<td>cheese</td>
<td>order</td>
</tr>
<tr>
<td>find</td>
<td>friendly</td>
<td>sweet</td>
<td>bread</td>
<td>ramen</td>
</tr>
<tr>
<td>visit</td>
<td>try</td>
<td>chocolate</td>
<td>salad</td>
<td>curry</td>
</tr>
<tr>
<td>experience</td>
<td>staff</td>
<td>toast</td>
<td>meal</td>
<td>chinese</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topic 11</th>
<th>Topic 12</th>
<th>Topic 13</th>
<th>Topic 14</th>
<th>Topic 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>General</td>
<td>Aspect</td>
<td>Thai</td>
<td>Menu</td>
</tr>
<tr>
<td>good</td>
<td>like</td>
<td>hotel</td>
<td>chicken</td>
<td>sandwich</td>
</tr>
<tr>
<td>food</td>
<td>place</td>
<td>live</td>
<td>sauce</td>
<td>salad</td>
</tr>
<tr>
<td>place</td>
<td>time</td>
<td>room</td>
<td>pork</td>
<td>lunch</td>
</tr>
<tr>
<td>price</td>
<td>get</td>
<td>casual</td>
<td>meat</td>
<td>chicken</td>
</tr>
<tr>
<td>like</td>
<td>eat</td>
<td>floor</td>
<td>fry</td>
<td>bread</td>
</tr>
<tr>
<td>pretty</td>
<td>know</td>
<td>view</td>
<td>good</td>
<td>order</td>
</tr>
<tr>
<td>service</td>
<td>good</td>
<td>casino</td>
<td>bbq</td>
<td>option</td>
</tr>
<tr>
<td>come</td>
<td>think</td>
<td>old</td>
<td>spicy</td>
<td>fresh</td>
</tr>
<tr>
<td>bit</td>
<td>want</td>
<td>free</td>
<td>flavour</td>
<td>soup</td>
</tr>
<tr>
<td>well</td>
<td>thing</td>
<td>open</td>
<td>order</td>
<td>vegetarian</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topic 16</th>
<th>Topic 17</th>
<th>Topic 18</th>
<th>Topic 19</th>
<th>Topic 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Location</td>
<td>Fast food</td>
<td>Time</td>
<td>Ambient</td>
</tr>
<tr>
<td>food</td>
<td>location</td>
<td>burger</td>
<td>wait</td>
<td>bar</td>
</tr>
<tr>
<td>order</td>
<td>wing</td>
<td>fry</td>
<td>table</td>
<td>beer</td>
</tr>
<tr>
<td>service</td>
<td>shop</td>
<td>cheese</td>
<td>order</td>
<td>drink</td>
</tr>
<tr>
<td>ask</td>
<td>counter</td>
<td>good</td>
<td>come</td>
<td>great</td>
</tr>
<tr>
<td>server</td>
<td>store</td>
<td>onion</td>
<td>time</td>
<td>place</td>
</tr>
<tr>
<td>come</td>
<td>buy</td>
<td>burrito</td>
<td>seat</td>
<td>night</td>
</tr>
<tr>
<td>time</td>
<td>drive</td>
<td>order</td>
<td>get</td>
<td>patio</td>
</tr>
<tr>
<td>table</td>
<td>market</td>
<td>bun</td>
<td>minute</td>
<td>game</td>
</tr>
<tr>
<td>take</td>
<td>find</td>
<td>dog</td>
<td>sit</td>
<td>music</td>
</tr>
<tr>
<td>experience</td>
<td>sub</td>
<td>mac</td>
<td>busy</td>
<td>nice</td>
</tr>
</tbody>
</table>

Table 6.1: Results 20 topics using Term frequency
Topics 7, 11, 12, 16 and 19 we found out that words such as *Time*, *Food* or *Good* are included at least in the fifty percent of the topic selection. In Figure 6.2 we can see the percentage of term repetition of the ten most frequent words in the group of topics mentioned above. In general terms, the ten words have a repetition percentage between the 66% and 33%. Finally, we may extract from the visualisation that there are two main groups.

- **Group 1**: The first group could be represented by Topic 12, being the most frequent for this group. Looking at the table of topics, we may conclude that surrounding topics refers to general aspects such as *Food*, *Place* and also *Good*. Considering, the terms composing these topics, we could assert that this group refers to non-specific aspects of the restaurants.

- **Group 2**: On the contrary, the second group is formed around Topic 10. Undoubtedly, the topics composing this group has an evident relationship. In this case, the topics basically describe different cuisines.

![Intertopic Distance Map (via multidimensional scaling)](image)

![Top-10 Most Relevant Terms for Topic 9](image)

Figure 6.1: Visualisation of 20 topics with term frequency

In the visualisation, we have selected Topic 9 (in color red). As we can see, the frequency of the terms in corpus changes but the term frequency inside the topic seem
to be more or less similar for all the terms. This case is not common but there exist two or three terms which predominate inside the topic. Certainly, better outcomes can be obtained if the model is tuned accordingly.

During this previous steps, we have only considered the topics distribution as a whole. Right now, we will split the corpus into positive or negative reviews to after that, describe the topic distribution for negative and positive reviews. The comparison of negative and positive topics are depicted in Figure 6.3, where positive texts are in the left and negative in the right part. As we can observe, in both cases there is a evident predominance of Topics 7, 9 11 and 12, which refer to general issues. However, the occurrence is quite different for both graphics. Whereas for positive texts Topic 7 is clearly the most repeated topic, in the other kind of texts there no exists such distinction. By and large, both topic distributions are quite similar. Apart from the general topics, some of the topics more relevant in the user community would be Topic 16 and 19 which refers to the waiting time. Certainly, this means that users appreciate positively efficient restaurants and penalise long waits and delays.

Now we are going to extract for each document the primary and secondary topic (i.e two most frequent topics inside a document) to describe the possible relationships in our data. To represent the results, we will use a co-ocurrence matrix to check the topics that usually are linked. In Figure 6.4, we observe the correlations of topics for both kind of texts (positive and negative texts correspond left and right images respectively). Regarding positive texts, Topic 7 is very correlated with the rest of topics as well as Topics 11 and 12. Specially, there is a strong level coincidence among the pair of topics (9, 7), (7, 11) and (7, 12). A similar tendency occurs with the co-occurrence matrix for negative reviews. However, if we observe closely we have that the set of topics (7, 19)
and (12, 16) are very common. From these results, we can mention that these pair of topics seem too general, hence, they overshadow less frequent topics, not contributing to obtain more specific results.

Finally, we are going to compute and analyse the probability of appearance of the primary and secondary topics at the same time. Respect to the positive graphic (see Figure 6.5), Topics 7, 12 and 11 are the most frequent as first and secondary topics. Regarding the rest of topics, they are more frequent as secondary topic than primary. We must remember that a document is defined through a mixture of topics, hence, we will consider a primary and secondary set of topics. Respect to negative reviews (see Figure 6.6), the distribution is a little distinct. Topics 12, 11 and 7 are equally frequent as primary topic. Interestingly, more that 50% of the reviews belong to the first four
topics as primary topic.

Figure 6.5: Frequency of primary and secondary positive topics

Figure 6.6: Frequency of primary and secondary negative topics

6.1.2. Results for 25 topics

In this section, we will perform the same experiment except for the number of topics, which will be changed from 20 to 25. The objective of this trial is to analyse whether significant changes occur or not.

The topic description is shown in Table 6.2, whose most remarkable aspect is that Topic 4 and 6 have not been properly categorised because we are not able to discover similar patterns inside the topic. Apart from that, the rest of topics are more specific and more understandable respect to the previous outcomes. For instance we have detected topics such as Topic 13, 15, 16, 20 and 23. The first one has been described with the word *Impression* due to the fact the corresponding terms detail how users are
surprised by the new changes introduced in the restaurant after some time. Analogous, reviews belonging to Topic 15 highlight alternative meals for vegetarians. Topic 20 is a bit difficult to labelled although we opted for relating with Canadian cuisine because *poutine* is a typical dish of that country. Respect to Topic 16, we have decide that this describes restaurants offers or discounts because the terms free and coupon are really likely in some business. Finally, it is likely that Topic 23 is a negative topic because refers to the word *slow, dry* or *bland*. Finally, most of the topics are the same as in the last results. In particular, *Mexican, Dessert, Menu, Location* and *Thai* are common topics in both results.

The next step would be the graphical representation of the topics. In this case, if we perform an analysis of the visualisations for 20 and 25 topics respectively, we may discuss some important contrasts. First and foremost, these topics are more separated from each other and located in the whole area, hence there are no group of topics so clearly. Nevertheless, there is a set of groups whose level of similarity is high. We are referring to Topic 3, 7, 11, 12 and 19. This group characterises for being mostly part of general terms.

![Intertopic Distance Map (via multidimensional scaling)](image)

![Top-10 Most Relevant Terms for Topic 3](image)

Figure 6.7: Visualisation 25 topics using Term Frequency

If we analyse the distribution of topics, taking into account positive or negative opinions in Figure 6.8, we observe that the general topics are the most frequent although there are some differences in both representations. For instance, Topic 7 is the most
### Table 6.2: Results 25 topics using Term frequency

<table>
<thead>
<tr>
<th>Topic 1</th>
<th>Topic 2</th>
<th>Topic 3</th>
<th>Topic 4</th>
<th>Topic 5</th>
<th>Topic 6</th>
<th>Topic 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffet</td>
<td>Mexican</td>
<td>General</td>
<td>Unknown</td>
<td>Dessert</td>
<td>Unknown</td>
<td>General</td>
</tr>
<tr>
<td>vegas</td>
<td>pizza</td>
<td>like</td>
<td>juice</td>
<td>cream</td>
<td>store</td>
<td>great</td>
</tr>
<tr>
<td>rib</td>
<td>taco</td>
<td>order</td>
<td>meatball</td>
<td>dessert</td>
<td>room</td>
<td>food</td>
</tr>
<tr>
<td>buffet</td>
<td>chip</td>
<td>place</td>
<td>sorry</td>
<td>coffe</td>
<td>card</td>
<td>place</td>
</tr>
<tr>
<td>ramen</td>
<td>salsa</td>
<td>time</td>
<td>update</td>
<td>chocolate</td>
<td>smell</td>
<td>good</td>
</tr>
<tr>
<td>las</td>
<td>mexican</td>
<td>ask</td>
<td>monday</td>
<td>ice</td>
<td>buy</td>
<td>service</td>
</tr>
<tr>
<td>station</td>
<td>good</td>
<td>food</td>
<td>medium</td>
<td>good</td>
<td>like</td>
<td>love</td>
</tr>
<tr>
<td>expectation</td>
<td>burrito</td>
<td>know</td>
<td>bother</td>
<td>like</td>
<td>hand</td>
<td>time</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu</td>
<td>Menu</td>
<td>Breakfast</td>
<td>General</td>
<td>Ambient</td>
<td>Impression</td>
<td>Italian</td>
</tr>
<tr>
<td>menu</td>
<td>steak</td>
<td>breakfast</td>
<td>good</td>
<td>bar</td>
<td>visit</td>
<td>pasta</td>
</tr>
<tr>
<td>flavour</td>
<td>restaurant</td>
<td>egg</td>
<td>food</td>
<td>beer</td>
<td>year</td>
<td>bread</td>
</tr>
<tr>
<td>flavor</td>
<td>good</td>
<td>brunch</td>
<td>place</td>
<td>drink</td>
<td>time</td>
<td>salad</td>
</tr>
<tr>
<td>dish</td>
<td>dinner</td>
<td>pancake</td>
<td>price</td>
<td>place</td>
<td>new</td>
<td>italian</td>
</tr>
<tr>
<td>crepe</td>
<td>dish</td>
<td>waffle</td>
<td>like</td>
<td>great</td>
<td>menu</td>
<td>sauce</td>
</tr>
<tr>
<td>unique</td>
<td>meal</td>
<td>bacon</td>
<td>pretty</td>
<td>good</td>
<td>variety</td>
<td>tomato</td>
</tr>
<tr>
<td>serve</td>
<td>order</td>
<td>coffee</td>
<td>service</td>
<td>night</td>
<td>know</td>
<td>garlic</td>
</tr>
<tr>
<td>orange</td>
<td>great</td>
<td>morning</td>
<td>quality</td>
<td>hour</td>
<td>find</td>
<td>dish</td>
</tr>
<tr>
<td>sweet</td>
<td>dessert</td>
<td>employee</td>
<td>restaurant</td>
<td>nice</td>
<td>day</td>
<td>cheese</td>
</tr>
<tr>
<td>ingredient</td>
<td>menu</td>
<td>sunday</td>
<td>think</td>
<td>happy</td>
<td>option</td>
<td>olive</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topic 15</th>
<th>Topic 16</th>
<th>Topic 17</th>
<th>Topic 18</th>
<th>Topic 19</th>
<th>Topic 20</th>
<th>Topic 21</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variety</td>
<td>Discount</td>
<td>Asian</td>
<td>Location</td>
<td>Time</td>
<td>Canadian</td>
<td>Drinks</td>
</tr>
<tr>
<td>lunch</td>
<td>burger</td>
<td>roll</td>
<td>location</td>
<td>order</td>
<td>sandwich</td>
<td>tea</td>
</tr>
<tr>
<td>salad</td>
<td>fry</td>
<td>sushi</td>
<td>parking</td>
<td>wait</td>
<td>baget</td>
<td>drink</td>
</tr>
<tr>
<td>option</td>
<td>bun</td>
<td>noodle</td>
<td>restaurant</td>
<td>table</td>
<td>sub</td>
<td>cup</td>
</tr>
<tr>
<td>quick</td>
<td>free</td>
<td>soup</td>
<td>area</td>
<td>come</td>
<td>poutine</td>
<td>milk</td>
</tr>
<tr>
<td>menu</td>
<td>double</td>
<td>order</td>
<td>locate</td>
<td>time</td>
<td>slider</td>
<td>attention</td>
</tr>
<tr>
<td>soup</td>
<td>shake</td>
<td>pho</td>
<td>lot</td>
<td>food</td>
<td>cookie</td>
<td>gem</td>
</tr>
<tr>
<td>bowl</td>
<td>patty</td>
<td>good</td>
<td>find</td>
<td>minute</td>
<td>bread</td>
<td>ice</td>
</tr>
<tr>
<td>veggie</td>
<td>gluten</td>
<td>place</td>
<td>street</td>
<td>seat</td>
<td>pour</td>
<td>green</td>
</tr>
<tr>
<td>vegetarian</td>
<td>girlfriend</td>
<td>rice</td>
<td>inside</td>
<td>take</td>
<td>les</td>
<td>water</td>
</tr>
<tr>
<td>meal</td>
<td>coupon</td>
<td>dinner</td>
<td>open</td>
<td>get</td>
<td>est</td>
<td>iced</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topic 22</th>
<th>Topic 23</th>
<th>Topic 24</th>
<th>Topic 25</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Negative</td>
<td>Food</td>
<td>Thai</td>
</tr>
<tr>
<td>chicken</td>
<td>star</td>
<td>cheese</td>
<td>sauce</td>
</tr>
<tr>
<td>good</td>
<td>slow</td>
<td>onion</td>
<td>rice</td>
</tr>
<tr>
<td>fry</td>
<td>reasonable</td>
<td>potato</td>
<td>chicken</td>
</tr>
<tr>
<td>order</td>
<td>bland</td>
<td>fry</td>
<td>pork</td>
</tr>
<tr>
<td>come</td>
<td>taste</td>
<td>mac</td>
<td>spicy</td>
</tr>
<tr>
<td>get</td>
<td>service</td>
<td>dog</td>
<td>thai</td>
</tr>
<tr>
<td>try</td>
<td>dry</td>
<td>good</td>
<td>dish</td>
</tr>
<tr>
<td>place</td>
<td>review</td>
<td>hot</td>
<td>order</td>
</tr>
<tr>
<td>delicious</td>
<td>fairly</td>
<td>meat</td>
<td>beef</td>
</tr>
<tr>
<td>definitely</td>
<td>flavour</td>
<td>grill</td>
<td>meat</td>
</tr>
</tbody>
</table>
frequent in positive reviews, whereas for negative reviews, this topic is the second most frequent.

Figure 6.8: Distribution of topics for negative and positive texts

Now we are going to analyse the relationship between primary and secondary topics. The most impressive features shown in Figure 6.9 are that for positive reviews, Topic 3 and 7 have a significant relationship with the rest of the topics either for primary or secondary topics. Nevertheless, for negatives reviews, there is a special predominance in the occurrence of the pair of topics (3, 19) whereas the rest of topics, although co-related, are not so important. The general conclusion of this, is that positive reviews assess several factors while negative reviews place special emphasis in Topic 19, which refers to the wait time.

However, if we analyse the frequency of primary and secondary topics, we conclude that their frequency is very similar according to Figure 6.10 and 6.11. In particular, in both cases the most frequent topics i.e Topic 7, 3 and 11 appear at least a 20% in our corpus. On the contrary, the least frequent topics i.e Topic 1, 4 and 20 does not exceed the occurrence probability of 2%.

6.2. LDA considering Term frequency-Inverse document frequency

6.2.1. Results for 20 topics

In this section, we are going to describe the LDA results in similar way we did in the previous section but performing a slight modification. In this case under consideration, we will replace the term frequency matrix with a TF-IDF matrix achieving that frequent words will be penalised during the classification process. One of the dilemmas of using this data structure in LDA is that the algorithm is based on probabilities not metrics which not ensure the correctness of the procedure.
Nonetheless, in Table 6.3 we provide the corresponding results to verify whether we have discovered an enhancement or not.

Using our criterion and Figure 6.12, four main groups are composed:

- **Group 1**: The members are Topic 1, 2, 3, 4, 5, 6, 7 and 9, describing aspects related to general features of the restaurant as well as distinct and specific cuisines. For instance, Topic 4 only focuses on the *Seafood* in the buffet and Topic 6 and 9 in *Indian* and *Mexican* food respectively.

- **Group 2**: It composed by Topic 8, 10 and 11. This first topic has been labelled as *Unknown* because uniquely captures meaningless words especially, French stop-words. On the contrary, the remaining the topics refer to food especially *Chinese*
<table>
<thead>
<tr>
<th>Topic 1</th>
<th>Topic 2</th>
<th>Topic 3</th>
<th>Topic 4</th>
<th>Topic 5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td><strong>Ambient</strong></td>
<td><strong>General</strong></td>
<td><strong>Seafood</strong></td>
<td><strong>Menu</strong></td>
</tr>
<tr>
<td>food</td>
<td>beer</td>
<td>good</td>
<td>buffet</td>
<td>steak</td>
</tr>
<tr>
<td>good</td>
<td>bar</td>
<td>order</td>
<td>crab</td>
<td>lobster</td>
</tr>
<tr>
<td>place</td>
<td>game</td>
<td>chicken</td>
<td>oyster</td>
<td>salad</td>
</tr>
<tr>
<td>great</td>
<td>great</td>
<td>like</td>
<td>seafood</td>
<td>pasta</td>
</tr>
<tr>
<td>service</td>
<td>drink</td>
<td>come</td>
<td>leg</td>
<td>filet</td>
</tr>
<tr>
<td>lunch</td>
<td>music</td>
<td>place</td>
<td>crawfish</td>
<td>wine</td>
</tr>
<tr>
<td>price</td>
<td>sport</td>
<td>food</td>
<td>calm</td>
<td>bread</td>
</tr>
<tr>
<td>time</td>
<td>fun</td>
<td>try</td>
<td>shrimp</td>
<td>dessert</td>
</tr>
<tr>
<td>friendly</td>
<td>good</td>
<td>sauce</td>
<td>dumpling</td>
<td>pasty</td>
</tr>
<tr>
<td>come</td>
<td>watch</td>
<td>dish</td>
<td>defiantly</td>
<td>foie</td>
</tr>
<tr>
<td><strong>Topic 6</strong></td>
<td><strong>Topic 7</strong></td>
<td><strong>Topic 8</strong></td>
<td><strong>Topic 9</strong></td>
<td><strong>Topic 10</strong></td>
</tr>
<tr>
<td><strong>Indian</strong></td>
<td><strong>General</strong></td>
<td><strong>Unknown</strong></td>
<td><strong>Mexican</strong></td>
<td><strong>Chinese</strong></td>
</tr>
<tr>
<td>indian</td>
<td>great</td>
<td>les</td>
<td>taco</td>
<td>sum</td>
</tr>
<tr>
<td>naan</td>
<td>place</td>
<td>est</td>
<td>salsa</td>
<td>dim</td>
</tr>
<tr>
<td>buffet</td>
<td>food</td>
<td>pour</td>
<td>mexican</td>
<td>pastrami</td>
</tr>
<tr>
<td>masala</td>
<td>good</td>
<td>que</td>
<td>burrito</td>
<td>deli</td>
</tr>
<tr>
<td>kabob</td>
<td>nice</td>
<td>pas</td>
<td>chip</td>
<td>roti</td>
</tr>
<tr>
<td>tikka</td>
<td>service</td>
<td>une</td>
<td>margarita</td>
<td>corned</td>
</tr>
<tr>
<td>lamb</td>
<td>restaurant</td>
<td>des</td>
<td>bean</td>
<td>reuben</td>
</tr>
<tr>
<td>calzone</td>
<td>bar</td>
<td>bon</td>
<td>tortilla</td>
<td>delivery</td>
</tr>
<tr>
<td>paneer</td>
<td>love</td>
<td>mais</td>
<td>enchilada</td>
<td>overpriced</td>
</tr>
<tr>
<td>chicken</td>
<td>drink</td>
<td>tres</td>
<td>guacamole</td>
<td>chinese</td>
</tr>
<tr>
<td><strong>Topic 11</strong></td>
<td><strong>Topic 12</strong></td>
<td><strong>Topic 13</strong></td>
<td><strong>Topic 14</strong></td>
<td><strong>Topic 15</strong></td>
</tr>
<tr>
<td><strong>Fast food</strong></td>
<td><strong>Service</strong></td>
<td><strong>Variety</strong></td>
<td><strong>Italian</strong></td>
<td><strong>Drinks</strong></td>
</tr>
<tr>
<td>burger</td>
<td>found</td>
<td>vegan</td>
<td>pizza</td>
<td>coffee</td>
</tr>
<tr>
<td>fry</td>
<td>place</td>
<td>pita</td>
<td>crust</td>
<td>tea</td>
</tr>
<tr>
<td>wing</td>
<td>order</td>
<td>tamale</td>
<td>slice</td>
<td>ice</td>
</tr>
<tr>
<td>cheese</td>
<td>good</td>
<td>eastern</td>
<td>topping</td>
<td>latte</td>
</tr>
<tr>
<td>onion</td>
<td>time</td>
<td>mesa</td>
<td>thin</td>
<td>milk</td>
</tr>
<tr>
<td>good</td>
<td>come</td>
<td>relleno</td>
<td>peperoni</td>
<td>cream</td>
</tr>
<tr>
<td>sandwich</td>
<td>like</td>
<td>ranchero</td>
<td>pie</td>
<td>boba</td>
</tr>
<tr>
<td>shake</td>
<td>service</td>
<td>ari</td>
<td>italian</td>
<td>smoothie</td>
</tr>
<tr>
<td>bbq</td>
<td>get</td>
<td>pricy</td>
<td>meatball</td>
<td>shop</td>
</tr>
<tr>
<td>chicken</td>
<td>wait</td>
<td>kebab</td>
<td>cheese</td>
<td>bubble</td>
</tr>
<tr>
<td><strong>Topic 16</strong></td>
<td><strong>Topic 17</strong></td>
<td><strong>Topic 18</strong></td>
<td><strong>Topic 19</strong></td>
<td><strong>Topic 20</strong></td>
</tr>
<tr>
<td><strong>Japanese</strong></td>
<td><strong>Negative</strong></td>
<td><strong>Thai</strong></td>
<td><strong>Breakfast</strong></td>
<td><strong>General</strong></td>
</tr>
<tr>
<td>sushi</td>
<td>baget</td>
<td>pho</td>
<td>breakfast</td>
<td>great</td>
</tr>
<tr>
<td>roll</td>
<td>alot</td>
<td>noodle</td>
<td>egg</td>
<td>love</td>
</tr>
<tr>
<td>sashimi</td>
<td>schnitzel</td>
<td>ramen</td>
<td>brunch</td>
<td>food</td>
</tr>
<tr>
<td>ayce</td>
<td>rooftop</td>
<td>soup</td>
<td>pancake</td>
<td>happy</td>
</tr>
<tr>
<td>tuna</td>
<td>hole</td>
<td>thai</td>
<td>toast</td>
<td>service</td>
</tr>
<tr>
<td>tempura</td>
<td>esp</td>
<td>broth</td>
<td>coffee</td>
<td>amazing</td>
</tr>
<tr>
<td>japanese</td>
<td>wall</td>
<td>good</td>
<td>waffle</td>
<td>hour</td>
</tr>
<tr>
<td>fish</td>
<td>reliable</td>
<td>rice</td>
<td>bacon</td>
<td>place</td>
</tr>
<tr>
<td>miso</td>
<td>junk</td>
<td>pork</td>
<td>good</td>
<td>good</td>
</tr>
<tr>
<td>teriyaki</td>
<td>bike</td>
<td>beef</td>
<td>french</td>
<td>awesome</td>
</tr>
</tbody>
</table>

Table 6.3: Results 20 topics using Tf-idf
6. Latent Dirichlet Allocation results

Figure 6.11: Frequency of primary and secondary negative topics

and Fast food.

- **Group 3:** It comprises Topic 13, 15 and 17. First, Topic 13 is not very clarifying respect to its content. In general, customers describe ingredients or dishes which come from places around the world. Particularly, *pita* is kind of round flatbread common in the Mediterranean, *tamale* is a typical dish from the Mesoamerican culture and *kebab* have its origin in Middle Easter cuisine. A similar problem occurs with Topic 17 due to its lack of specificity. In this topic, we consider the terms relate to negative features as consequence of the occurrence of the term *junk*. Finally, in this group the Topic 15 is the most evident topic because it describes several drinks.

- **Group 4:** The last group is formed by Topic 14, 16, 18, 19 and 20. Except for Topic 20, the rest of the topics mainly denote dishes for either a concrete cuisine or specific time of the day. Regarding cuisine, the topics are easily identifiable either the corresponding term occurs inside the topic or denote terms strongly related. In particular, the *Italian*, *Japanese* and *Thai* food. Finally, in Topic 19 users can explain the variety or the features of a restaurant although it seems there is buffet during breakfast time. Finally, we will assume that Topic 20 is general.

From those groups we can summarise that its members share some categories. Moreover, each group has a different level of occurrence. In this scenario, the topics of group one are more frequent that the rest of groups. On the other hand, group four will be less frequent that group 1, 2, and 3.

As usual, we will verify the topic frequency regarding the bad or good quality of the restaurants (see Figure 6.13). In this case, there are only four relevant topics for
both cases. The topics sorted in descending order are Topic 12, 3, 7 and 1. From the terms belonging to Topic 20, we can tell that customers consider the service provided by the restaurant totally essential. This includes the availability to find table as well as the wait time. The other three topics describe general ideas but each one has its own peculiarities. First of all, Topic 3 tends to suggest users recommendations. Topic 7 highlights the Service and the Drinks. However, Topic 1 provides feedback about the Time, Price and Service. A small proportion of users likes to assess features contained in Topic 18, 19 and 20. Whereas Topic 20 seems to assess the Service offered, the Topic 18 and 19 focus on Thai food restaurants and the variety of food served during breakfast time respectively.

In this moment, we will proceed to extract and compare the relationship between primary and the secondary topics (see Figure 6.14). Comparing both co-occurence matrices, we reach a conclusion on the analogy of the graphs. In this case, the most frequent topics mentioned before (i.e Topic 12, 3, 7 and 1) are highly correlated with the rest of the topics. Nevertheless, there is a special relationship among the most frequent topic. For instance, we could emphasise the pair of topics (3,12), (7,12) or (12,9).

Finally, we are going to analyse the frequency of the primary and secondary topics. Similar to the previous results, the outcome for positive and negative differs slightly. From the graphic, we have discovered that in at least 50% of the documents appear
6. Latent Dirichlet Allocation results

Figure 6.13: Distribution of topics for negative and positive texts

the Topic 12 as either primary or secondary topic for both positive or negative texts. The following topic would be Topic 3 which occurs at least a 35% in our corpus. The next two topics are Topic 7 and 1 but their percentage of occurrence does not exceed 20%. These four topics tend to occur more likely as a primary topic. On the contrary, the rest of topics have at most a frequency of 10% and they usually act as a secondary topic.

Figure 6.14: Co-occurrence matrix for positive and negative documents
6.2.2. Results for 25 topics

Similar to what we have done, we will comment the result obtained by computing 25 topics, which are shown in Table 6.4. Apart from the general topics, there are confusing topics which cannot be accurately classified. Some of the controversial topics are Topic 12, 16 and 17. We have tagged the first two topics as *Experience* because it seems the description of a specific user but there are no common terms to exactly assert a concrete category. Respect to Topic 17, we have the same problem with other language than English stopwords. In general terms, these topics are quite similar to the previous results although there are more specific and rare terms such as *Kabob* or *souvlaki*.

When it comes to the visualisation of the topic (Figure 6.17), we observe that there exists an evident group composed by topics such as Topic 3, 9 or 6.
Table 6.4: Results 25 topics using Tf-idf
Figure 6.17: Visualisation 25 topics using Tf-idf

Respect to the distribution of topics in terms of positive or negative review (see Figure 6.18), we should mention that the probabilities tend to be very similar. In fact, the only main difference is that Topic 3 and 23 are the first and second most frequent topics for positive reviews. On the contrary, Topic 23 and 3 are the first and second most frequent topics for negative reviews. These two topics refer to general aspects such the Place, Service or Food. The next topics would be Topic 5 and 9, which try to describe specific cuisines (fast food and Asian).

Finally, we will describe the relationship among topics and the two levels of quality. According the Figure 6.19, we can conclude that both co-relation matrices are identical. In particular, Topic 1, 3, 7 and 12 are very related with the rest of the topics and curiously, the pair of topics (3, 12) is the most frequent. This is curious because although Topic 3 is very general, Topic 12 is very specific, hence, the topics should not be very co-related. Another important relation is the pair (7, 12), which combines a general opinion with the variety of a specific food.

To conclude, we can discuss that the TF-IDF approach has overshadowed some topics. However, we expected that the most general terms and topics would have been penalised, but this is not the case.
6. Latent Dirichlet Allocation results

Figure 6.18: Distribution of topics for negative and positive texts

Figure 6.19: Co-occurrence matrix for positive and negative documents

6.3. LDA filtering most common terms

To end up with this chapter, we are going to describe two techniques to delete noise by removing the most frequent terms. The first approach consists of using the LDA algorithm but ignoring terms that have a document frequency strictly higher than the given threshold. In this trial, the threshold will be 50%, obtaining the results shown in Figure 6.5. In general, we have obtained similar results respect to the previous experiments. However, the results are better because topics share few general terms in common. Moreover, these shown terms seem more usual, in contrast to the outcomes obtained when we have used the TF-IDF matrix.
<table>
<thead>
<tr>
<th>Topic 1</th>
<th>Topic 2</th>
<th>Topic 3</th>
<th>Topic 4</th>
<th>Topic 5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Experience</strong></td>
<td><strong>Menu</strong></td>
<td><strong>Time</strong></td>
<td><strong>Mexican</strong></td>
<td><strong>Italian</strong></td>
</tr>
<tr>
<td>vegas</td>
<td>chicken</td>
<td>order</td>
<td>taco</td>
<td>salad</td>
</tr>
<tr>
<td>buffet</td>
<td>fry</td>
<td>time</td>
<td>chip</td>
<td>cheese</td>
</tr>
<tr>
<td>room</td>
<td>order</td>
<td>come</td>
<td>salsa</td>
<td>bread</td>
</tr>
<tr>
<td>hotel</td>
<td>bean</td>
<td>ask</td>
<td>mexican</td>
<td>sauce</td>
</tr>
<tr>
<td>las</td>
<td>rice</td>
<td>table</td>
<td>burrito</td>
<td>tomato</td>
</tr>
<tr>
<td>stay</td>
<td>sauce</td>
<td>get</td>
<td>fish</td>
<td>fresh</td>
</tr>
<tr>
<td>casino</td>
<td>salad</td>
<td>wait</td>
<td>margarita</td>
<td>order</td>
</tr>
<tr>
<td>station</td>
<td>lunch</td>
<td>take</td>
<td>tortilla</td>
<td>garlic</td>
</tr>
<tr>
<td>prime</td>
<td>get</td>
<td>say</td>
<td>guacamole</td>
<td>slice</td>
</tr>
<tr>
<td>leg</td>
<td>meal</td>
<td>minute</td>
<td>order</td>
<td>italian</td>
</tr>
<tr>
<td><strong>Topic 6</strong></td>
<td><strong>Topic 7</strong></td>
<td><strong>Topic 8</strong></td>
<td><strong>Topic 9</strong></td>
<td><strong>Topic 10</strong></td>
</tr>
<tr>
<td><strong>Asian</strong></td>
<td><strong>General</strong></td>
<td><strong>Breakfast</strong></td>
<td><strong>Menu</strong></td>
<td><strong>Asian</strong></td>
</tr>
<tr>
<td>roll</td>
<td>place</td>
<td>cream</td>
<td>restaurant</td>
<td>wing</td>
</tr>
<tr>
<td>sushi</td>
<td>great</td>
<td>ice</td>
<td>great</td>
<td>meat</td>
</tr>
<tr>
<td>thai</td>
<td>love</td>
<td>tea</td>
<td>dish</td>
<td>bbq</td>
</tr>
<tr>
<td>fish</td>
<td>service</td>
<td>dessert</td>
<td>steak</td>
<td>pork</td>
</tr>
<tr>
<td>order</td>
<td>time</td>
<td>chocolate</td>
<td>dinner</td>
<td>rib</td>
</tr>
<tr>
<td>fresh</td>
<td>friendly</td>
<td>sweet</td>
<td>menu</td>
<td>sauce</td>
</tr>
<tr>
<td>lunch</td>
<td>try</td>
<td>cake</td>
<td>meal</td>
<td>pull</td>
</tr>
<tr>
<td>place</td>
<td>amazing</td>
<td>waffle</td>
<td>service</td>
<td>beef</td>
</tr>
<tr>
<td>restaurant</td>
<td>staff</td>
<td>try</td>
<td>wine</td>
<td>dim</td>
</tr>
<tr>
<td>salmon</td>
<td>delicious</td>
<td>flavour</td>
<td>delicious</td>
<td>brisket</td>
</tr>
<tr>
<td><strong>Topic 11</strong></td>
<td><strong>Topic 12</strong></td>
<td><strong>Topic 13</strong></td>
<td><strong>Topic 14</strong></td>
<td><strong>Topic 15</strong></td>
</tr>
<tr>
<td><strong>General</strong></td>
<td><strong>Breakfast</strong></td>
<td><strong>Negative</strong></td>
<td><strong>Variety</strong></td>
<td><strong>Menu</strong></td>
</tr>
<tr>
<td>ramen</td>
<td>sandwich</td>
<td>place</td>
<td>dish</td>
<td>option</td>
</tr>
<tr>
<td>bowl</td>
<td>breakfast</td>
<td>like</td>
<td>noodle</td>
<td>vegetarian</td>
</tr>
<tr>
<td>not</td>
<td>egg</td>
<td>think</td>
<td>soup</td>
<td>veggi</td>
</tr>
<tr>
<td>wrap</td>
<td>coffee</td>
<td>pretty</td>
<td>rice</td>
<td>veggan</td>
</tr>
<tr>
<td>sized</td>
<td>brunch</td>
<td>price</td>
<td>order</td>
<td>healthy</td>
</tr>
<tr>
<td>cup</td>
<td>pancake</td>
<td>taste</td>
<td>sauce</td>
<td>complaint</td>
</tr>
<tr>
<td>orange</td>
<td>toast</td>
<td>well</td>
<td>spicy</td>
<td>free</td>
</tr>
<tr>
<td>stomach</td>
<td>bacon</td>
<td>star</td>
<td>beef</td>
<td>platter</td>
</tr>
<tr>
<td>awhile</td>
<td>sausage</td>
<td>bad</td>
<td>pork</td>
<td>poutine</td>
</tr>
<tr>
<td>hawaiian</td>
<td>french</td>
<td>eat</td>
<td>come</td>
<td>gluten</td>
</tr>
<tr>
<td><strong>Topic 16</strong></td>
<td><strong>Topic 17</strong></td>
<td><strong>Topic 18</strong></td>
<td><strong>Topic 19</strong></td>
<td><strong>Topic 20</strong></td>
</tr>
<tr>
<td><strong>Experience</strong></td>
<td><strong>Location</strong></td>
<td><strong>Fast food</strong></td>
<td><strong>Time</strong></td>
<td><strong>Ambient</strong></td>
</tr>
<tr>
<td>restaurant</td>
<td>location</td>
<td>burger</td>
<td>place</td>
<td>pizza</td>
</tr>
<tr>
<td>menu</td>
<td>find</td>
<td>fry</td>
<td>come</td>
<td>bar</td>
</tr>
<tr>
<td>visit</td>
<td>parking</td>
<td>cheese</td>
<td>time</td>
<td>beer</td>
</tr>
<tr>
<td>experience</td>
<td>area</td>
<td>order</td>
<td>great</td>
<td>drink</td>
</tr>
<tr>
<td>dining</td>
<td>street</td>
<td>onion</td>
<td>wait</td>
<td>place</td>
</tr>
<tr>
<td>item</td>
<td>walk</td>
<td>sandwich</td>
<td>service</td>
<td>great</td>
</tr>
<tr>
<td>card</td>
<td>look</td>
<td>bun</td>
<td>hour</td>
<td>night</td>
</tr>
<tr>
<td>understand</td>
<td>open</td>
<td>mac</td>
<td>lunch</td>
<td>like</td>
</tr>
<tr>
<td>offer</td>
<td>shop</td>
<td>dog</td>
<td>nice</td>
<td>game</td>
</tr>
<tr>
<td>review</td>
<td>lot</td>
<td>hot</td>
<td>get</td>
<td>music</td>
</tr>
</tbody>
</table>

Table 6.5: Results 20 topics omitting most frequent terms
In fact, this strategy had a tendency to be very specific on its results. A similar approach to omitting most frequent terms, the tool LDAvis is able to control the relevance of a term inside a topic. This task is done by modifying a parameter $\lambda$ in Equation 6.1. In this case, by selecting an adequate value of $\lambda$ the outcomes should be quite analogous.

$$p(\text{term } w|\text{topic } t) = \lambda \ p(w|t) + \frac{p(w|t)}{p(w)}$$ (6.1)

6.4. Conclusions

In this machine learning algorithm, we continue having the same issues or restrictions that appeared during the EM algorithm phase. This means that frequent terms overshadow the overall performance of the algorithm. Moreover, we have to mention that we have found little relevant relationship between the topics and the level of quality of the restaurants. However, we may highlight that this topic modelling technique works perfectly because the vast majority of the obtained topics can be described with few words and reflect ideas that are related to real aspects of the restaurant industry.
7. Conclusions and future work

In this last chapter, we will comment in general terms our perception about the development and progress reached during the time devoted to this project. Furthermore, more improvements could be done in order to complement the information gathered in this report and that may be used as a starting point for a future MSc degree.

7.1. Conclusions

From the point of view of researching, this project has been quite rewarding because we have discovered, learned and acquired new skills as a consequence of several fruitless actions. In particular, the Expectation Maximisation algorithm, Topic Modelling methods and Natural Processing Language have been some of the aspects we had to investigate. Apart from that knowledge acquired, processing of huge volumes of data has involved the use of efficient programming techniques to solve problems in short time windows.

On the other hand, from the point of view of the project, there exist some discrepancies between the goals stated at the beginning of the project and the goals reached at the end of the project. As we have explained during the project, we have not been able to create a probabilistic machine learning model to predict the quality of restaurants using some predictive variables of potential users. Nevertheless, the use of Topic Modelling could be labelled as satisfactory. These results show how users has a tendency to describe specific aspects or ideas of a restaurant such as the type of food. Finally, we can conclude that these ideas or groups are more likely to represent categories of restaurants than a level of quality or negative/positive issues. Definitely, our project has reached the objectives related to analysis and summary of data but not the objectives regarding classification.
7.2. Future work

Obviously, time constraints have provoked that not all the features are implemented in the project. In addition, we are going to mention ideas or tasks that could be helpful to increase the quality of the obtained outcomes such as further tests or analysis.

One of the main tasks performed in this data mining is the exploratory data analysis. During this phase, we have extracted a lot of useful information to understand the problem we have coped with. However, this analysis can be extended in order to go into further details, in particular aspects or properties that have been omitted because of its medium-low level of relevance for the project. More specifically, the dataset containing information about businesses include lots of properties such as Price range or the availability of Table service in the restaurant. As a consequence, this information could not be essential for our machine learning models but valuable for the stages of Business understanding and Data understanding belonging to the methodology CRISP-DM. In general, this task will be intended to discover information that we may have overlooked.

As a part of the project, it was necessary to perform a data cleaning task. More specifically, NLP is the most significant part of our data cleaning, on the grounds that our project is totally focus on the textual user reviews. Therefore, an approach could consist on improving, adding or discover more phases to our NLP pipeline. One of these actions could be the use of the Named Entity Recognition so that the final idea of these changes is to remove noisy information, achieving at the end better and clearer results.

One of the most interesting features to discuss in the project is EM algorithm due to its powerful way to infer unknown features. In the view of trials performed, it would be valuable a further investigation of the EM algorithm applied to our specific problematic. In general, we have discovered that the algorithm works perfectly according to the proof of concept but wrongly in real data. As a consequence, prospective projects could enhance the results by means of two parts or sections. First, one action would be the discovery and application of new techniques or methods that could improve the accuracy of the EM classification model. One of these techniques could be the assessments of results using synthetic or custom datasets of distinct sizes. The second action to be done would be a thorough analysis of old and new results. The main objective of this action is to verify what kind of results we have achieved. In early steps, the intention was to predict the quality of a restaurant, but the information could reflect any other kind of patterns such as user profiles or tendencies when writing reviews.

Regarding the section of topic modelling, we should study carefully the documentation provided by the tool called LDAvis because the fundamentals of this tool can
be used for other purposes. In addition, we could perform other kind of modifications such as the selection of the best number of topics or cleaning the data using other techniques in order to avoid noisy results.

Finally, it could be really interesting to explore another machine learning algorithms such as neural networks or support vector machines, and compare the corresponding outcomes to decide the best algorithms for this task.


The content of the CD that accompanies the memory we can find the following resources:

- Report of the project in PDF format inside the directory Report.
- Some books and articles which have been used during this project and are considered as bibliography. This content can be found in the directory Bibliography.