EXPLORING THE EFFECTS OF CONSUMERS' TRUST: A PREDICTIVE MODEL FOR SATISFYING BUYERS' EXPECTATIONS BASED ON SELLERS' BEHAVIOUR IN THE MARKETPLACE

Sinan AlSheikh

Thesis Submitted in Partial Fulfilment of the Requirements of

the Degree of Doctor of Philosophy

School of Computing, Science and Engineering

University of Salford, Salford, UK

October 2019

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

DECLARATION

I hereby declare that this submission is my own work and that it contains no material that was previously published or written by another person, or that has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.

ACKNOWLEDGEMENT

First and foremost, I would like to express my deepest appreciation to my supervisors **Prof. Farid Meziane** and **Prof. Khaled Shaalan** for their invaluable contributions, encouragement, and guidance through the period of my study. Without their support, this thesis would not have been possible.

I would also like to thank IBM as a company who gave me access to many beautiful brains who I proudly call colleagues. It's very addictive to work with smart people; it pushes you to do more and more all the time. I feel proud just to be working with them. I will not be able to thank everyone personally, but I have to mention a few people who were key for my journey towards completing this thesis. **Naveed Ahmad**, big thanks for that moment when I told you about my decision to stop pursuing my degree and focus on my career in IBM. In one second, you flipped from a smiling Naveed into someone else. I still remember the look in your eyes and firm words when you said "Sinan, I felt proud when you started your degree, I will not let you stop yours. You are doing something that I wish I did myself. I stopped my PhD when I was your age, and I still regret it". Special thanks to **Mario Goen, Mohammed Shehadeh, and Andie Brodowski** for ever-lasting moral support when I needed it the most.

I deeply thank my parents **Sameer AlSheikh** and **Essaf Hanna** for their endless prayers and follow-ups on progress. No one could deny their contribution in all my successes in my entire life, but their keen interest in the progress of my PhD research felt a little bit different. It felt as if they wanted me to get the degree not for myself, but more for them. I do not know if they wanted to feel proud of their son, or they just wanted to brag about it among relatives and friends. In both cases, it helped me move forward to this point.

Last but not least, no words can express how much I want to thank My wife **Juman Jijakli**. Without you being in my life, I would not have even applied to a PhD program. When Juman and I got married, she made me write in our marriage certificate that I would get my PhD degree and dedicate it to her as her marriage dowry. Today, I fulfill my promise and dedicate this degree to her. I feel that it's my turn to help her get her PhD degree and raise the bar high for our kids.

LIST OF PUBLICATIONS

1. AlSheikh, S. S., Shaalan, K., & Meziane, F. (2019). Exploring the effects of consumers' trust: A predictive model for satisfying buyers' expectations based on sellers' behavior in the marketplace. IEEE Access, 7, 73357–73372. doi:10.1109/ACCESS.2019.2917999

2. AlSheikh, S. S., Shaalan, K., & Meziane, F. (2017). Consumers' trust and popularity of negative posts in social media: A case study on the integration between B2C and C2C business models. *2017 International Conference on Behavioral, Economic, Socio-Cultural Computing (BESC)*, 1–6. doi:10.1109/BESC.2017.8256364

ABBREVIATIONS

UGC	User-Generated Content, which includes any form of content created by users in digital ecosystems that is publicly shared and available for the consultation and access of other users
K-means	k-means clustering is a method of vector quantization that is popular for cluster analysis in data mining
BTRM-WSN	Bio-inspired Trust and Reputation Model for Wireless Sensor Network
JSON	JavaScript Object Notation is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types
POS Tagging	POS tagging, tags words in the document sentences into structural elements like verbs, nouns, adjectives, and adverbs
IBM Watson	IBM Watson is a group of multipurpose online tools provided by IBM over BlueMix (eCloud). As part of this thesis, we used one tool called Tone Analyser.
C2C	Consumer-to-Consumer
B2C	Business-to-Consumer
P2P	Peer-to-Peer
Lyft	Company Name specialized in C2C transportation services
Uber	Company Name specialized in C2C transportation services
Hailo	Company Name specialized in C2C transportation services
Hackney Carriage	Company Name specialized in B2C transportation services
GrabTaxi	Company Name specialized in B2C transportation services
VRBO	Company Name specialized in C2C Hospitality services
Expedia	Company Name specialized in B2C Hospitality services
HomeAway	Company Name specialized in C2C Hospitality services
Trivago	Company Name specialized in B2C Hospitality services
CouchSurfing	Company Name specialized in C2C Hospitality services

Airbnb	Company Name specialized in C2C Hospitality services
Booking	Company Name specialized in C2C Hospitality services
Kayak	Company Name specialized in C2C Hospitality services
Tripping	Company Name specialized in C2C Hospitality services
Aldi	Company Name specialized in B2C Online retail services
Radio Shack	Company Name specialized in B2C Online retail services
Carrefour	Company Name specialized in B2C Online retail services
BestBuy	Company Name specialized in B2C Online retail services
eBay	Company Name specialized in C2C Online retail services
Amazon	Company Name specialized in C2C/B2C Online retail services

ABSTRACT

In recent years, Consumer-to-Consumer (C2C) marketplaces have become very popular among Internet users. However, compared to traditional Business-to-Consumer (B2C) stores, most modern C2C marketplaces are reported to be associated with stronger negative sentiments among consumers. On the other hand, these negative sentiments are a result of sellers' inability to meet buyers' expectations. These negative emotions are also linked to the low trust relationship among sellers and buyers in C2C marketplaces. The growth of these negative emotions might jeopardize buyers' decisions to opt for C2C marketplaces in their future purchase intentions.

In the present study, the concept of trust is explained in a situation characterised by the following aspects: One party (the trustor) is willing to rely on the actions of another party (the trustee) in a situation in the future to meet his/her expectations. Based on the buyer's and seller's behaviour in the C2C marketplace, we were able to quantify the trust emotion found in text. We also performed text mining on Airbnb, a rich source of data in C2C interactions, to quantify the trust level in host descriptions of offered facilities. Specifically, the research questions addressed the possibility to infer trust from C2C interactions on Airbnb, as well as whether it is possible to infer trust from emotions such as joy and fear. The data are acquired from Ashville, and Boston in the USA, Vancouver in Canada and Manchester in the UK. In line with our expectations, the results of the analysis demonstrate that negative guest feedback in Airbnb reviews is stronger when the description of the host's property expresses the emotion of joy only. Conversely, negative guest sentiments in reviews are the weakest when the host sentiment expressed in Airbnb listings is mixed and expresses different balanced emotions (e.g., joy and fear).

Keywords— Trust; Social Media; Sentiment Analysis; B2C; C2C; tone analyser

CHAPTER ONE: INTRODUCTION

1.1 Introduction & Motivation

Today's environment is characterized by several trends, including globalisation, which entails moving from local economies to a global economy; diversity of markets and marketplaces; flexibility entailing fewer rules and more need for a flexible workforce; and, arising from the advances in new information technology, the emergence of networks that enables more strategic alliances and direct communication among parties (Adams et al. 2018).

Accordingly, recent years have also witnessed an unparalleled growth of the spectrum of services offered at Customer-to-Customer (C2C) marketplaces (Head & Hassanein, 2002; Sahney, 2008). In modern C2C marketplaces, such as Uber and Airbnb, almost any individual can offer a product or a service, such as sharing a ride or renting out a coach in a living room. The broad range of currently available C2C services has also led to an increase of the complexity surrounding finalising a deal online (Head & Hassanein, 2002). Trust between buyers and sellers is a prerequisite of a successful completion of a deal online. Therefore, in essence, modern C2C marketplaces are becoming an industry of trust (Sahney, 2008; Wu & Lin, 2016).

The concept of trust, conventionally defined as the expectation of trustors towards trustees to meet certain expectation (e.g., quality of a product/service or payment on time) has been extensively addressed in previous research (Pennanenet al., 2006; Mui, 2002; Head & Hassanein, 2002). Varying in their aspects, most definitions of trust involve the following three main parts: trustor, trustee, and expectations. The probability of the trustee meeting the expectations of the trustor is referred to as the level of trust. This study targets the third part of trust definition—namely, setting the "expectation" right. Our overarching goal is to build a framework that will help the trustee to use the right tone while describing the facility in order to set the right expectation of the trustor in a C2C marketplace.

In the context of trust in C2C interactions, it is necessary to refer to an extensive body of previous research on commercial reputation or rating systems in online communities (e.g., Mui, 2002; Meziane, & Kasiran, 2008; Dang, 2018; AlSheikh et al., 2017). These studies convincingly demonstrated that, compared to user interactions using the traditional Business-to-Consumer (B2C) marketplaces, C2C marketplaces are characterized by much stronger negative sentiments on social media (AlSheikh et al., 2017). In this body of research, trust was quantified based on who members of a social network choose to partner with or avoid. However, despite the growing number of relevant studies, this field still lacks a quantitative model to estimate trust levels among buyers and sellers on the transaction level, which warrants further research to better meet user expectations and to better control C2C marketplaces.

For instance, in the hospitality services industry, there is a tendency for hosts to fall into the trap of over-promoting their facilities, which leads to higher expectations on part of their guests. Only the host knows whether and, if so, to what extent the description of a property differs from the reality. Many hosts work hard to meet the high expectations of their guests, but not all of them succeed, which leads to disappointment on both sides. Therefore, anticipating this type of transactions ahead of time can prevent hosts and guests from having disappointing transactions and increase the number of trusted transactions.

On previous research in emotion psychology, trust was conceptualised as one of basic human emotions. Specifically, in Plutchik’s (2001) taxonomy of eight basic emotions, which serve as the foundation for all other emotions, trust is deemed to be one of the eight basic emotions.

Although other proposals of basic emotions, such as the one formulated by Ekman (1972, 1992), do not include trust into the list of basic emotions, both Plutchik's (2001) and Ekman (1972, 1992) converge in thinking that non-basic emotions are combinations of the basic emotions, which may be called 'blended' or 'mixed' emotion. Regardless whether trust is considered as a basic or non-basic emotion, in the present study, we will elaborate on the idea that trust may be a combination of different emotions. Based on this assumption, we will propose a model to quantify trust.

1.2 Definition of Trust

Trust is a multi-dimensional concept that has attracted a considerable scholarly interest from a wide variety of perspectives (Kramer & Tyler, 1996; Bigley & Pearce, 1998; Rousseau et al., 1998; Kramer, 1999). Specifically, trust has been regarded as a major construct in research predicting various individual-level and organizational-level outcomes (Davis et al., 2000; Simons et al., 2002; Roy et al., 2006; Colquitt et al., 2007). A distinct trend in the body of previous research on trust has focused on the concept of trust within e-commerce. In e-commerce interactions, some of these concepts overlap at various points in time, which contributes to the success or failure of online transactions. Each concept has a different impact on the decisions of either buyers or sellers. According to Mayer et al. (1995), the concept of trust can be better explained in a situation characterised by the following aspects:

> One party (the trustor) is willing to rely on the actions of another party (the trustee) in some situation in the future. Additionally, the trustor (voluntarily or otherwise) abandons control over the actions performed by the trustee. Therefore, the trustor is uncertain of the outcome of the trustee's actions.

> This uncertainty involves the risk of failure or harm to the trustor if the trustee does not behave as expected.

While there is no consensual definition of trust in the literature, the many and varied definitions of trust rely on the following three aspects pertinent to trust: trustor, trustee, and expectations (Hurley, 2006). The trustor abandons control and builds expectations based on the results from the trustee. In the digital domain, trust has been defined as follows:

> Trust is the confidence placed in an organisation (trustee) to collect, store, and use the digital information of others (trustors) in a manner that benefits and protects (expectations) those to whom the information pertains. (Accenture, 2015; PwC, 2014)

1.3 Research Aim and Objectives

This thesis aims to define an approach to quantify trust from joy and fear that are detected in text published by sellers in C2C marketplace. This will help to prevent deals that might result in customer dissatisfaction. Specifically, the study will focus on the following:

Trust among individuals engaging in monetary transactions online. Therefore, the thesis will not cover nonmonetary transactions or transactions performed by organizations.

The B2C and C2C business models. Buyers and sellers in these e-commerce models have psychological, social, and cultural characteristics that can influence their decisions in establishing or finalizing a deal.

To achieve the overarching aim formulated above, the following objectives will be addressed:

- **Objective 1:** To provide a concise review of the trust in the digital domain, as well as how C2C marketplaces are governing trust in today's markets;
- **Objective 2:** To investigate and examine the concept and levels of trust in the physical and digital domains;
- **Objective 3:** To compare consumers' trust and popularity of negative posts about C2C marketplaces in social media as compared to sentiments expressed about traditional B2C marketplaces;
- **Objective 4:** To develop and present a conceptual framework to quantify the trust emotion found in text
- **Objective 5:** To test and evaluate the proposed conceptual framework in a case study based on the analysis of the data gathered from multiple cities on Airbnb;

1.4 Research Questions

The output of C2C online interactions comprises large amounts of textual data, such as reviews on social platforms. Accordingly, in order to detect (basic) emotions, such as joy, anger, fear, disgust, and sadness, in various types of texts, sentiment analysis has been widely used. In essence, sentiment analysis focuses on word choice and frequency of occurrence of a given phrase near a set of positive or negative words (Sindhwani & Melville, 2008). In the present study, we rely on Plutchik's (2001) Wheel of Emotions where trust is deemed to be one of the eight basic emotions, positioned between joy and fear. Accordingly, the two key research questions addressed in the present study are as follows:

Research Question 1 (RQ1): Can trust, one of the eight basic emotions, exist in C2C texts, such as Airbnb accommodation descriptions?

Research Question 2 (RQ2)**:** If it exists, can trust be inferred from detecting joy and fear?

1.5Structure of the thesis

The remainder of this study is structured as follows. Chapter Two provides the theoretical background of the present study, including working definitions of major concepts, such as trust in the digital domain. This chapter also discusses trust as an emotion and how it impacts the behaviour of buyers and sellers in e-commerce deals. We conclude Chapter Two with a review of currently available models of quantifying trust (opinion mining).

Chapter Three summarizes the results of a sentiment analysis study (AlSheikh et al., 2017) conducted on three industries (taxi, hospitality, and retail) where we aimed to identify which business model attracts the most negative sentiments in user-generated content (UGC) published online (AlSheikh et al., 2017). Based on multiple companies from each industry, we analyse the content published on social media about each company and then classify this content based on the sentiment of each post.

Chapter Four explains trust relationship in physical world and compares that with the online trust relationship. In Chapter Five, based on the review of the literature and the findings reviewed in previous chapters, we introduce the proposed conceptual framework to measure trust as a basic emotion in text. This chapter also describes the data collection process, presents the data model, and describes the text-mining techniques used in the framework to measure trust level in text. As an illustration of the model, the chapter discusses two examples of two different listings published on Airbnb. One of them is a trusted listing that attracted multiple positive reviews, while the other is a not trusted listing that received multiple negative reviews.

In Chapter Six, we present the results of four case studies for selected cities (Ashville and Boston in the US, Manchester in the UK, and Vancouver in Canada). Finally, in Chapter Seven, conclusions are drawn, and directions of further research are outlined.

CHAPTER TWO: BACKGROUND

In this chapter, we introduce the definitions of trust as an emotion (Section 2.1). Furthermore, we outline and discuss several models of trust (Sections 2.3) and highlight the weakness/shortcomings in each model in relation to calculating trust in e-commerce. Those models are considered the most relevant and innovative computational models to calculate trust.

2.1Trust as an Emotion

According to the appraisal theory of emotions, emotions are elicited by certain acts or events, also called emotion antecedents (Lazarus, 1991; see also Arnold, 2013; Ekman, 1972; Lazarus & Folkman, 1986; Scherer, 2005; Scherer et al., 2001). Richard Lazarus, a pioneer in cognitive emotion, states that appraising a situation occurs prior to experiencing an emotion (Lazarus, 1991). According to the appraisal theory of emotions, the series of activities are first elicited by a stimulus, followed by the thought which then ends in the immediate experience of a physiological reaction and the emotion. For example, reading a story can elicit the reader's emotion based on the writer's phrases and selection of words. The frequency of occurrence for a set of positive or negative words stimulates the reader's brain, which then turns into a thought followed by an immediate experience of an emotion. Another example of how an emotion could be elicited is a threatening sight of a tiger.

Plutchik (1986, 2001) argues that, in human experience, there are eight basic emotions that form the foundation for all other human emotions. According to Plutchik (1986, 2001), along with joy, fear, surprise, sadness, disgust, anger, and anticipation, trust is such basic emotions (see Figure 2.1). In Plutchik's (2001) classification, each basic emotion has a stronger and a weaker form. In the case of trust, its weaker form is acceptance, while its stronger form is admiration (see Table 2.1).

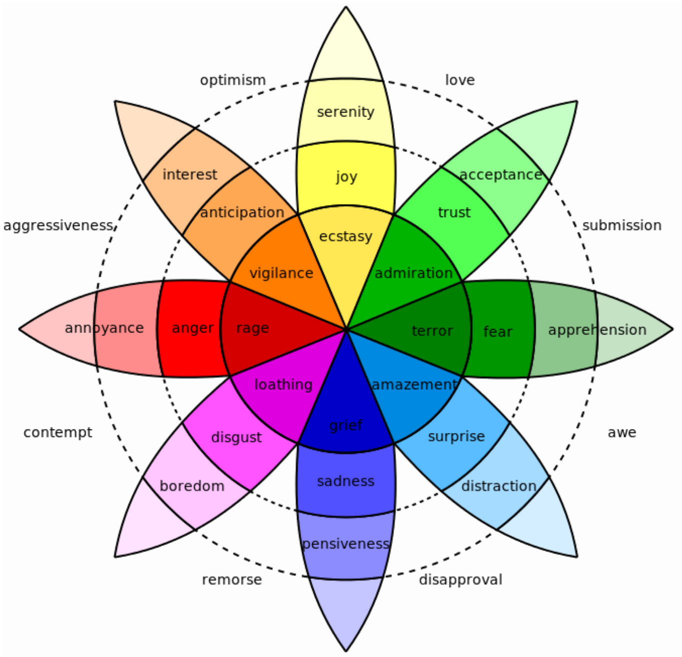


Figure 0.1: Plutchik's (2001) Wheel of Emotions. Layers show forms of emotions as basic, weaker, and stronger. The wheel of emotions should form a diamond shape in 3D.

On the other hand, other basic emotion theorists, such as Ekman (1972, 1992) did not consider trust to be a basic emotion. However, both Ekman (1972, 1992) and Plutchik (1986, 2001) agreed that non-basic emotions are combinations of the basic emotions, which may be called 'blended' or 'mixed' emotions.

Table 0-1: List of 8 Basic Emotions (Plutchik 2001)

Weaker	**Normal (Basic)**	**Stronger**
Serenity	Joy	Ecstasy
Acceptance	Trust	Admiration
Apprehension	Fear	Terror
Distraction	Surprise	Amazement
Pensiveness	Sadness	Grief
Boredom	Disgust	Loathing

Annoyance	Anger	Rage
Interest	Anticipation	Vigilance

In reading online reviews, just like in reading a story, the writer's selection of words and phrases triggers readers' brain to build a thought, which then leads to the emergence of an emotional experience. Overall, there is a tendency for Airbnb hosts for example to fall into the trap of over-promoting their facilities, which leads to higher expectations from their guests. The higher the guest's expectation, the higher the trust level is built. However, only the host knows whether and, if so, to what extent the description of a property differs from the reality. Many hosts work hard to meet the high expectations of their guests, but not all of them succeed, which leads to disappointment on both sides. Anticipating this type of transactions ahead of time can help the hosts to write realistic descriptions of their property and thus prevent hosts and guests from disappointing transactions and increase the number of trusted transactions.

2.2 Buyers and Sellers attitude in e-Commerce Deals

Buyers and sellers are essential to any deal in both offline and e-commerce. Both parties have their own wants and needs that must be satisfied to finalise a deal. The process of finalising a deal is also known as the process of trade-offs between buyers and sellers to reach a state that satisfies both sides (Burnett, 2012; Beck et al., 2016).

When a buyer or a seller is represented by an organisation, behaviours and trade-offs may be structured and documented by the organisation. For instance, an organisation may have a rule to engage in potential deals only if the profit margin exceeds or is equal to 10%. In contrast, if the buyer or seller is an individual or simple group of individuals, corresponding wants and needs may vary, and trade-offs may not be defined in a structured form. This variance adds ambiguity to the deal (Arnold, 2013; Niranjanamurthy et al., 2013; Burnett, 2012). In the present study, we focus on the deals between individuals.

According to several previous studies, the following aspects highlight the main characteristics that influence individual consumers' behaviour in approaching deals (Burnett, 2012; Yoldas, 2011):

- ***Personal/demographic characteristics***, e.g., gender, age, weight, occupation, income status, education, and lifestyle. For instance, a buyer might make or break a deal if seller is from the opposite gender, income status, education, or lifestyle.

- ***Psychological characteristics***, i.e. consumers' psychological state(s) at the time of finalising the deal. Here, an individual emotion (e.g., joy, anger, trust, or fear) can be a deal maker or breaker.

- ***Social characteristics***, i.e. aspects that include, but are not limited to, previous feedback to a similar transaction. Specifically, other buyers' reviews and comments can exert pressure on the consumer or bias his/her decision as to whether or not finalise a deal (Lee et al., 2018).

- ***Cultural characteristics***, i.e. collective mental programming of the mind for an individual or group. Cultural characteristics distinguish members of one group of people from another. For example, individual's nationality, religion, as well as favourite political party or football team can be a deal maker or breaker.

In the present thesis, trust between buyers and sellers is considered to be one of the *psychological* characteristics that influence the decision-making processes.

2.3 Opinion Mining

As mentioned in the previous section, trust influences the decision-making processes between buyers and sellers, however the uncertainty about quality of the offerings provided by users in

C2C markets works against that. It is important to calculate the trust level of its users before initiating any transaction. In general, users in C2C marketplaces have to know how much trust to give to other users with whom they might have had no earlier transaction. These algorithms are also known as reputation models (see Braga et al., 2018, for a recent review).

An issue that has recently emerged in the hospitality services industry is that user reviews tend to be positive, which has led to the emergence of the problem called "all good reputation problem" (Fradkin et al., 2015, 2018; Cao et al., 2011; Lee et al., 2018; Resnick et al., 2000). The predominantly positive evaluation in those reviews is explained by the fact that, when tempted to write a negative review, guests in C2C marketplaces fear that hosts might provide a similarly negative feedback on them, which might damage their own reputation and risk their future deals with other hosts. Accordingly, the users who wrote a positive feedback about a particular host on one platform might go to another social media platform and share there a very negative experience, posting a more truthful and objective description of their experience. However, this time, it will be a generic negative post about the C2C marketplace in general, rather than a negative evaluation of the specific host. Accordingly, the number of negative posts about C2C marketplaces in general has been steadily growing in the last several years (AlSheikh et al., 2017). The problem that this creates is that, for an outsider, it becomes more difficult to identify a specific host who is responsible for the negative evaluation. Therefore, the host's setting a high expectation by over-promoting his/her facilities can results in not only disappointment or frustration among the many guests, but also a growth in the number of negative posts randomly published about the C2C marketplace in general.

In response to the issue outlined above, many studies have sought to elaborate algorithms to quantify trust in textual content. The following table summarizes the approaches and models found in literature:

Category	Author	Summary
Trust and reputation algorithms	Zhang et al., 2014; Rangari & Waghmare, 2015; Brody & Elhadad, 2010	aspect-based opinion mining, which aims to automatically discover whether a guest free text review expresses a positive or a negative opinion towards the host. This approach is discussed in further details in Section 2.4.1.
Trust and reputation algorithms	O'Donovan et al., 2007; see Section 2.4.2 for further discussion	AuctionRules algorithm, which suggests a classification set of rules tailored to capture the signs of negativity in text reviews provided by C2C users
Trust and reputation models	Zhou and Hwang (2007)	PowerTrust and reputation model, which was established is based on the analysis of UGC (user-generated content) provided by 10,000 eBay users. UGC includes any form of content created by users in digital ecosystems that is publicly shared and available for the consultation and access of other users (Saura & Bennett, 2019). According to this model, users with a very big number of feedback comments are extremely rare (power users). Those users can be used as the basis to calculate reputation for others who belong to the same network (see Section 2.4.3 for further details)
Trust and reputation models	Egger, 2003	EigenTrust and reputation model. The aim of this model is to reduce malicious and fraud contributors in the network. It can be employed to reduce fraud and malicious reviews and feedbacks given automatically to a specific product in order to increase its reputation in the system (see Section 2.4.4 for an outlined of this model).
Trust and reputation models	Wang & Singh, 2006	both the parallel network of acquaintance and the real network of acquaintance enable calculating the trust between two individuals (a) and (b) using the reputation between the chains of individuals with the hypothetical link (a)-(b). In Section 2.2.5, we discuss an ideal scenario, while a more realistic scenario and its limitation in our

		day-to-day marketplaces (Wang & Singh, 2006) is presented in Section 2.2.6.
Trust and reputation models	Mui, 2002; Mármol, & Pérez, 2011	Chernoff bound-based trust model which is based on the number of encounters between the buyer and the seller during a transaction. This model assumes that the guest and the host will interact between each other before finalizing a deal (e.g., in a chat). Section 2.2.7 discusses this model in further detail, showing the limitations of this model in C2C marketplaces.
Trust and reputation models	Mármol and Pérez 2011	bio-inspired trust and reputation model for wireless sensor networks. This model was inspired by how ants find their way searching for food, and how they navigate back to their colony. In Section 2.2.8, we discuss how we can learn from the ant's trust algorithm, and how similar it can be to human purchase behaviour. Section 2.2.8 also lists the limitations of this algorithm with regard to calculating trust.

2.3.1 Aspect-Based Opinion Mining

Aspect-based opinion mining, also known as sentiment analysis (Rangari & Waghmare, 2015), aims to automatically discover whether a given piece of text expresses positive or negative opinion towards a subject (Chuang et al., 2012). Sentiment analysis can be looked at as a general text categorization problem. It combines the techniques of natural language processing, data retrieval, text analytics, and computational linguistics (Piorkowski & Zhou, 2011; Latif & Jaffry, 2013). Opinion mining is basically a supervised method in which one needs to train a classifier on the training set before it is performed on a test set. It can analyse people's feedback, reviews, and appraisals to find out emotions towards specific subjects, such as products, offerings, sellers, or buyers. Sentiment analysis has been extensively used in research on user-generated content in the hospitality and tourism sectors (e.g., Berezina et al., 2017; Bjørkelund et al., Han et al., 2016; Hu et al., 2017; Ye et al., 2009).

Aspect-based opinion mining is also known as phrase-level opinion mining. While aspect-based opinion mining works on three levels (document-level, sentence-level, and phrase-level), document-level and sentence-level usually return a generalized opinion about a subject. However, phrase-level opinion mining can return a more granular evaluation of the opinions towards a specific aspect in the product or service. This algorithm is mainly used to discover sentiments on aspects of items. Aspects that are explicitly mentioned using nouns or noun phrases in a sentence are called explicit aspects. For example, the cleanness aspect in a review sentence such as "The house was very clean" is considered to be an explicit aspect. On the other hand, there are also implicit aspects that are not explicitly mentioned in a sentence, but are implied. For example, "The room rate was overpriced" implies the price aspect of the room.

Applying this algorithm to reviews in the C2C hospitality industry makes it possible to identify explicit and implicit aspects that would make or break a future deal. The negative aspect can then be highlighted to the host as a feedback to improve.

Despite its overall effectiveness, a limitation of this approach is that it does not work effectively unless there have been multiple reviews on the facility. Fraud review comments can mislead this algorithm, hiding the negative aspects. Moreover, whenever a host does not have any review recorded in the system, guests should take leap of faith to try their luck. Therefore, in order for this algorithm to work effectively, some guests have to go through the experience of the facility's not meeting their expectations built based on the host facility description.

2.3.2 The AuctionRules Algorithm

The AuctionRules algorithm was initially proposed by O'Donovan et al. (2007) to deal with the problem of unnaturally high trust ratings on C2C marketplaces. The algorithm suggests a classification set of rules tailored to capture the signs of negativity in the text review comments provided by C2C users. In that feedback, despite a positive score, the commenter may still voice some complaint inside the free text feedback field.

The aim of the AuctionRules algorithm is to correctly classify users' comments into positive or negative according to a predefined threshold. AuctionRules is built on the fact that the online markets are restricted in nature, and the actions are limited to the workflow defined by the marketplace. Having said that, there are few silent factors that the buyer or seller cares about which are reflected in their comments. The output of the algorithm is a summarized sentence from the marketplace with a set of core features in order to set the expectation correctly for any future deal (trust level)

For example, in a C2C marketplace such as eBay, the following seven core features are taken in consideration in order to calculate the trust in the user feedback text. (The terms in brackets are contents of each feature set).

- Item—The quality/condition of the product being bought or sold (item, product);
- Person—The person who the user makes the transaction with (buyer, seller, dealer);
- Cost—Cost of the item, shipping, hidden costs, and other similar keywords (expense, cost);
- Shipping—Delivery of the item, security, time, and other similar keywords (delivery, shipping);
- Response—Communication with the other party (response, comment, email, communication);
- Packaging—The packaging quality/condition of the item (packaging);
- Payment—Method of payment to the seller, or back to buyer for return (payment);

- Transaction—The overall transaction quality (service, transaction, business).

For example, after analysing all the comments provided on an individual user on eBay, the algorithm will produce the following sentence: "User X is trusted when it comes to payment, but shipping has been unsatisfactory in the past".

However, similarly to aspect-based opinion mining, the limitation of the AutionRules algorithm is that it requires multiple reviews in the system in order to calculate the trust level. Yet, unlike aspect-based opinion mining, AuctionRules pre-defines a set of aspects that can fit a specific industry or marketplace. The algorithm searches for user text reviews with the focus on only the aforementioned 7 core features (aspects) and discards all other aspects.

2.3.3 PowerTrust and Reputation Model

PowerTrust is another P2P trust and reputation model based on distributed peer feedback This model was initially proposed by Zhou and Hwang (2007) who studied the feedback provided by 10,000 eBay users. Users with few feedback comments were quite common; however, users with a very high number of feedback comments were extremely rare (power users).

The model starts with the analysis of the feedback comments of power users. After aggregating all the feedback of power users, the model calculates the global reputation score $v_i \in [0,1]$ of every peer i. To this end, it first collects all reputation scores for v_j and the normalized local trust score $r_{ij} \in [0,1]$; where j are peers who have interacted with i in the past. r_{ij} is defined by equation 1:

$$r_{ij} = \frac{s_{ij}}{\sum_j s_{ij}} \tag{1}$$

where S_{ij} represents the satisfaction level between peers i & j based on a previous transaction. Said differently, if the feedback from peer i is positive, following a previous transaction with peer j, the global reputation score v_i can be calculated using Eq. (2).

$$v_i = (1 - \alpha). \sum_j (v_j X \, r_{ji}) \tag{2}$$

where α is the greedy factor calculated based on the status of the power user.

In a PowerTrust network, each peer has a global reputation score v_i calculated based on the degree of satisfaction associated with his/her historical transactions with other peers in the network. This model takes the feedback between peers into consideration. This model has reported to be effective in identifying fraudulent peers in the P2P network (Zhou & Hwang, 2007). It is also highly scalable to network with a large number of peers.

The limitation of this model is that it assumes that all members have some interaction with others before. New joiners will need to build their interactions one transaction at a time. Moreover, this model keeps the highly trusted peers trusted regardless of their future transactions. It will take many bad transactions for a highly trusted peer to lose its score.

2.3.4 EigenTrust and Reputation Model

EigenTrust (Kamvar et al., 2003; Egger, 2003) is another trust and reputation model built to be used in Peer-to-Peer (P2P) networks. This model determines the trust value for each peer based on successful historical transactions. The aim of this model is to prune down malicious and fraud contributors in the network.

Each peer i in an EigenTrust network of peers holds a vector of trust values at every point in time for all peers in the network. The trust value is calculated as shown in Eq. (3).

$$t_i^{(k+1)} = (1-a)\,.\,C^T\,.\,t^k + a\,.\,p^{\rightarrow} \tag{3}$$

where $t_i^{(k+1)}$ is the trust value for peer i in specific time $(k+1)$, $a \in [0,1]$ is a constant to calculate the global trust value, C^T is the transposed matrix of $[C_{ij}]$, and C_{ij} represents the trust from peer i towards peer j based on the historically successful transactions between them. However, if peer i does not know anyone or has not had any previous successful transactions, s/he will choose to trust pre-trusted peers. Furthermore, $\overrightarrow{P_i}$ is the distribution over pre-trusted peers ($p_i^{\rightarrow} = 1/P$ if $i \in P$ and $p_i^{\rightarrow} = 0$). Otherwise, P is the pre-trusted peers.

This model is built on the assumption that each EigenTrust network has several known trusted peers with high trust values. Presumably, this helps other peers in the network to rapidly build their trust values. Eq. (3) is repeated for every peer in the network until all trust values are calculated. After calculating all the trust values, each peer can select who to transact with. A simple way is to select the peer with the highest trust value in the vector of trust; this is called the deterministic selection process. On the other hand, there is the probabilistic process where selection is based on a probability of 10% random peer with a low trust in the network.

While this approach helps to solve real-world problems, its limitation is that it is not computationally efficient. Calculating trust in big EigenTrust networks can grow exponentially. In order to calculate the trust for a single node, the trust for all other nods in the network has to be calculated first. Moreover, if an EigenTrust network has no highly trusted nodes, all other members will not have high trust values. On the other hand, calculating trust in micro-EigenTrust networks can yield insignificant results.

2.3.5 Parallel Network of Acquaintances

Parallel network of acquaintances (Mui, 2002; Mármol, & Pérez, 2011) is another model to calculate trust—specifically, within a network of acquaintances. This approach is based on the assumption that the social network between the trustor and the trustee can indicate the probability of the trustee to meet the expectation of the trustor based on the trustee's reputation in social network.

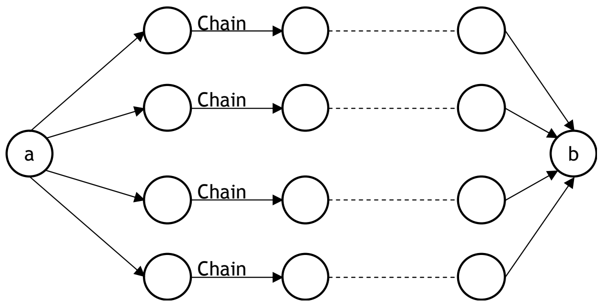


Figure 0.2: The Relationship Chains Between a Trustor (a) and a Trustee (b) in Parallel Network of Acquaintances (Mui, 2002)

Figure 2.2 shows K chains between the trustor (a) and the trustee (b). Each chain consists of at least one link between two agents in the network. The reputation between two people can be considered as a function of a number of cooperative events in the chain divided by the number of encounters. If we assign the reputation to be the weight of the link, then, in theory, we can calculate the reputation between the trustor (a) and the trustee (b). The estimate of the trustee's (b) reputation across the entire parallel network can be calculated as a weighted sum across all the chains.

The limitation of this model is that it is theatrical rather than realistic, and it is usually used to explain the real network of acquaintances (see Section 2.4.6). The limitation of the parallel network of acquaintances is that it assumes that the nodes between (a) and (b) do not intersect; in other words, that the people form one of the chains between (a) and (b) do not know anyone from the next chain. In real life, however, this is not usually the case. Moreover, to compute this model, all nodes between (a) and (b) should be known, and all interactions between all nodes should be captured.

2.3.6 Real Network of Acquaintances

This model is built on top of parallel network of acquaintances model. Real network of acquaintances forms an arbitrary chain that overlaps between the trustor and the trustee. Figure 2.3 shows a generalised representation of a social network of acquaintances in real life.

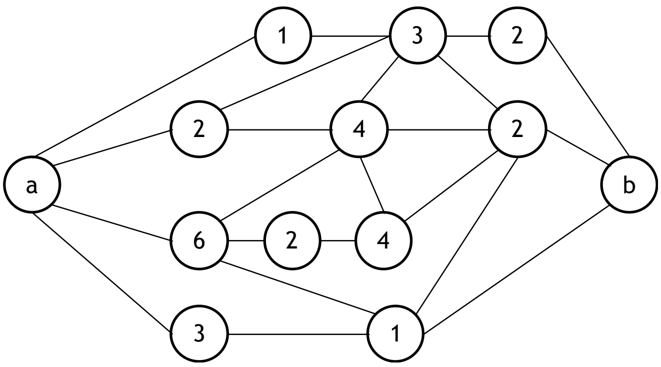


Figure 0.3: The Relationship Chains Between a Trustor (a) and a Trustee (b) in Teal Network of Acquaintances (Mui, 2002)

The entirety of these links can be considered to constitute a Bayesian Network which grows exponentially with an increase of the number of nodes. However, in solving real-world problems, this approach is not computationally efficient. To estimate the reputation of the trustee (b) in a real social network, all possible paths should be taken in consideration. Any new node introduced between (a) and (b) will increase the complexity to calculate the trust level. However, several assumptions and techniques to simplify and reduce the complexity of this problem to an acceptable computational level are available (Mui, 2002; Mármol, & Pérez, 2011).

The social network of acquaintances assumes that every trustor (a) has a chain of links to the trustee (b). However, the limitation of this model of trust is that, while its key assumption might be true, capturing the network and all the events among people is rather challenging. Another limitation of this approach is that it does not account for trusted people who are densely surrounded with people who are not trustworthy:

> "Would Mahatma Ghandi get a lower reputation because of his social network and how they used to interact with him?"

This question raises a concern that, according to this model, Mahatma Ghandi will not be considered as a trusted person. His network of acquaintances was full of people with conflicts; yet, their interactions did not lead to low trusted relationships.

In order to use this trust model in e-commerce to calculate the trust between buyers and sellers, all relationships that connects buyers and sellers should be identified. Moreover, each relationship that connects a buyer with a seller and their network of acquaintances should be identified and ranked. Collecting all these data would make this model challenging to use, considering also that buyers and sellers can be from different co-tenant networks. Even if this data were identified in a way or another, the network would be considered as Bayesian network, where the complexity of calculating the trust level between buyers and sellers grows exponentially with an increase of the number of nodes in the network, which makes this model computationally un-friendly.

2.3.7 Chernoff Bound-Based Trust and Reputation Model

Chernoff bound-based trust model is based on the reputation of the trustee to the trustor (Mui, 2002). The reputation of the trustee is considered as a function of cooperative events towards the trustor divided by the number of encounters. Each cooperative event adds to the overall probability of trustee meeting the expectation of the trustor. Let $X_{ab}(1)$, $X_{ab}(2)$, … $X_{ab}(m)$ be a sequence of m independent encounters, each one being the probability of success. The minimum number of encounters necessary to achieve the desired level of confidence and error is represented by (m).

The result of Eq. (4) will be a random variable representing the portion of success of the trust relationship between trustor (a) and trustee (b).

$$\alpha = \left(x_{ab}(1) + x_{ab}(2) + \cdots + x_{ab}(m)\right)/m \tag{4}$$

However, in most C2C marketplaces, this approach would have a limitation, as it is assumed that the trustor and the trustee have interacted before the transaction. However, in most C2C marketplaces, this is not the case (e.g., the first time you interact with an Uber driver is when you ride a car towards your destination).

Another weakness of this approach is that the impact of the negative events is equal to that of the positive ones. However, in everyday life, this assumption is unrealistic. Moreover, each trustee has to perform negative events in the first place towards the trustor to de-cumulate the portion of success.

2.3.8 Bio-Inspired Trust and Reputation Model for Wireless Sensor Network

The final model that we will briefly evaluate is the bio-inspired trust and reputation model for wireless sensor network (BTRM-WSN) proposed by Mármol and Pérez (2011). This trust and reputation model were inspired by observations of the behaviour of ants (Nguyen, 2017; Mármol & Pérez, 2011).

In a nutshell, while ants are sent to discover a new route, they leave trails of pheromone for other ants to follow. Since not all paths are worth being followed, ants build a trust matrix for all the paths that they go through. When multiple paths cross, the path with the strongest pheromone level gets higher points than those with less pheromone. Moreover, when an ant reaches the desired destination, it will consider this path as the most trusted path and will always use it in future journeys to reach the desired destination. Other ants also produce pheromone in the process of selecting their trusted paths. This makes the trusted path even more trustworthy for other ants. On the other hand, other paths lose their pheromones over time. As a result, all ants can easily decide which path to select, since less optimal paths lose significant parts of their pheromone, while a single path (the one with the strongest pheromone level) has been consistently used by other ants.

Based on these observations of how ants find a trusted path, searching for food, and navigate back to their colony, Mármol and Pérez (2011) developed a trust and reputation model that can be used in the distributed sensor networks. The trusted path is not necessarily the shortest or the fastest, but it is the path that can be trusted to take the sensor to the desired destination.

When the ant model is extrapolated to e-commerce, a similar pattern observed in human buyers/sellers is the so-called bandwagon effect. In essence, buyers/sellers prefer to use a marketplace that many other buyers/sellers have previously used, despite the fact that there might be other marketplaces with better processes or workflows. Similarly, buyers tend to buy from sellers who have recorded more successful deals or who have higher stars ranking in the system.

In order to calculate trust using BTRM-WSN in e-commerce, both buyers and sellers need to have multiple previous transactions. However, in e-commerce, this can be considered as a limitation, since calculating trust using BTRM-WSN will work against new sellers or buyers. In fact, this approach would only help well established sellers who have a long and successful history of transactions. In other words, while already trusted sellers will become more trusted, regardless of their future conduct behaviour, new sellers or buyers will be forced to fake a historical track of transactions just to be looked at as trusted resource.

2.4 Summary

In recent years, the newly emerging C2C marketplaces have been developing into an industry of trust (Wu & Lin, 2016). In view of the uncertainty about quality of the offerings provided by users, it is important for C2C marketplaces to calculate the trust level of its users before initiating any transaction. In the hospitality services industry, hosts tend to build higher expectations from their guests by over-promoting their facilities, which may lead to disappointment and frustration on part of guests. However, despite their disappointment and/or frustration, guests tend to provide a very positive feedback to hosts, for the fear of hosts' writing a similarly negative feedback in response. This circularity brings about the so-called "all good reputation problem" (Resnick et al., 2000; Fradkin et al. 2015, 2018; Lee et al., 2018). However, some disappointed guests would turn to other social media to share their disappointing experiences, targeting not the specific host, but rather the entire C2C marketplace. Another strategy that guests use is giving a 5-star feedback to the host but expressing their disappointment implicitly in the free text feedback form, a careful reader might be able to detect it.

Seeking to solve the “all good feedback” problem and calculate hosts’ real trust level based on the guests’ free text feedback, several studies used sentiment analysis (also known as opinion mining) in order to identify the hidden message in the guests’ feedback (Rangari & Waghmare, 2015); O‘Donovan et al., 2007); Zhou & Hwang 2007). However, while this approach can enrich the existing feedback system, it is based on the assumption that multiple reviews are given to an offering. Therefore, this approach requires that many users go through many disappointing experiences and write about them in the marketplace feedback form. Moreover, hosts can always create new offerings for the same facility and start all over again.

Several other computational trust and reputation models which we have reviewed in this chapter are parallel and real network of acquaintances, Chernoff bound-based trust model **(**Mui, 2002), as well as EigenTrust and reputation model (Egger, 2003). Unlike review-based models that use sentiment analysis, these models aim to calculate trust and reputation before a transaction is finalized and feedback is provided. They are built on different assumptions, some of which might be hard to achieve. For example, for the network of acquaintance reputation mode to work, it might be difficult to identify the full network of people that links hosts and guests. Furthermore, it is very difficult to calculate the reputation between each pair in the network in order to estimate the trust level between host and guest before they finalize a transaction. For instance, Chernoff bound-based trust model requires that all interactions between hosts and guests are captured and analysed before the parties finalise a transaction. Given that most transactions can be finalised in one click, the interaction between hosts and guests can occur outside the marketplace.

Accordingly, in the present thesis, we focus on managing guests’ expectations, rather than on the analysis of their feedback. To this end, we propose a model that can help C2C hospitality marketplaces to automatically identify the trust level in the host description for any offering. This model will help to identify cases when hosts set excessively high expectation among guests, which will likely lead to a disappointing experience for both parties. By managing trust level in facility descriptions, C2C marketplaces can avoid many disappointing experiences by just calculating the trust level in those descriptions. This model also help hosts to edit their offerings in order to appropriately set the right expectations, which can lead to positive guest experiences. This model was trained on Airbnb data acquired from the US city of Ashville, Alabama, and tested on Airbnb data acquired from Manchester in the UK.

CHAPTER THREE: SENTIMENT ANALYSIS STUDY ON SELECTED INDUSTRIES

In this chapter, we analyse consumers' trust towards C2C marketplaces as compared to traditional B2C marketplaces. Specifically, we focus on three e-commerce industries: the taxi industry, the hospitality industry, and the retail industry. To this end, user-generated content was collected using the SocialMention tool. The data were collected from the publicly available UGC from Twitter, Reddit, Photobucket, Topix, as well as other blogs and sites. Discussion of the accuracy and precision of the SocialMention tool is beyond the scope of the present thesis. However, this tool was used for all search queries to provide fair comparisons. Then, sentiment analysis was automatically applied to analyse the data in terms of positive and negative sentiments expressed by users.

3.1 Taxi Industry

Uber, Hailo, Lyft, and GrabTaxi were selected as representative examples of modern taxi companies built on the C2C model. Hackney Carriage and Taxicab were selected to represent traditional taxi companies built on the B2C model. We assumed that any mention (post) that contains one of the selected brand names is referring to the company itself. Several other companies (e.g., Ola and Via) were excluded from the analysis, as their brand names can be used in a context other than the brand itself.

Table 3.1 summarizes the frequency (per minute) of positive and negative sentiments for all companies. There is at least one mention (post) per minute for each company on Twitter, Reddit, Photobucket, Topix, and many other blogs.

Table 0-1: Sentiment Analysis Results (Taxi Companies)

	Model	Positive	Neutral	Negative	Negative to Positive Sentiment Ratio
Taxicab	B2C	138	59	7	5%
GrabTaxi	C2C	84	36	12	14%
Hackney Carriage	B2C	117	50	16	14%
Lyft	C2C	156	67	31	20%
Hailo	C2C	114	49	29	25%
Uber	C2C	148	63	37	25%

Note: Numbers normalised per minute

As can be seen in Table 3.1, the percentage of negative sentiments expressed by users towards modern taxi companies (Uber, Hailo, and Lyft) is greater than that for traditional taxi companies. Uber and Hailo had 25% negative posts (1 negative post per 3 positive posts), while Hackney Carriage had only 14% negative posts (1 negative post per 6 positives posts). This means the sentiments towards Hackney Carriage are twice as positive as those towards Uber and Hailo. Taxicab had only 5% negative posts (1 negative post per 20 positive posts), which makes the sentiment towards Taxicab six times more positive as compared to that towards Uber and Hailo. Of note, neutral posts with neutral sentiments were removed from this calculation.

Figure 3.1 presents the main categories of negative tweets related to the Uber company. The lowest percentage of negative tweets focused on price. The highest percentage focused on Uber drivers (Metropolitan Police, 2017), with complaints directed towards unexpected behaviour and poor navigation experience. For instance, some drivers cancelled requests from passengers who were waiting for half an hour to get a ride to the airport, leaving them in a situation where they risked missing their flight. Other passengers missed their flights while drivers were trying to find directions to the airport. In general, Uber drivers did not meet the passengers' expectations and made many mistakes before, during, and after the ride. Uber passengers (trustors) hired Uber drivers (trustees) to pick them up at a time X and transport them from point A to point B within a time period Y, which is determined by the Uber application (expectation). Not meeting this expectation negatively affects the trust relationship, so trustors are negatively impacted and openly express their dissatisfaction through negative posts. In contrast, traditional taxis use a simplified expectation between trustors and trustees—that of transporting passengers from point A into point B, excluding other commitments offered by a sophisticated application. Therefore, it is uncommon to find negative tweets or comments on other social media from drivers towards passengers.

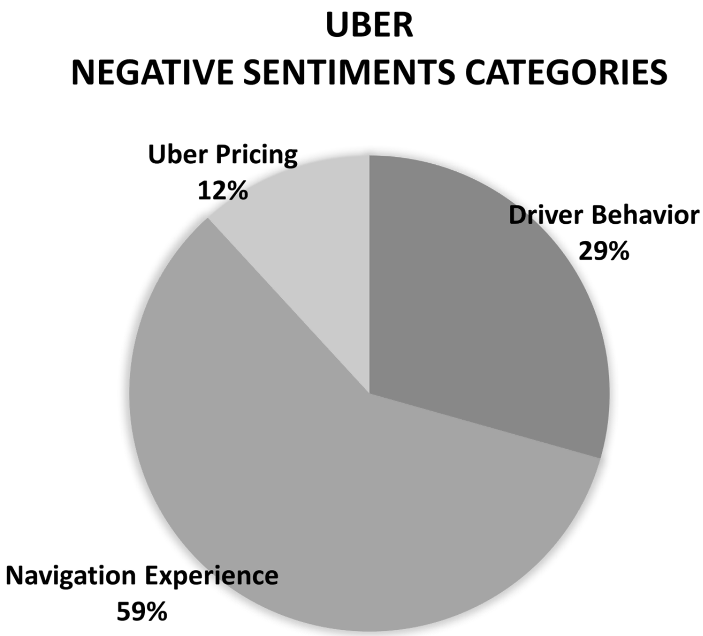


Figure 0.1: Negative Sentiments Towards UBER

An interesting finding regarding GrabTaxi, a modern taxi company following the C2C business model, is that sentiments towards this company were twice as positive as those towards Uber, making them equivalent to traditional taxis that follow the B2C business model, such as Hackney Carriage.

In further analysis and detailed comparisons between all taxi companies, we found that the difference between GrabTaxi and other modern companies, such as Uber, lies in the process of hiring drivers. GrabTaxi has limited their hiring process to accepting drivers through official government channels and transportation authorities within the country. In contrast, Uber, Hailo, and Lyft have a relatively relaxed hiring process.

To become a taxi driver for GrabTaxi, one must possess a government trade license and register as a taxi company. Therefore, the car registration should be changed from being individually owned into company property that is used as a taxi. In addition, one should post GrabTaxi logos on the car to identify it as a GrabTaxi car. These processes make GrabTaxi more similar to the B2C model than to the C2C model. In contrast, Uber, Hailo, and Lyft require only a police report and a valid driver license to start working for them.

Another interesting detail is that, whenever a driver is officially identified as a professional taxi driver, his/her behaviour changes accordingly. For instance, the driver may avoid taking certain actions, as s/he knows any misbehaviour may have legal consequences and result in job loss.

In summary, the results of sentiment analysis of user generated content about online vs. traditional sectors in the taxi industry revealed that sentiments towards B2C companies are more positive than those towards C2C companies. People tend to complain about C2C drivers' behaviours and lack of navigation experience on local roads. However, a notable exception in this respect is GrabTaxi. Although the company is built on the C2C model, sentiments towards GrabTaxi were much more positive as compared to those towards other C2C taxi companies, such as Uber, Lyft and Hailo. A reason underlying this difference is that, unlike other C2C taxi companies, GrabTaxi has a stricter hiring process that requires all drivers' registration with local government agencies.

3.2 Online Hospitality Services Industry

Next, we also analysed several C2C companies in the hospitality industry, including Airbnb, CouchSurfing, HomeAway, and VRBO, which are C2C marketplaces where people offer rental spaces. For the purpose of comparison, we used online marketplaces that offer accommodation in registered hotels and hotel apartments, such as Trivago and Expedia, which mostly follow the B2C business model. Several other companies (e.g., Booking, Kayak, and Tripping) were excluded from the analysis, because their brand names are common word in the English language.

Table 0-2: Sentiment Analysis Results (Hospitality Companies)[1]

	Model	**Positive**	**Neutral**	**Negative**	**Negative to Positive Sentiment Ratio**
VRBO	C2C	98	42	5	5%
Expedia	B2C	63	27	4	6%
HomeAway	C2C	84	36	6	7%
Trivago	B2C	82	35	14	17%
CouchSurfing	C2C	102	44	25	25%
Airbnb	C2C	123	53	41	33%

Table 3.2 summarizes the frequency of positive and negative sentiments per minute based on the SocialMention analysis. It also shows the frequency of mentions (posts) for all companies.

[1] Note: Numbers normalised per minute.

There is at least one mention (post) per minute for each company on Twitter, Reddit, Photobucket, Topix, and many other blogs.

The sentiments towards C2C marketplaces in the hotel industry, such as Airbnb and CouchSurfing, were more negative than those towards B2C marketplaces that specialize in registered hotels, such as Expedia and Trivago. Airbnb had 33% negative posts (1 negative post per 2 positive posts), while Trivago had 17% negative posts (1 negative post per 5 positives posts). This means the sentiments towards Trivago are 2.5 times more positive that those towards Airbnb. Expedia had 6% negative posts, which makes its assessment by users seven times more positive than that of Airbnb.

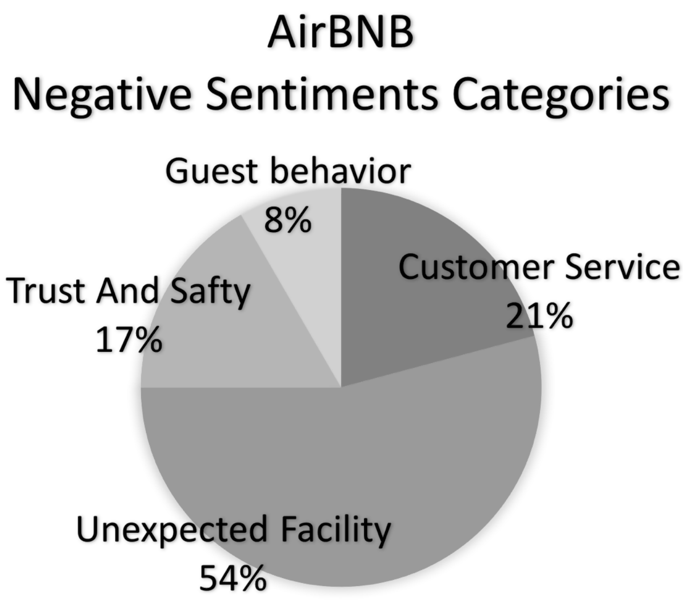


Figure 0.2: Negative Sentiments Towards Airbnb

Table 3.2 also shows that two of the C2C-based companies (VRBO and HomeAway) have the same sentiment levels as two of the B2C companies (Expedia and Trivago). Upon deeper analysis, we found that VRBO and HomeAway have firmer rules for listing properties online as compared to the rules applied by other C2C-based companies. Specifically, VRBO and HomeAway do not allow any property to be listed on the public domain until the owners properly identify themselves. All properties are suspended until the host links a valid payment method, such as a credit card or a bank account. In contrast, Airbnb accepts almost any property owners with the minimal quality assurance and identity verification. To verify this finding, we created a virtual property in Dubai and filled all required fields with a description of a house. We then uploaded fake photos and randomly assigned a location. Within minutes, this fake property listing appeared on the Airbnb public site. We also received a confirmation email from Airbnb congratulating us on our first published property (see Annex A for a screenshot and the email confirmation). To conclude, it appears that the reason of a more positive evaluation of some C2C marketplaces in the hospitality industry, such as VRBO and HomeAway, is related to these marketplaces' more stringent rules of host registration. Whenever a host is fully identified by the site, s/he feels obligated to avoid dishonest behaviours that could lead to financial loss or discontinuity of their listing.

Figure 3.2 presents the main categories of negative tweets related to the Airbnb company. The highest percentage of negative tweets focused on unexpected homes/rooms provided by the host. The online description does not always match reality. The second most popular negative tweet category focused on Airbnb customer service. Finally, the third most popular category was related to the lack of trust and security provided by the host.

In summary, the results of sentiment analysis of online hospitality businesses revealed that sentiments towards B2C marketplaces that offer accommodation in registered hotels are generally more positive than those towards C2C companies. Users tend to produce negative posts about fake listings, dirty homes/rooms, and untrustworthy hosts. However, HomeAway and VRBO were exceptions from this pattern. A possible reason for this difference could be the stricter host registration process required on these websites before a host is allowed to publish a property listing on HomeAway and VRBO.

3.3 Online Retail Industry

Finally, we also performed sentiment analysis of UGC on several companies in the retail industry. To this end, we selected Amazon and eBay, which are C2C sellers and resellers of goods online with no physical stores, versus Best Buy, Radio Shack, Carrefour, and Aldi, which are traditional brick-and-mortar B2C stores that sell and resell similar goods.

Table 3.3 summarizes the frequency of positive and negative sentiments per minute based on the results of applying the SocialMention tool. The table also shows the frequency of mentions (posts) for all companies. There is at least one mention (post) per minute for each company on Twitter, Reddit, Photobucket, Topix, and many other blogs.

Table 0-3: Sentiment Analysis Results (Retail Companies)

	Model	**Positive**	**Neutral**	**Negative**	**Negative to Positive Sentiment Ratio**
Aldi	B2C	93	40	4	4%
Radio Shack	B2C	63	27	3	5%
Carrefour	B2C	132	56	7	5%
Best Buy	B2C	142	61	14	10%
eBay	C2C	92	39	18	20%
Amazon	B2C, C2C	105	45	26	25%

Figure 0.3: Negative Sentiments Towards Amazon

As in the analysis of the taxi and hospitality industries, the sentiments expressed in UGC towards C2C online e-commerce businesses with no physical stores (Amazon and eBay) were more negative than those towards e-commerce businesses with physical stores (brick-and-mortar stores). Amazon had 25% negative sentiments (1 negative per 3 positive posts), while Best Buy had 10% negative sentiments (1 negative per 9 positives posts). This demonstrates that the sentiments towards Best Buy are three times more positive as compared to those towards Amazon. Carrefour had 5% negative sentiments (1 negative per 19 positive posts), suggesting that the sentiments towards Carrefour are six times more positive than those towards Amazon.

Figure 3.3 presents the main categories of negative tweets related to the Amazon company. The lowest percentage of negative posts was focused on product issues and prices, while the highest percentage was related to delivery issues.

In summary, the sentiment analysis of the online retail industry revealed that sentiments towards online stores with physical locations, such as Best Buy, Carrefour, and Aldi, are more positive than those towards online-only stores, such as Amazon and eBay. Most of the negative sentiments shared publicly were related to delivery issues.

3.4 Summary

The results of sentiment analysis of UGC about C2C vs. B2C businesses in three online industries—taxi, hospitality, and retail—convincingly demonstrate that C2C marketplaces garnered more negative sentiments as compared to B2C marketplaces. An interesting detail suggested by our analysis is that those C2C e-commerce businesses that used physical validation of their users (such as the requirement of registration with a state authority for taxi drivers, or the need to add a valid payment method for property owners in the hospitality industry) garnered more positive sentiments on part of users.

CHAPTER FOUR: UNDERSTANDING ONLINE TRUST RELATIONSHIPS

The results of sentiment analysis of hundreds of negative posts reported in Chapter Three demonstrated that the three aspects of a trust relationship (trustor, trustee, and expectation) were visible in B2C and C2C e-commerce businesses. However, if expectations are not met, there are changes in the perception of defining the level of trust between two parties and the impact on both parties. In this chapter, we aim to explain trust relationship in physical world and compare it with online trust relationship. To this end, we consider four scenarios (Sections 4.1-4.4).

4.1 Scenario 1: Father Playing with Baby

Consider the definition of trust discussed earlier in this thesis and apply it to a real-world scenario: a father playing with his son by tossing him into the air. As illustrated in Figure 4.1, the baby plays the role of **trustor,** while the father plays the role of **trustee**. The **expectation** is that the father catches the baby before the baby hits the ground. This leaves the baby highly vulnerable and dependent on the success of the father. Meeting the expectation not only has a high impact on the baby, but also on his father (assuming he is a caring father).

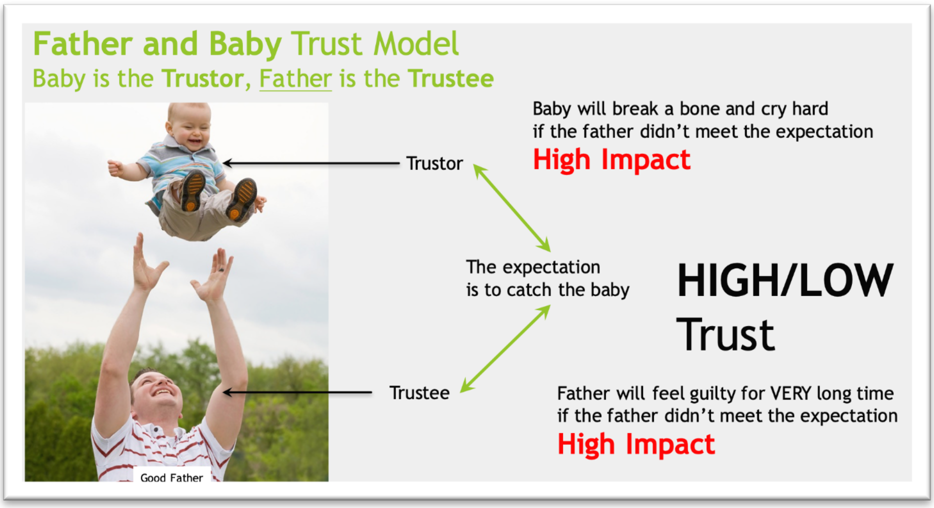


Figure 0.1: Father and Son High Trust Model

It is very important to separate the impact of meeting the expectation on the baby versus the impact on the father. This relationship is considered to be a low trust relationship if the safety of the baby is not a top priority for the father. Therefore, the success of this relationship depends on the trustee (father) more than on the trustor (baby).

4.2 Scenario 2: Product-to-Money Trust Model

Buying groceries can be used as an example that demonstrates a trust relationship in our daily lifestyle. In grocery stores, one would typically collect items and pay for them at the listed price before leaving the shop.

In this example (see Figure 4.2), the cashier is playing the role of **trustor,** and the customer is playing the role of **trustee**. The **expectation** is that the customer will exchange money for products before leaving the store. The cashier is dependent on the customer meeting this expectation and thus is subject to a significant impact if the customer pays/does not pay for the products.

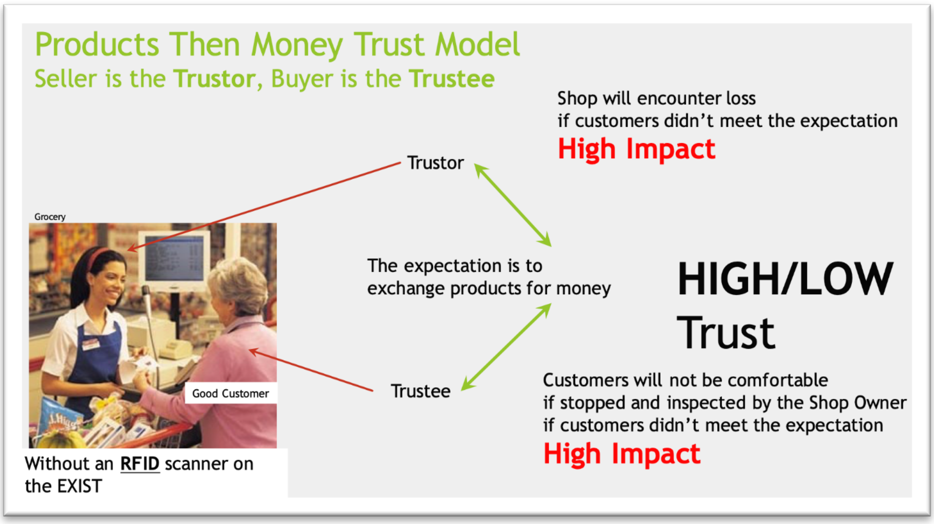


Figure 0.2: Product-to-Money High Trust Model

The success of this relationship does not depend on how important it is to the casher, but rather on how much it will impact the customer reputation if s/he does not pay for the products (assuming s/he is a good customer). In some parts of the world, there are documented cases of stores that operate without any cashier control. Customers are requested to leave the payment for goods in a safe (or use an automatic payment machine) and then enjoy the items they purchased. However, in other parts of the word, store owners introduced RFID gates to increase the probability of customers' meeting the expectations.

Although the impact of not meeting the expectation is high on the cashier/store owner, the actual determination of the level of trust depends on the probability of the trustee meeting the expectation of the trustor. In this example, changing the trustee from a customer who worries about his/her reputation into a customer who does not worry much will reverse the trust level.

4.3 Scenario 3: Traditional B2C E-Commerce Trust Model

In this scenario (see Figure 4.3), the example focuses on the trust relationship when engaging in a transaction online. Amazon was chosen because it is a popular retail brand around the world. The company cares about its reputation and branding.

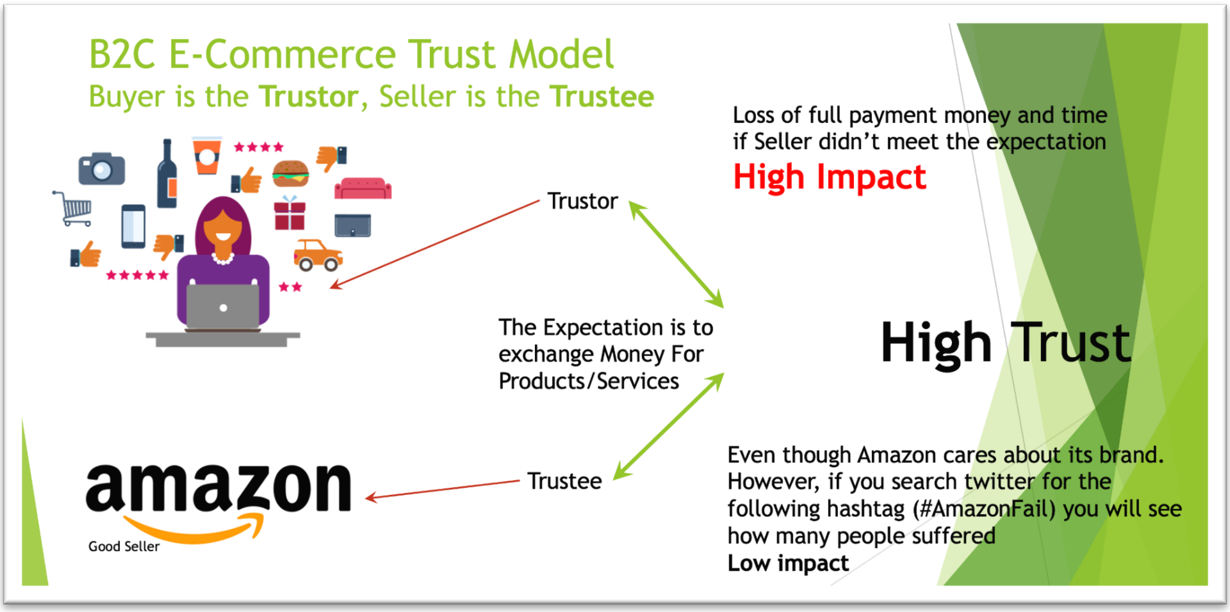


Figure 0.3: Traditional E-Commerce B2C High Trust Model

In this example, the user plays the role of **trustor,** and Amazon plays the role of **trustee**. The **expectation** is to exchange products for money before leaving the site. The user typically pays the full price before receiving the items. In this example, the user is highly dependent on Amazon to meet the expectation of delivering the item. The user is also vulnerable to Amazon's decision on whether or not to deliver the purchased products or services. On the other hand, Amazon brand is big enough to fail several transactions without any significant impact on its reputation.

In the sentiment analysis on the online retail industry in Chapter Three, the percentage of negative sentiments towards Amazon's online-stores was higher compared to that of other break-and-motor stores. Therefore, unless Amazon is highly impacted by fulfilling the expectation of the trustor, just like a traditional break-and-motor store (e.g. Aldi, or Radio Shack), the trust model will always be high**.**

4.4 Scenario 4: Modern C2C E-Commerce Trust Model

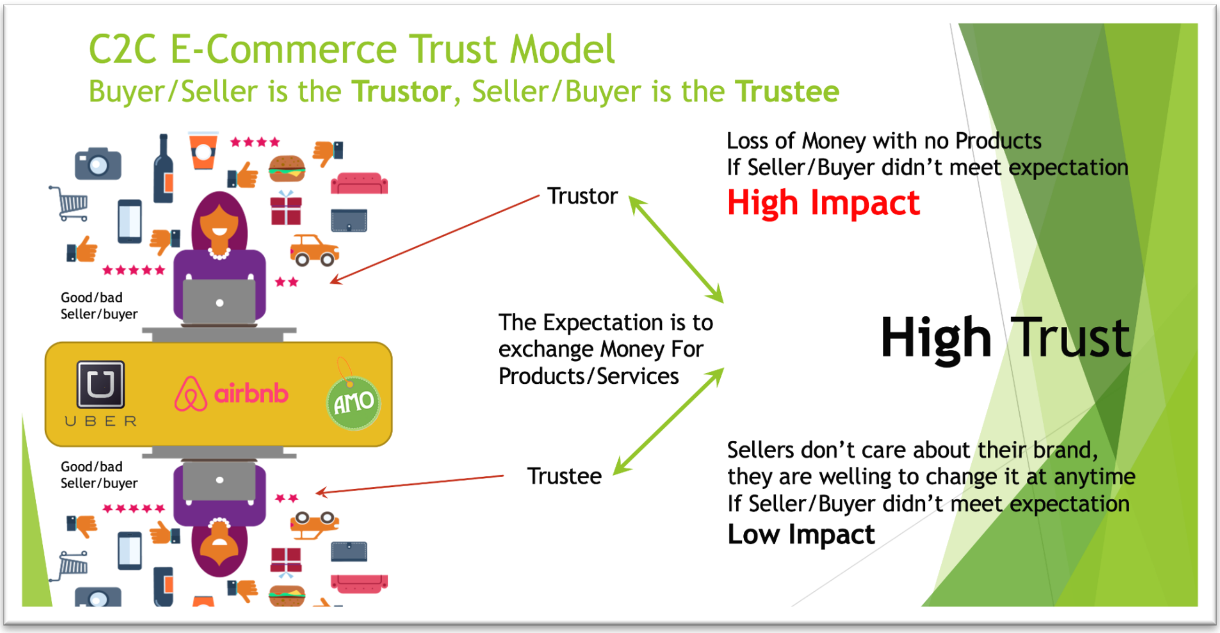


Figure 0.4: Modern E-Commerce High Trust Model

In traditional B2C e-commerce, a business employee is responsible for meeting the expectations of a trustor. Any misbehaviour that does not satisfy expectations will have a serious impact on the employee's career. Unlike traditional B2C e-commerce, modern C2C e-commerce involves a wide variety of trustees who do not necessarily have a firm commitment to the marketplace (see Figure 4.4). Typically, the impact of any misbehaviour of trustee in C2C is minimal as compared to the traditional B2C model. Hence, the trust relationship is significantly high.

4.5 Summary

In the sentiment analysis of the taxi industry and hospitality services in Chapter Three, the percentage of negative sentiments towards Uber and Airbnb were higher as compared to other physical companies (e.g., Yellow cab, or traditional hotels). This shows how the sentiment affects the trust which indicates that e-commerce transactions are considered to be high trust transactions in general.

The three aspects of a trust relationship (trustor, trustee, and expectation) are visible in all four scenarios discussed in Sections 4.1-4.4. Unlike in traditional break-and-motor stores, in e-commerce (B2C or C2C), the level of trust is considered to be high due to the fact that the impact of not meeting the expectation is low on the trustee and high on the trustor. Moreover, there is no guaranty that the expectations of the trustor will be met (nobody can say for sure that the red is really red).

CHAPTER FIVE: APPLICATION OF THE RESEARCH METHODS TO THE PROBLEM

In this chapter, we define an approach to quantify trust from joy and fear that are detected in UGC published by sellers in C2C marketplace. In Section 5.1, we provide further detail about the data collection process, while Section 5.2. Pertinent definitions of key terms and an outline of the process of text mining are provided in Sections 5.3-5.5. Finally, in Sections 5.6 and 5.7, as an illustration of the model, we present two examples of two different listings published on Airbnb. One of them is a not trusted listing that received multiple negative reviews (Sample A), while the other is a trusted listing that attracted multiple positive reviews (Sample B).

The motivation of this research is to help guests, who are looking to rent a new facility and want to establish a first-time relationship with hosts, to quantify the trust found in the text written by the seller. The approach used in quantifying trust in the text, can also be used by the host to edit the description about their facilities and set the right expectation before any guest get into a disappointing experience.

5.1 Data Collection

In the present study, we used Airbnb data available to us under a Creative Commons CC0 1.0 Universal (CC0 1.0) "Public Domain Dedication" license (Appendix B). Specifically, we focused on the following cities:

1. Asheville, North Carolina, United States. Data published on April 18, 2017

2. Manchester, England, United Kingdom. Data published on April 10, 2017

3. Boston, Massachusetts, United States. Data published on Oct 6, 2017
4. Vancouver, Canada. Data Published on Oct 6, 2017

Those four cities were selected because of their similarity in size and the number of rooms/homes/apartments listed on Airbnb (at the time of data collection). Asheville data were used to train the model, while Manchester, Boston, and Vancouver data were used for evaluation.

The data for analysis were collected from the Inside Airbnb website. Table 5.1 lists several representative cities datasets published by Inside Airbnb. The first column is the city name, while the second column ("Listings") shows the number of rooms/homes/apartments offered in that city. The column "Occupied Nights/Year" is the average number of nights each listing is occupied per year, thus providing information on how active the city is. The column "Reviews" shows the total number of reviews received from all guests who booked accommodation in that city. The last column ("Review/Listing") shows the average number of reviews received per listing in that city. Considering that writing a review is not mandatory on Airbnb, the "Review/Listing" varies across cities depending on how active/keen guests in writing reviews of hosts/accommodation on Airbnb are. The cities compared in the present study appear in bold.

Table 0-1: Selected Cities (Inside Airbnb Website)

City	Listings	Occupied Nights/Year	Reviews	Review /Listing	Size (MB)
Amsterdam	18547	84	337,118	18	
Antwerp	1227	99	26,547	22	
Asheville	**742**	**130**	**27,721**	**32**	**158.3**
Athens	5,127	96	124,227	24	
Austin	8,808	70	140,479	16	181.2
Barcelona	17,369	99	388,184	22	

Berlin	20,576	95	265,631	13	
Boston	**4,870**	**107**	**120,737**	**25**	**112.5**
Brussels	6,192	81	111,676	18	
Chicago	5,207	118	132,147	25	
Copenhagen	20,545	57	220,347	11	
Edinburgh	9,638	126	259,251	27	403.4
Geneva	2,408	71	25,479	11	9.9
Hong Kong	6,474	67	82,393	13	39.9
London	49,348	89	564,297	11	306.5
Vancouver	**4,838**	**151**	**160,138**	**33**	**24.6**
Los Angeles	31,253	93	651,392	21	
Madrid	12,775	99	290,810	23	
Málaga	4,853	88	97,811	20	
Mallorca	14,858	37	109,522	7	
Manchester	**865**	**103**	**14,880**	**17**	**147.8**
Melbourne	12,174	85	182,120	15	101.7
Montréal	10,619	55	97,204	9	73.8

As shown in Table 5.1, while Asheville has a low number of listings on the Airbnb site as compared to other cities (at the time of data collection), its average number of reviews per listing is one of the highest (32 reviews). For instance, Austin has 10 times more listings on Airbnb than Asheville (8,808 vs. 742, respectively). However, the average number of reviews per listing in Austin is half that of Asheville (16 vs. 32, respectively). Manchester is similar to Asheville in terms of the number of listings (865 vs. 742, respectively), but has a two times lower average number of reviews per listing (32 vs. 17 reviews, respectively).

Similarly, if we compare Boston and Vancouver, Boston has a higher average number of listings on the Airbnb site compared to other cities (at the time of data collection); however, its average number of reviews per listing is 25 reviews per listing. On the other hand, Vancouver is similar to Boston in terms of the number of listings (4,838 vs. 4,870, respectively), but it has a higher average number of reviews per listing (33 vs. 25 reviews, respectively).

Figures 5.1-5.4 captured from Inside Airbnb (http://InsideAirbnb.com), show the densities and distributions of Airbnb listings in each of the four cities. Red dots represent all homes/apartments offered on Airbnb, while green dots represent private rooms offered on Airbnb. As can be seen in Figures 5.1-5.4, the densities and distribution of homes/apartments in Ashville and Manchester are similar. They are also similar in Boston and Vancouver.

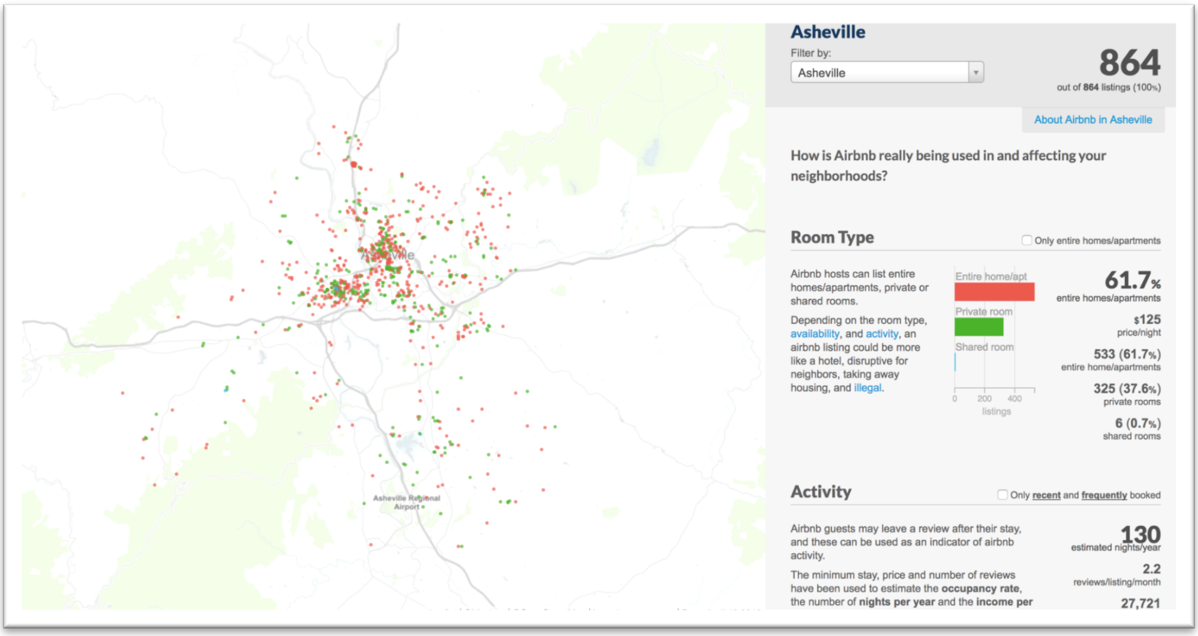


Figure 0.1: Accommodation Distribution (Asheville, USA)

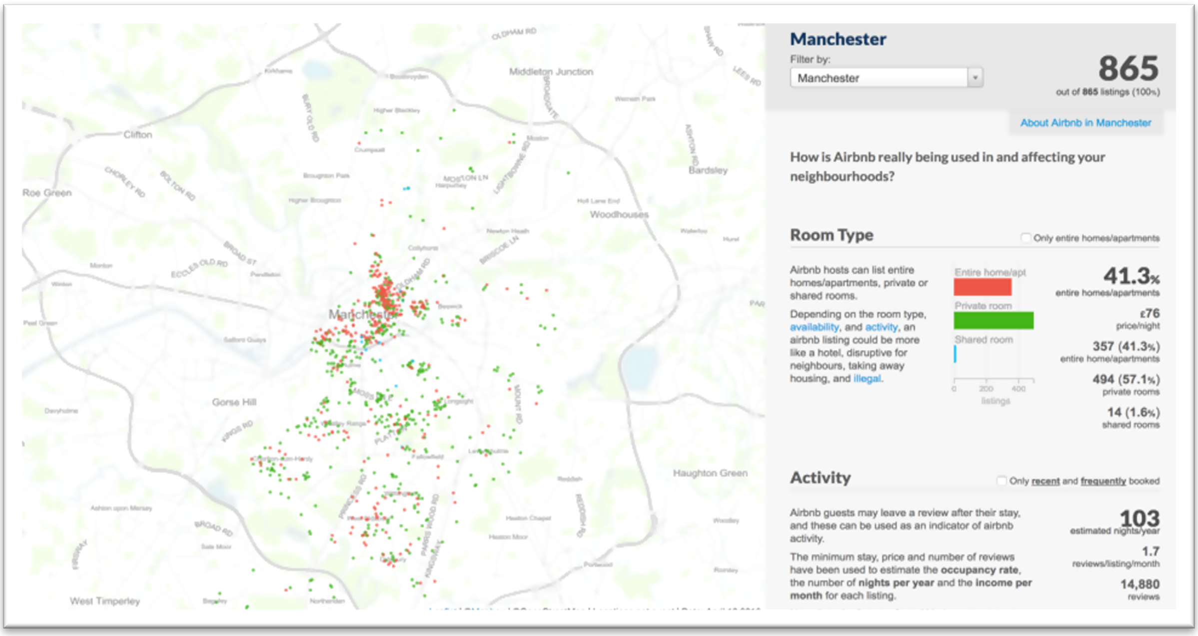


Figure 0.2: Manchester, UK Accommodation Distribution (Manchester, UK)

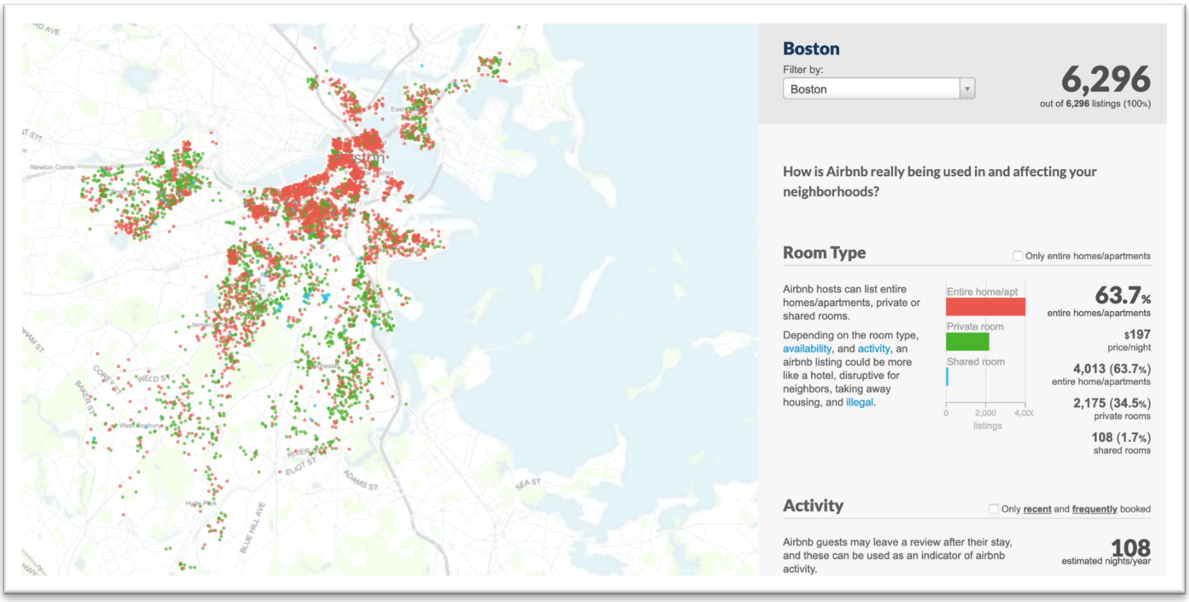


Figure 0.3: Accommodation Distribution (Boston, USA)

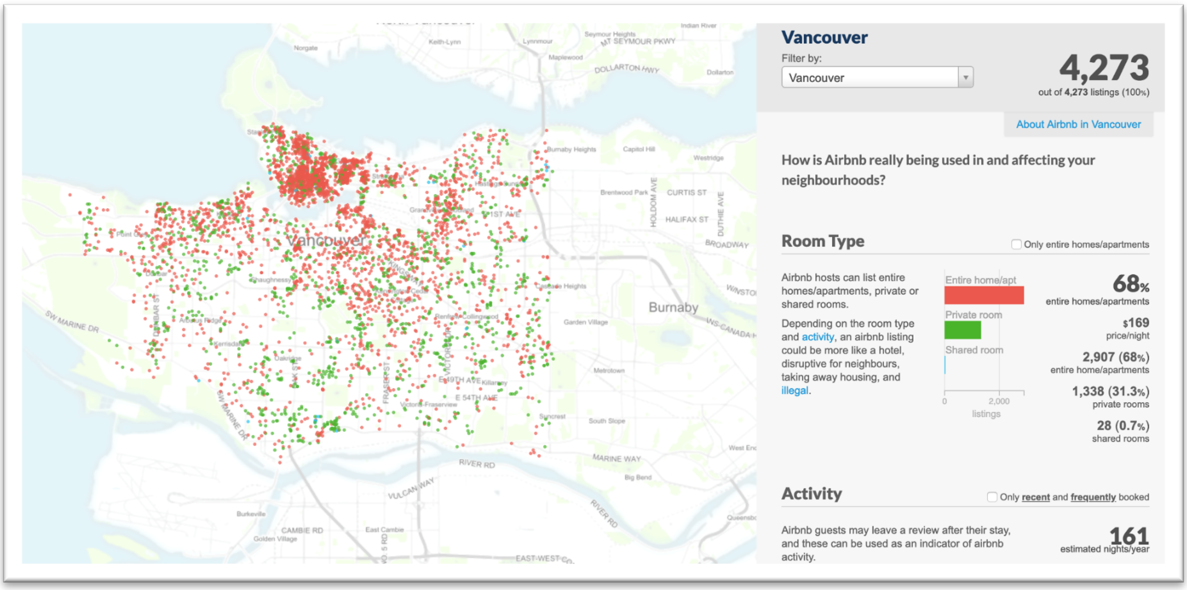


Figure 0.4: Accommodation Distribution (Vancouver, Canada)

5.2 Data Model

The Airbnb data model published for those cities (Asheville and Manchester) consists of the following five data components (see Figure 5.5).

- *Listings* include summarised versions of the listed properties.
- *Listings_details* include full details regarding listed properties, including a description from the host and directions to the nearest subway station. This is one of the main files used in the present study
- *Review_details* include all guests' reviews of the properties they used. Review details are linked to listings and listings details through a foreign key (listing_id). This is another main type of files used in the present study
- *Neighbourhoods* include segmentations of the city and link the properties to the segments they belong to.

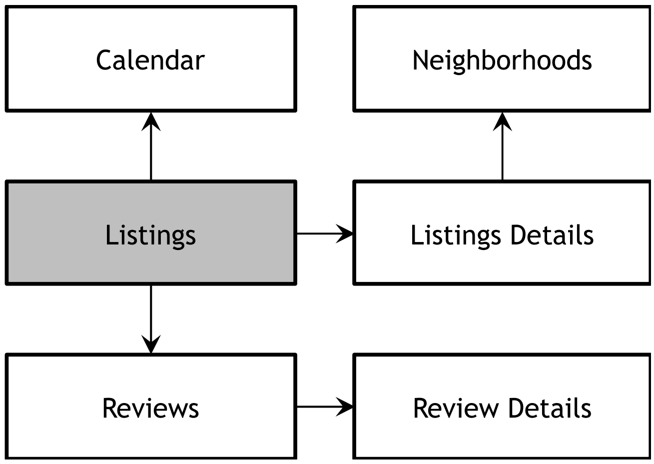


Figure 0.5: Relationship Data Model (Inside Airbnb Website)

5.3 Trust Definition

Trust is a basic emotion (Plutchik, 1986, 2001) that has a psychological impact on the decision-making processes in e-commerce. Specifically, trust can influence individual behaviour and decisions when finalising deals and performing actions.

In the present thesis, we adopt the definition of trust where trust is assumed to consist of the following three main parts: trustor, trustee, and expectations (see Figure 5.6).

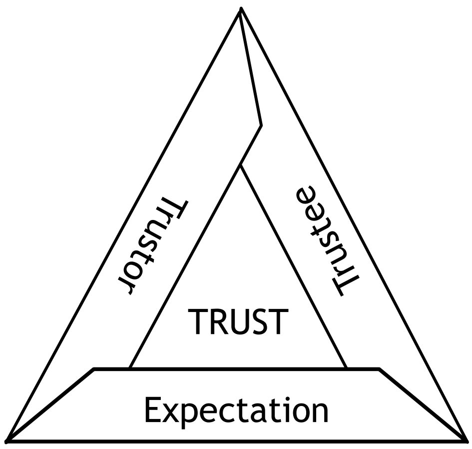


Figure 0.6: Trust Triangle.

5.4 Text Mining

This section describes the text mining and conceptual framework proposed to measure trust in Airbnb host listings. As mentioned in Section 2.2, basic emotions of joy, anger, fear, disgust, and sadness can be detected in texts using sentiment analysis tools. Trust is also one of the eight basic emotions (Plutchik 1986, 2001). Therefore, the conceptual framework is designed to identify trust using UGC written by the hosts (i.e. listing descriptions). Figure 5.7 shows a flowchart outlining the stages of the text mining steps used in the present study.

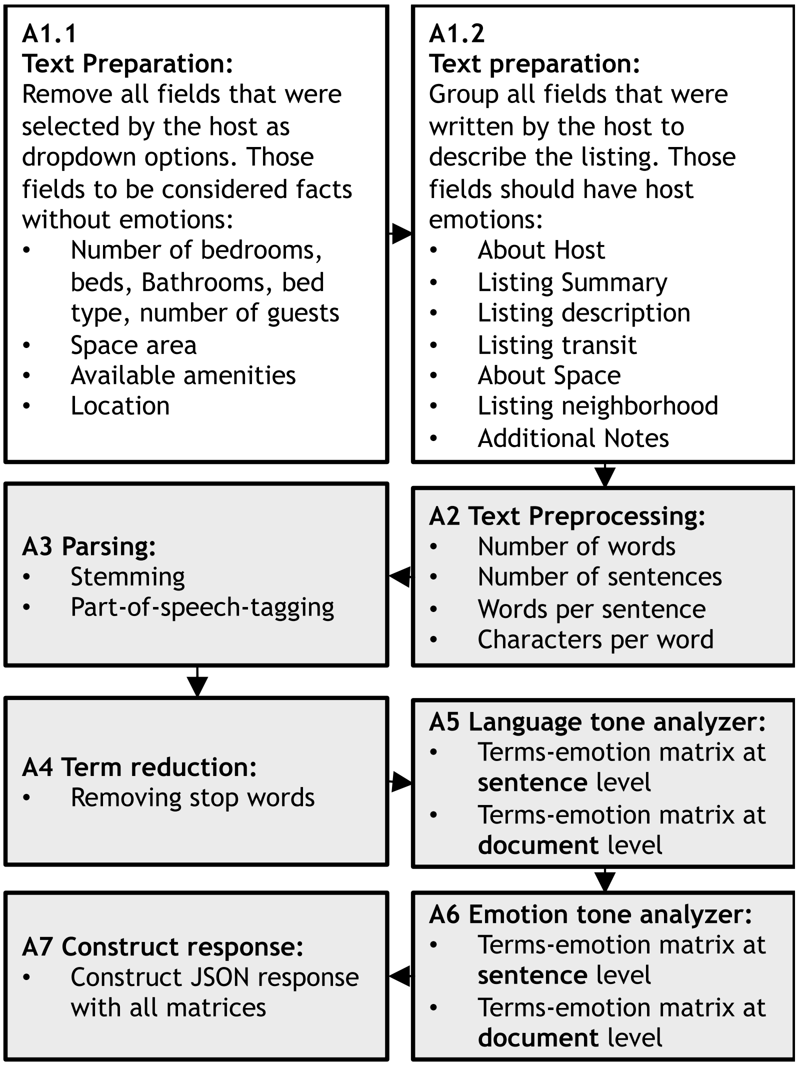


Figure 0.7: Text Mining Steps

Furthermore, Figure 5.8 shows the steps of the conceptual framework we used to train the analyser and generate trust rules. The first two steps, B1 and B2, are the text mining steps performed on the Airbnb data (see Figure 5.7). In the analysis of the sentiment used in UGC of host listings, each listing was assigned five emotional sentiments (anger, disgust, fear, joy, and sadness). We selected several strongest sentiments found in the text and then performed Principal Component analysis for the dimension reduction. This reduced the output to a two-dimensional representation. Finally, hosts' emotional sentiments were classified using a K-means classifier.

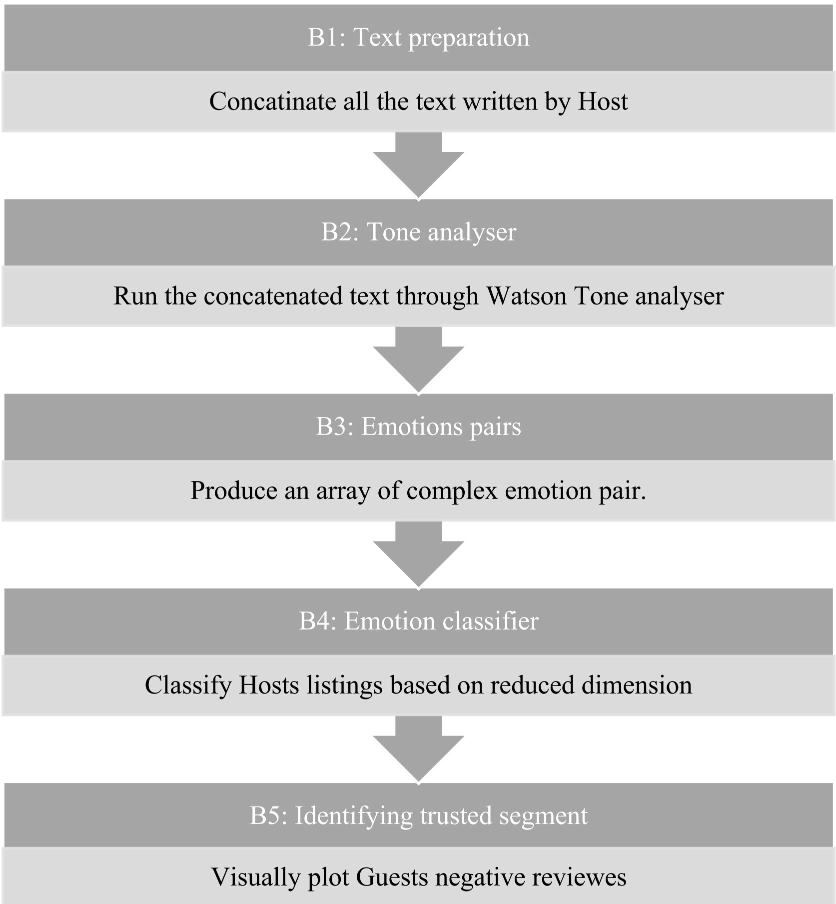


Figure 0.8: Conceptual Framework

The IBM Watson Tone Analyzer service was used to detect emotional and language tones in written text (Gain & Hotti, 2017). The service is based on psycholinguistics theory, an area of research that explores the connection between linguistic behaviour and psychological theories. To develop scores for each of these tone dimensions, the service uses linguistic analysis and the correlation between the linguistic characteristics of written text and emotional and language tones.

In their daily communications, for example, individuals exhibit different tones: joyful or sad, open or conservative, analytical or informal (Gou and others, 2014, and Jian and others, 2014). In various contexts, these tones can impact the perception of the online identity of a person and the effectiveness of their communications.

Watson Tone Analyzer can analyse tone at both document and sentence levels. It is trained to analyse large corpora to predict the tone of new texts. For each of the tone, Watson trains its model independently using the One-Vs-Rest multi-class classifications. Which is one type of binary classifications for multi classes. It starts with splitting the dataset into multiple classes and comparing each one of them to the rest of the data set. This is where the name came from, One-vs-Rest. For example:

- Joy vs [Sad, Anger, Fear]
- Anger vs [Joy, Sad, Fear]
- Sad vs [Joy, Anger, Fear]

IBM Watson Tone Analyzer has a dedicated model for Each emotion. During prediction, the tones predicted with at least 0.5 probability are taken as the final tones. In the present study, Watson Tone Analyzer service was used to perform steps A5-A6 shown in Figure 5.7.

Our proposed conceptual framework classified guest reviews of Airbnb hosts into the following two groups: (1) negative; (2) positive. The principles of classification were as follows. If the review gave 1, 2, 3, or 4 stars to the host/accommodation, it was classified as a negative review; by contrast, a 5-star review was considered a positive review. Previous studies demonstrated that guest reviews on Airbnb tend to be biased and are mostly positive (Fradkin et al. 2015, 2018; Lee et al., 2018). This trend is due to the fact that Airbnb guests want the host to write a similarly positive review of them. This, in turn, guarantees that the guest will be accepted by other hosts and will get better deals in the future.

5.5 Sample A: Untrusted Host Iteration

In this section, we present an example of a not trusted listing that attracted multiple negative reviews.

5.5.1 Host Listing Details

The image below is for a house in Vancouver Canada with 7 rooms that can fit 14 or more guests (see Table 5.2). However, regardless of the amazing photos and description of the host, this listing has a very low rating of 3 stars based previous guests' experience. This listing will be considered as an untrusted host listing, and we will run our proposed method in order to classify this listing as untrustworthy before any guest has a disappointing experience.

Figure 0.9: Sample A (Airbnb Public Website)

Table 0-2: Untrusted Host Listing Details Published on Airbnb

Field Name	Value
URL on Airbnb	https://www.airbnb.com/rooms/29889054
Listing title	Dreaming house
House details	14 guests / 7 bedrooms / 9 beds / 4 baths
Listing summary	Located in Mackenzie St & W 49th Ave, Kerrisdale district, Vancouver, downtown area. Nearby shopping mall, bank, restaurant, golf course, post office, library, nature park, cinema, community center, gas station and other facilities have everything.
Listing space	Living room and kitchen, private garden and free parking
Listing description	Located in Mackenzie St & W 49th Ave, Kerrisdale district, Vancouver, downtown area. Nearby shopping mall, bank, restaurant, golf course, post office, library, nature park, cinema, community center, gas station and other facilities have everything. Living room and kitchen, private garden and free parking Huge private garden, hot tub pool, model kitchen and lovely living space NO shoes in house and NO smoking around house There are golf course, post office, library, park, cinema, community center, gas station and other facilities have everything. It is convenient to walk 3 minutes to 49

	(UBC to Metrotown Station), 16 (via DT to Nanaimo and 29th streets), 15 minutes by bus to Oakridge Mall, Kerrisdale business district, 30 minutes by bus to Richmond times square, night market, and only 15 minutes by bus from McArthurGlen outlets and Vancouver airport. Free Fresh daily fruits
Neighbourhood overview	There are golf course, post office, library, park, cinema, community center, gas station and other facilities have everything.
Notes	Free Fresh daily fruits
Transit	It is convenient to walk 3 minutes to 49 (UBC to Metrotown Station), 16 (via DT to Nanaimo and 29th streets), 15 minutes by bus to Oakridge Mall, Kerrisdale business district, 30 minutes by bus to Richmond times square, night market, and only 15 minutes by bus from McArthurGlen outlets and Vancouver airport.
Access	Huge private garden, hot tub pool, model kitchen and lovely living space
House rules	NO shoes in house and NO smoking around house

5.5.2 Tone Analysis for Host Listing

After removing all fields selected by the host from the dropdown list, we concatenated all fields written by the host to describe the listing, e.g., listing title, listing summary, listing space, listing description, neighbourhood overview, listing transit, access, house rules. Then the description was submitted to IBM Watson Tone Analyser to extract the emotions expressed in the data (see Table 5.3).

Table 0-3: Watson Tone Analysis (Sample A)

Joy	< 0.5	0.5 – 0.75	> 0.75
Fear	< 0.5	0.5 – 0.75	> 0.75

Dreaming house. Located in Mackenzie St & W 49th Ave, Kerrisdale district, Vancouver, downtown area. Nearby shopping mall, bank, restaurant, golf course, post office, library, nature park, cinema, community center, gas station and other facilities have everything. Living room and kitchen, private garden and free parking. Located in Mackenzie St & W 49th Ave, Kerrisdale district, Vancouver, downtown area. Nearby shopping mall, bank, restaurant, golf course, post office, library, nature park, cinema, community center, gas station and other facilities have everything. Living room and kitchen, private garden and free parking. Huge private garden, hot tub pool, model kitchen and lovely living space.

There are golf course, post office, library, park, cinema, community center, gas station and other facilities have everything. It is convenient to walk 3 minutes to 49 (UBC to Metrotown Station), 16 (via DT to Nanaimo and 29th streets), 15 minutes by bus to Oakridge Mall, Kerrisdale business district, 30 minutes by bus to Richmond times square, night market, and only 15 minutes by bus from McArthurGlen outlets and Vancouver airport. Free Fresh daily fruits. There are golf course, post office, library, park, cinema, community center, gas station and other facilities have everything.Free Fresh daily fruits. It is convenient to walk 3 minutes to 49 (UBC to Metrotown Station), 16 (via DT to Nanaimo and 29th streets), 15 minutes by bus to Oakridge Mall, Kerrisdale business district, 30 minutes by bus to Richmond times square, night market, and only 15 minutes by bus from McArthurGlen outlets and Vancouver airport. Huge private garden, hot tub pool, model kitchen and lovely living space. NO shoes in house and NO smoking around house

Note: Concatenated text written by the host, each emotion is color coded based on the intensity of the emotion found in the text

5.5.3 Selected Guest Review

However, despite the very positive description of the property (see Section 5.6.1), this listing has received numerous negative comments from users. A sample review of the property is provided below.

Guest Feedback
Hello Future Guests, whatever you do, please do not stay at this home!!!!! It's a scam. The photos on the website are all staged. Please look at each one and you will notice that the same props are set up in every photo. Unfortunately, I fell for this. I walked in and it was NOT AT ALL what you see on the Airbnb website. The hot tub pool was empty and dirty and the tiles in the house were moldy, grey and disgusting and had to wear shoes in the home. The host will play dumb when you ask him when you arrive and say the electrical is not good and that's why the hot tub doesn't work and say he has already taken this feature out of Airbnb (you will see, it is not). You will not find this out until you arrive. The beds are rock hard. It is only box springs on the beds with a small piece of foam on it. We all had sore necks, backs and did not sleep comfortably at all. There was no toilet paper and not enough towels for all the guests. The carpets upstairs had dirt all over them. The oven does not work and the dishwasher leaks. This place is NOT worth what he is asking, nor does it look like the photos on the website. This is not a Dream House this is a NIGHTMARE house! He will not refund your money and he will scam as many people as he can. Speaking to neighbors they have been trying to sell and can't. This is because it's dirty, moldy and smells like crap and we needed to air out the whole house. What this host is doing is not acceptable and I am warning people to not waste their money and

advising people DO NOT STAY HERE!!!!!! **You don't want to be as disappointed as I was.**	

5.6 Sample B: Trusted Host Iteration

In this section, we present an example of a trusted listing that attracted multiple positive reviews.

5.6.1 Host Listing Details

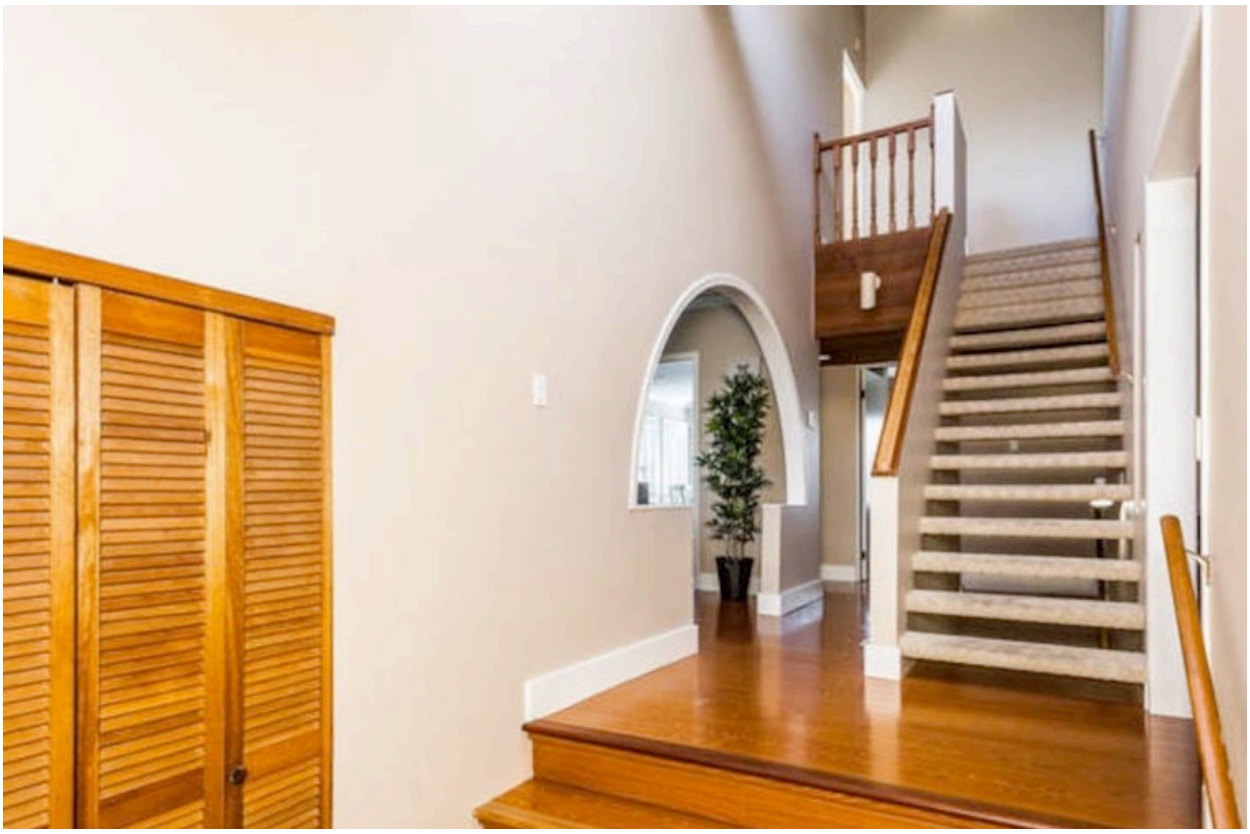


Figure 0.10: Sample B (Airbnb Public Website)

This listing had the following description (see Table 5.4).

Table 0-4: Untrusted Host's Listing Details Published on Airbnb

Field Name	Value
URL on Airbnb	https://www.airbnb.com/rooms/23896457
Listing title	West Coast Style, Deck, Transit: 15 Min Dtwn/Beach

House details	16+ guests / 7 bedrooms / 7 beds / 4 baths
Listing summary	Gorgeous, updated 7-bedroom, 4-bathroom whole house. Cedar ceilings add a cozy, West Coast feel to this spacious home. Across from a park on a quiet street but just two blocks to restaurants and shopping on Granville St. Free street parking with room for up to 4 vehicles.
Listing space	Three level home with spacious deck and backyard, built in 1945 and recently renovated. Garage and street parking available. The main level has new laminate floor, updated kitchen, dining room, living room, two bedrooms, and one full bathroom. The kitchen was renovated less than 10 years ago with stone counters, wood cabinets, tile floor, and stainless-steel appliances. Please note the tile has a few cracks. All dishes, pots, pans etc. are provided, so you just bring the food! Eat in the kitchen with an extendable table that seats 4-6. The spacious back deck with BBQ has a door from the kitchen. The dining room is next to the kitchen and has an extendable table that seats up to 10. The living room has a sofa bed (double) and a wood burning fireplace. The bathroom has been renovated with tile floor, stone counters, and tiled shower. Both bedrooms on this level have a queen size bed, nightstands, lamps, and mirror. One room has a closet and one room has a wardrobe. The upstairs is carpeted
Listing description	Gorgeous, updated 7-bedroom, 4-bathroom whole house. Cedar ceilings add a cozy, West Coast feel to this spacious home. Across from a park on a quiet street but just two blocks to restaurants and shopping on Granville St. Free street parking with room for up to 4 vehicles. Three level home with spacious deck and backyard, built in 1945 and recently renovated. Garage and street parking available. The main level has new laminate floor, updated kitchen, dining room, living room, two bedrooms, and one full bathroom. The kitchen was renovated less than 10 years ago with stone counters, wood cabinets, tile floor, and stainless-steel appliances. Please note the tile has a few cracks. All dishes, pots, pans etc. are provided, so you just bring the food! Eat in the kitchen with an extendable table that seats 4-6. The spacious back deck with BBQ has a door from the kitchen. The dining room is next to the kitchen and has an extendable table that seats up to 10. The living room has a sofa bed.
Neighbourhood overview	Granville street has a fantastic little village between W 63 and W 70. The house is just one block from this walkable shopping district. Many flavors of restaurants, including sushi, Chinese, pizza, Subway and much more! Starbucks is just two blocks from the house at Granville and W 64 and Safeway grocery

	store and the BC Liquor Store is at W 70. The Marple neighborhood has includes a public library. The house is across the street from a park, on a quiet residential street but just half a block to Granville.
Notes	Families are welcome and many families have enjoyed staying at the house. However, we have not child proofed the house. Please ask any safety questions you may have before booking. Group BBQs and family dinners are allowed, and if you are part of a wedding party we do allow a rehearsal dinner or similar at the house. However, we do not allow noisy parties or rowdy behavior. It's crucial to maintain a good relationship with the neighbors.
Transit	Less than 15 minutes to downtown or beaches by car. The #10 or the #16 bus takes you directly downtown in less than 30 minutes. Street parking available for free in front of the house.
Access	The whole house is for the exclusive use of your group.
interaction	Please use the Airbnb messaging system to ask for any assistance you may need. A friend will often help us and have access to this account. Please only call in an emergency as a friend will usually be answering the phone. The cleaning company may send someone to assist you if needed, but please plan to be self-reliant.
House rules	- Please remove shoes at entry doors. - Please keep the house clean and put dishes into dishwasher when you leave. Housekeeping is available for an additional charge. - Please be considerate of the neighbors with noise at all times of day. Keep the house quiet after 10 pm.

5.6.2 Tone Analysis for Host Listing

As in the analysis of Sample A (see Section 5.6.2), after removing all fields selected by the host from the dropdown list, we concatenated all fields written by the host to describe the listing, e.g., listing title, listing summary, listing space, listing description, neighbourhood overview, listing transit, access, house rules. Then the description was submitted to IBM Watson Tone Analyser to extract the emotions expressed in the data (see Table 5.5).

Table 0-5: Watson Tone Analysis (Sample B)

Joy	< 0.5	0.5 – 0.75	> 0.75
Fear	< 0.5	0.5 – 0.75	> 0.75

West Coast Style, Deck, Transit: 15 Min Dtwn/Beach. Gorgeous, updated 7-bedroom, 4-bathroom whole house. Cedar ceilings add a cozy, West Coast feel to this spacious home. Across from a park on a quiet street but just two blocks to restaurants and shopping on Granville St. Free street parking with room for up to 4 vehicles. Three level home with spacious deck and backyard, built in 1945 and recently renovated. Garage and street parking available. The main level has new laminate floor, updated kitchen, dining room, living room, two bedrooms, and one full bathroom. The kitchen was renovated less than 10 years ago with stone counters, wood cabinets, tile floor, and stainless-steel appliances. Please note the tile has a few cracks. All dishes, pots, pans etc are provided, so you just bring the food! Eat in the kitchen with an extendable table that seats 4-6. The spacious back deck with BBQ has a door from the kitchen. The dining room is next to the kitchen and has an extendable table that seats up to 10. The living room has a sofa bed (double) and a wood burning fireplace. The bathroom has been renovated with tile floor, stone counters, and tiled shower. Both bedrooms on this level have a queen size bed, nightstands, lamps, and mirror. One room has a closet and one room has a wardrobe. The upstairs is carpeted. Granville street has a fantastic little village between W 63 and W 70. The house is just one block from this walkable shopping district. Many flavors of restaurants, including sushi, Chinese, pizza, Subway and much more! Starbucks is just two blocks from the house at Granville and W 64 and Safeway grocery store and the BC Liquor Store is at W 70. The Marple neighborhood has includes a public library. The house is across the street from a park, on a quiet residential street but just half a block to Granville. Families are welcome and many families have enjoyed staying at the house. However, we have not child proofed the house. Please ask any safety questions you may have before booking. Group BBQs and family dinners are allowed, and if you are part of a wedding party, we do allow a rehearsal dinner or similar at the house. However, we do not allow noisy parties or rowdy behavior. It's crucial to maintain a good relationship with the neighbors. Less than 15 minutes to downtown or beaches by car. The #10 or the #16 bus takes you directly downtown in less than 30 minutes. Street parking available for free in front of the house. The whole house is for the exclusive use of your group. Please use the Airbnb messaging system to ask for any assistance you may need. A friend will often help us and have access to this account. Please only call in an emergency as a friend will usually be answering the phone. The cleaning company may send someone to assist you if needed, but please plan to be self-reliant. - Please remove shoes at entry doors. Please keep the house clean and put dishes into dishwasher when you leave. Housekeeping is available for an additional charge. - Please be considerate of the neighbors with noise at all times of day. Keep the house quiet after 10 pm.

Note: Concatenated text written by the host, each emotion is color coded based on the intinsity of the emotion found in the text

5.6.3 Selected Guest Reviews

Contrary to the case with Sample A, the listing in Sample B attracted many positive reviews. A sample review is provided below.

Guest Feedback
Massive house with upgraded kitchen and bathrooms on the main and upper levels. 4 very large rooms upstairs with high ceilings. The basement area was a little dark with some close quarters and small added bathroom, but there was also another kitchen/living area down there. The keypad entry was super convenient for a big group and Ella was quick to respond to any questions we had. The front/outside is a little unkept and could use some TLC but the inside is very clean and welcoming. The home was a lovely 7 bedrooms. We were in town for a conference, so mostly we were at the home to rest and eat breakfast, but we did take advantage of the surrounding neighborhood and the back porch. Ella was extremely accommodating about a luggage *request* as well, can't recommend this place enough.

5.7 Case Study A: Training the Model (Ashville)

In Case Study A, we used the data collected from Ashville to train the model.

5.7.1 Identifying Hosts' Sentiments in Airbnb Listing

The first step in calculating the sentiment of the hosts was to concatenate all texts written for each listing into one single document. This included the texts written under the following seven columns from the data model: Summary, Description, Space, Notes, Neighbourhood Overview, and Transit. In the next step, we parsed the document into fundamental Parts of Speech (POS tagging). In general, POS tagging tags words in document sentences into structural elements, such as verbs, nouns, adjectives, adverbs, and so forth. Next, we analysed each sentence both in isolation and in conjunction with the remaining sentences. The selections of words and the frequency of occurrence of a given phrase occurring near a set of positive or negative words was used to establish whether the phrase was in general positive or negative. The IBM Watson™ Tone Analyzer was used to analyse the emotional sentiments in the text data.

5.7.2 Classifying host sentiments in Airbnb listings

K-means classifier was used because the data is unlabelled (i.e., data without defined categories or groups). The algorithm classified the host emotional sentiments found in the texts of the listings. The classification process unfolded in the following three steps.

Step 1: In this step, we combined the emotional sentiments into pairs, for example (joy and sadness), (joy and disgust), (joy and anger), and (joy and fear). The combinations resulted in 25 pairs of emotions. After plotting all those pairs together, we obtained the diagrams for all emotions in the host listings (see Figure 6.1).

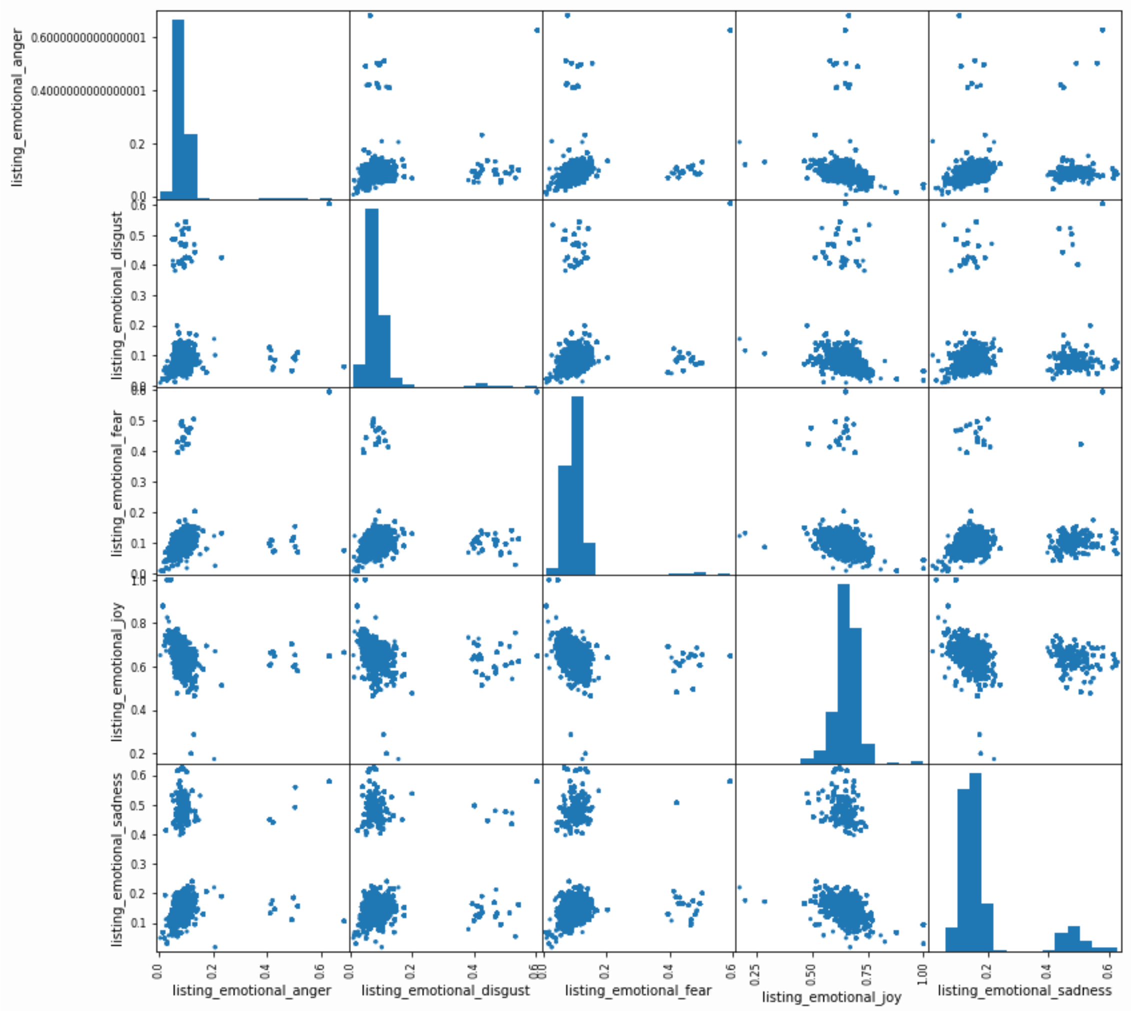


Figure 0.11: Host Emotions in Airbnb Listings (Ashville). Each emotion is paired with each other to form 25 quadrants. The diagonal quadrants form a histogram chart that represents when an emotions was coupled with itself. The histogram shows the average intensity of the emotion in the city.

As can be seen in Figure 6.1, in Ashville, most hosts had dual emotions in their descriptions. Most of the emotion pairs can be classified into two clusters. Therefore, we assumed that each cluster had a central point called centroid. Assuming that the central points of two clusters are c1, c2 with random values, then (see Eq. (5)):

$$\mathrm{C} = [c_1, c_2] \tag{5}$$

where C is the set of all centroids.

The diagonal histogram graphs show the matching emotional pairs—for example, (joy and joy) or (sadness and sadness). The histogram shows the frequency of that emotion and its intensity. As can be seen in Figure 6.1, joy was the most frequent emotion with a high intensity across all host listings, followed by sadness, fear, disgust, and, finally, anger.

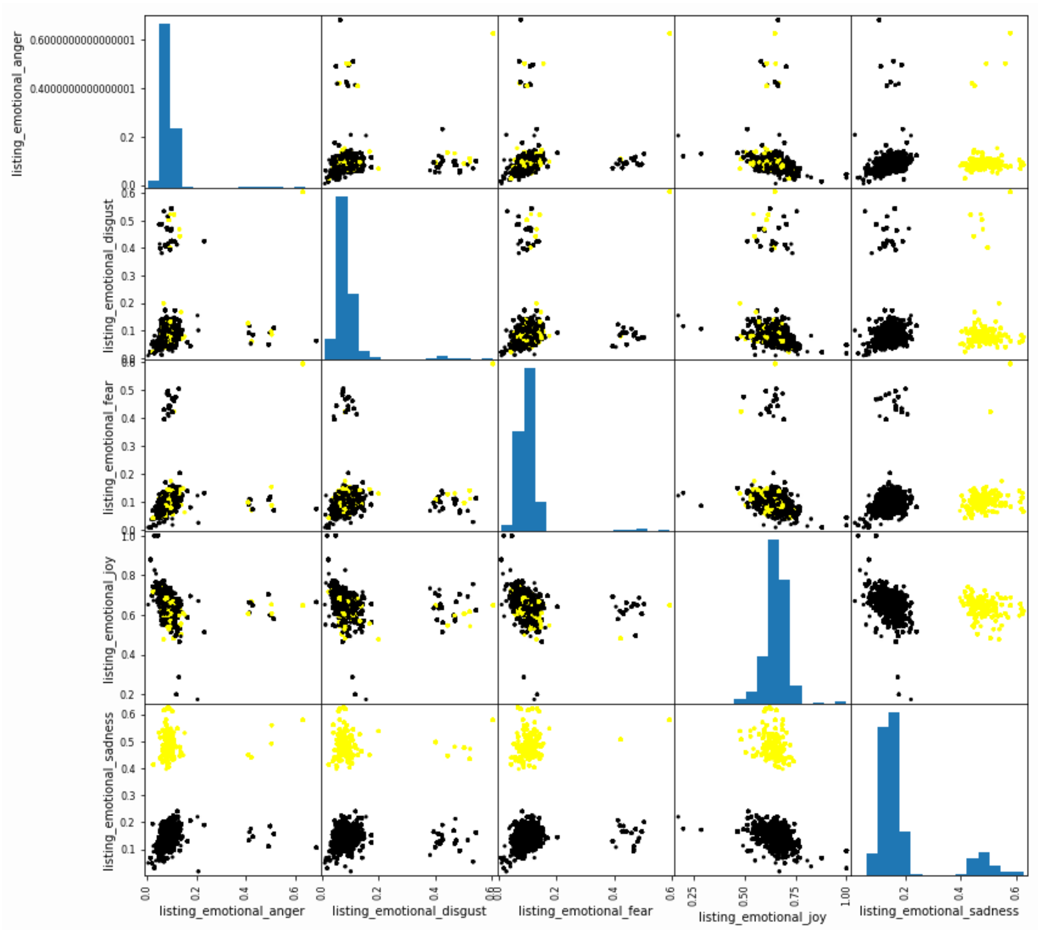


Figure 0.12: K-means Cluster for Hosting Listing Emotion Pair (Joy and Sadness), Ashville

*Step 2***:** In order to classify each host listing, the emotional pair (joy and sadness) was selected as the base of the classification. Using Eq. (6), we calculated the Euclidean distance between each emotional pair to the centroid that was nearest to it.

$$min\ dist(c_i, X)^2 \ (6)$$

where $dist(c_i, X)^2$ is the Euclidean distance, and X is the emotional pair point.

Step 3: Next, after calculating the distance between all emotional pair points with the nearest centroid, we updated the centroid location to best match the centre of all points belonging to it (see Eq. (7)).

$$c_i = \frac{1}{|P_i|} \sum_{X_i \in P_i} X_i \ (7)$$

where P_i is the set of all points assigned to the c_i cluster.

The algorithm was repeated until the clusters assigned to each emotional pair did not change.

5.7.3 Identifying Guest Sentiments in Airbnb Reviews (Ashville)

Joy was the most prominent emotion in all hosts' Airbnb listings in Ashville. Figure 6.3 visualises the relationship between all four possible emotional pairs, on the one hand, and joy, on the other hand. The K-means classifier was used separately on each diagram. The classifier classified each diagram in isolation from other pairs. Each diagram consisted of 27,721 points with transparency equal to half. Each guest review was mapped to its host. Host sentiment was duplicated according to the number of reviews it received. The darker the point shown on the diagram, the more reviews it received.

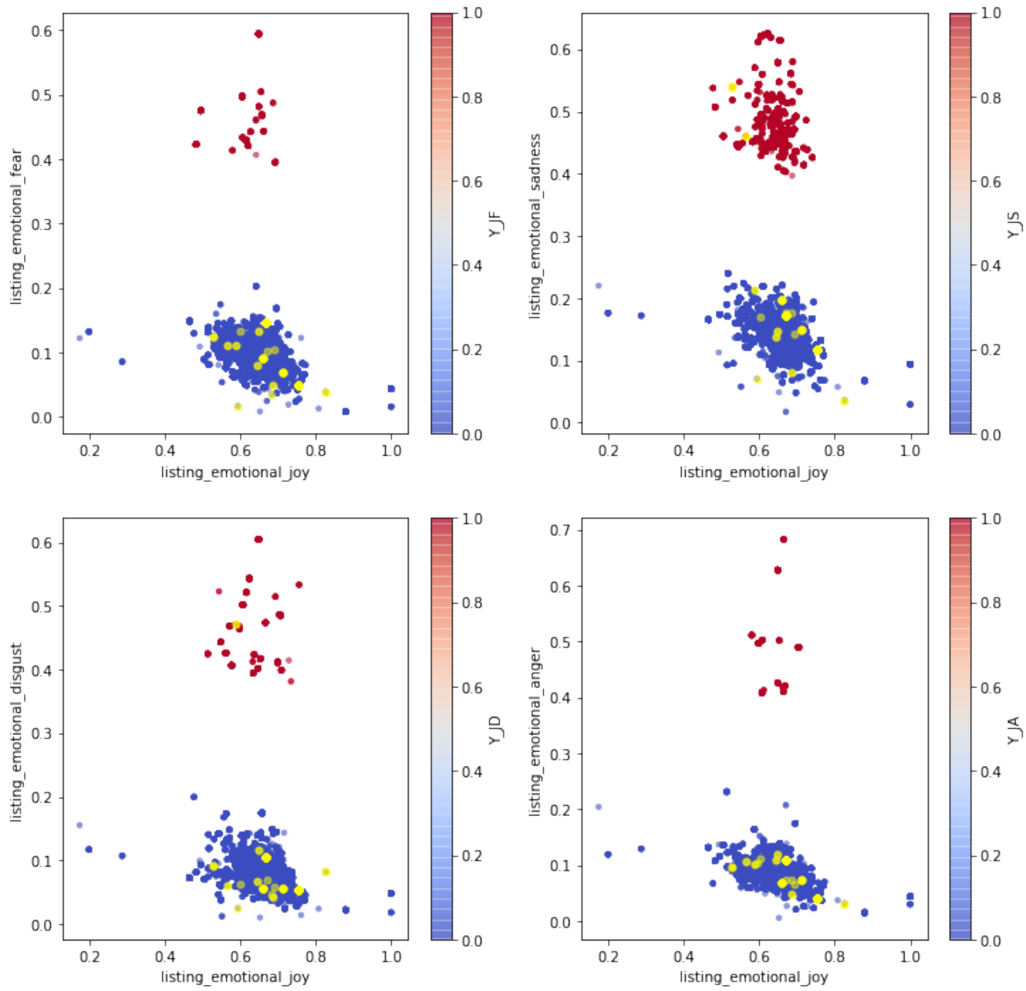


Figure 0.13: Emotional Pairs Combinations with Joy (Ashville). The red and blue points represent host's listings, while the yellow points represent guest negative review provided on top of host listings.

The yellow points in Figure 6.3 represent the guest reviews with listing of four or fewer stars. As demonstrated in several previous studies, Airbnb guests tend to give five stars to hosts more frequently than lower ratings (Fradkin et al., 2015, 2018; Teubner et al., 2017). This tendency, as discussed in Chapter Two, is linked to the fact that guests want the host to reciprocally give them a high rating too, as high ratings on Airbnb help guests to be more readily accepted by future hosts and, therefore, to get better deals. Accordingly, it was assumed the ratings of four or fewer stars to be negative reviews.

Figure 6.3 shows that the first emotional pair (joy and fear) was clearly segmented. The percentage of the yellow points on the red segment is lower than that on the blue segment. Table 6.1 reports the values of reviews and listings in each segment.

Table 0-6: Joy and Fear Segment Reviews Analysis (Ashville)

Class	Listings	Reviews	Reviews/Listing	Negative Reviews	%Negative Reviews
High joy & high fear	19	748	39.3	0	0%
High joy & low fear	723	26,973	37.3	86	100%
Total	742	27,721	37.3	86	100%

The percentage of negative reviews was calculated based on the number of negative reviews for a particular segment over all negative reviews given to all segments. In this case, 0/86 resulted in zero.

In order to confirm that trust can be inferred from Joy and Fear as mentioned by Plutchik's (1986, 2001) Wheel of Emotions, radar charts in Figures 6.4-6.7 show eight basic emotions. As discussed in Section 2.2, Ekman (1972, 1992) did not consider trust to be a basic emotion; however, he agreed with Plutchik (2001) that a combination of two emotions leads to other emotions. At the time when the present study was conducted, the IBM Watson™ Tone Analyzer service was capable of measuring the values of only five emotions from textual content (joy, fear, sadness, disgust, and anger). Therefore, we considered the values of the remaining three emotions (trust, surprise, and anticipation) to be a function derived from the 'neighbour' basic emotions (see Table 2.1 in Section 2.2). Accordingly, the value of the remaining three emotions was obtained by averaging the value of the nearest two emotions specified in Table 2.1 (see Section 2.2). For instance, trust was computed as the average of joy and fear, while anticipation was computed as the average of joy and anger.

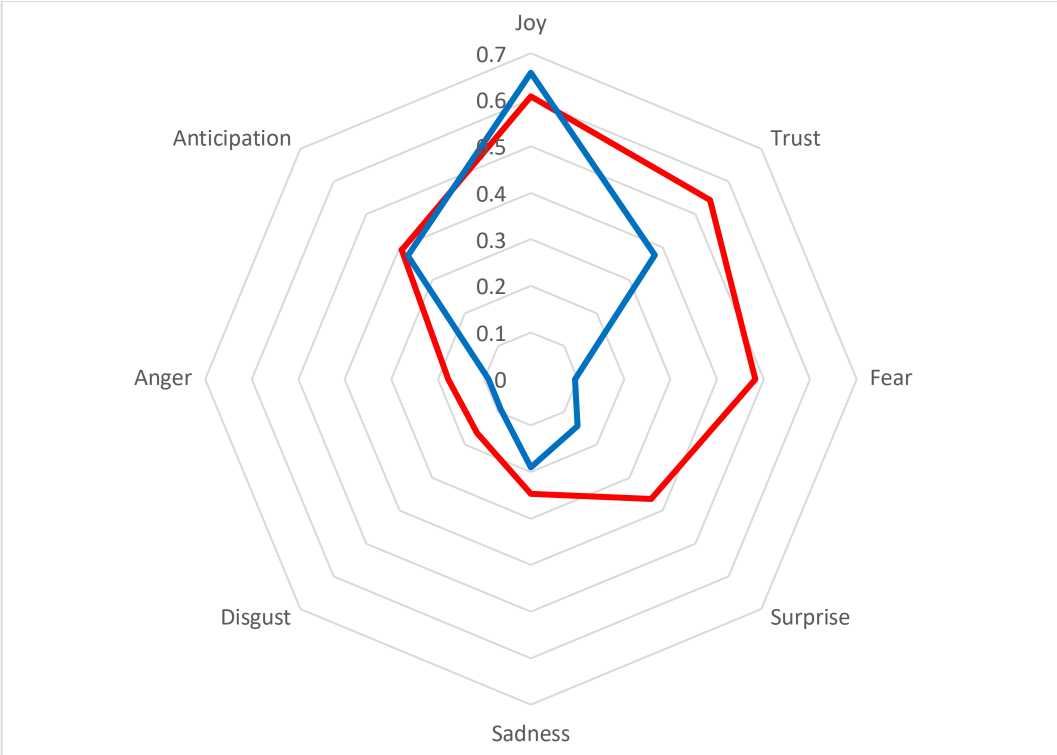


Figure 0.14: Joy and Fear Radar Chart (Ashville)

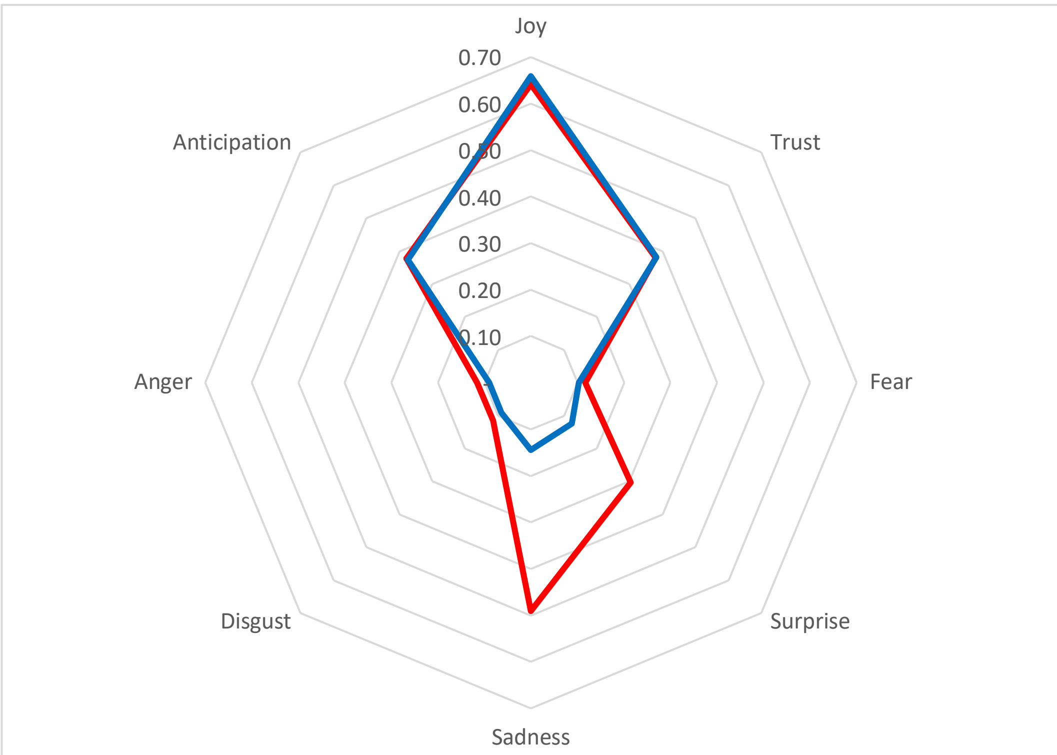


Figure 0.15: Joy and Sadness Radar Chart (Ashville)

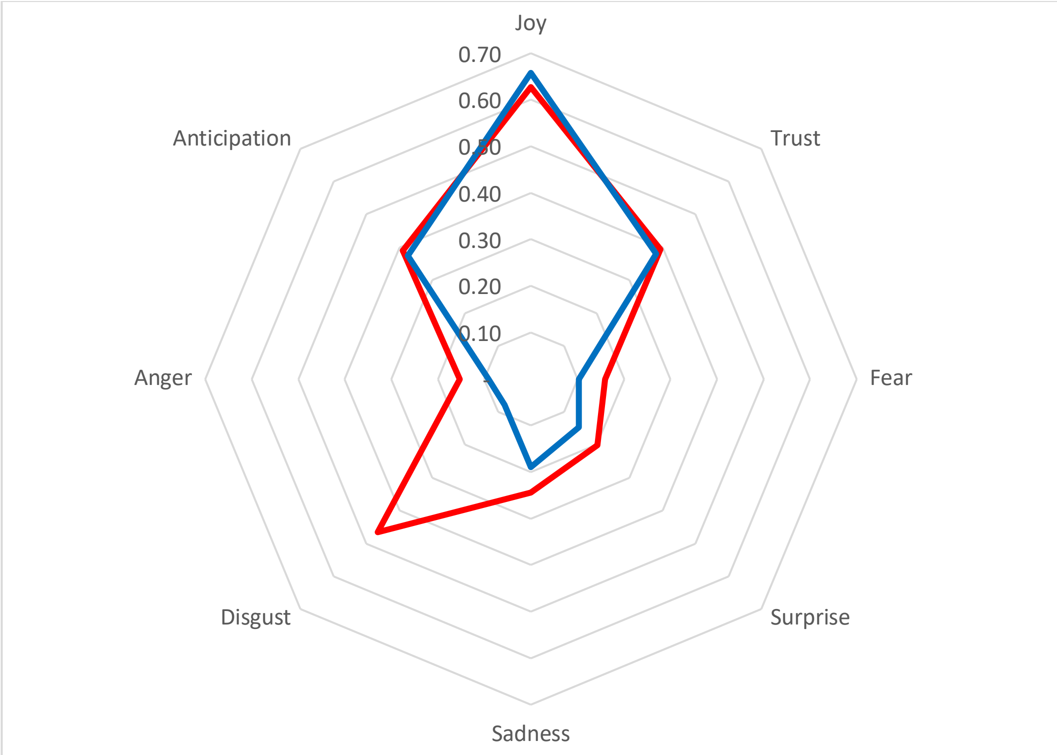


Figure 0.16: Joy and Disgust Radar Chart (Ashville)

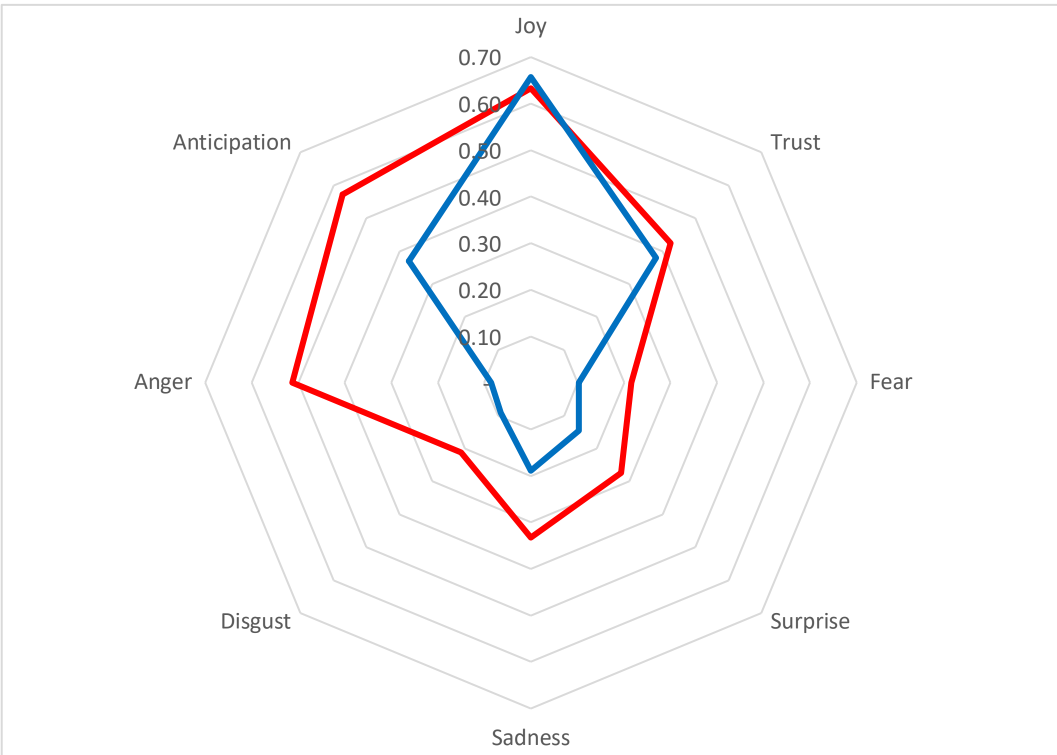


Figure 0.17: Joy and Anger Radar Chart for All Basic Emotions (Ashville)

According to Figures 6.4-6.7, the highest value for trust was in the red cluster in Figure 6.4. This finding is consistent with the results shown Figure 6.3. The blue cluster in all emotional pairs was dominated by joy only. All other emotions appeared to be of a low intensity. The shape of the radar chart for the blue cluster did not change considerably in any of the combinations.

5.8 Case Study B: Evaluating the Model (Manchester)

In Case Study B, we used the data collected from Manchester to evaluate the model.

5.8.1 Identifying Hosts Sentiments in an Airbnb Listing

Sentiments expressed in the listing descriptions from Manchester were analysed using the same approach as the one outlined in Section 6.1 for the Ashville data. The five basic emotions found in the text were used to categorise the listings. As discussed previously in Section 2.2 each emotional pair reveals a more complex emotion. Figure 6.8 shows all combinations of emotional pairs extracted from the Manchester dataset. The diagonal in the figure shows the histogram of the frequency of a single emotion.

5.8.2 Classifying Host Sentiments in Airbnb Listings (Manchester)

As can be seen in Figure 6.8, similarly to the Ashville case, joy was also the most prominent emotion in the Manchester Airbnb listings as well. However, unlike in the Ashville data, sadness level in Manchester was also high. Figure 6.8 shows that most emotion pairs could be classified into three clusters. Figure 6.9 provides further detail on all emotional pairs with respect to joy in the Manchester dataset.

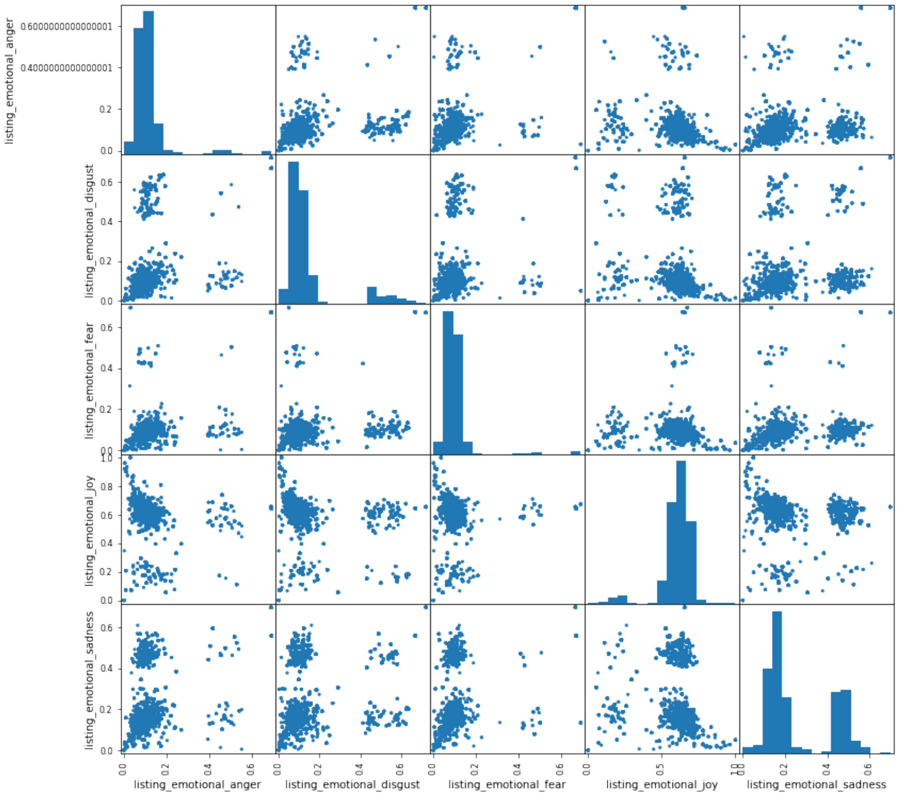


Figure 0.18: Host Emotion Listings (Manchester). Each emotion is paired with each other to form 25 quadrants. The diagonal quadrants form a histogram chart that represents when the emotion coupled with itself. The histogram shows the average intensity of the emotion in the city.

5.8.3 Identifying Guest Sentiments in Airbnb Reviews (Manchester)

As it was demonstrated in Section 6.2.2, that joy was the dominant emotion in all Airbnb host listings from Manchester. Figure 6.9 visualises the relationship between all four possible emotional pairs with joy. The K-means classifier was used separately on each emotional pair. The classifier classified each emotional pair in isolation from other pairs. Each emotional pair diagram comprised 14,880 points with transparency equal to half. Each review was mapped to its host. Host sentiment was duplicated according to the number of reviews it received. The darker the point shown in the diagram, the more reviews it received.

The yellow points represent those guest reviews that evaluated listing accuracy with four or fewer stars. We considered a review with four or fewer stars to be a bad review.

As can be seen in Figure 6.9, the emotional pair (joy and fear) was clearly segmented, and the percentage of the yellow points on the red segment was very low. Table 6.2 provides the values for each emotional pair.

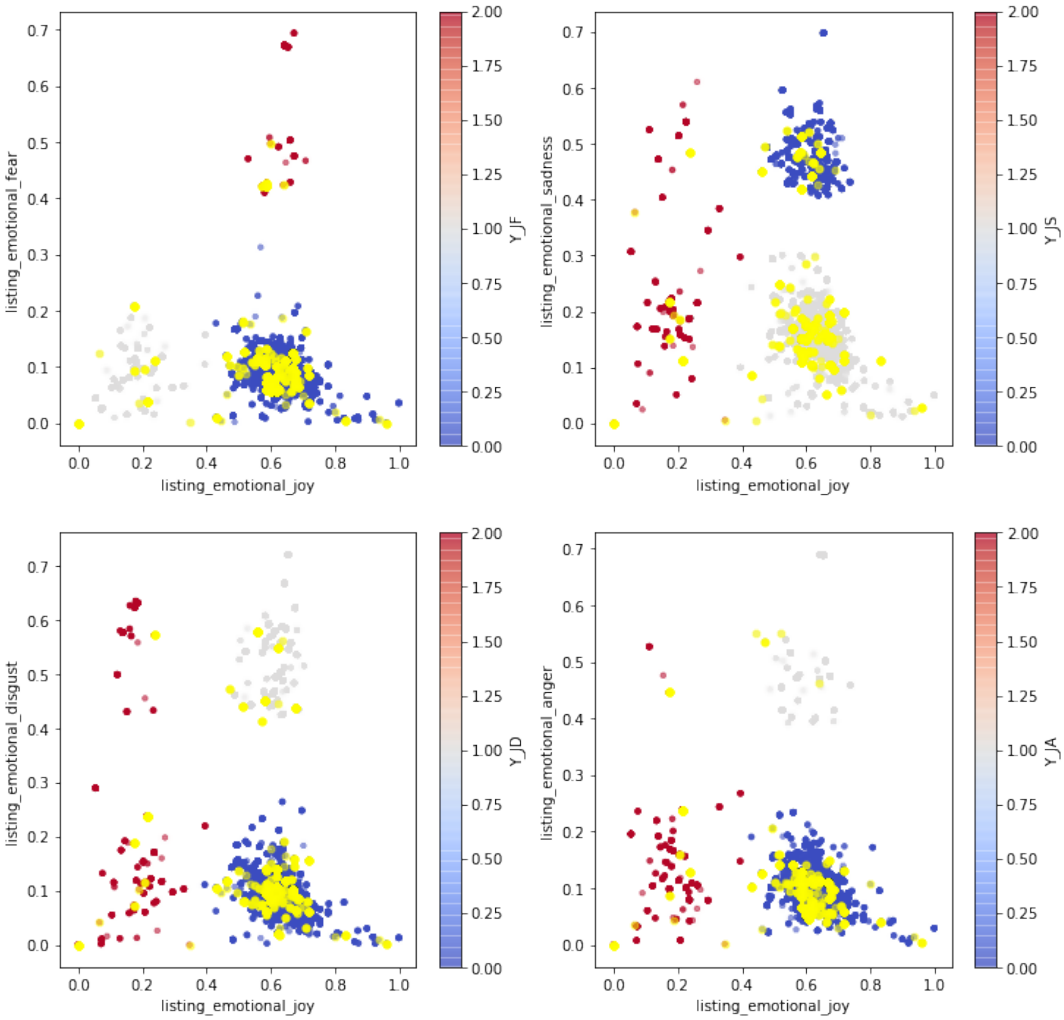


Figure 0.19: All Possible Emotional Pair Combinations with Joy (Manchester). The red, grey, and blue points represent host's listings, while the yellow points represent guest negative review provided on top of host listings.

Table 0-7: Joy and Fear segment Reviews Analysis (Manchester)

Class	Listings	Reviews	Reviews/Listing	Negative Reviews	%Negative Reviews
High joy & high fear	20	278	13.9	36	3.1%
Low joy & low fear	60	748	12.4	77	6.7%
High joy & low fear	596	13854	23.2	1038	90.2%
Total	676	14880	22.0	1151	100%

The percentage of negative reviews was calculated based on the number of negative reviews for a particular segment over all negative reviews given to all segments. In this case, 36/1151 resulted in 3% of all negative reviews given to listings in Manchester.

As can be seen in Table 6.2, the red cluster in the joy and fear emotional pair had high joy and high fear. In order to investigate what other emotions were in the red cluster in this emotional pair, radar charts (Figures 6.10-6.13) were created. The charts show the average of all emotions found per cluster per emotional pair.

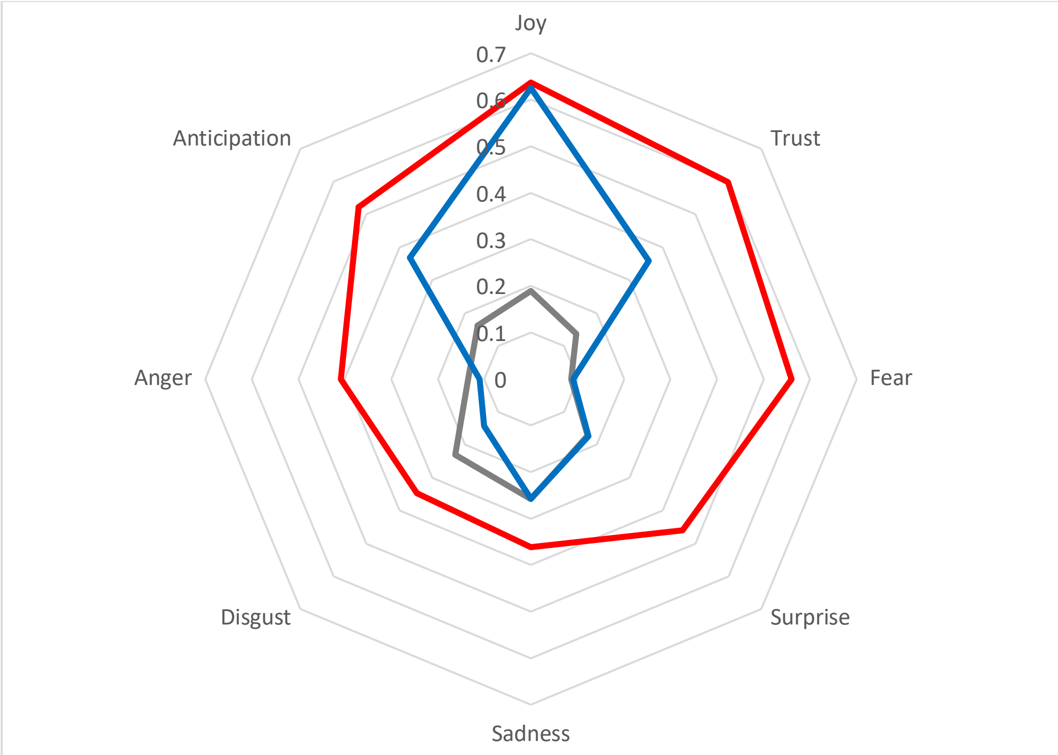


Figure 0.20: Joy and Fear Radar Chart (Manchester)

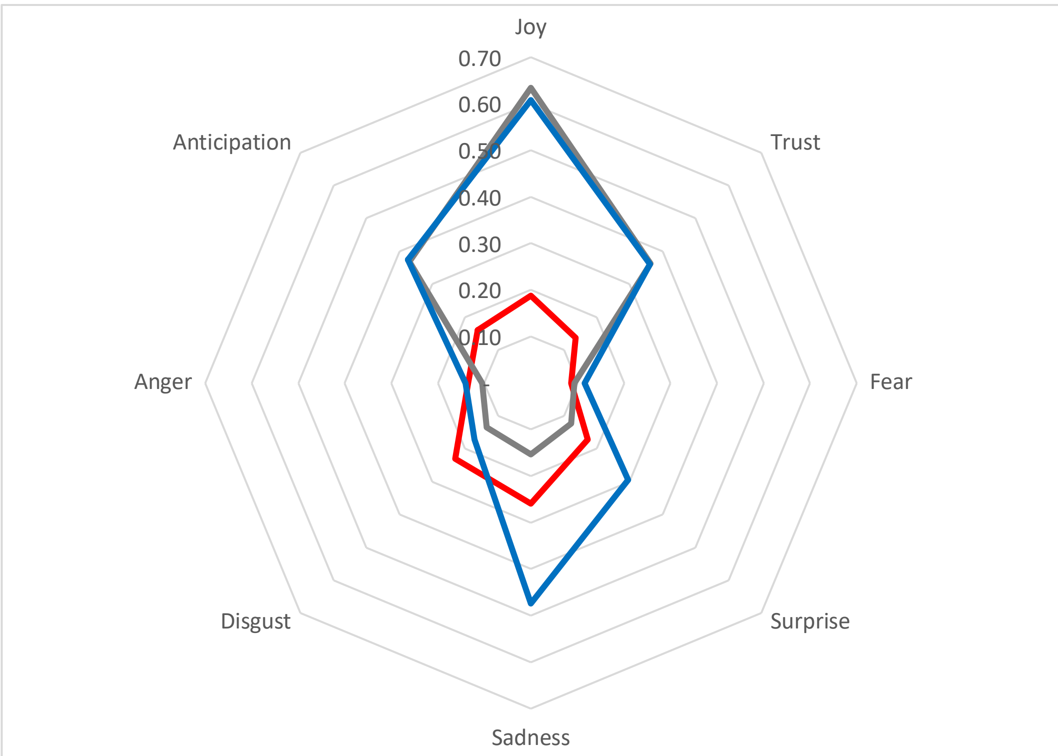


Figure 0.21: Joy and Sadness Radar Chart (Manchester)

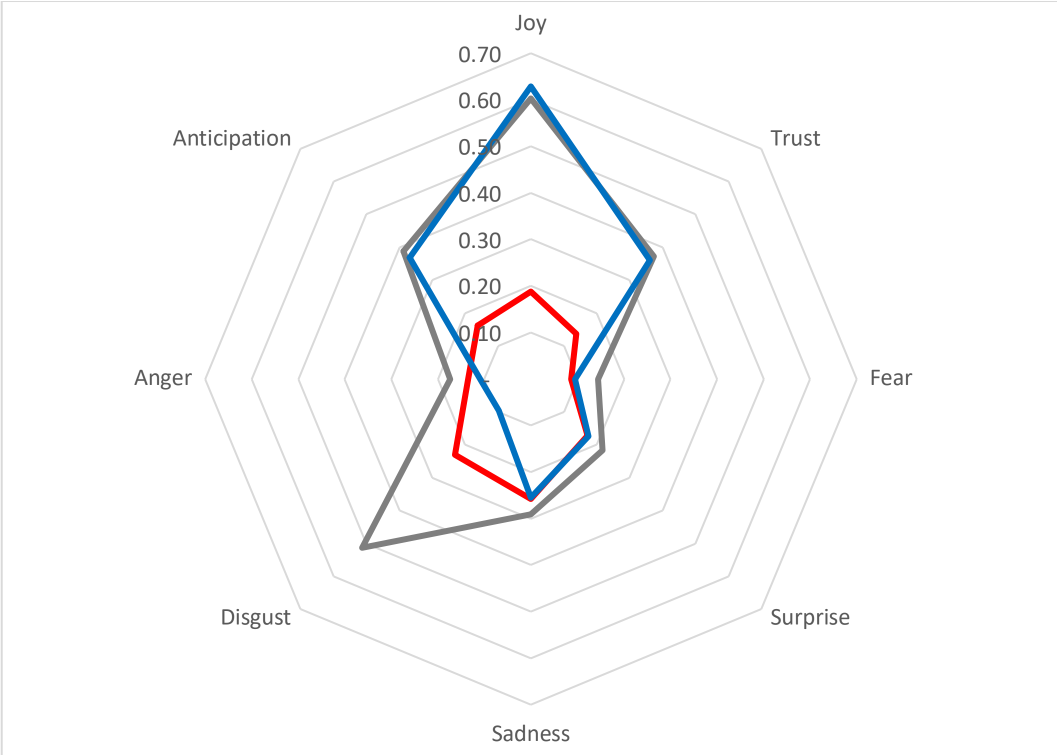


Figure 0.22: Joy and Disgust Radar Chart (Manchester)

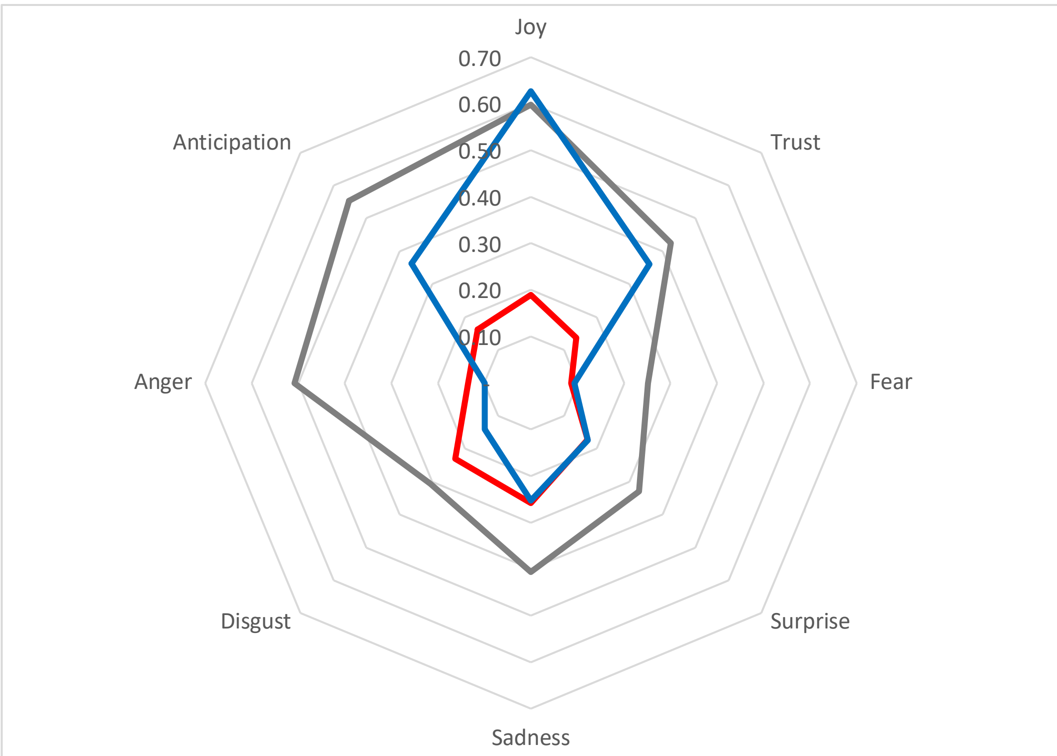


Figure 0.23: Joy and Anger Radar Chart (Manchester)

As it can be seen in Figures 6.10-6.13, the highest value for trust appeared in the red cluster in Figure 6.10. This finding is consistent with the results shown in Figure 6.9. The blue cluster in all emotional pairs was dominated by joy only. All other emotions had a low intensity. The shape of the radar chart for the blue cluster did not change considerably in any combination.

5.9 Case Study C: Evaluating the Model (Vancouver)

Similarly to Case Study B, in Case Study C, we used the data collected from Vancouver to evaluate the model.

5.9.1 Identifying Hosts Sentiments in an Airbnb Listing

Sentiments expressed in the listing descriptions from Vancouver were analysed using the same approach as the one outlined in Sections 6.1 and 6.2 for the Ashville and Manchester datasets, respectively. The five basic emotions found in the text were used to categorise the listings. Figure 6.14 shows all combinations of emotional pairs extracted from the Vancouver dataset. The diagonal in the figure shows the histogram of the frequency of a single emotion.

5.9.2 Classifying host sentiments in Airbnb listings (Vancouver)

According to the results in Figure 6.14, similarly to the pattern observed in Case Studies A and B, joy was the most prominent emotion in the Vancouver Airbnb listings. However, unlike in the Ashville data, and similarly to the Manchester data, sadness level in Vancouver was also high. Figure 6.14 also shows that most emotion pairs could classified into three clusters. Figure 35 provides further detail on all emotional pairs with respect to joy in the Vancouver dataset.

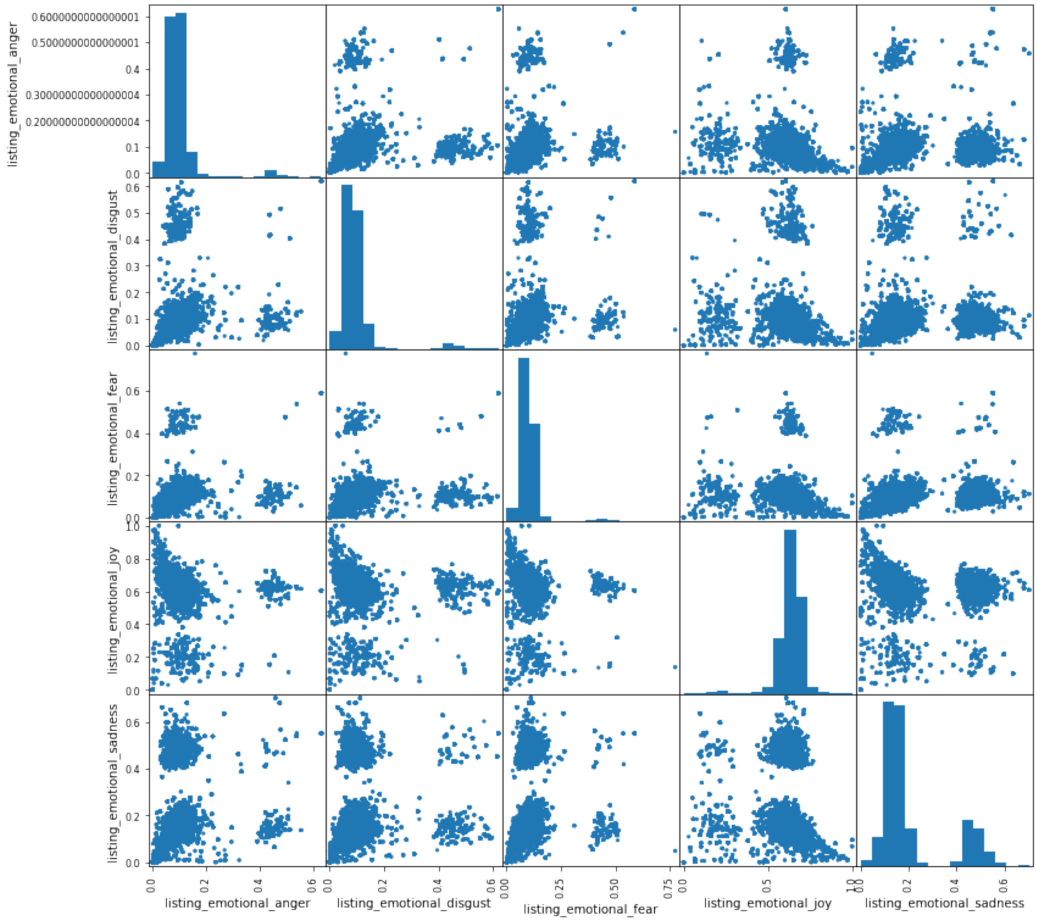


Figure 0.24: Host Emotion Listings (Vancouver). Each emotion is paired with each other to form 25 quadrants. The diagonal quadrants form a histogram chart that represents when the emotion is coupled with itself. The histogram shows the average intensity of the emotion in the city.

5.9.3 Identifying Guest Sentiments in Airbnb reviews (Vancouver)

As discussed above in Section 6.2.2, joy was the dominant emotion in all host sentiments in Vancouver Airbnb listings. Figure 6.15 visualises the relationship between all four possible emotional pairs with joy. The K-means classifier was used separately on each emotional pair. The classifier classified each emotional pair in isolation from other pairs. Each emotional pair diagram comprised 160,138 points with transparency equal to half. Each review was mapped to its host. Host sentiment was duplicated according to the number of reviews it received. The darker the point shown in the diagram, the more reviews it received.

The yellow points represent those guest reviews that evaluated listing accuracy as equal to and fewer than four stars. We considered a review with four or fewer stars to be a bad review.

As can be seen in Figure 6.15, the emotional pair (joy and fear) was clearly segmented, and the percentage of the yellow points on the red segment was very low. Table 6.3 reports the values for each emotional pair.

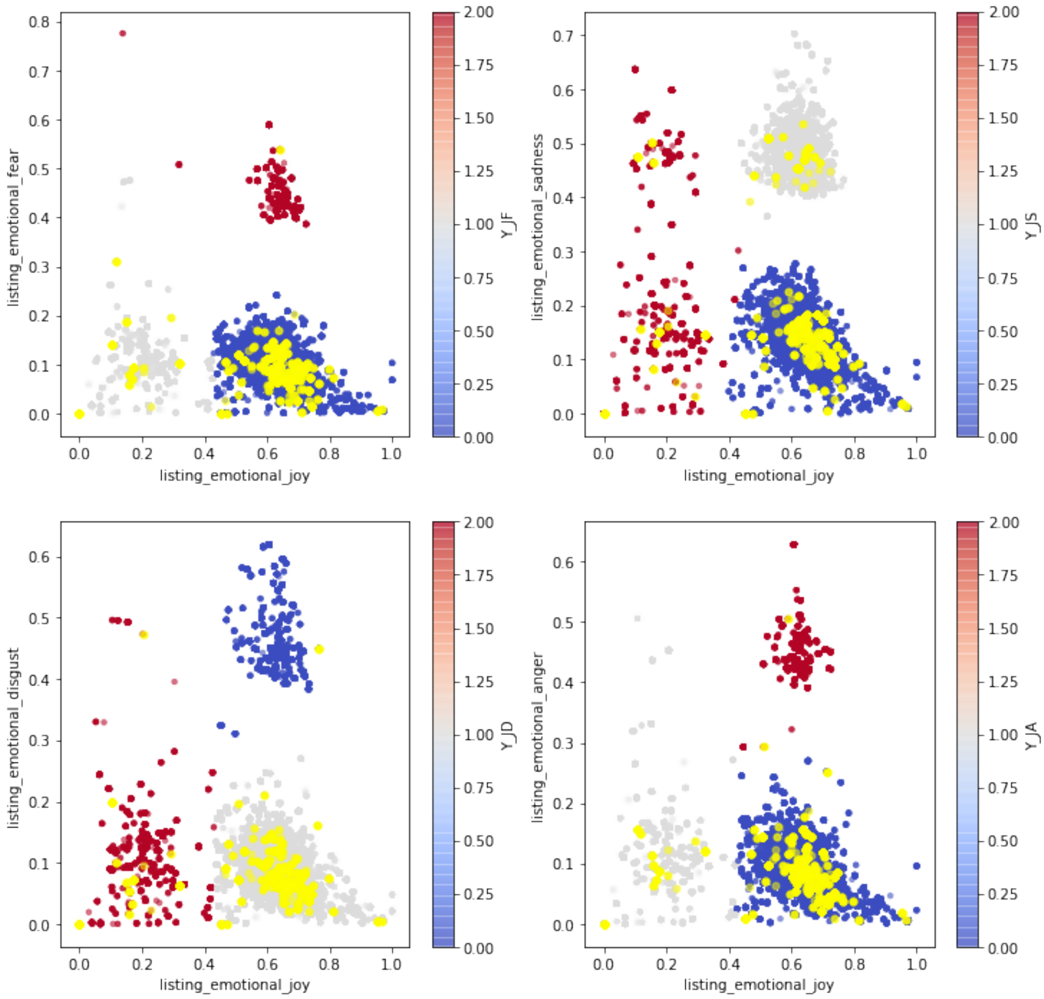


Figure 0.25: All Possible Emotional Pair Combinations with Joy (Vancouver). The red, grey, and blue points represent hostlistings, while the yellow points represent guest negative reviewsprovided on top of host listings.

Table 0-8: Joy and Fear Segment Reviews Analysis (Vancouver)

Class	**Listings**	**Reviews**	**Reviews/Listing**	**Negative Reviews**	**%Negative Reviews**
High joy & high fear (red)	66	3,348	50.7	3	0.2%
Low joy & low fear (Grey)	196	4,264	21.7	92	6.4%
High joy & low fear (Blue)	3,519	152,525	43.3	1,326	93.3%
Total	3,781	160,137	42.3	1,421	100%

The percentage of negative reviews was calculated based on the number of negative reviews for a particular segment over all negative reviews given to all segments. In this case, 3/1421 resulted in 0.2% of all negative reviews given to listings in Vancouver.

As it can be seen in Table 6.3, the red cluster in the joy and fear emotional pair had high joy and high fear. Radar charts (see Figures 6.16-6.19) were created to show the average of all emotions found per cluster per emotional pair.

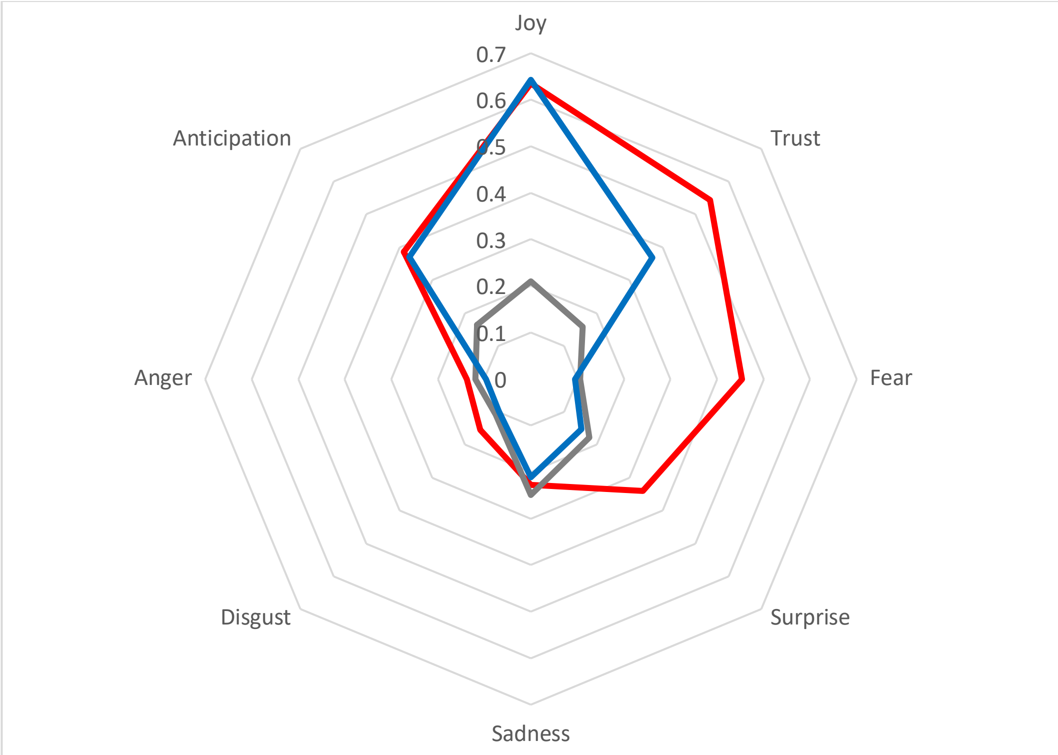


Figure 0.26: Joy and Fear Radar Chart (Vancouver)

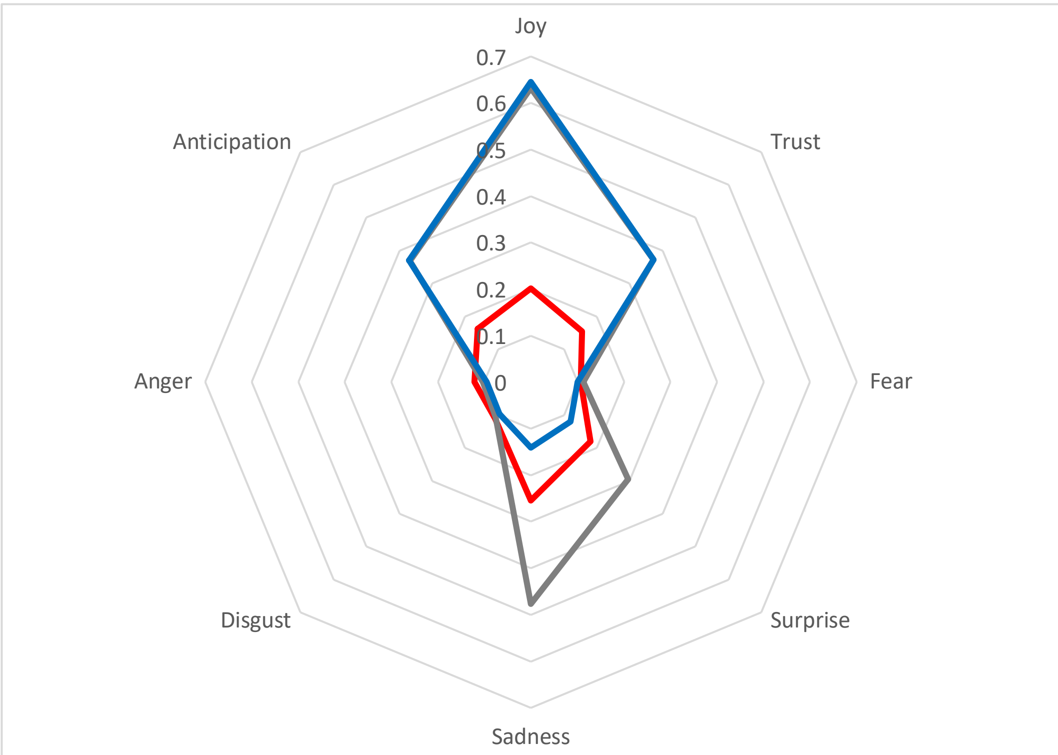


Figure 0.27: Joy and Sadness Radar Chart (Vancouver)

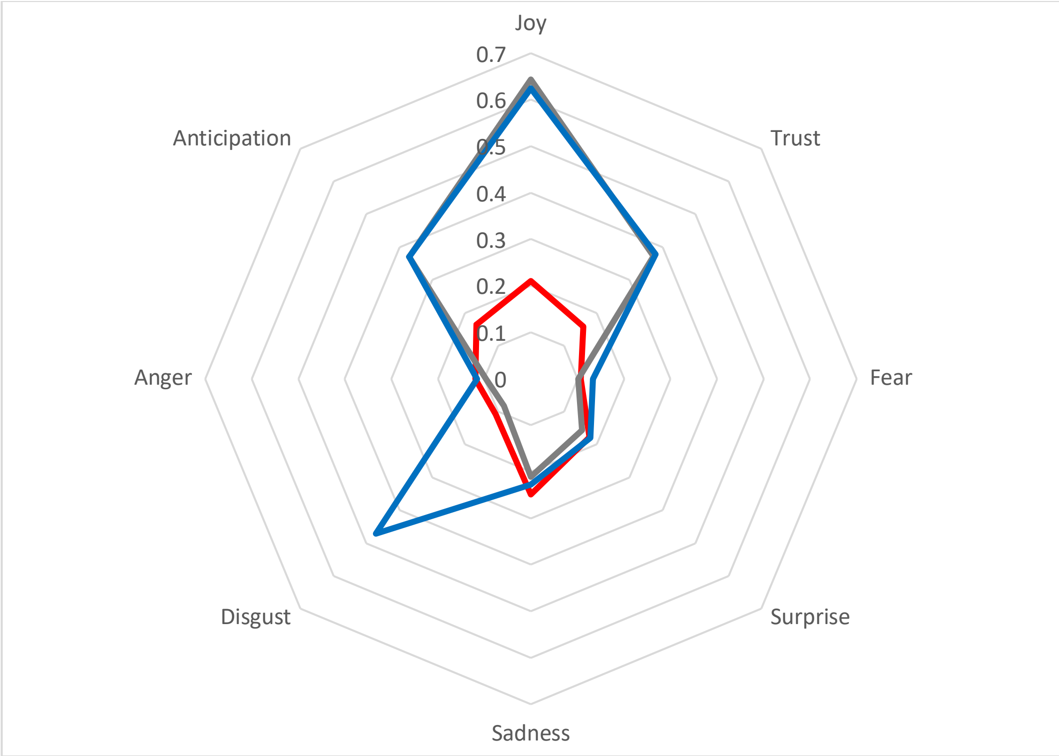


Figure 0.28: Joy and Disgust Radar Chart (Vancouver)

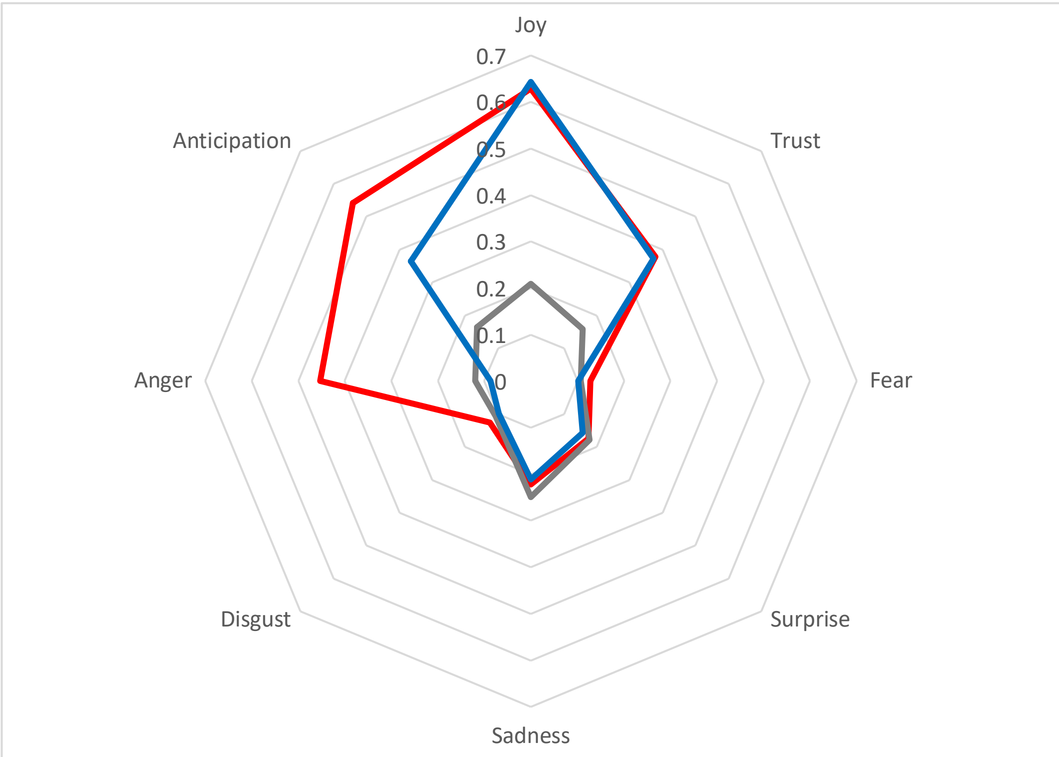


Figure 0.29: Joy and Anger Radar Chart (Vancouver)

The highest value for trust appeared in the red cluster in Figure 6.16. This finding is in line with the results reported in Figure 6.15. The blue cluster in all emotional pairs was dominated by joy only. All other emotions had a low intensity. The shape of the radar chart for the blue cluster did not change considerably in any combination.

5.10 Case Study D: Evaluating the Model (Boston)

In Case Study D, we used the data collected from Boston to evaluate the model.

5.10.1 Identifying Host Sentiments in an Airbnb Listing

Sentiments expressed in the listing descriptions from Boston were analysed using the same approach as in Cases A-C outlined in Section 6.1-6.3 for the Ashville, Manchester, and Vancouver data. The five basic emotions found in the text were used to categorise the listings. Figure 6.20 shows all combinations of emotional pairs extracted from the Boston dataset. The diagonal in the figure shows the histogram of the frequency of a single emotion.

5.10.2 Classifying Host Sentiments in Airbnb Listings (Boston)

Figure 6.20 shows that, similarly to the Ashville, Manchester, and Vancouver data, joy was the most prominent emotion in the Boston Airbnb listings as well. However, unlike in the Ashville data, sadness level in Boston was also high. Figure 40 also shows that most emotion pairs could classified into three clusters. Figure 6.21 provides further detail on all emotional pairs with respect to joy in the Boston dataset.

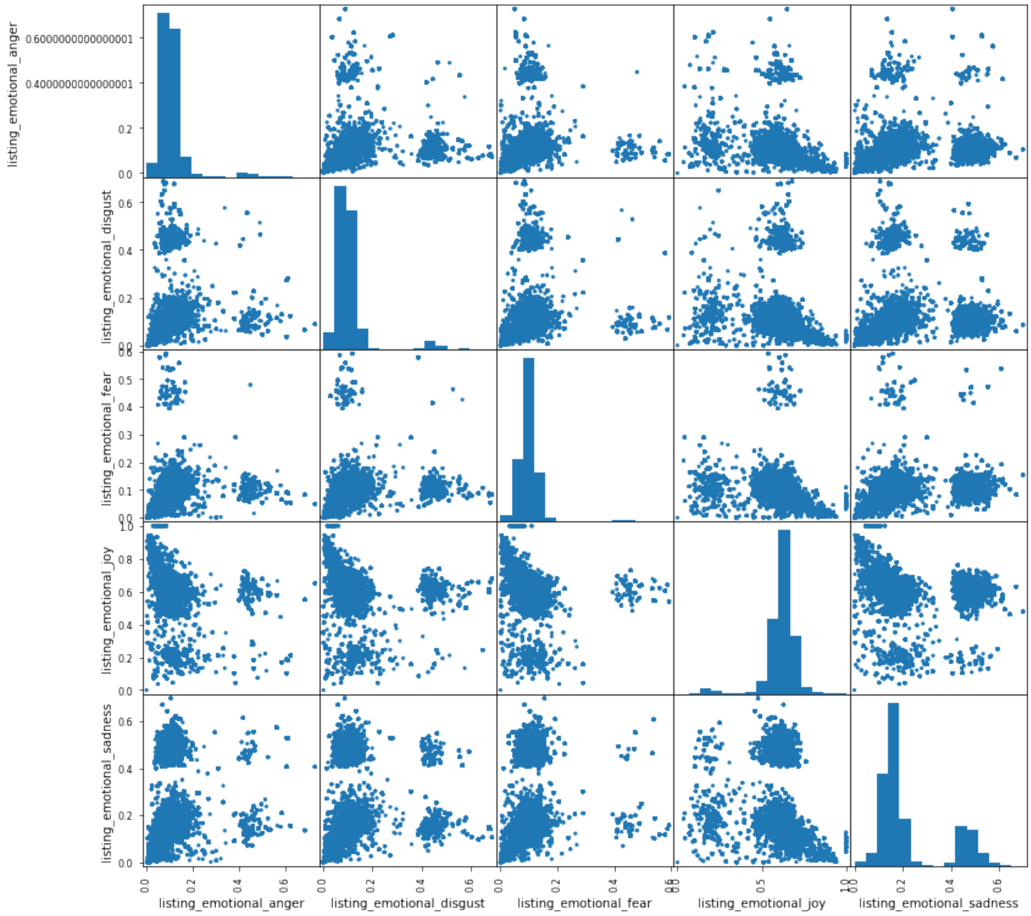


Figure 0.30: Host Emotion Listings (Boston). Each emotion is paired with each other to form 25 quadrants. The diagonal quadrants form a histogram chart that represents when the emotion is with itself. The histogram shows the average intensity of the emotion in the city.

5.10.3 Identifying Guest Sentiments in Airbnb Reviews (Boston)

Considering that, as demonstrated in Section 6.4.2, joy was the dominant emotion in all host sentiments in Boston Airbnb listings, Figure 6.21 visualises the relationship between all four possible emotional pairs with joy. The K-means classifier was used on each emotional pair separately. The classifier classified each emotional pair in isolation from other pairs. Each emotional pair diagram comprised 120,737 points with transparency equal to half. Each review was mapped to its host. Host sentiment was duplicated according to the number of reviews it received. The darker the point shown in the diagram, the more reviews it received.

The yellow points represent those guest reviews that evaluated listing accuracy as equal to or fewer than four stars. We considered four or fewer stars as a bad review.

As it can be seen in Figure 6.21, the emotional pair (joy and fear) was clearly segmented, and the percentage of the yellow points on the red segment was very low. Table 6.4 provides the values for each emotional pair.

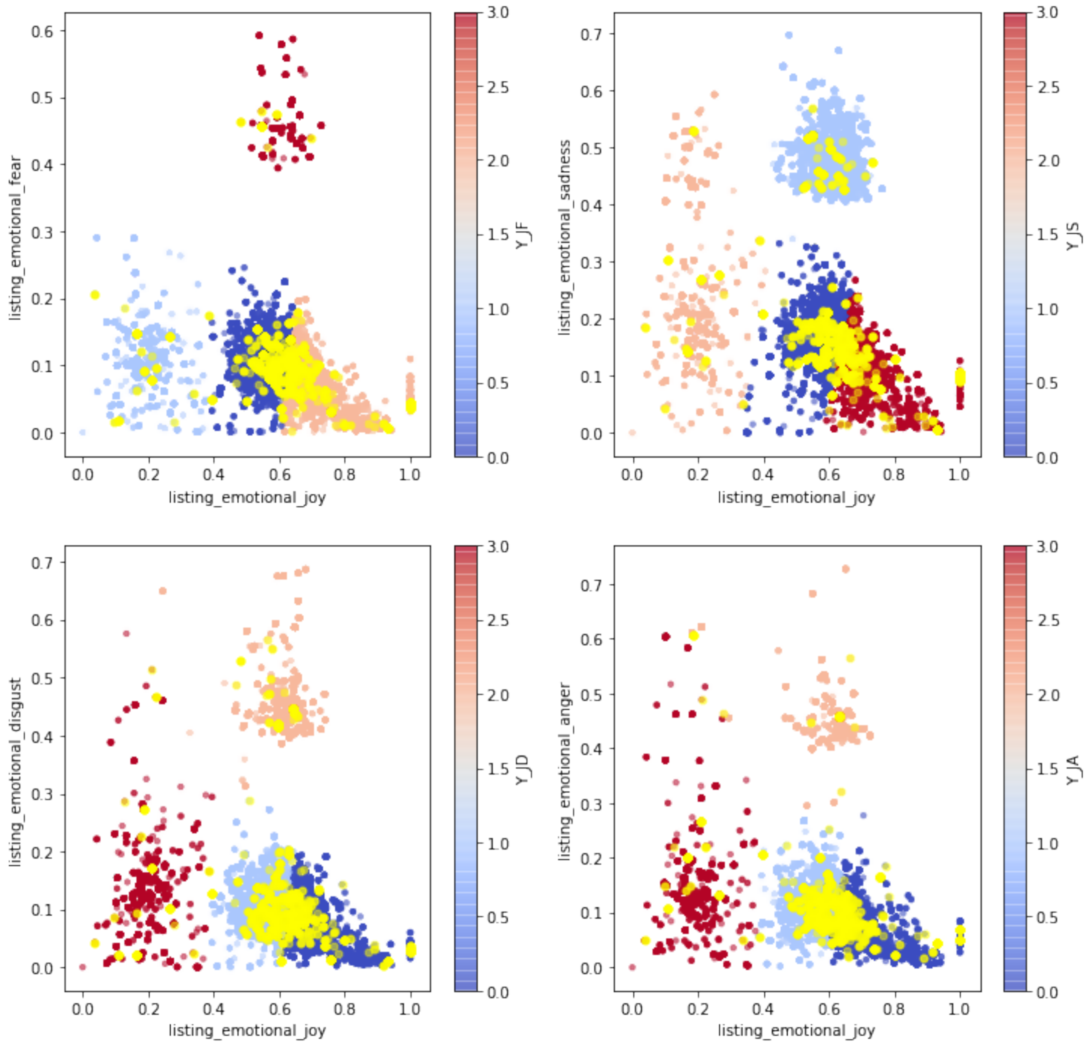


Figure 0.31: All Possible Emotional Pair Combinations with Joy (Boston). The red, skin, light blue, and dark blue points represent host's listings, while the yellow points represent guest negative review provided on top of host listings.

Table 0-9: Joy and Fear Segment Reviews Analysis (Boston)

Class	Listings	Reviews	Reviews/Listing	Negative Reviews	%Negative Reviews
High joy & high fear	51	1713	33.5	33	1.4%
Low joy & low fear	209	4485	41.45	95	4%
High joy & low fear	3,726	114,589	30.75	2,170	94.4%
Total	3,986	120,787	30.30	2298	100%

The percentage of negative reviews was calculated based on the number of negative reviews for a particular segment over all negative reviews given to all segments. In this case, 33/2298 resulted in 1.4% of all negative reviews given across Manchester.

As can be seen in Table 6.4, the red cluster in the joy and fear emotional pair had high joy and high fear. Radar charts (Figures 6.22-6.25) were created to show the average of all emotions found per cluster per emotional pair.

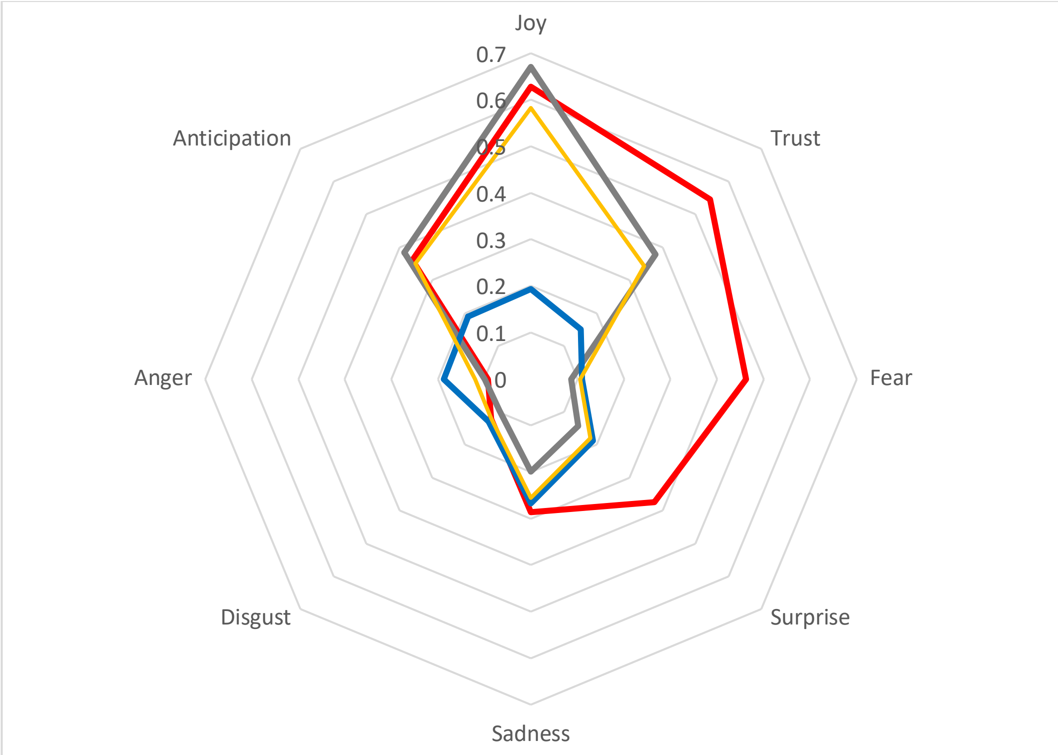


Figure 0.32: Joy and Fear Radar Chart (Boston)

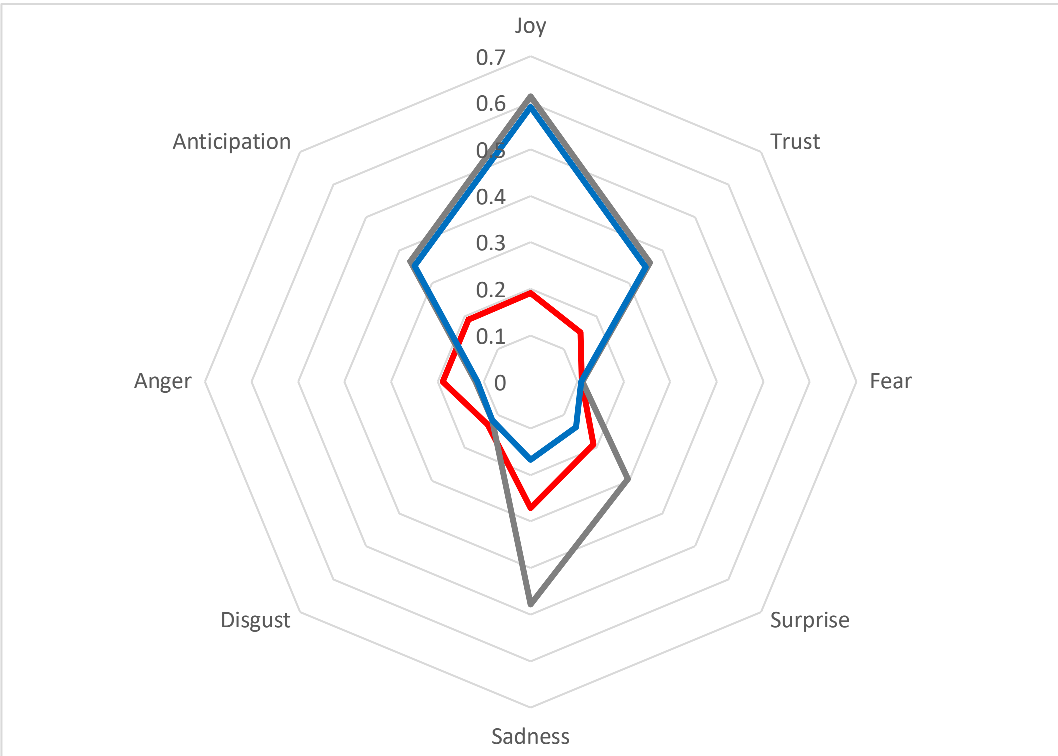


Figure 0.33: Joy and Sadness Radar Chart (Boston)

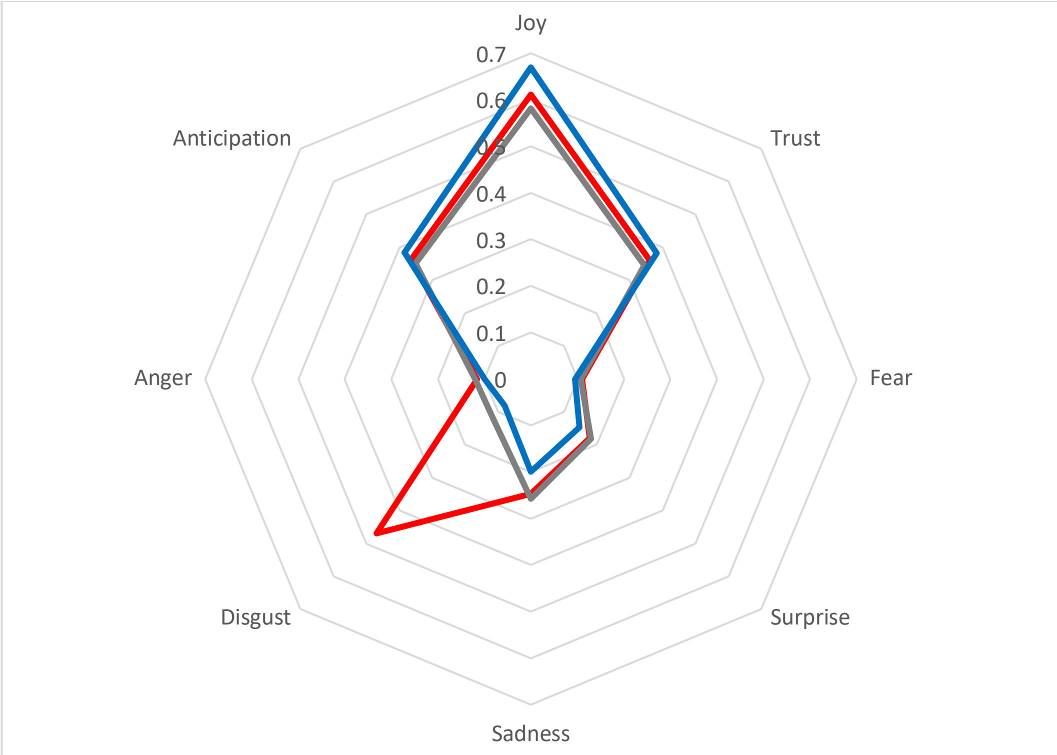


Figure 0.34: Joy and Disgust Radar Chart (Boston)

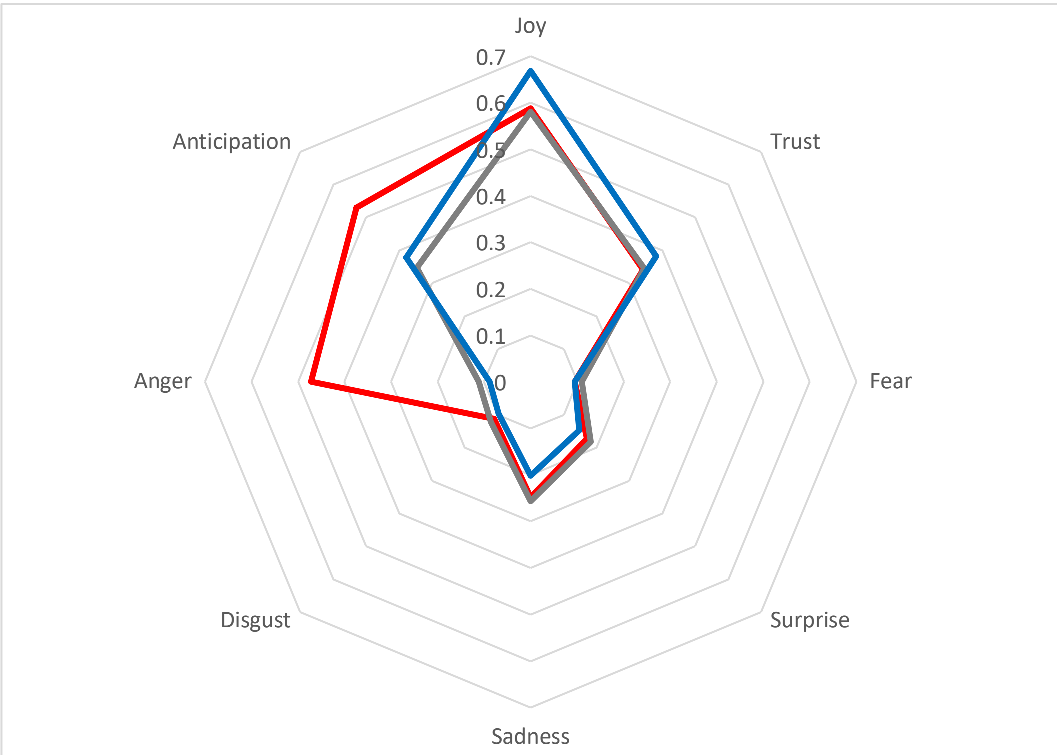


Figure 0.35: Joy and Anger Radar Chart (Boston)

The highest value for trust appeared in the red cluster in Figure 6.22, and this finding is congruent with the results in Figure 6.21. The blue cluster in all emotional pairs was dominated by joy only. All other emotions had a low intensity. The shape of the radar chart for the blue cluster did not change considerably in any combination.

5.11 Summary

In this chapter, two host listings examples were raised to demonstrate the trusted host and untrusted host style of writing. An interesting finding suggested that Guest reviews tend to be more negative when the emotion found in the property description text is dominated by Joy as a single emotion. On the other hand, when the text contains Joy and Fear as mixed emotion, guest negative reviews tend to reduce to the minimum.

After showing the two examples, four case studies showed that Joy is the dominating emotion across all hosts on Airbnb. This is normal since all hosts are trying to show the best out of their property. However, when the host emotion is mixed with another emotion like (fear, anger, sadness, disgust) the guest negative review differs significantly. Negative guest reviews tend to be the lowest when the host emotion is a mix between Joy and Fear. In other words, guests' expectations where met as per the text presented on Airbnb.

CHAPTER SIX: DIMENTIONS REDUCTION

Following to the previous chapter, when the host emotion is mixed with another emotion like (joy, fear, anger, sadness, disgust) the guest negative review differs significantly. In particular, when the host uses Joy and Fear in the listing description, guest negative reviews are the minimum. In this chapter, another approach was followed to validate if mixing all 5 emotions would give similar results.

This approach assumes that all five emotions are five dimensions to explain the host real intentions while writing the description of the Airbnb listing. We used Principal component analysis (PCA) which is a Linear dimensionality reduction to project it to a lower dimensional space. We reduced the five dimensions into three dimensions and then two dimensions.

6.1 Case Study E: Ashville USA

Figure 46 represent the Host Emotions in Airbnb Listings (Ashville) – reduced using Principal Component Analysis, from five dimensions (i.e. joy, anger, fear, disgust, and sadness) into three dimensions (three emotions). After the reductions, the three emotions can't be humanly named. On top of the 3D figure, the red circles represent the listings that received negative Guest reviews less than 4 stars. It might not be visible on figure 46, therefore figure 47 shows only those red circles.

From the figure 6.1, there is no clear segment of hosts with minimal number of negative reviews. However, our next attempt was to reduce the dimension into two dimensions and test again.

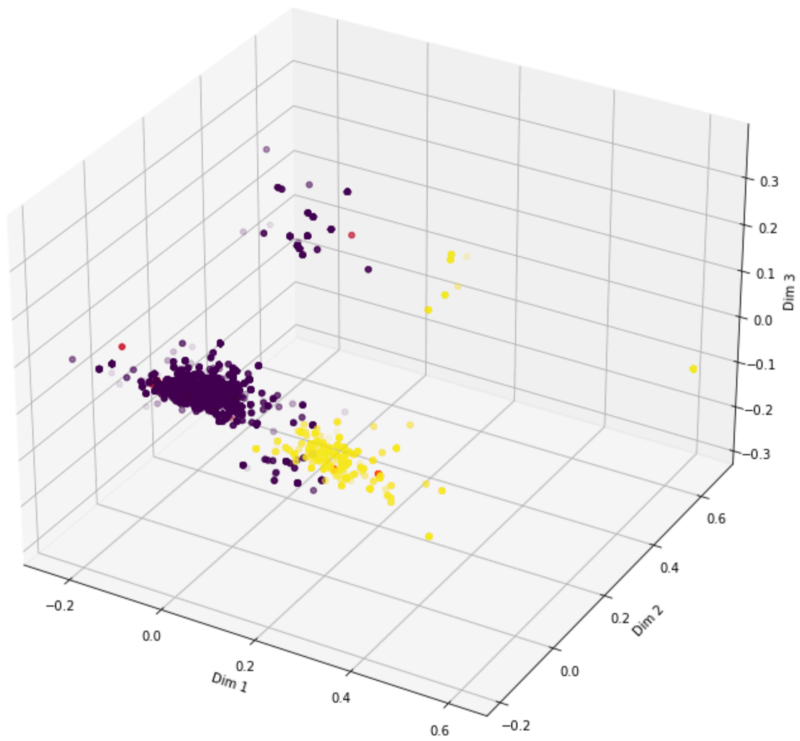


Figure 6.1: Host Emotions in Airbnb Listings (Ashville) – reduced using Principal Component Analysis, from 5 dimensions (emotions) into 3 dimensions (emotions)

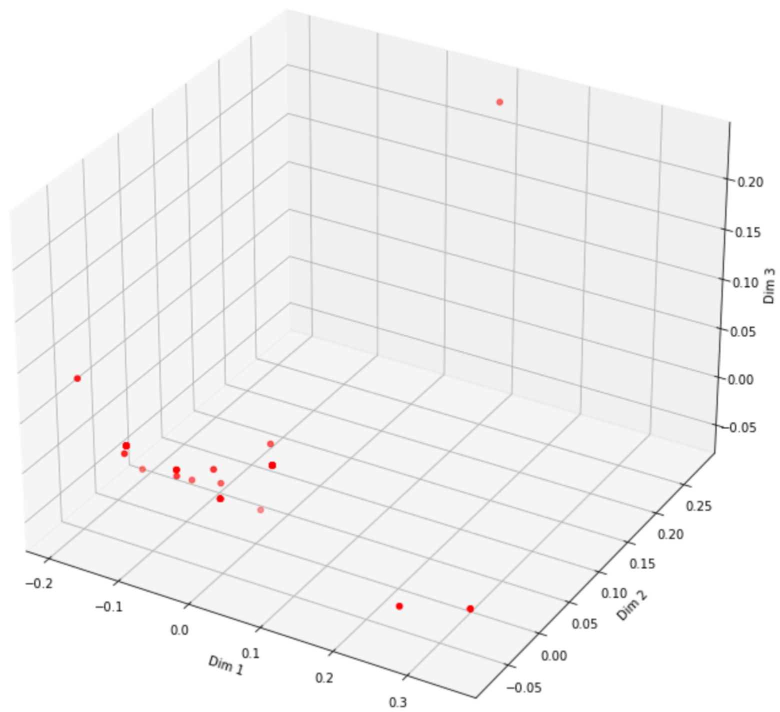


Figure 6.2: Listings with Guest negative review (4stars or less)

Similarly, Figure 48 represent the Host Emotions in Airbnb Listings (Ashville) – reduced using Principal Component Analysis, from five dimensions (i.e. joy, anger, fear, disgust, and sadness) into two dimensions. As mentioned previously, after the reductions, the two dimensions can't be humanly named.

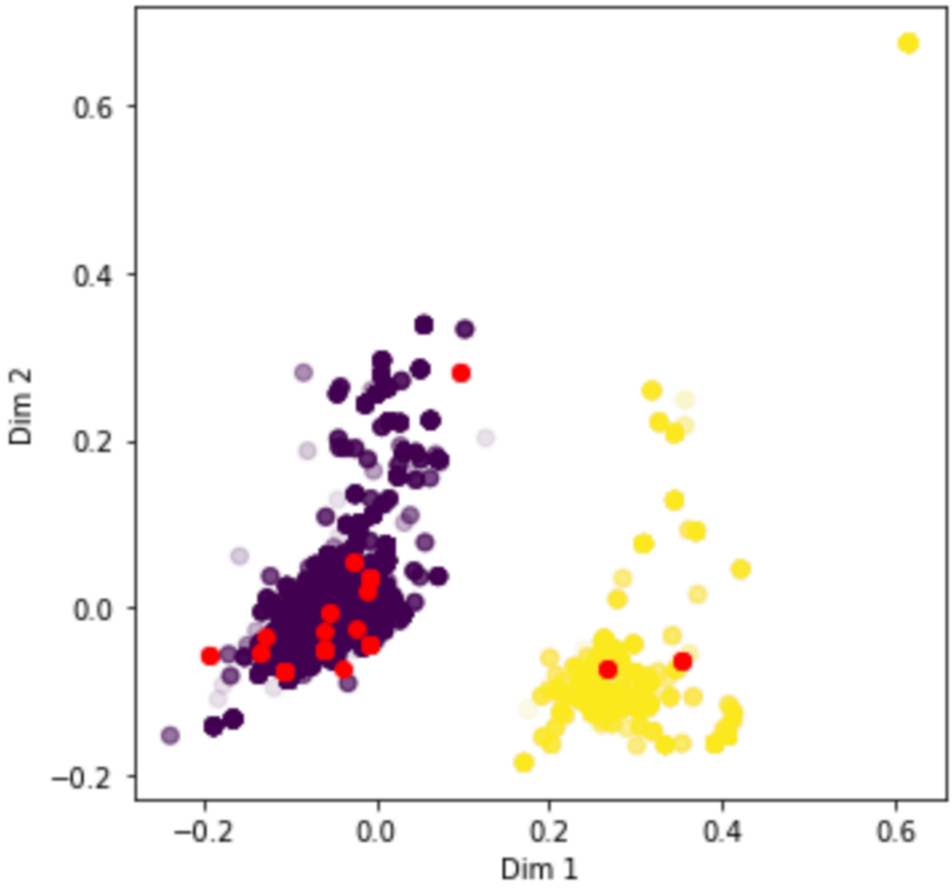


Figure 6.3: Host Emotions in Airbnb Listings (Ashville) – reduced using Principal component analysis, from 5 dimensions (emotions) into 2 Dimensions (emotions)

The red points in Figure 48 represent the guest reviews with listing of four or fewer stars. As demonstrated in several previous studies, Airbnb guests tend to give five stars to hosts more frequently than lower ratings (Fradkin et al., 2015, 2018; Teubner et al., 2017). It was assumed the ratings of four or fewer stars to be negative reviews. Despite the fact that the percentage of the red points on the Yellow segment is lower than red points on the Purple segment, but the results found in Case study A using 2 basic emotions (Joy and Fear) Ashville listing was segmented in a better way compared to current case study (Case study E)

Table 0-1: Dimension reduction Reviews Analysis (Ashville)

Class	Listings	Reviews	Reviews/Listing	Negative Reviews	%Negative Reviews
Yellow	142	3763	26.5	5	5.8%
Purple	600	23,958	39.9	81	94.2%
Total	742	27,721	37.3	86	100%

6.2 Case Study F: Manchester UK

Figure 49 represent the Host Emotions in Airbnb Listings (Manchester) – reduced using Principal Component Analysis, from 5 dimensions (5 emotions) into 3 dimensions (3 emotions). After the reductions, the 3 emotions can't be humanly named. On top of the 3D figure, the red circles represent the listings that received negative Guest reviews less than 4 stars. It might not be visible on figure 49, therefore figure 50 shows only those red circles.

From the figure 6.4, there is no clear segment of hosts with minimal number of negative reviews. However, our next attempt was to reduce the dimension into 2 dimensions and test again.

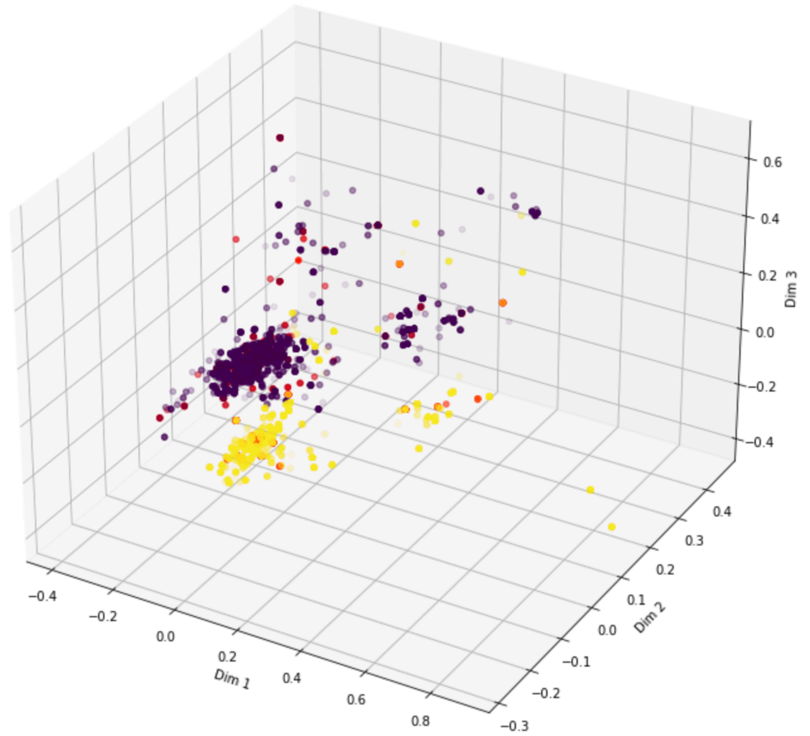


Figure 6.4: Host Emotions in Airbnb Listings (Manchester) – reduced using Principal component analysis, from 5 dimensions (emotions) into 3 Dimensions (emotions)

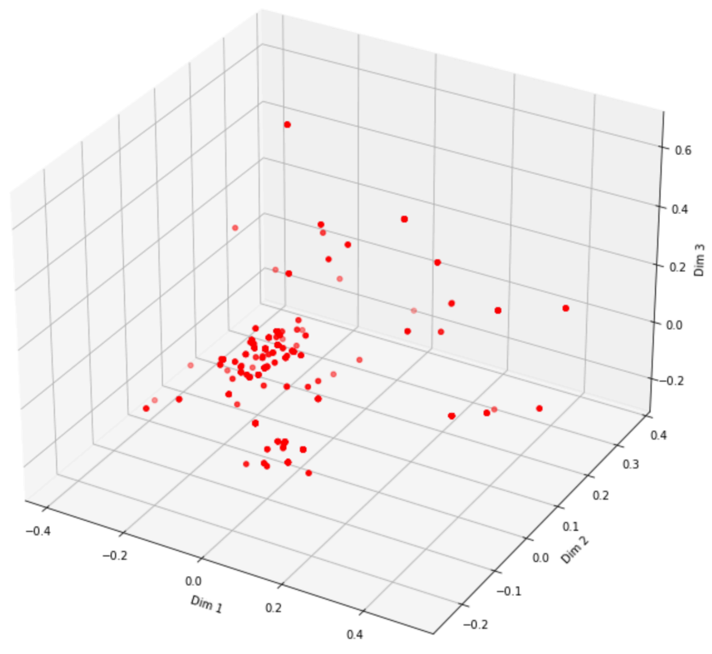


Figure 6.5: Listings with Guest negative review (4stars or less)

Similarly, Figure 51 represent the Host Emotions in Airbnb Listings (Manchester) – reduced using Principal Component Analysis, from 5 dimensions (i.e. joy, anger, fear, disgust, and sadness) into 2 dimensions.

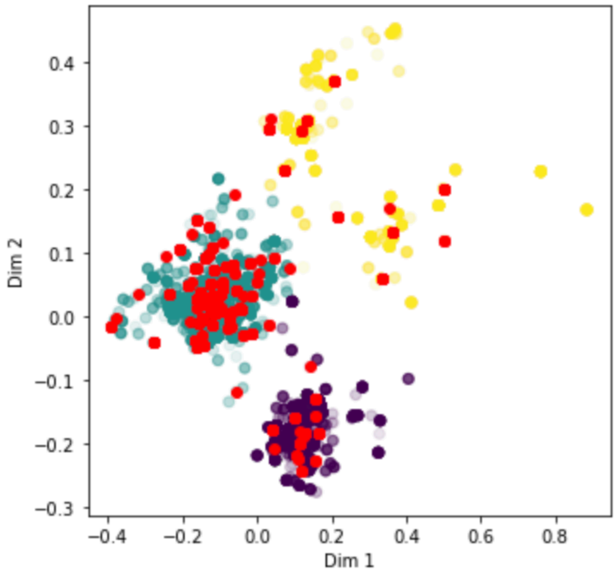


Figure 6.6: Host Emotions in Airbnb Listings (Manchester) – reduced using Principal component analysis, from 5 dimensions (emotions) into 2 Dimensions (emotions)

The red points in Figure 51 represent the guest reviews with listing of four or fewer stars. Despite the fact that the percentage of the red points on the Yellow segment is lower than red points on the Purple and Green segments, but the results found in Case study B using 2 basic emotions (Joy and Fear) Manchester listing was segmented in a better way compared to current case study (Case study F).

Table 0-2: Dimension reduction Reviews Analysis (Manchester)

Class	Listings	Reviews	Reviews/Listing	Negative Reviews	%Negative Reviews
Yellow	87	1,978	22.74	160	13.90%
Green	438	8,714	19.89	859	74.63%
Purple	151	4,188	27.74	132	11.47%
Total	676	14,880	22.00	1151	100%

6.3 Summary

In this chapter, another approach was followed to validate if mixing all 5 emotions would give similar results to mixing Joy and Fear only. This approach assumes that all five emotions are represented as five. We used Principal component analysis (PCA) to reduce the five dimensions into three dimensions and then two dimensions. From the results shown, reducing 5 dimensions (emotions) into 3 dimensions (emotions) for Ashville USA and Manchester UK didn't improve the segmentation of hosts vs guests' feedback. It wasn't possible to find a clear segment of hosts who received the least number of negative feedbacks from guests. However, reducing 5 dimensions (emotions) into 2 dimensions (emotions) showed better results than 3 dimensions, however, it's still not better than the approach in chapter 5 (i.e. using Joy and Fear emotions only)

CHAPTER 7: CONCLUSIONS, AND FUTURE WORK

7.1 Research Summary

Today, almost any individual can make use of C2C marketplaces to offer a product or provide a service, such as sharing a ride or renting out a coach in a living room. In the last decade that has witnessed a revolutionary growth of modern C2C marketplaces, the spectrum of trust has become broader and increasingly complex. However, a prerequisite of any online transaction in C2C marketplaces, such as Uber and Airbnb, is that buyers and sellers must trust each other (Head & Hassanein, 2002). Therefore, modern C2C marketplaces largely depend on trust among their users (Wu & Lin, 2016).

In response to this need, in the present study, we performed text mining and subsequent sentiment analysis of the Airbnb host descriptions of listing and guests reviews to predict the trust level based on the hosts' descriptions of their listed facilities. The data acquired from the Inside Airbnb website on the cities of Ashville in Alabama, the US, Manchester, the UK, Vancouver, Canada, and Boston, the US, were used for the analysis. The results from both cities were highly comparable. After detecting 5 basic of the basic emotions (i.e. joy, anger, fear, disgust, and sadness), in host texts using existing tools, we were able to calculate the trust , which is the 6th basic emotion from text.

The five emotions were combined into pairs to produce 25 pairs. Joy was found to be the dominant emotion in all hosts' sentiments in both cities, followed by sadness and fear. A K-means classifier was used to classify the host emotional sentiments found in the text. Each pair was interesting to study; however, after plotting negative guest reviews on top of all pairs, the emotional pair of joy and fear was decided to be the most interesting classification to measure trust. The results showed that negative guest reviews were higher when the host sentiment in the descriptions was only joy. By contrast, negative guest sentiments were at their minimum when the host sentiment hinted at a mixture of joy and fear.

7.2 Review of the research questions

In this thesis, we aimed to propose an approach to quantify trust as an emotion found in text. We were able to quantify the trust emotion based on seller's written text in the C2C marketplace. To this end, based on Plutchik's (2001) Wheel of Emotions that conceptualises trust as one of eight basic emotions, positioned between joy and fear, we addressed the following two questions:

Research Question 1 (RQ1): Can trust, one of the eight basic emotions, exist in C2C texts, such as Airbnb accommodation descriptions?

Research Question 2 (RQ2): If it exists, can trust be inferred from detecting joy and fear?

The answers to both RQ1 and RQ2 are affirmative. First, trust can be detected in text written by hosts describing their facilities. Second, following the approach demonstrated in the present study, detecting joy and fear in host textual content can serves as foundation to infer trust.

7.3 Conclusions

In conclusion, due to the uncertainty about quality of C2C offerings provided by hosts (Trustors), it is important that C2C marketplaces (e.g., Airbnb) maintain the trust triangle (see Figure 5.6 in Section 5.4) so that to detect a disappointing transaction ahead of time. Therefore, the take-home message of this thesis is that, for better achievement and stronger customer trust, C2C marketplaces should analyse hosts' (Trustors') sentiments in the listing descriptions

(Expectation) before releasing those descriptions to the public (Trustors). In addition, the present study proposes a model that can help C2C hospitality marketplaces to better advise hosts on how to appropriately set guest expectations while describing their facilities. Implementing this model in practice will reduce guests' (Trustors') disappointment with transactions and, subsequently, reduce the number of negative posts published about C2C marketplace in general.

7.4Future Works

Quantifying trust in text was one of the main contributions of this study, however, the study also opened the door for many future development and further researches. This chapter list few of them: If trust in text was inferred from joy and fear, can trust be inferred if the joy and fear emotions were detected in human voice. This should be a research question for a future thesis.

Another future development is to study the emotional pair (Joy and Anger). The charts in this thesis, shows that High Joy and High Anger found in text also attract less negative experience by the guests of Airbnb, however, the reason was not justified. It might be that high joy and high anger pair of emotions infer the (Anticipation) basic emotion. Which leads the guest into anticipating the expectations before it happiness. This yield into a more successful transaction on Airbnb.

APPENDICES

8.1 Appendix A – Unvalidated Airbnb listing

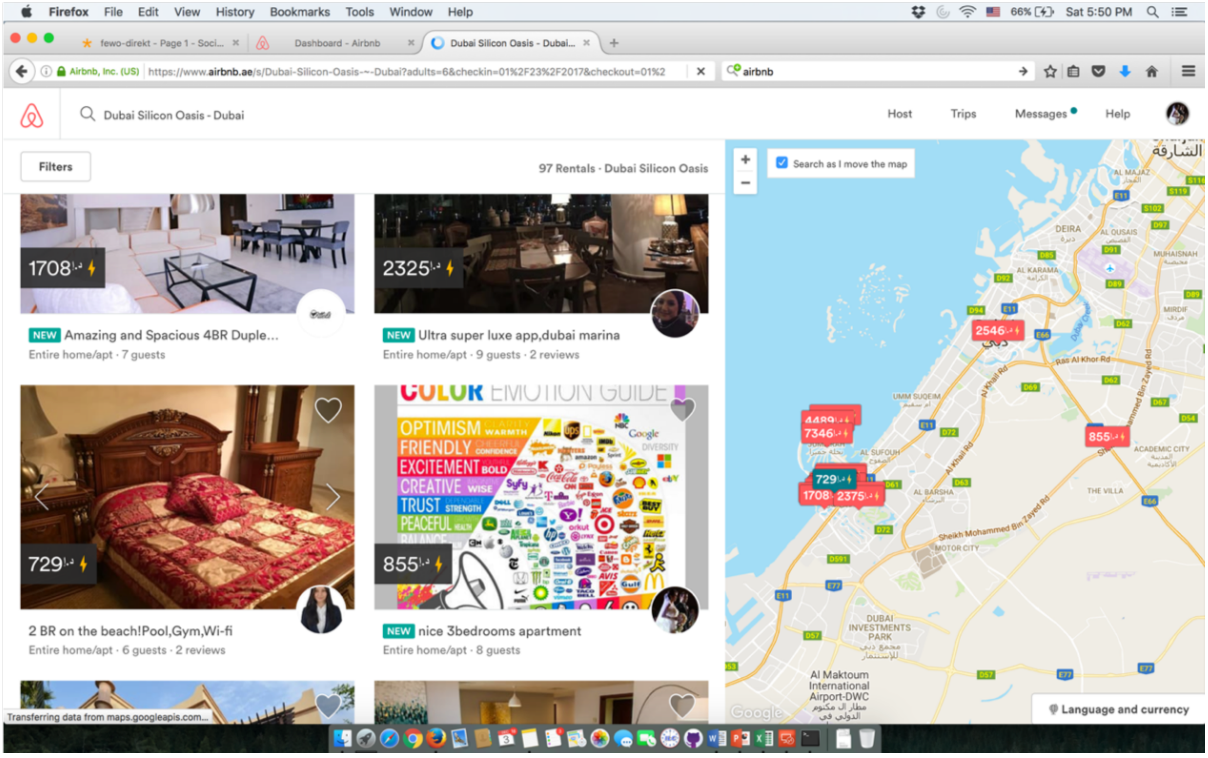


Figure 0.1: screen shot of Airbnb with a fake listing added for research purposes (nice 3 bedrooms apartment)

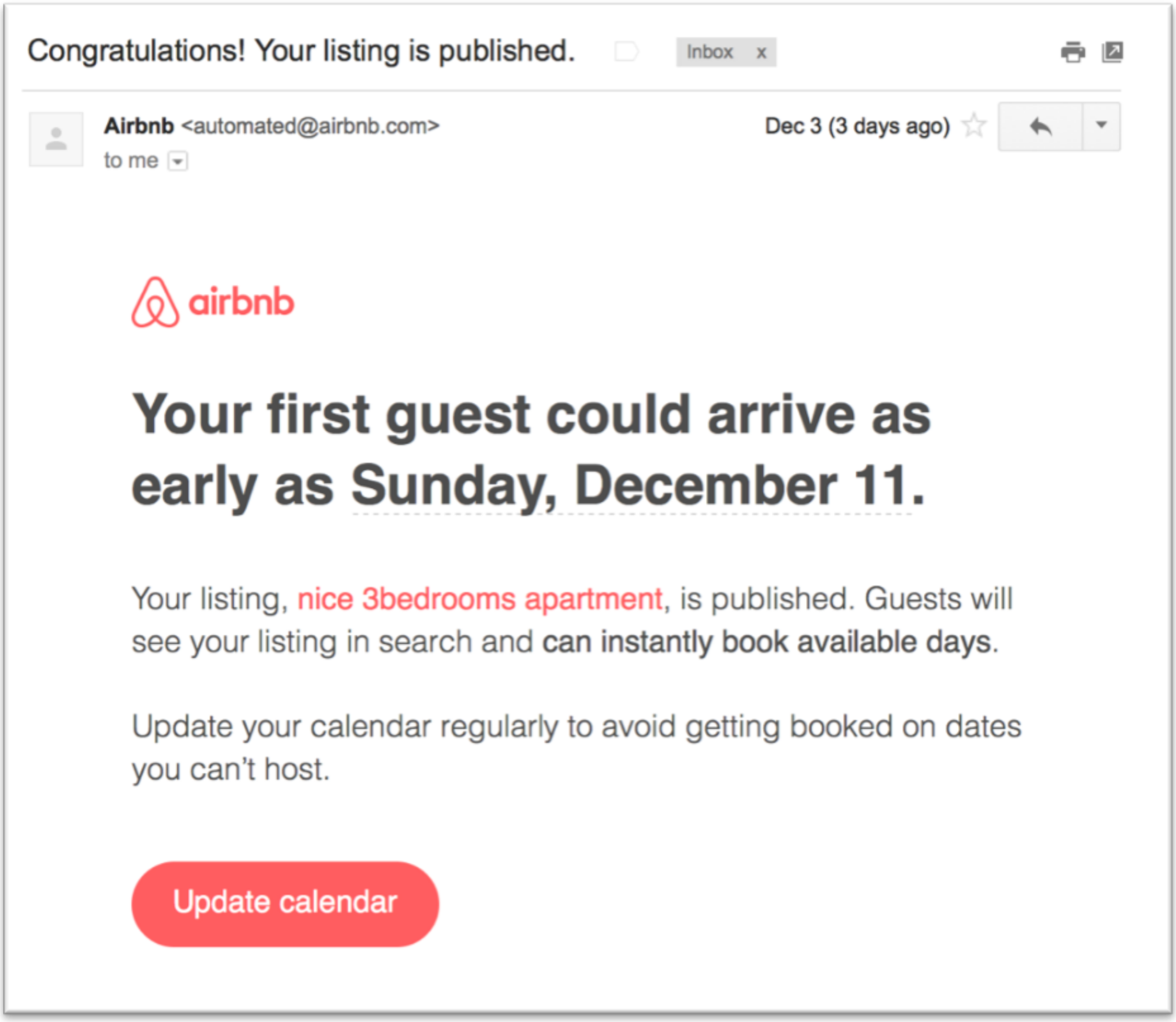


Figure 0.2: Airbnb email confirmation that the fake listing is published, and guests can instantly book

8.2 Appendix B - Universal (CC0 1.0) Public Domain Dedication

No Copyright, the person who associated a work with this deed has dedicated the work to the public domain by waiving all of his or her rights to the work worldwide under copyright law, including all related and neighbouring rights, to the extent allowed by law.

You can copy, modify, distribute and perform the work, even for commercial purposes, all without asking permission. In no way are the patent or trademark rights of any person affected by CC0, nor are the rights that other persons may have in the work or in how the work is used, such as publicity or privacy rights.

Unless expressly stated otherwise, the person who associated a work with this deed makes no warranties about the work, and disclaims liability for all uses of the work, to the fullest extent permitted by applicable law.

When using or citing the work, you should not imply endorsement by the author or the affirmer.

8.3 Appendix C – Source Code Step One: Tone analyser

```
# STEP 1
# Preparing the environement. Execute it cell only once
# Initial code was taken from https://github.com/IBM/pixiedust-facebook-ana
lysis/blob/master/notebooks/pixiedust_facebook_analysis.ipynb

# Watson Devloper libraries
!pip install --upgrade watson-developer-cloud

# Beautiful Soup to extract data from XML and HTML. Parse Watson response
!pip install --upgrade beautifulsoup4

# pixiedust to visualize few numbers
!pip install --user --upgrade pixiedust
```

```
In []:
```

```
# STEP 2
# Importing Watson Analyser libraries
# ToneAnalyzer will be used to analyse Host descriptions and Guest reviews
# VisualRecognitionV3 might be used to analyse Host and Guest thumbnail (op
tional)
import watson_developer_cloud
from watson_developer_cloud import ToneAnalyzerV3, VisualRecognitionV3
import watson_developer_cloud.natural_language_understanding.features.v1 as
features

# Supporting libraries
import math
```

```python
import operator
from operator import itemgetter
import json

# Libraries for calculations
import pandas as pd
import numpy as np
np.random.seed(2600)

# Libraries for making plots
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:

```python
# Watson Tone Analyzer Token
TONE_ANALYZER_USERNAME = '<UserName>'
TONE_ANALYZER_PASSWORD = '<Password>'

Tone_Analyzer = ToneAnalyzerV3(version='2016-05-19',
                               username=TONE_ANALYZER_USERNAME,
                               password=TONE_ANALYZER_PASSWORD)
```

In [ ]:

```python
temp_json = Tone_Analyzer.tone({
  "text": "Don't miss out on this modern home with luxuriously comfy king b
ed, 50 Inch flat screen TV with Netflix, wrap around decks with 180 mountai
n views, simple relaxation, great kitchen, Jacuzzi tub, and outdoor hot tub
. Space: I will not accept reservations more than 30 days in advance of the
date you would like to stay on Airbnb due to their set cancellation policie
s. You can book more than 30 Days in advance on th(URL HIDDEN)website  List
ing 613068  Beautiful Nightly Rental with Spectacular mountain views and ou
tdoor entertaining space with lounge chairs, decks, outdoor entertainment b
ar, and comfortable patio set for the ultimate relaxation. The home is 2800
square feet, huge private decks and 180 Mountain Views. The master bedroom
features mountain views, large jetted Jacuzzi tub, shower, double vanity, m
odern, simple feel, and 5lb density 12 inch thick temperpedic memory foam m
attress.  7 miles to World Famous Sunny Point Cafe, West Asheville, and Dow
ntown. The Space There is a fabulous temperpedic memory foam KING mattress
5.0 density. All linens, soap, shampoo, fully equipped kitchen, sofa, chair
s, yoga mat and more. High Speed 30MB internet, 50 inch flat screen smart T
V equipped with all you can watch. Don't miss out on this modern home with
luxuriously comfy king bed, 50 Inch flat screen TV with Netflix, wrap aroun
d decks with 180 mountain views, simple relaxation, great kitchen, Jacuzzi
tub, and outdoor hot tub. I will not accept reservations more than 30 days
in advance of the date you would like to stay on Airbnb due to their set ca
ncellation policies. You can book more than 30 Days in advance on th(URL HI
DDEN)website  Listing 613068  Beautiful Nightly Rental with Spectacular mou
```

```
ntain views and outdoor entertaining space with lounge chairs, decks, outdo
or entertainment bar, and comfortable patio set for the ultimate relaxation
. The home is 2800 square feet, huge private decks and 180 Mountain Views.
The master bedroom features mountain views, large jetted Jacuzzi tub, showe
r, double vanity, modern, simple feel, and 5lb density 12 inch thick temper
pedic memory foam mattress.  7 miles to World Famous Sunny Point Cafe, West
Asheville, and Downtown. The Space There is a fabulous temp. The neighborho
od is quiet and peaceful, surrounded by mountain views and rolling pictures
que landscape. You have complete privacy on the back deck.  Traveling all o
ver the world inspired me to get my degree in Culinary Arts and inspired me
to move to Asheville.  Great food and unique culture is hard to find in one
place.  Somehow Asheville has managed to combine culture, art, great food,
locally brewed beer, local farms, and local businesses into a charming town
nestled into ancient mountains full of waterfalls, hiking, history, and art
. Asheville is one of the few places that has four beautiful seasons and so
me of the best coffee and restaurants around.  I am excited to open up my h
ome to travelers and I'm looking forward to share all there is to experienc
e in and around Asheville."
}, 'text/html')
```

In [ ]:

```
# Those are the files from NothCarolina
calendar = pd.read_csv("https://github.com/la7oon/PhD_Trust_Repo/blob/maste
r/AirBnb/Asheville_20170402/calendar.csv?raw=true")
neighbourhoods = pd.read_csv("https://github.com/la7oon/PhD_Trust_Repo/blob
/master/AirBnb/Asheville_20170402/neighbourhoods.csv?raw=true")
reviews = pd.read_csv("https://github.com/la7oon/PhD_Trust_Repo/blob/master
/AirBnb/Asheville_20170402/reviews.csv?raw=true")
reviews_details = pd.read_csv("https://github.com/la7oon/PhD_Trust_Repo/blo
b/master/AirBnb/Asheville_20170402/reviews_details.csv?raw=true")
listings = pd.read_csv("https://github.com/la7oon/PhD_Trust_Repo/blob/maste
r/AirBnb/Asheville_20170402/listings.csv?raw=true")
listings_details = pd.read_csv("https://github.com/la7oon/PhD_Trust_Repo/bl
ob/master/AirBnb/Asheville_20170402/listings_details.csv?raw=true")
```

In [ ]:

```
print len(reviews_details)
```

In [ ]:

```
# Analysied Data
listings_details_analysied = pd.read_csv("https://github.com/la7oon/PhD_Tru
st_Repo/blob/master/AirBnb/Asheville_20170402/Asheville_listings_details_an
alysed.csv?raw=true")
reviews_details_analysied = pd.read_csv("https://github.com/la7oon/PhD_Trus
t_Repo/blob/master/AirBnb/Asheville_20170402/Asheville_review_details_analy
sed.csv?raw=true")
```

In [ ]:

```
# no need to add +1 here, the header is counted as +1
howManyRowsToAnalise = len(listings_details)

# print listings_details
listings_details.loc[:,'AllStrings'] = pd.Series("-", index=listings_detail
s.index)
for i in range(0,howManyRowsToAnalise):
    listings_details.loc[i,'AllStrings'] = str(listings_details.summary[i])
+ str(listings_details.space[i]) + str(listings_details.notes[i]) + str(lis
tings_details.description[i]) + str(listings_details.neighborhood_overview[
i]) + str(listings_details.transit[i]) + str(listings_details.host_about[i]
)
    if(i%100 == 0):
        print i
```

In [ ]:

```
#  EXPENSIVE METHOD - this method will cost money to run, please use it car
efully
# this cell might take few minutes
# each listing is being analysed seperatly
# listings_details.loc[:,'WatsonResponse'] = pd.Series("-", index=listings_
details.index)
for I in range(0,howManyRowsToAnalise):
    listings_details.loc[I,'WatsonResponse'] = json.dumps(Tone_Analyzer.ton
e({"text": listings_details.AllStrings[i]}, 'text/html'), separators=(',','
:'))
    if(i%100 == 0):
        print i
```

In [ ]:

```
# Save Analyised city listing into a file
listings_details.to_csv("Asheville_listings_details_analysed.csv")
```

In [ ]:

```
#  EXPENSIVE METHOD - this method will cost money to run, please use it car
efully
# this cell might take few minutes

# Preparing the review strings
# no need to add +1 here, the header is counted as +1
howManyRowsToAnalise = len(reviews_details)

# each listing is being analysed seperatly
# reviews_details.loc[:,'WatsonResponse'] = pd.Series("-", index=reviews_de
tails.index)
for i in range(23000,howManyRowsToAnalise):
```

```
    reviews_details.loc[i,'WatsonResponse'] = json.dumps(Tone_Analyzer.tone
({"text": str(reviews_details.comments[i])}, 'text/html'), separators=(',',
':'))
    if(i%100 == 0):
        print i
    if(i%500 == 0):
        reviews_details.to_csv("Asheville_review_details_analysed.csv")
```

In [ ]:

```
# Save Analyised review details into a file
reviews_details.to_csv("Asheville_review_details_analysed.csv")
```

8.4 Appendix D – Source Code Step two: Data Analysis using Jupiter

```
#You might need to install zope.interface in case you faced any zope.interf
ace issues
!pip install 'zope.interface==4.4.3' --force-reinstall
#Sometimes pip is outdated, you might want to update them on the docer assi
gned
!pip install --upgrade pip
```

In []:

```
# Preparing the environement. Execute it cell only once
# Initial code was taken from https://github.com/IBM/pixiedust-facebook-ana
lysis/blob/master/notebooks/pixiedust_facebook_analysis.ipynb
# Watson Devloper libraries
!pip install --upgrade watson-developer-cloud

# Beautiful Soup to extract data from XML and HTML. Parse Watson response
!pip install --upgrade beautifulsoup4

# pixiedust to visualize few numbers
!pip install --user --upgrade pixiedust
```

In []:

```
# STEP 2
# Importing Watson Analyser libraries
# ToneAnalyzer will be used to analyse Host descriptions and Guest reviews
# VisualRecognitionV3 might be used to analyse Host and Guest thumbnail (op
tional)
import watson_developer_cloud
from watson_developer_cloud import ToneAnalyzerV3, VisualRecognitionV3
import watson_developer_cloud.natural_language_understanding.features.v1 as
features

# Supporting libraries
```

```
import math
import operator
from operator import itemgetter
import json

# Libraries for calculations
import pandas as pd
from pandas.tools.plotting import scatter_matrix

import matplotlib.pyplot as plt
from matplotlib import cm

import numpy as np
np.random.seed(2600)

from sklearn.cluster import KMeans

import seaborn as sns
import pylab as pl

# Libraries for making plots
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns

# Load the PCA Library. This one is available on scikit-learn
from sklearn.decomposition import PCA
```

In [ ]:

```
# Watson Tone Analyzer Token
TONE_ANALYZER_USERNAME = '747f6ebe-0618-4700-823d-6251141de14c'
TONE_ANALYZER_PASSWORD = 'wIngUaRbByDZ'


Tone_Analyzer = ToneAnalyzerV3(version='2016-05-19',
                               username=TONE_ANALYZER_USERNAME,
                               password=TONE_ANALYZER_PASSWORD)
```

In [ ]:

```
# Load analyised files from GITHUB
listings_details_analyised = pd.read_csv("https://github.com/la7oon/PhD_Tru
st_Repo/blob/master/AirBnb/Asheville_20170402/Asheville_listings_details_an
alysed.csv?raw=true")
reviews_details_analyised = pd.read_csv("https://github.com/la7oon/PhD_Trus
t_Repo/blob/master/AirBnb/Asheville_20170402/Asheville_reviews_details_anal
ysed.csv?raw=true")
```

```
listing_vs_reviews_tones_matrix = pd.read_csv("https://github.com/la7oon/PhD_Trust_Repo/blob/master/AirBnb/Asheville_20170402/Asheville_listing_vs_reviews_tones_matrix.csv?raw=true")
```

In []:

```
# Configurations
len(reviews_details_analyised)
```

Out[3]:

```
27721
```

In []:

```
# Airbnb reviews Step 1
# Extract the emotional, language, and social tones and save them in new columns for ease of parsing
howManyRowsToAnalise = len(reviews_details_analyised)

# print listings_details
for i in range(0,howManyRowsToAnalise):
    temp_json = eval(reviews_details_analyised.WatsonResponse[i])
    reviews_details_analyised.loc[i,'EmotionalToneObject'] = json.dumps(sorted(temp_json["document_tone"]["tone_categories"][0]["tones"], key = itemgetter('score')))
    reviews_details_analyised.loc[i,'LanguageToneObject'] = json.dumps(sorted(temp_json["document_tone"]["tone_categories"][1]["tones"], key = itemgetter('score')))
    reviews_details_analyised.loc[i,'SocialToneObject'] = json.dumps(sorted(temp_json["document_tone"]["tone_categories"][2]["tones"], key = itemgetter('score')))
#     print reviews_details_analyised.EmotionalToneObject[i]
    if(i%1000==0):
        print i
```

In []:

```
# Airbnb reviews
#  Extract each value from emotional, language, and social tones and save them as seperate columns for visualization
howManyRowsToAnalise = len(reviews_details_analyised)

for i in range(0,howManyRowsToAnalise):
    EmotionalToneObject = eval(reviews_details_analyised.EmotionalToneObject[i])
    LanguageToneObject = eval(reviews_details_analyised.LanguageToneObject[i])
    SocialToneObject = eval(reviews_details_analyised.SocialToneObject[i])

    for j in range(0, len(EmotionalToneObject)):
```

```
        reviews_details_analysised.loc[i,'review_emotional_'+EmotionalToneOb
ject[j]['tone_id']] = EmotionalToneObject[j]['score']
    for j in range(0, len(LanguageToneObject)):
        reviews_details_analysised.loc[i,'review_language_'+LanguageToneObje
ct[j]['tone_id']] = LanguageToneObject[j]['score']
    for j in range(0, len(SocialToneObject)):
        reviews_details_analysised.loc[i,'review_social_'+SocialToneObject[j
]['tone_id']] = SocialToneObject[j]['score']
    if(i%1000==0):
        print i
```

In []:

```
#Save a copy of reviews_details_analysised after splitting the Watson Result
s
reviews_details_analysised.to_csv("Asheville_reviews_details_analysised.csv")
```

In [ ]:

```
# Airbnb listings
# Extract the emotional, language, and social tones and save them in new co
lumns for ease of parsing
howManyRowsToAnalise = len(listings_details_analysised)

# print listings_details
for i in range(0,howManyRowsToAnalise):
    temp_json = eval(listings_details_analysised.WatsonResponse[i])
    listings_details_analysised.loc[i,'EmotionalToneObject'] = json.dumps(so
rted(temp_json["document_tone"]["tone_categories"][0]["tones"], key = itemg
etter('score')))
    listings_details_analysised.loc[i,'LanguageToneObject'] = json.dumps(sor
ted(temp_json["document_tone"]["tone_categories"][1]["tones"], key = itemge
tter('score')))
    listings_details_analysised.loc[i,'SocialToneObject'] = json.dumps(sorte
d(temp_json["document_tone"]["tone_categories"][2]["tones"], key = itemgett
er('score')))
#     print reviews_details_analysised.EmotionalToneObject[i]
    if(i%500==0):
        print i
```

In [ ]:

```
# Airbnb Listing
# Extract each value from emotional, language, and social tones and save th
em as seperate columns for visualization
howManyRowsToAnalise = len(listings_details_analysised)

for i in range(0,howManyRowsToAnalise):
    EmotionalToneObject = eval(listings_details_analysised.EmotionalToneObje
ct[i])
```

```
    LanguageToneObject = eval(listings_details_analysed.LanguageToneObject
[i])
    SocialToneObject = eval(listings_details_analysed.SocialToneObject[i])

    for j in range(0, len(EmotionalToneObject)):
        listings_details_analysed.loc[i,'listing_emotional_'+EmotionalTone
Object[j]['tone_id']] = EmotionalToneObject[j]['score']
    for j in range(0, len(LanguageToneObject)):
        listings_details_analysed.loc[i,'listing_language_'+LanguageToneOb
ject[j]['tone_id']] = LanguageToneObject[j]['score']
    for j in range(0, len(SocialToneObject)):
        listings_details_analysed.loc[i,'listing_social_'+SocialToneObject
[j]['tone_id']] = SocialToneObject[j]['score']
    if(i%500==0):
        print i
```

In [ ]:

```
#Save a copy of listings_details_analysed after splitting the Watson Resul
ts
listings_details_analysed.to_csv("Asheville_listings_details_analyised.csv
")
```

In [ ]:

```
#  this nested loop is very expensive and grows exponensially
#  take than in mind when playing it
howManyReviews = len(reviews_details_analysed)
howManyListings = len(listings_details_analysed)
# listings_details_analysed
# reviews_details_analysed
for i in range(0,howManyReviews):
    if(i%1000==0):
        print i
    for j in range(0,howManyListings):
        if(listings_details_analysed.id[j] == reviews_details_analysed.li
sting_id[i]):
            listinID = j
            break

    listing_vs_reviews_tones_matrix.loc[i,'listing_id'] = listings_details_
analyised.id[listinID]

    listing_vs_reviews_tones_matrix.loc[i,'listing_emotional_anger'] = list
ings_details_analysed.listing_emotional_anger[listinID]
    listing_vs_reviews_tones_matrix.loc[i,'listing_emotional_disgust'] = li
stings_details_analysed.listing_emotional_disgust[listinID]
    listing_vs_reviews_tones_matrix.loc[i,'listing_emotional_fear'] = listi
ngs_details_analysed.listing_emotional_fear[listinID]
```

```
    listing_vs_reviews_tones_matrix.loc[i,'listing_emotional_joy'] = listings_details_analysed.listing_emotional_joy[listinID]
    listing_vs_reviews_tones_matrix.loc[i,'listing_emotional_sadness'] = listings_details_analysed.listing_emotional_sadness[listinID]
    listing_vs_reviews_tones_matrix.loc[i,'listing_language_analytical'] = listings_details_analysed.listing_language_analytical[listinID]
    listing_vs_reviews_tones_matrix.loc[i,'listing_language_confident'] = listings_details_analysed.listing_language_confident[listinID]
    listing_vs_reviews_tones_matrix.loc[i,'listing_language_tentative'] = listings_details_analysed.listing_language_tentative[listinID]
    listing_vs_reviews_tones_matrix.loc[i,'listing_social_openness_big5'] = listings_details_analysed.listing_social_openness_big5[listinID]
    listing_vs_reviews_tones_matrix.loc[i,'listing_social_conscientiousness_big5'] = listings_details_analysed.listing_social_conscientiousness_big5[listinID]
    listing_vs_reviews_tones_matrix.loc[i,'listing_social_extraversion_big5'] = listings_details_analysed.listing_social_extraversion_big5[listinID]
    listing_vs_reviews_tones_matrix.loc[i,'listing_social_agreeableness_big5'] = listings_details_analysed.listing_social_agreeableness_big5[listinID]
    listing_vs_reviews_tones_matrix.loc[i,'listing_social_emotional_range_big5'] = listings_details_analysed.listing_social_emotional_range_big5[listinID]

    listing_vs_reviews_tones_matrix.loc[i,'review_emotional_anger'] = reviews_details_analysed.review_emotional_anger[i]
    listing_vs_reviews_tones_matrix.loc[i,'review_emotional_disgust'] = reviews_details_analysed.review_emotional_disgust[i]
    listing_vs_reviews_tones_matrix.loc[i,'review_emotional_fear'] = reviews_details_analysed.review_emotional_fear[i]
    listing_vs_reviews_tones_matrix.loc[i,'review_emotional_joy'] = reviews_details_analysed.review_emotional_joy[i]
    listing_vs_reviews_tones_matrix.loc[i,'review_emotional_sadness'] = reviews_details_analysed.review_emotional_sadness[i]
    listing_vs_reviews_tones_matrix.loc[i,'review_language_analytical'] = reviews_details_analysed.review_language_analytical[i]
    listing_vs_reviews_tones_matrix.loc[i,'review_language_confident'] = reviews_details_analysed.review_language_confident[i]
    listing_vs_reviews_tones_matrix.loc[i,'review_language_tentative'] = reviews_details_analysed.review_language_tentative[i]
    listing_vs_reviews_tones_matrix.loc[i,'review_social_openness_big5'] = reviews_details_analysed.review_social_openness_big5[i]
    listing_vs_reviews_tones_matrix.loc[i,'review_social_conscientiousness_big5'] = reviews_details_analysed.review_social_conscientiousness_big5[i]
    listing_vs_reviews_tones_matrix.loc[i,'review_social_extraversion_big5'] = reviews_details_analysed.review_social_extraversion_big5[i]
    listing_vs_reviews_tones_matrix.loc[i,'review_social_agreeableness_big5'] = reviews_details_analysed.review_social_agreeableness_big5[i]
```

```
    listing_vs_reviews_tones_matrix.loc[i,'review_social_emotional_range_bi
g5'] = reviews_details_analysised.review_social_emotional_range_big5[i]
```

In [ ]:

```
#Save a copy of listing_vs_reviews_tones_matrix
listing_vs_reviews_tones_matrix.to_csv("Asheville_listing_vs_reviews_tones_
matrix.csv")
```

In [ ]:

```
howManyListings = len(listings_details_analyised)
howManyRowsToAnalise = len(listing_vs_reviews_tones_matrix)

j = 0
for i in range(0,howManyRowsToAnalise):

    if(listings_details_analyised.id[j] != listing_vs_reviews_tones_matrix.
listing_id[i]):
        for Z in range(0,howManyListings):
            if(listings_details_analyised.id[Z] == listing_vs_reviews_tones
_matrix.listing_id[i]):
                j = Z
                break

    listing_vs_reviews_tones_matrix.loc[i,'host_is_superhost'] = listings_d
etails_analyised.host_is_superhost[j].replace("f","0").replace("t","1")
    listing_vs_reviews_tones_matrix.loc[i,'host_identity_verified'] = listi
ngs_details_analyised.host_identity_verified[j].replace("f","0").replace("t
","1")
    listing_vs_reviews_tones_matrix.loc[i,'is_location_exact'] = listings_d
etails_analyised.is_location_exact[j].replace("f","0").replace("t","1")
    listing_vs_reviews_tones_matrix.loc[i,'property_type'] = listings_detai
ls_analyised.property_type[j]
    listing_vs_reviews_tones_matrix.loc[i,'room_type'] = listings_details_a
nalyised.room_type[j]
    listing_vs_reviews_tones_matrix.loc[i,'accommodates'] = listings_detail
s_analyised.accommodates[j]
    listing_vs_reviews_tones_matrix.loc[i,'bathrooms'] = listings_details_a
nalyised.bathrooms[j]
    listing_vs_reviews_tones_matrix.loc[i,'bedrooms'] = listings_details_an
alyised.bedrooms[j]
    listing_vs_reviews_tones_matrix.loc[i,'beds'] = listings_details_analyi
sed.beds[j]
    listing_vs_reviews_tones_matrix.loc[i,'price'] = listings_details_analy
ised.price[j].replace("$","")
    listing_vs_reviews_tones_matrix.loc[i,'cleaning_fee'] = listings_detail
s_analyised.cleaning_fee[j]
    listing_vs_reviews_tones_matrix.loc[i,'number_of_reviews'] = listings_d
etails_analyised.number_of_reviews[j]
```

```
    listing_vs_reviews_tones_matrix.loc[i,'review_scores_rating'] = listing
s_details_analysed.review_scores_rating[j]
    listing_vs_reviews_tones_matrix.loc[i,'review_scores_accuracy'] = listi
ngs_details_analysed.review_scores_accuracy[j]
    listing_vs_reviews_tones_matrix.loc[i,'review_scores_cleanliness'] = li
stings_details_analysed.review_scores_cleanliness[j]
    listing_vs_reviews_tones_matrix.loc[i,'review_scores_checkin'] = listin
gs_details_analysed.review_scores_checkin[j]
    listing_vs_reviews_tones_matrix.loc[i,'review_scores_communication'] =
listings_details_analysed.review_scores_communication[j]
    listing_vs_reviews_tones_matrix.loc[i,'review_scores_location'] = listi
ngs_details_analysed.review_scores_location[j]
    listing_vs_reviews_tones_matrix.loc[i,'review_scores_value'] = listings
_details_analysed.review_scores_value[j]
    listing_vs_reviews_tones_matrix.loc[i,'instant_bookable'] = listings_de
tails_analysed.instant_bookable[j].replace("f","0").replace("t","1")
    listing_vs_reviews_tones_matrix.loc[i,'cancellation_policy'] = listings
_details_analysed.cancellation_policy[j]
    listing_vs_reviews_tones_matrix.loc[i,'require_guest_profile_picture']
= listings_details_analysed.require_guest_profile_picture[j].replace("f","
0").replace("t","1")
    listing_vs_reviews_tones_matrix.loc[i,'require_guest_phone_verification
'] = listings_details_analysed.require_guest_phone_verification[j].replace
("f","0").replace("t","1")
    listing_vs_reviews_tones_matrix.loc[i,'reviews_per_month'] = listings_d
etails_analysed.reviews_per_month[j]

    if(i%500==0):
        print i
```

In [ ]:

```
# listings_details.to_csv("Manchester_listings_details_analysed.csv")
listing_vs_reviews_tones_matrix = pd.read_csv("https://github.com/la7oon/Ph
D_Trust_Repo/blob/master/AirBnb/Manchester_20170410/Manchester_listing_vs_r
eviews_tones_matrix.csv?raw=true")
```

In []:

```
# Airbnb Listing
#  Try to build a complex feeling from mix of basic feeling

howManyRowsToAnalise = len(listing_vs_reviews_tones_matrix)
for i in range(0,howManyRowsToAnalise):
    listing_vs_reviews_tones_matrix.loc[i,'GuestFeeling'] = "0" # nutral

    if(
        listing_vs_reviews_tones_matrix.review_emotional_joy[i] >= listing_
vs_reviews_tones_matrix.review_emotional_anger[i]
```

```
    and listing_vs_reviews_tones_matrix.review_emotional_joy[i] >= listing_
vs_reviews_tones_matrix.review_emotional_disgust[i]
    and listing_vs_reviews_tones_matrix.review_emotional_joy[i] >= listing_
vs_reviews_tones_matrix.review_emotional_fear[i]
    and listing_vs_reviews_tones_matrix.review_emotional_joy[i] >= listing_
vs_reviews_tones_matrix.review_emotional_sadness[i]
    and listing_vs_reviews_tones_matrix.review_language_confident[i] >= lis
ting_vs_reviews_tones_matrix.review_language_analytical[i]
    and listing_vs_reviews_tones_matrix.review_language_confident[i] >= lis
ting_vs_reviews_tones_matrix.review_language_tentative[i]
    ):
        listing_vs_reviews_tones_matrix.loc[i,'GuestFeeling'] = "1" # confi
dant of joy

    if(
        listing_vs_reviews_tones_matrix.review_emotional_anger[i] >= listin
g_vs_reviews_tones_matrix.review_emotional_joy[i]
    and listing_vs_reviews_tones_matrix.review_emotional_anger[i] >= listin
g_vs_reviews_tones_matrix.review_emotional_disgust[i]
    and listing_vs_reviews_tones_matrix.review_emotional_anger[i] >= listin
g_vs_reviews_tones_matrix.review_emotional_fear[i]
    and listing_vs_reviews_tones_matrix.review_emotional_anger[i] >= listin
g_vs_reviews_tones_matrix.review_emotional_sadness[i]
    and listing_vs_reviews_tones_matrix.review_language_confident[i] >= lis
ting_vs_reviews_tones_matrix.review_language_analytical[i]
    and listing_vs_reviews_tones_matrix.review_language_confident[i] >= lis
ting_vs_reviews_tones_matrix.review_language_tentative[i]
    ):
        listing_vs_reviews_tones_matrix.loc[i,'GuestFeeling'] = "-1" # conf
idant of anger

    if(i%500==0):
        print i
```

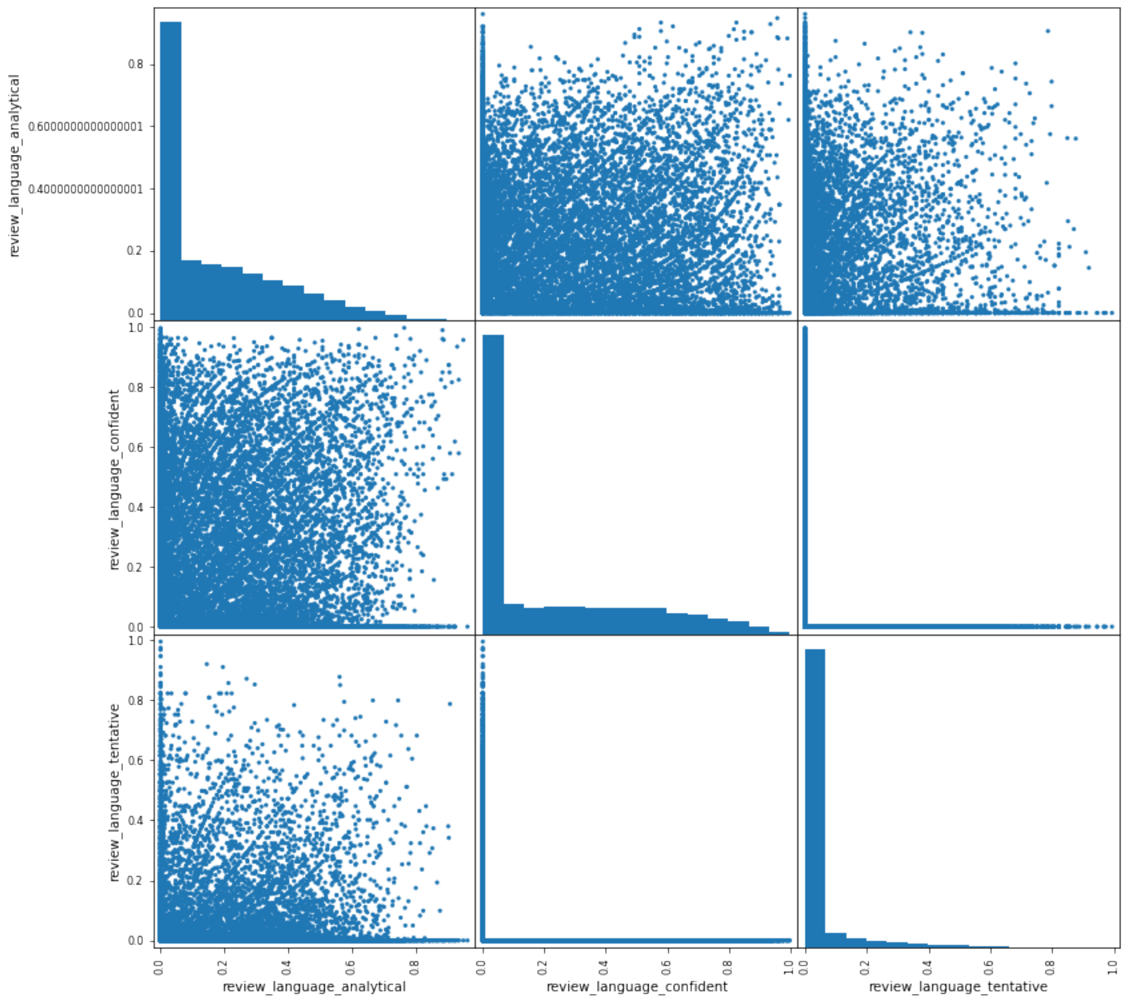
In [15]:

```
#Save a copy of listing_vs_reviews_tones_matrix
listing_vs_reviews_tones_matrix.to_csv("Asheville_listing_vs_reviews_tones_
matrix.csv")
```

In [4]:

```
### Profiling Listing emotions
# feature_names = ['listing_emotional_anger','listing_emotional_disgust','l
isting_emotional_fear','listing_emotional_joy','listing_emotional_sadness']
# feature_names = ['review_emotional_anger','review_emotional_disgust','rev
iew_emotional_fear','review_emotional_joy','review_emotional_sadness']
feature_names = ['review_language_analytical','review_language_confident','
review_language_tentative']
```

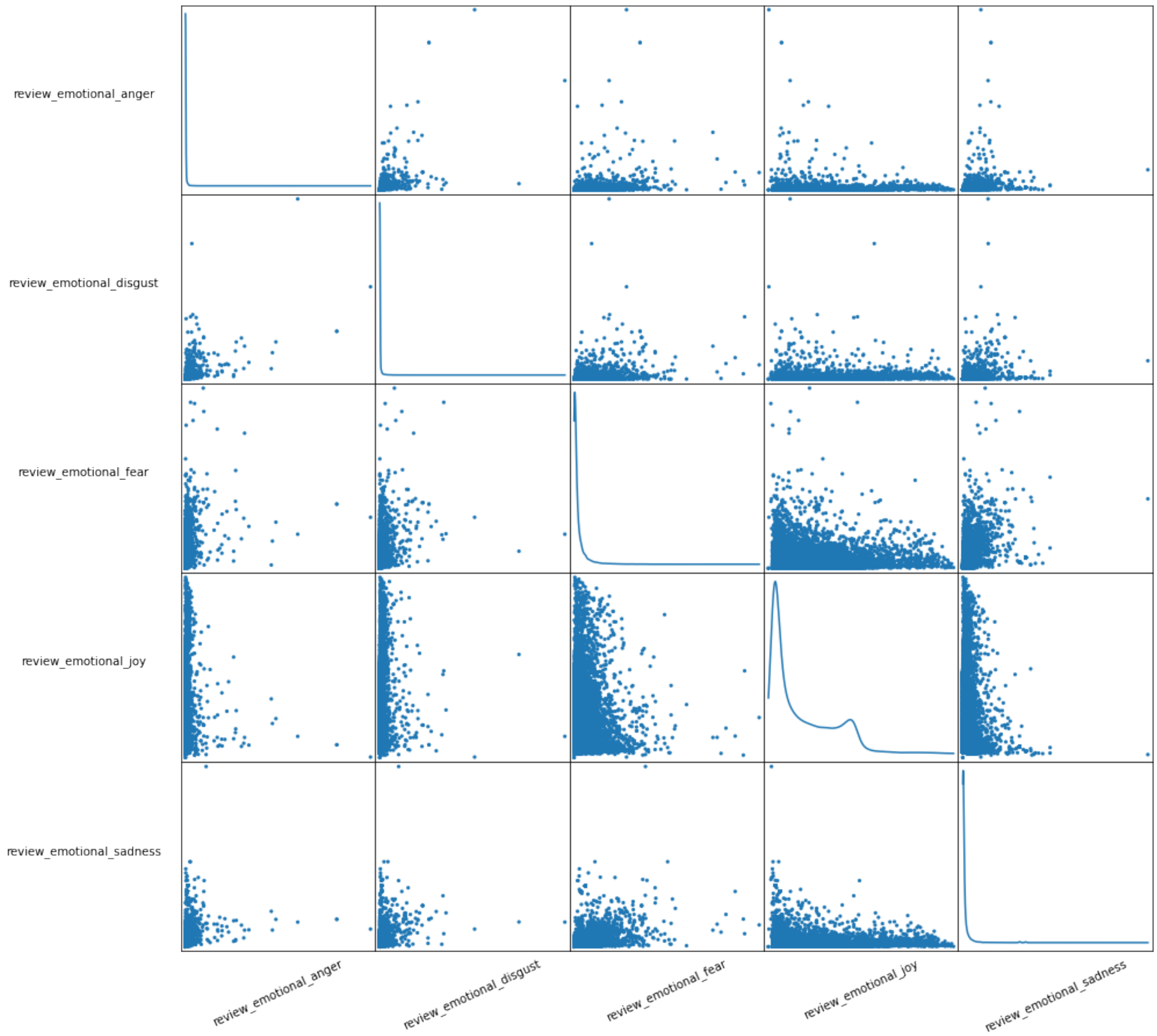
```
X = listing_vs_reviews_tones_matrix[feature_names]
# Y = KMeans(n_clusters=2, random_state=0).fit_predict(listing_vs_reviews_t
ones_matrix[['listing_emotional_fear','listing_emotional_joy']])


# This is a good figure ... you can see the purpule is dominating in multip
le squars... keep this code
# feature_names = ['listing_language_confident','listing_emotional_joy', 'r
eview_emotional_anger','review_emotional_disgust','review_emotional_fear','
review_emotional_joy','review_emotional_sadness','review_language_analytica
l','review_language_confident','review_language_tentative','review_social_o
penness_big5','review_social_conscientiousness_big5','review_social_extrave
rsion_big5','review_social_agreeableness_big5','review_social_emotional_ran
ge_big5']
# X = listing_vs_reviews_tones_matrix[feature_names]
# Y = KMeans(n_clusters=4, random_state=0).fit_predict(listing_vs_reviews_t
ones_matrix[['listing_language_confident','listing_emotional_joy']])

cmap = cm.get_cmap('gnuplot')
sm = pd.plotting.scatter_matrix(X, alpha=1, marker = 'o', s=10, hist_kwds={
'bins':15}, figsize=(13, 13), cmap=cmap, diagonal='hist')

# #Change label rotation
# [s.xaxis.label.set_rotation(25) for s in sm.reshape(-1)]
# # [s.yaxis.label.set_rotation(0) for s in sm.reshape(-1)]

# #May need to offset label when rotating to prevent overlap of figure
# # [s.get_yaxis().set_label_coords(-0.5,0.5) for s in sm.reshape(-1)]

# #Hide all ticks
# [s.set_xticks(()) for s in sm.reshape(-1)]
# # [s.set_yticks(()) for s in sm.reshape(-1)]

plt.show()
plt.suptitle('Scatter-matrix for Asheville Airbnb Host Language tone vs Gus
t Emotional Tones')
plt.savefig('Asheville Airbnb Host Language tone vs Gust Emotional Tones')
```



In [25]:

```
### Profiling Review emotions
feature_names = ['review_emotional_anger','review_emotional_disgust','revie
w_emotional_fear','review_emotional_joy','review_emotional_sadness']
X = reviews_details_analyised[feature_names]
# Y = KMeans(n_clusters=2, random_state=0).fit_predict(reviews_details_anal
yised[['review_emotional_joy','review_emotional_sadness']])

cmap = cm.get_cmap('gnuplot')
sm = pd.plotting.scatter_matrix(X, alpha=1, marker = 'o', s=10, hist_kwds={
'bins':15}, figsize=(15, 15), cmap=cmap, diagonal='kde')

#Change label rotation
[s.xaxis.label.set_rotation(25) for s in sm.reshape(-1)]
[s.yaxis.label.set_rotation(0) for s in sm.reshape(-1)]

#May need to offset label when rotating to prevent overlap of figure
[s.get_yaxis().set_label_coords(-0.5,0.5) for s in sm.reshape(-1)]
```

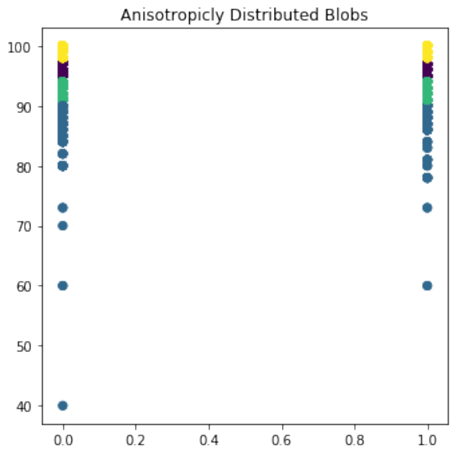
```
#Hide all ticks
[s.set_xticks(()) for s in sm.reshape(-1)]
[s.set_yticks(()) for s in sm.reshape(-1)]

plt.show()
plt.suptitle('Scatter-matrix for Asheville Airbnb Guest Enotions')
plt.savefig('Asheville Airbnb Guest Enotions')
```



In [18]:

```
plt.figure(figsize=(12, 12))

# # Incorrect number of clusters
# plt.subplot(221)
# plt.scatter(X[:, 0], X[:, 1], c=y_pred)
# plt.title("Incorrect Number of Blobs")

# Anisotropicly distributed data
```

```
Y = KMeans(n_clusters=4, random_state=0).fit_predict(listing_vs_reviews_ton
es_matrix[['host_identity_verified','host_is_superhost']])

plt.subplot(222)
plt.scatter(listing_vs_reviews_tones_matrix['host_identity_verified'], list
ing_vs_reviews_tones_matrix['review_scores_rating'], c=Y)
plt.title("Anisotropicly Distributed Blobs")

# # Different variance
# X_varied, y_varied = make_blobs(n_samples=n_samples,
#                                 cluster_std=[1.0, 2.5, 0.5],
#                                 random_state=random_state)
# y_pred = KMeans(n_clusters=3, random_state=random_state).fit_predict(X_va
ried)

# plt.subplot(223)
# plt.scatter(X_varied[:, 0], X_varied[:, 1], c=y_pred)
# plt.title("Unequal Variance")

# # Unevenly sized blobs
# X_filtered = np.vstack((X[y == 0][:500], X[y == 1][:100], X[y == 2][:10])
)
# y_pred = KMeans(n_clusters=3,
#                 random_state=random_state).fit_predict(X_filtered)

# plt.subplot(224)
# plt.scatter(X_filtered[:, 0], X_filtered[:, 1], c=y_pred)
# plt.title("Unevenly Sized Blobs")

plt.show()
```



In [27]:

```
# This creates data-blocks that we can manipulate.
# X = listing_vs_reviews_tones_matrix[listing_vs_reviews_tones_matrix.colum
ns[1:]].values
data = listing_vs_reviews_tones_matrix #listings_details_analyised
```

```
# X = data[['listing_emotional_anger','listing_emotional_disgust','listing_
emotional_fear','listing_emotional_joy','listing_emotional_sadness','listin
g_language_analytical','listing_language_confident','listing_language_tenta
tive','listing_social_openness_big5','listing_social_conscientiousness_big5
','listing_social_extraversion_big5','listing_social_agreeableness_big5','l
isting_social_emotional_range_big5']].values
# X = data[['listing_emotional_anger','listing_emotional_disgust','listing_
emotional_fear','listing_emotional_joy','listing_emotional_sadness','listin
g_language_analytical','listing_language_confident','listing_language_tenta
tive']].values
X = data[['listing_social_openness_big5','listing_social_conscientiousness_
big5','listing_social_extraversion_big5','listing_social_agreeableness_big5
','listing_social_emotional_range_big5']].values

# X = data[['review_emotional_anger','review_emotional_disgust','review_emo
tional_fear','review_emotional_joy','review_emotional_sadness','review_lang
uage_analytical','review_language_confident','review_language_tentative']].
values
y = data.review_scores_rating.values
```

In [28]:

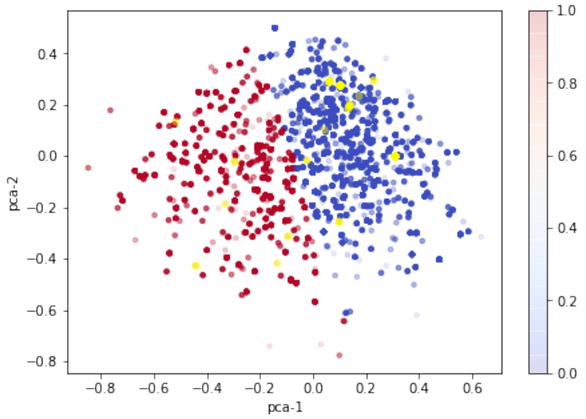
```
# Create the PCA, give the data to the PCA and `fit` the analysis.
pca = PCA(n_components=2)
pca.fit(X)

# Transform the original data to new data.
X_pca = pca.transform(X)

# Store the data in the original data-frame.
data['pca-1'], data['pca-2'] = X_pca[:,0], X_pca[:,1]
```

In [30]:

```
SELECTED_COLUMN_FOR_COLOR = 'listing_language_confident'

Y = KMeans(n_clusters=2, random_state=0).fit_predict(listing_vs_reviews_ton
es_matrix[['pca-1','pca-2']])

plt.figure(figsize=(16,5))
ax1 = plt.subplot(1, 2, 1)
data.plot.scatter(x='pca-1', y='pca-2', c=Y, s=12, alpha=0.1, cmap=plt.get_
cmap('coolwarm'), ax=ax1)
# Good results for 75 guest with tentative reviews
# data[data.review_language_tentative >= 0.7].plot.scatter(x='pca-1', y='pc
a-2', s=24, alpha=0.9, c='yellow', edgecolors='black', ax=ax1)
data[data.review_scores_accuracy <= 8].plot.scatter(x='pca-1', y='pca-2', s
=24, alpha=0.4, c='yellow', edgecolors='yellow', ax=ax1)
```

```
# ax2 = plt.subplot(1, 2, 2)
# data.plot.scatter(x='pca-1', y='pca-3', s=12, c=Y, alpha=0.2, cmap=plt.ge
t_cmap('coolwarm'), ax=ax2)
# # data[data.host_is_superhost !='1'].plot.scatter(x='pca-1', y='pca-3', s
=24, alpha=0.9, c='none', edgecolors='black', ax=ax2)
# # data[data.host_identity_verified=='1'].plot.scatter(x='pca-1', y='pca-3
', s=24, alpha=0.9, c='none', edgecolors='black', ax=ax2)
# data[data.review_language_tentative >= 0.75].plot.scatter(x='pca-1', y='p
ca-3', s=24, alpha=0.9, c='none', edgecolors='black', ax=ax2)
```

Out[30]:



In [38]:

```
# data.review_emotional_sadness.describe()
# len(data[data.review_language_tentative >= 0.3])
len(listings_details_analysed[listings_details_analysed.listing_emotional
_fear > 0.3])
```

Out[]:

24

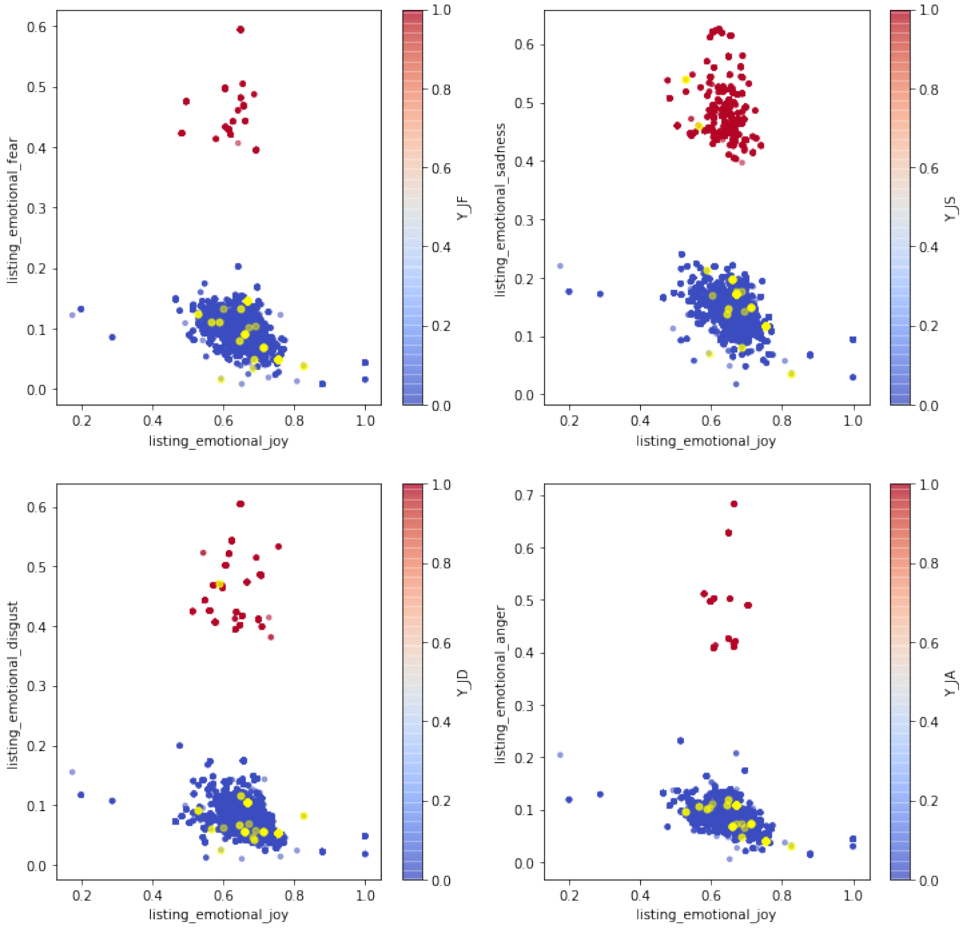
In []:

```
# This creates data-blocks that we can manipulate.
data = listing_vs_reviews_tones_matrix #listings_details_analysed
X = data[['listing_emotional_fear','listing_emotional_joy']].values
y = data.review_scores_rating.values

# Create the PCA, give the data to the PCA and `fit` the analysis.
pca = PCA(n_components=1)
pca.fit(X)
# Transform the original data to new data.
X_pca = pca.transform(X)
# Store the data in the original data-frame.
data['trust-1'] = X_pca[:,0]

SELECTED_COLUMN_FOR_COLOR = 'trust-1'
```

```
plt.figure(figsize=(12,12))
ax1 = plt.subplot(221)
data['Y_JF'] = KMeans(n_clusters=2, random_state=0).fit_predict(data[['list
ing_emotional_joy','listing_emotional_fear']])
data.plot.scatter(x='listing_emotional_joy', y='listing_emotional_fear', c=
'Y_JF', s=12, alpha=0.5, cmap=plt.get_cmap('coolwarm'), ax=ax1)
# Good results for 75 guest with tentative reviews
data[data.review_scores_accuracy <= 8].plot.scatter(x='listing_emotional_jo
y', y='listing_emotional_fear', s=24, alpha=0.5, c='yellow', edgecolors='ye
llow', ax=ax1)
# data[data.review_emotional_sadness >= 0.1].plot.scatter(x='listing_emotio
nal_joy', y='listing_emotional_fear', s=24, alpha=0.9, c='green', edgecolor
s='black', ax=ax1)


ax1 = plt.subplot(222)
data['Y_JS'] = KMeans(n_clusters=2, random_state=0).fit_predict(data[['list
ing_emotional_joy','listing_emotional_sadness']])
data.plot.scatter(x='listing_emotional_joy', y='listing_emotional_sadness',
c='Y_JS', s=12, alpha=0.5, cmap=plt.get_cmap('coolwarm'), ax=ax1)
# Good results for 75 guest with tentative reviews
data[data.review_scores_accuracy <= 8].plot.scatter(x='listing_emotional_jo
y', y='listing_emotional_sadness', s=24, alpha=0.5, c='yellow', edgecolors=
'yellow', ax=ax1)
# data[data.review_emotional_sadness >= 0.1].plot.scatter(x='listing_emotio
nal_joy', y='listing_emotional_fear', s=24, alpha=0.9, c='green', edgecolor
s='black', ax=ax1)


ax1 = plt.subplot(223)
data['Y_JD'] = KMeans(n_clusters=2, random_state=0).fit_predict(data[['list
ing_emotional_joy','listing_emotional_disgust']])
data.plot.scatter(x='listing_emotional_joy', y='listing_emotional_disgust',
c='Y_JD', s=12, alpha=0.5, cmap=plt.get_cmap('coolwarm'), ax=ax1)
# Good results for 75 guest with tentative reviews
data[data.review_scores_accuracy <= 8].plot.scatter(x='listing_emotional_jo
y', y='listing_emotional_disgust', s=24, alpha=0.5, c='yellow', edgecolors=
'yellow', ax=ax1)
# data[data.review_emotional_sadness >= 0.1].plot.scatter(x='listing_emotio
nal_joy', y='listing_emotional_fear', s=24, alpha=0.9, c='green', edgecolor
s='black', ax=ax1)


ax1 = plt.subplot(224)
data['Y_JA'] = KMeans(n_clusters=2, random_state=0).fit_predict(data[['list
ing_emotional_joy','listing_emotional_anger']])
data.plot.scatter(x='listing_emotional_joy', y='listing_emotional_anger', c
='Y_JA', s=12, alpha=0.5, cmap=plt.get_cmap('coolwarm'), ax=ax1)
```

```
# Good results for 75 guest with tentative reviews
data[data.review_scores_accuracy <= 8].plot.scatter(x='listing_emotional_jo
y', y='listing_emotional_anger', s=24, alpha=0.5, c='yellow', edgecolors='y
ellow', ax=ax1)
# data[data.review_emotional_sadness >= 0.1].plot.scatter(x='listing_emotio
nal_joy', y='listing_emotional_fear', s=24, alpha=0.9, c='green', edgecolor
s='black', ax=ax1)
```

Out[4]:



In [5]:

```
# Data for the tables in the report
print("Red Joy and Fear")
print("Total of yellow dots ontop of Red class", len(data[data.review_score
s_accuracy <= 8][data.Y_JF == 1]))
print("Total reviews for class Red in general", len(data[data.Y_JF == 1]))
print("Total Listing for class Red", len(set(data[data.Y_JF == 1].listing_i
d)))
print("Blue Joy and Fear")
print("Total of yellow dots ontop of Blue class", len(data[data.review_scor
es_accuracy <= 8][data.Y_JF == 0]))
print("Total reviews for class Blue in general", len(data[data.Y_JF == 0]))
```

```
print("Total Listing for class Blue", len(set(data[data.Y_JF == 0].listing_
id)))
print("Red Joy and Sadness")
print("Total of yellow dots ontop of Red class", len(data[data.review_score
s_accuracy <= 8][data.Y_JS == 1]))
print("Total reviews for class Red in General", len(data[data.Y_JS == 1]))
print("Total Listing for class Red", len(set(data[data.Y_JS == 1].listing_i
d)))
print("Blue Joy and Sadness")
print("Total of yellow dots ontop of Blue class", len(data[data.review_scor
es_accuracy <= 8][data.Y_JS == 0]))
print("Total reviews for class Blue in General", len(data[data.Y_JS == 0]))
print("Total Listing for class Blue", len(set(data[data.Y_JS == 0].listing_
id)))
```

```
Red Joy and Fear
('Total of yellow dots ontop of Red class', 0)
('Total reviews for class Red in general', 748)
('Total Listing for class Red', 19)
Blue Joy and Fear
('Total of yellow dots ontop of Blue class', 86)
('Total reviews for class Blue in general', 26973)
('Total Listing for class Blue', 723)
Red Joy and Sadness
('Total of yellow dots ontop of Red class', 5)
('Total reviews for class Red in General', 3763)
('Total Listing for class Red', 142)
Blue Joy and Sadness
('Total of yellow dots ontop of Blue class', 81)
('Total reviews for class Blue in General', 23958)
('Total Listing for class Blue', 600)
```

In [10]:

```
# Data for the spider chart
print("Joy and Fear Red Class")
print("Avg Joy", data[data.Y_JF == 1].listing_emotional_joy.mean())
print("Avg Fear", data[data.Y_JF == 1].listing_emotional_fear.mean())
print("Avg Sadness", data[data.Y_JF == 1].listing_emotional_sadness.mean())
print("Avg Disgust", data[data.Y_JF == 1].listing_emotional_disgust.mean())
print("Avg Anger", data[data.Y_JF == 1].listing_emotional_anger.mean())
print('')
print("Joy and Fear Blue Class")
print("Avg Joy", data[data.Y_JF == 0].listing_emotional_joy.mean())
print("Avg Fear", data[data.Y_JF == 0].listing_emotional_fear.mean())
print("Avg Sadness", data[data.Y_JF == 0].listing_emotional_sadness.mean())
print("Avg Disgust", data[data.Y_JF == 0].listing_emotional_disgust.mean())
print("Avg Anger", data[data.Y_JF == 0].listing_emotional_anger.mean())
print('')
print("Joy and Sadness Red Class")
```

```
print("Avg Joy", data[data.Y_JS == 1].listing_emotional_joy.mean())
print("Avg Fear", data[data.Y_JS == 1].listing_emotional_fear.mean())
print("Avg Sadness", data[data.Y_JS == 1].listing_emotional_sadness.mean())
print("Avg Disgust", data[data.Y_JS == 1].listing_emotional_disgust.mean())
print("Avg Anger", data[data.Y_JS == 1].listing_emotional_anger.mean())
print('')
print("Joy and Sadness Blue Class")
print("Avg Joy", data[data.Y_JS == 0].listing_emotional_joy.mean())
print("Avg Fear", data[data.Y_JS == 0].listing_emotional_fear.mean())
print("Avg Sadness", data[data.Y_JS == 0].listing_emotional_sadness.mean())
print("Avg Disgust", data[data.Y_JS == 0].listing_emotional_disgust.mean())
print("Avg Anger", data[data.Y_JS == 0].listing_emotional_anger.mean())
print('')
print("Joy and Disgust Red Class")
print("Avg Joy", data[data.Y_JD == 1].listing_emotional_joy.mean())
print("Avg Fear", data[data.Y_JD == 1].listing_emotional_fear.mean())
print("Avg Sadness", data[data.Y_JD == 1].listing_emotional_sadness.mean())
print("Avg Disgust", data[data.Y_JD == 1].listing_emotional_disgust.mean())
print("Avg Anger", data[data.Y_JD == 1].listing_emotional_anger.mean())
print('')
print("Joy and Disgust Blue Class")
print("Avg Joy", data[data.Y_JD == 0].listing_emotional_joy.mean())
print("Avg Fear", data[data.Y_JD == 0].listing_emotional_fear.mean())
print("Avg Sadness", data[data.Y_JD == 0].listing_emotional_sadness.mean())
print("Avg Disgust", data[data.Y_JD == 0].listing_emotional_disgust.mean())
print("Avg Anger", data[data.Y_JD == 0].listing_emotional_anger.mean())
print('')
print("Joy and Anger Red Class")
print("Avg Joy", data[data.Y_JA == 1].listing_emotional_joy.mean())
print("Avg Fear", data[data.Y_JA == 1].listing_emotional_fear.mean())
print("Avg Sadness", data[data.Y_JA == 1].listing_emotional_sadness.mean())
print("Avg Disgust", data[data.Y_JA == 1].listing_emotional_disgust.mean())
print("Avg Anger", data[data.Y_JA == 1].listing_emotional_anger.mean())
print('')
print("Joy and Anger Blue Class")
print("Avg Joy", data[data.Y_JA == 0].listing_emotional_joy.mean())
print("Avg Fear", data[data.Y_JA == 0].listing_emotional_fear.mean())
print("Avg Sadness", data[data.Y_JA == 0].listing_emotional_sadness.mean())
print("Avg Disgust", data[data.Y_JA == 0].listing_emotional_disgust.mean())
print("Avg Anger", data[data.Y_JA == 0].listing_emotional_anger.mean())
print('')
```

```
Joy and Fear Red Class
('Avg Joy', 0.6072930788770052)
('Avg Fear', 0.4821497486631016)
('Avg Sadness', 0.24724770588235293)
('Avg Disgust', 0.16456848262032087)
('Avg Anger', 0.1782727232620321)
```

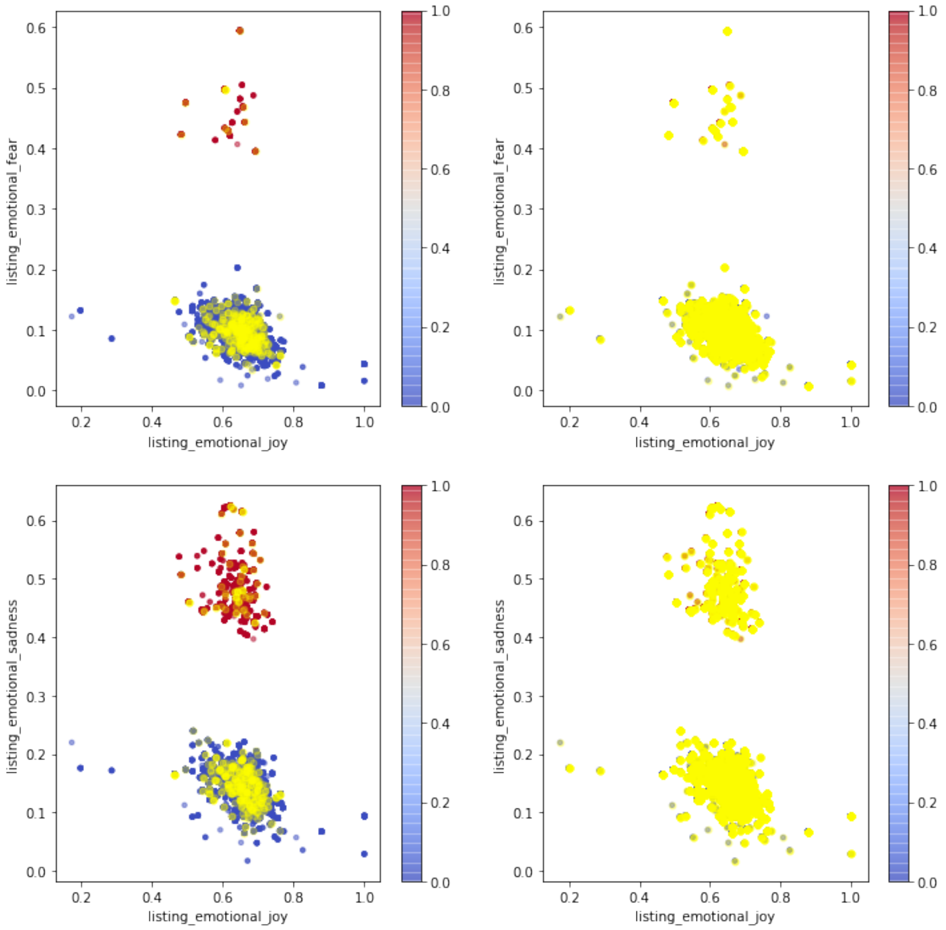
```
Joy and Fear Blue Class
('Avg Joy', 0.6580221829236644)
('Avg Fear', 0.09480621354688022)
('Avg Sadness', 0.18998208145182216)
('Avg Disgust', 0.09200995610425237)
('Avg Anger', 0.09048958451043636)

Joy and Sadness Red Class
('Avg Joy', 0.641647153335105)
('Avg Fear', 0.1170569798033484)
('Avg Sadness', 0.4907360494286474)
('Avg Disgust', 0.11508670156789794)
('Avg Anger', 0.11554323863938346)

Joy and Sadness Blue Class
('Avg Joy', 0.659010323274063)
('Avg Fear', 0.10340473307454712)
('Avg Sadness', 0.1445316058519075)
('Avg Disgust', 0.09065074350947491)
('Avg Anger', 0.08929519797145005)

Joy and Disgust Red Class
('Avg Joy', 0.6272621235632184)
('Avg Fear', 0.15954789272030653)
('Avg Sadness', 0.24470935153256704)
('Avg Disgust', 0.4657307241379311)
('Avg Anger', 0.15311845210727967)

Joy and Disgust Blue Class
('Avg Joy', 0.6578035725906212)
('Avg Fear', 0.1031333362072197)
('Avg Sadness', 0.18944601731828917)
('Avg Disgust', 0.0794189337256813)
('Avg Anger', 0.09049997735877346)

Joy and Anger Red Class
('Avg Joy', 0.6329436536964981)
('Avg Fear', 0.21558412062256813)
('Avg Sadness', 0.3325085680933852)
('Avg Disgust', 0.21175813229571985)
('Avg Anger', 0.5128219844357976)

Joy and Anger Blue Class
('Avg Joy', 0.6571012800014702)
('Avg Fear', 0.10317366016098796)
('Avg Sadness', 0.18886384250376742)
('Avg Disgust', 0.09174249608556621)
('Avg Anger', 0.08492421288638954)
```

In [43]:

```
# This creates data-blocks that we can manipulate.
data = listing_vs_reviews_tones_matrix #listings_details_analyised
X = data[['listing_emotional_fear','listing_emotional_joy']].values
y = data.review_scores_rating.values

# Create the PCA, give the data to the PCA and `fit` the analysis.
pca = PCA(n_components=1)
pca.fit(X)
# Transform the original data to new data.
X_pca = pca.transform(X)
# Store the data in the original data-frame.
data['trust-1'] = X_pca[:,0]

SELECTED_COLUMN_FOR_COLOR = 'trust-1'

plt.figure(figsize=(12,12))
ax1 = plt.subplot(221)
Y = KMeans(n_clusters=2, random_state=0).fit_predict(data[['listing_emotion
al_joy','listing_emotional_fear']])
data.plot.scatter(x='listing_emotional_joy', y='listing_emotional_fear', c=
Y, s=12, alpha=0.5, cmap=plt.get_cmap('coolwarm'), ax=ax1)
# Good results for 75 guest with tentative reviews
data[data.review_language_tentative >= 0.5].plot.scatter(x='listing_emotion
al_joy', y='listing_emotional_fear', s=24, alpha=0.3, c='yellow', edgecolor
s='yellow', ax=ax1)


ax1 = plt.subplot(222)
Y = KMeans(n_clusters=2, random_state=0).fit_predict(data[['listing_emotion
al_joy','listing_emotional_fear']])
data.plot.scatter(x='listing_emotional_joy', y='listing_emotional_fear', c=
Y, s=12, alpha=0.5, cmap=plt.get_cmap('coolwarm'), ax=ax1)
# Good results for 75 guest with tentative reviews
data[data.review_language_tentative < 0.5].plot.scatter(x='listing_emotiona
l_joy', y='listing_emotional_fear', s=24, alpha=0.3, c='yellow', edgecolors
='yellow', ax=ax1)


ax1 = plt.subplot(223)
Y = KMeans(n_clusters=2, random_state=0).fit_predict(data[['listing_emotion
al_joy','listing_emotional_sadness']])
data.plot.scatter(x='listing_emotional_joy', y='listing_emotional_sadness',
c=Y, s=12, alpha=0.5, cmap=plt.get_cmap('coolwarm'), ax=ax1)
# Good results for 75 guest with tentative reviews
data[data.review_language_tentative >= 0.5].plot.scatter(x='listing_emotion
al_joy', y='listing_emotional_sadness', s=24, alpha=0.3, c='yellow', edgeco
lors='yellow', ax=ax1)
```

```
ax1 = plt.subplot(224)
Y = KMeans(n_clusters=2, random_state=0).fit_predict(data[['listing_emotion
al_joy','listing_emotional_sadness']])
data.plot.scatter(x='listing_emotional_joy', y='listing_emotional_sadness',
c=Y, s=12, alpha=0.5, cmap=plt.get_cmap('coolwarm'), ax=ax1)
# Good results for 75 guest with tentative reviews
data[data.review_language_tentative < 0.5].plot.scatter(x='listing_emotiona
l_joy', y='listing_emotional_sadness', s=24, alpha=0.3, c='yellow', edgecol
ors='yellow', ax=ax1)
```

Out[43]:

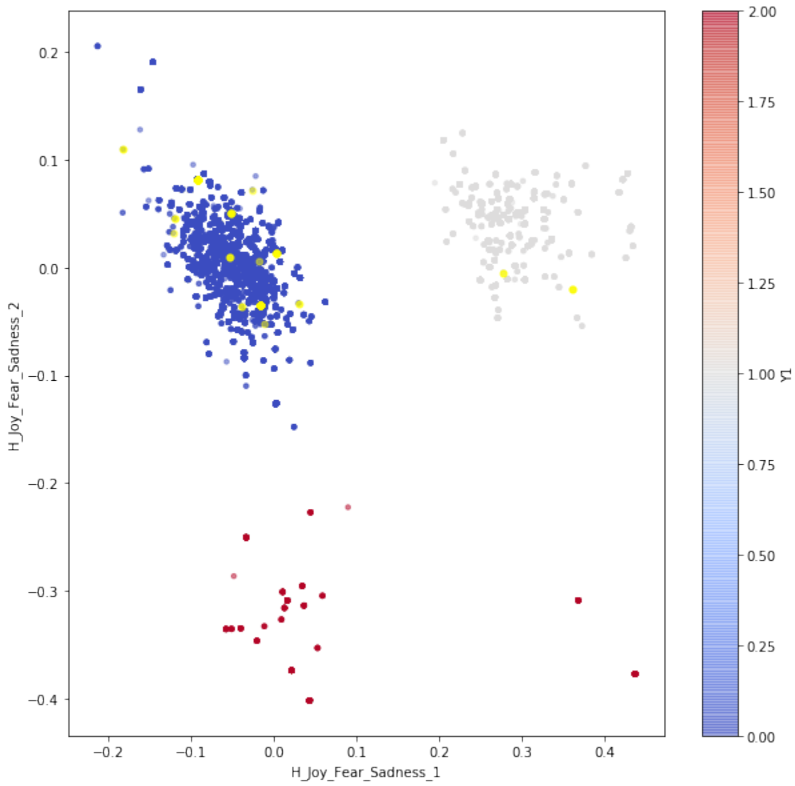


In [3]:

```
# This creates data-blocks that we can manipulate.
data = listing_vs_reviews_tones_matrix #listings_details_analysed
X1 = data[['listing_emotional_sadness','listing_emotional_fear','listing_em
otional_joy']].values
X2 = data[['review_emotional_sadness','review_emotional_fear','review_emoti
onal_joy']].values
y = data.review_scores_rating.values
# Create the PCA, give the data to the PCA and `fit` the analysis.
pca1 = PCA(n_components=2)
```

```
pca2 = PCA(n_components=2)
pca1.fit(X1)
pca2.fit(X2)
# Transform the original data to new data.
X1_pca = pca1.transform(X1)
X2_pca = pca2.transform(X2)
# Store the data in the original data-frame.
data['H_Joy_Fear_Sadness_1'], data['H_Joy_Fear_Sadness_2'] = X1_pca[:,0], X
1_pca[:,1]
data['G_Joy_Fear_Sadness_1'], data['G_Joy_Fear_Sadness_2'] = X2_pca[:,0], X
2_pca[:,1]


SELECTED_COLUMN_FOR_COLOR = 'trust-1'
data['Y1'] = KMeans(n_clusters=3, random_state=0).fit_predict(data[['H_Joy_
Fear_Sadness_1','H_Joy_Fear_Sadness_2']])
data['Y2'] = KMeans(n_clusters=3, random_state=0).fit_predict(data[['G_Joy_
Fear_Sadness_1','G_Joy_Fear_Sadness_2']])

plt.figure(figsize=(22,22))
# ax1 = plt.subplot(221)
# data.plot.scatter(x='Joy_Fear_Sadness_1', y='Joy_Fear_Sadness_2', c=Y, s=
12, alpha=0.5, cmap=plt.get_cmap('coolwarm'), ax=ax1)
# Good results for 75 guest with tentative reviews
# data[data.review_language_tentative >= 0.5].plot.scatter(x='Joy_Fear_Sadn
ess_1', y='Joy_Fear_Sadness_2', s=24, alpha=0.3, c='black', edgecolors='yel
low', ax=ax1)

ax1 = plt.subplot(221)
data.plot.scatter(x='H_Joy_Fear_Sadness_1', y='H_Joy_Fear_Sadness_2', c='Y1
', s=12, alpha=0.5, cmap=plt.get_cmap('coolwarm'), ax=ax1)
# Good results for 75 guest with tentative reviews
# data[data.Y2 == 2][data.review_language_tentative >=0.5].plot.scatter(x='
H_Joy_Fear_Sadness_1', y='H_Joy_Fear_Sadness_2', s=24, alpha=0.3, c='yellow
', edgecolors='yellow', ax=ax1)
data[data.review_scores_accuracy <= 8].plot.scatter(x='H_Joy_Fear_Sadness_1
', y='H_Joy_Fear_Sadness_2', s=24, alpha=0.5, c='yellow', edgecolors='yello
w', ax=ax1)

# ax1 = plt.subplot(222)
# data.plot.scatter(x='G_Joy_Fear_Sadness_1', y='G_Joy_Fear_Sadness_2', c='
Y2', s=12, alpha=0.5, cmap=plt.get_cmap('coolwarm'), ax=ax1)
```

Out[3]:



In [4]:

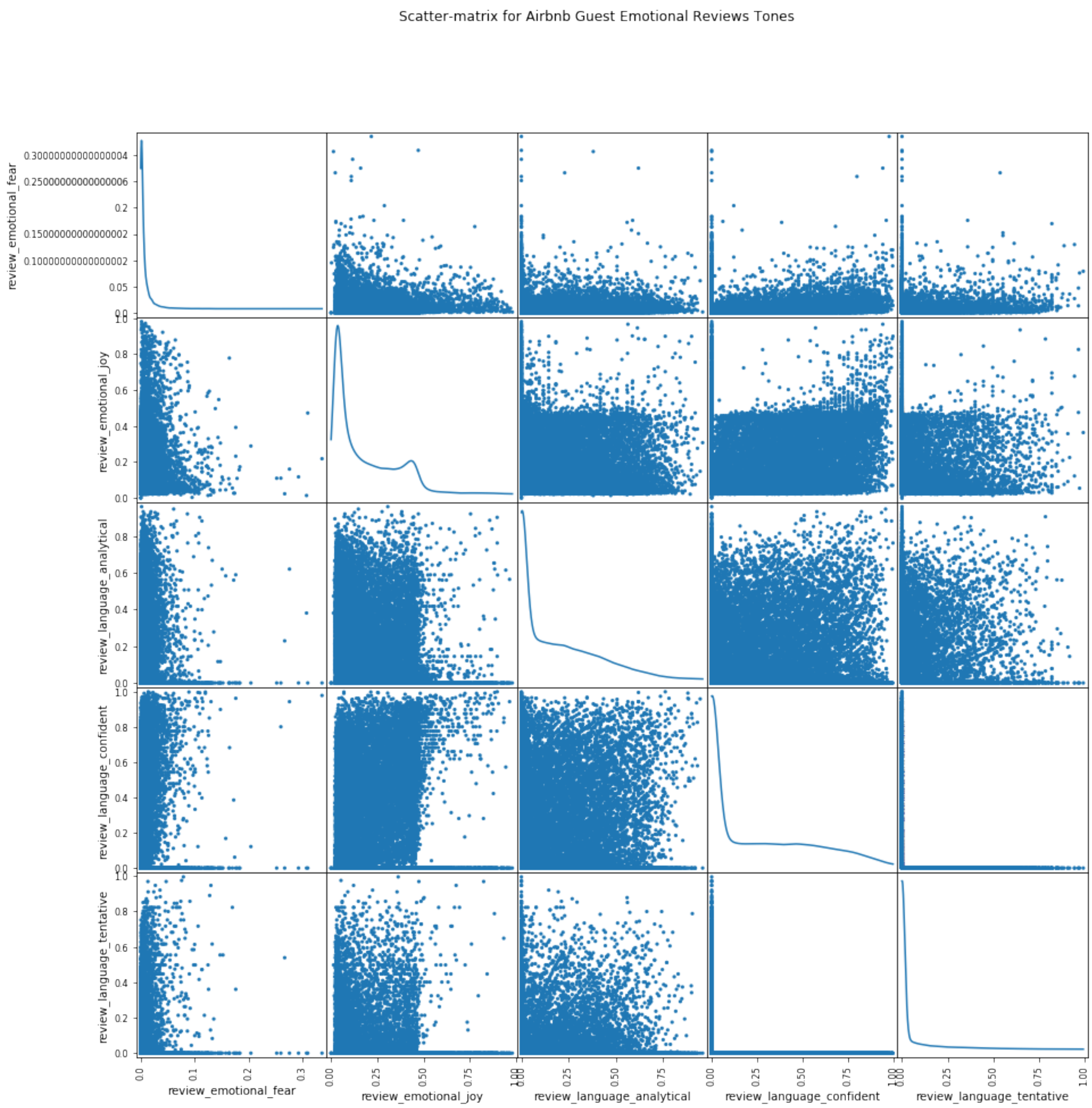
```
print("Joy Fear and Sadness Red Class")
print("Avg Joy", data[data.Y1 == 2].listing_emotional_joy.mean())
print("Avg Fear", data[data.Y1 == 2].listing_emotional_fear.mean())
print("Avg Sadness", data[data.Y1 == 2].listing_emotional_sadness.mean())
print("Avg Disgust", data[data.Y1 == 2].listing_emotional_disgust.mean())
print("Avg Anger", data[data.Y1 == 2].listing_emotional_anger.mean())
print('')
print("Joy Fear and Sadness Gray Class")
print("Avg Joy", data[data.Y1 == 1].listing_emotional_joy.mean())
print("Avg Fear", data[data.Y1 == 1].listing_emotional_fear.mean())
print("Avg Sadness", data[data.Y1 == 1].listing_emotional_sadness.mean())
print("Avg Disgust", data[data.Y1 == 1].listing_emotional_disgust.mean())
print("Avg Anger", data[data.Y1 == 1].listing_emotional_anger.mean())
print('')
print("Joy Fear and Sadness Blue Class")
print("Avg Joy", data[data.Y1 == 0].listing_emotional_joy.mean())
print("Avg Fear", data[data.Y1 == 0].listing_emotional_fear.mean())
print("Avg Sadness", data[data.Y1 == 0].listing_emotional_sadness.mean())
print("Avg Disgust", data[data.Y1 == 0].listing_emotional_disgust.mean())
print("Avg Anger", data[data.Y1 == 0].listing_emotional_anger.mean())
```

```
Joy Fear and Sadness Red Class
('Avg Joy', 0.6008097526315789)
('Avg Fear', 0.47661015921052635)
('Avg Sadness', 0.24618500657894735)
```

```
('Avg Disgust', 0.16386698289473686)
('Avg Anger', 0.17745376710526317)

Joy Fear and Sadness Gray Class
('Avg Joy', 0.6433625152862702)
('Avg Fear', 0.09727582073374097)
('Avg Sadness', 0.4875770063924402)
('Avg Disgust', 0.09905037854363535)
('Avg Anger', 0.09840571734296831)

Joy Fear and Sadness Blue Class
('Avg Joy', 0.6605167924067971)
('Avg Fear', 0.09440713461456147)
('Avg Sadness', 0.14415637088558833)
('Avg Disgust', 0.0909112529212858)
('Avg Anger', 0.08925201926122502)
```

In [29]:

```
feature_names = ['review_emotional_fear','review_emotional_joy','review_lan
guage_analytical','review_language_confident','review_language_tentative']
X = reviews_details_analyised[feature_names]
cmap = cm.get_cmap('gnuplot')
scatter = pd.plotting.scatter_matrix(X, alpha=1, marker = 'o', s=10, hist_k
wds={'bins':15}, figsize=(15, 15), cmap=cmap, diagonal='kde')
plt.suptitle('Scatter-matrix for Airbnb Guest Emotional Reviews Tones')
plt.savefig('Airbnb_Guest_Emotional_Reviews_Tones')
```



In [9]:

```
# data = listing_vs_reviews_tones_matrix #listings_details_analyised

# print("number of listings : ", len(set(data.listing_id)))
# print("number of Reviews review_language_tentative > 0.5: ", len(data[dat
a.review_language_tentative >= 0.5]))
# print('')
# print("number of Listings listing_emotional_fear >= 0.35: ", len(set(data
[data.listing_emotional_fear >= 0.35].listing_id)))
# print("number of Listings listing_emotional_fear < 0.35: ", len(set(data[
data.listing_emotional_fear < 0.35].listing_id)))
# print("number of Reviews review_language_tentative >= 0.5 AND listing_emo
tional_fear >= 0.35: ", len(data[data.review_language_tentative >= 0.5][dat
a.listing_emotional_fear >= 0.35]))
```

```
# print("number of Reviews review_language_tentative >= 0.5 AND listing_emo
tional_fear < 0.35: ", len(data[data.review_language_tentative >= 0.5][data
.listing_emotional_fear < 0.35]))

# print('')
# print("number of Listings listing_emotional_sadness >= 0.35: ", len(set(d
ata[data.listing_emotional_sadness >= 0.35].listing_id)))
# print("number of Listings listing_emotional_sadness < 0.35: ", len(set(da
ta[data.listing_emotional_sadness < 0.35].listing_id)))
# print("number of Reviews review_language_tentative >= 0.5 AND listing_emo
tional_sadness >= 0.35: ", len(data[data.review_language_tentative >= 0.5][
data.listing_emotional_sadness >= 0.35]))
# print("number of Reviews review_language_tentative >= 0.5 AND listing_emo
tional_sadness < 0.35: ", len(data[data.review_language_tentative >= 0.5][d
ata.listing_emotional_sadness < 0.35]))
# print('')
# print("number of Reviews for Listings when listing_emotional_fear >= 0.3:
", len(data[data.listing_emotional_fear >= 0.3]))
# print("number of Reviews for Listings when listing_emotional_fear >= 0.35
AND review_language_tentative > 0.5: ", len(data[data.listing_emotional_fea
r >= 0.35][data.review_language_tentative >= 0.5]))
# print("number of Reviews for Listings when review_emotional_fear >= 0.2 A
ND review_emotional_joy > 0.4: ", len(data[data.review_emotional_fear >= 0.
2][data.review_emotional_joy >= 0.4]))


# print("Total of yellow dots ontop of Red class", len(data[data.review_lan
guage_tentative >= 0.5][data.Y1 == 2]))
# print("Total reviews for class Red in general", len(data[data.Y1 == 2]))
# print("Total Listing for class Red", len(set(data[data.Y1 == 21].listing_
id)))
# print('')
# print("Total of yellow dots ontop of Gray class", len(data[data.review_la
nguage_tentative >= 0.5][data.Y1 == 1]))
# print("Total reviews for class Gray in general", len(data[data.Y1 == 1]))
# print("Total Listing for class Gray", len(set(data[data.Y1 == 1].listing_
id)))
# print('')
# print("Total of yellow dots ontop of Blue class", len(data[data.review_la
nguage_tentative >= 0.5][data.Y1 == 0]))
# print("Total reviews for class red in Blue", len(data[data.Y1 == 0]))
# print("Total Listing for class Blue", len(set(data[data.Y1 == 0].listing_
id)))
# print('')

# print("Total of yellow dots ontop of Red class", len(data[data.review_lan
guage_tentative >= 0.3][data.Y1 == 2][data.Y2 == 2]))
# print("Total of yellow dots ontop of Red class", len(data[data.review_lan
guage_tentative >= 0.3][data.Y1 == 2][data.Y2 == 1]))
```

```
# print("Total of yellow dots ontop of Red class", len(data[data.review_lan
guage_tentative >= 0.3][data.Y1 == 2][data.Y2 == 0]))
# print("Total reviews for class Red in general", len(data[data.Y1 == 2]))
# print("Total Listing for class Red", len(set(data[data.Y1 == 2].listing_i
d)))
# print('')
# print("Total of yellow dots ontop of Gray class", len(data[data.review_la
nguage_tentative >= 0.3][data.Y1 == 1][data.Y2 == 2]))
# print("Total of yellow dots ontop of Gray class", len(data[data.review_la
nguage_tentative >= 0.3][data.Y1 == 1][data.Y2 == 1]))
# print("Total of yellow dots ontop of Gray class", len(data[data.review_la
nguage_tentative >= 0.3][data.Y1 == 1][data.Y2 == 0]))
# print("Total reviews for class Gray in general", len(data[data.Y1 == 1]))
# print("Total Listing for class Gray", len(set(data[data.Y1 == 1].listing_
id)))
# print('')
# print("Total of yellow dots ontop of Blue class", len(data[data.review_la
nguage_tentative >= 0.3][data.Y1 == 0][data.Y2 == 2]))
# print("Total of yellow dots ontop of Blue class", len(data[data.review_la
nguage_tentative >= 0.3][data.Y1 == 0][data.Y2 == 1]))
# print("Total of yellow dots ontop of Blue class", len(data[data.review_la
nguage_tentative >= 0.3][data.Y1 == 0][data.Y2 == 0]))
# print("Total reviews for class Blue in General", len(data[data.Y1 == 0]))
# print("Total Listing for class Blue", len(set(data[data.Y1 == 0].listing_
id)))

print("Total of yellow dots ontop of Red class", len(data[data.review_score
s_accuracy <= 9][data.Y1 == 2]))
print("Total reviews for class Red in general", len(data[data.Y1 == 2]))
print("Total Listing for class Red", len(set(data[data.Y1 == 2].listing_id)
))
print('')
print("Total of yellow dots ontop of Gray class", len(data[data.review_scor
es_accuracy <= 9][data.Y1 == 1]))
print("Total reviews for class Gray in general", len(data[data.Y1 == 1]))
print("Total Listing for class Gray", len(set(data[data.Y1 == 1].listing_id
)))
print('')
print("Total of yellow dots ontop of Blue class", len(data[data.review_scor
es_accuracy <= 9][data.Y1 == 0]))
print("Total reviews for class Blue in General", len(data[data.Y1 == 0]))
print("Total Listing for class Blue", len(set(data[data.Y1 == 0].listing_id
)))

print("")
# print("Total reviews in Red", len(data[data.review_language_tentative >=
0.3][data.Y2 == 2]))
# print("Total reviews in Gray", len(data[data.review_language_tentative >=
0.3][data.Y2 == 1]))
```

```
# print("Total reviews in Blue", len(data[data.review_language_tentative >=
0.3][data.Y2 == 0]))
print("Total reviews score <= 9", len(data[data.review_scores_accuracy <= 9
]))

print("Avg reviews", data.review_language_tentative.mean())

data.review_scores_accuracy.mean()
('Total of yellow dots ontop of Red class', 36)
('Total reviews for class Red in general', 760)
('Total Listing for class Red', 21)

('Total of yellow dots ontop of Gray class', 655)
('Total reviews for class Gray in general', 3598)
('Total Listing for class Gray', 140)

('Total of yellow dots ontop of Blue class', 4302)
('Total reviews for class Blue in General', 23363)
('Total Listing for class Blue', 581)

('Total reviews score <= 9', 4993)
('Avg reviews', 0.041682734677768118)
```

REFERENCES

Accenture. (2015). Unleashing business value in a digital world. Available : https://wecfinland.fi/wp-content/uploads/2016/11/Accenture-for-World-Energy-Conuncil-v1.0-final.pdf

Adams, B.D., Thomson, M., Brown, A., Sartori, J., Taylor,T., & Waldherr, S. (2008). Organizational trust in the Canadian forces. DRDC Toronto No. CR-2008-038.

AlSheikh, S. S., Shaalan, K., & Meziane, F. (2017). Consumers' trust and popularity of negative posts in social media: A case study on the integration between B2C and C2C business models. 2017 International Conference on Behavioral, Economic, Socio-cultural Computing (BESC), pp. 1-6. doi:10.1109/besc.2017.8256364

Arnold, M.B. (ed.). (2013). Feelings and emotions: The Loyola symposium, vol. 7. New York: Academic Press.

Beck, R., Czepluchm J. S, Lollike, N., & Malone, S. (2016). Blockchain–The gateway to trust-free cryptographic transactions. Available: https://pdfs.semanticscholar.org/ee1e/fd77e8b6287438d312b244177bb143f7a072.pdf

Berezina, K., Bilgihan, A., Cobanoglu, C., & Okumus, F. (2016). Understanding satisfied and dissatisfied hotel customers: text mining of online hotel reviews. Journal of Hospitality Marketing & Management, vol. 25, no.1, pp. 1-24.

Bigley, G.A. & Pearce, J.L. (1998). Straining for shared meaning in organization science: Problems of trust and distrust. Academy of Management Review, vol. 23, pp. 405-421.

Braga, D.D., Niemann, M., Hellingrath, B., & Neto, F.B. (2018). Survey on computational trust and reputation models. ACM Computing Surveys, vol. 51, no 101, pp.1-40.

Brody, S., & Elhadad, N. (2010). An unsupervised aspect-sentiment model for online reviews. Proceedings of HLT, Los Angeles, CA, USA, pp. 804-812. Available: https://acl-arc.comp.nus.edu.sg//~antho/N/N10/N10-1122.pdf

Bjørkelund, E., Burnett, T. H., & Nørvåg, K. (2012A study of opinion mining and visualization of hotel reviews. Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services (pp. 229-238). ACM.

Burnett, J. (2012). Understanding buyer behaviour. Chapter 4, pp. 1-26. Available: https://resources.saylor.org/wwwresources/archived/site/wp-content/uploads/2012/12/BUS203-3.2_Understanding-buyer-behavior.pdf

Cao, Q., Duan, W., & Gan, Q. (2011). Exploring determinants of voting for the "helpfulness" of online user reviews: A text-mining approach. Decision Support Systems, vo. 50, no. 2, pp. 511-521. doi:10.1016/j.dss.2010.11.009

Chuang, J., Ramage, D., Manning, C. D., & Heer, J. (2012). Interpretation and trust: Designing model-driven visualizations for text analysis. CHI'12, May 5–10, 2012, Austin, Texas, USA. Available: http://vis.stanford.edu/files/2012-InterpretationTrust-CHI.pdf

Colquitt, J.A., Scott, B.A., & LePine, J.A. (2007). Trust, trustworthiness, and trust propensity: A meta-analytic test of their unique relationships with risk taking and job performance. Journal of Applied Psychology, vol. 92, pp. 909-927.

Corbitt, B. J., Thanasankit, T., Yi, H. (2003). Trust and e-commerce: A study of consumer perceptions. Electronic Commerce Research and Applications, vol. 2, no. 3, pp. 203-215. doi:10.1016/S1567-4223(03)00024-3

Dang, Q.-V. (2018). Trust assessment in large-scale collaborative systems. PhD Thesis. Lorrain University. Available : https://hal.archives-ouvertes.fr/tel-01634377v3/document

Davis, J. H., Schoorman, F. D., Mayer, R. C., & Tan, H. H. (2000). The trusted general manager and business unit performance: Empirical evidence of a competitive advantage. Strategic Management Journal, vol. 21, pp. 563-576.

Egger, F. N. (2003). F Rom I Nteractions to T Ransactions : Designing the Trust Experience for Business-to-Consumer Electronic Commerce. Personal Relationships. http://doi.org/ISBN 90-386-1778-X.

Ekman, P. (1992). An argument for basic emotions. Cognition and Emotion, vol. 6, no. 3-4, pp. 169–200

Ekman, P. (1972). Universals and cultural differences in facial expressions of emotion. In J. Cole (Ed.), Nebraska Symposium on Motivation, vol. 19, pp. 207–282. Lincoln: University of Nebraska Press.

Fradkin, A., Grewal, E., & Holtz, D. (2018). The determinants of online review informativeness: Evidence from field experiments on Airbnb. MIT Sloan School of Management, pp. 1–61. doi:10.2139/ssrn.2939064

Fradkin, A., Grewal, E., Holtz, D., & Pearson, M. (2015). Bias and reciprocity in online reviews : Evidence From Field Experiments on Airbnb. MIT Sloan School of Management, pp. 1–38. doi:10.1145/2764468.2764528

Gain, U., & Hotti, V. (2017). Tones and traits: Experiments of text-based extractions with cognitive services. Finnish Journal of e-Health and e-Welfare, vol. 9, no. 2-3, pp. 82-93. doi: 10.23996/fjhw.61001

Gefen, D., Benbasat, I., & Pavlou, P. (2014). A research agenda for trust in online environments. Journal of Management Information Systems, vol. 24, no. 4, pp. 275-286, doi: 10.2753/MIS0742-1222240411

Gómez Mármol, F., & Martínez Pérez, G. (2011). Providing trust in wireless sensor networks using a bio-inspired technique. Telecommunication Systems, vol. 46, no. 2, pp. 163–180. doi:10.1007/s11235-010-9281-7

Han, H. J., Mankad, S., Gavirneni, N., & Verma, R. (2016). What guests really think of your hotel: Text analytics of online customer reviews. Cornell Hospitality Report, vol. 16, no. 2, pp. 3-17.

Head, M., & Hassanein, K. (2002). Trust in e-commerce: Evaluating the impact of third-party seals. Quarterly Journal of Electronic Commerce, vol. 3, no. 3, pp. 307-325.

Hu, Y. H., Chen, Y. L., & Chou, H. L. (2017). Opinion mining from online hotel reviews–A text summarization approach. Information Processing & Management, vol. 53, no. 2, pp. 436-449.

Hurley, R.F. (2006). The decision to trust. Harvard Business Review. Available: https://hbr.org/2006/09/the-decision-to-trust

Kamvar, S. D., Schlosser, M. T., & Garcia-Molina, H. (2003). The eigentrust algorithm for reputation management in P2P networks. WWW2003, May 20–24, 2003, Budapest, Hungary. Available: http://ilpubs.stanford.edu:8090/562/1/2002-56.pdf

Keser, C. (2002). Trust and reputation building in e-commerce. CIRANO Working Papers 2002s-75. Available: https://cirano.qc.ca/files/publications/2002s-75.pdf

Kramer, R. 1999. Trust and distrust in organizations: Emerging perspectives, enduring questions. Annual Review of Psychology, vol. 50, pp. 569-598

Kramer, R. M., T. R. Tyler (Eds.). 1996. *T*rust in organizations: Frontiers of theory and research. Thousand Oaks, CA: Sage Publications

Latif, I., & Jaffry, S. W. (2013). Trust evaluation mechanisms for Wikipedia. Proceedings of the IJCNLP 2013 Workshop on Natural Language Processing for Social Media (SocialNLP), October, pp. 36-42. Available: https://www.aclweb.org/anthology/W13-4206.pdf

Lazarus, R.S. (1991). *Emotion and adaptation*. New York: Oxford University Press.

Lazarus, R. S., & Folkman, S. (1986). Cognitive theories of stress and the issue of circularity. In M. H. Appley and R. Trumbull (Eds.), Dynamics of Stress. Physiological, Psychologcal, and Social Perspectives (pp. 63–80). New York: Plenum.

Lee, P. J., Hu, Y. H., & Lu, K. T. (2018). Assessing the helpfulness of online hotel reviews: A classification-based approach. Telematics and Informatics, vol. 35, no. 2, pp. 436-445. doi:1016/j.tele.2018.01.001

Mármol, F. G., & Pérez, G. M. (2011). Trust and reputation models comparison. Internet Research, vol. 21, no. 2, pp. 138-153. doi:10.1108/10662241111123739

Mayer, R. C., Davis, J. H., & Schoorman, F. D. (1995). An integrative model of organizational trust. Academy of Management Review, vol. 20, no. 3, pp. 709–734.

Metropolitan Police, M. (2017). Freedom of information request. Available: https://tfl.gov.uk/corporate/safety-and-security/security-on-the-network/tph-related-sexual-offences

Meziane, F., & Kasiran, M. K. (2008). Evaluating trust in electronic commerce: A study based on the information provided on merchants' websites. Journal of the Operational Research Society, vol. 59, no. 4, pp. 464-472. doi:10.1057/palgrave.jors.2602430

Mui, L. (2002). Computational models of trust and reputation: Agents, evolutionary games, and social networks. PhD thesis. MIT. Available: https://groups.csail.mit.edu/medg/people/lmui/docs/phddissertation.pdf

Nguyen, T. (2017). Trust management for complex agent groups. PhD thesis. doi:10.13140/RG.2.2.18070.80967

Niranjanamurthy, M., Kavyashree, N., & Chahar, S. J. D. (2013). Analysis of e-commerce and m-commerce: Advantages, limitations, and security issues. International Journal of Advanced Research in Computer and Communication Engineering, vol. 2, no. 6, pp. 2360-2370.

O'Donovan, J., Smyth, B., Evrim, V., & McLeod, D. (2007). Extracting and visualizing trust relationships from online auction feedback comments. Proceedings of IJCAI, San Francisco, CA, USA, pp. 2826-2831.

Pennanen, K., Kaapu, T., & Paakki, M. K. (2006). Trust, risk, privacy, and security in e-Commerce. Proceedings of the International Conference on Electronic Business (ICEB). Available: https://pdfs.semanticscholar.org/346b/bd5d14338d4d5cbaa5e91c94f089480456a5.pdf

Piorkowski, J., & Zhou, L. (2011). A text-based model for identifying online trust relationships. AMCIS 2011 Proceedings. Paper 459. https://aisel.aisnet.org/amcis2011_submissions/459

Plutchik, R. (1980). A general psychoevolutionary theory of emotion. In R. Plutchik & H. Kellerman (Eds.). *Emotion. Theory, research, and experience* (vol.1, pp. 3-33*).* New York, NY: Academic Press.

Plutchik, R. (2001). The nature of emotions: Human emotions have deep evolutionary roots. American Scientist, vol. 89, no. 4, pp. 344-350.

PwC. (2014). The confidence to take risks. Available: https://www.pwc.co.uk/issues/cyber-security-data-privacy/insights/building-digital-trust-to-take-risks.html

Rangari, P. A., & Waghmare, K. C. (2015). A survey on aspect-based opinion mining. International Journal of Science, Engineering and Technology Research (IJSETR), vol. 4, no. 3, pp. 517-520. Available: http://ijsetr.org/wp-content/uploads/2015/03/IJSETR-VOL-4-ISSUE-3-517-520.pdf

Resnick, P., Zeckhauser, R., Friedman, E., & Kuwabara, K. (2000). Reputation systems: Facilitating trust in internet interactions. Communications of the ACM, vol. 43, no.12, pp. 45-48. Available: http://onemvweb.com/sources/sources/reputations.pdf

Rousseau, D., Sitkin, S., Burt, R., & Camerer, C. (1998). Not so different after all: A cross-discipline view of trust. Academy of Management Review, vol. 23, pp. 387-392.

Roy, J., Lewicki, E. Tomlinson, C., & Gillespie, N. (2006). Models of interpersonal trust development: Theoretical approaches, empirical evidence, and future directions. Journal of Management, vol. 32, pp. 991-1022.

Sahney, S. (2008). Critical success factors in online retail – An application of quality function deployment and interpretive structural modeling. International Journal of Business and Information, vol. 3, no. 1, pp. 144-163.

Saura, J.R. and Bennett, D. (2019). A three-stage methodological process of data text mining: A UGC business intelligence analysis. *Symmetry-Basel* vol. 11, no. 4, p. 519. doi:10.13140/RG.2.2.11093.06880

Scherer, K. (2005).What are emotions? And how can they be measured? Social Science Information, vol. 44, no. 4, pp. 695-729.

Scherer, K.R., Schorr, A., Johnstone, T. (Eds.) (2001) Appraisal processes in emotion. Oxford: Oxford University Press.

Simons, T., & McLean Parks, J. (2002). Empty words: The impact of perceived managerial integrity on employees, customers and profits. Paper presented at the annual meeting of the Academy of Management, Denver, CO.

Sindhwani, V., & Melville, P. (2008). Document-word co-regularization for semi-supervised sentiment analysis. Proceedings - IEEE International Conference on Data Mining, ICDM, pp. 1025–1030. doi:10.1109/ICDM.2008.113

Teubner, T., Hawlitschek, F., and Dann, D. (2017). Price determinants on Airbnb: How reputation pays off in the sharing economy. Journal of Self-Governance and Management Economics, vol. 5, no. 4, pp. 53-80.

Wang, Y., & Singh, M. (2006). Trust representation and aggregation in a distributed agent system. Proceedings of AAAI, pp. 1425-1430. Available: https://www.aaai.org/Papers/AAAI/2006/AAAI06-224.pdf

Wu, T-Y., & Lin, C.A. (2016). Predicting the effects of eWOM and online brand messaging: Source trust, bandwagon effect and innovation adoption factors. Telematics and Informatics, vol. 34, no. 2, pp. 470-480. doi:10.1016/j.tele.2016.08.001

Ye, Q., Zhang, Z., & Law, R. (2009). Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. Expert Systems with Applications, vol. 36, no.3, pp. 6527-6535.

Yoldas, S. (2011). Research about buying behaviour among cnline customers: Comparison of Turkey with UK. Master's thesis. University of Roehampton. Available: http://schillermadrid.edu/library-catalog/opac_css/doc_num.php?explnum_id=16

Zhang, X., Cui, L., & Wang, Y. (2014). Commtrust: Computing multi-dimensional trust by mining e-commerce feedback comments. IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 7, pp. 1631-1643. doi:10.1109/TKDE.2013.177

Zhou, R., & Hwang, K. (2007). PowerTrust: A robust and scalable reputation system for trusted peer-to-peer computing. IEEE Transactions on Parallel and Distributed Systems, vol. 18, no. 4, pp. 460-73.

Jian, Zhao, Liang Gou, Wang Fei, and Michelle X. Zhou. PEARL: An Interactive Visual Analytic Tool for Understanding Personal Emotion Style Derived from Social Media. In Proceedings of IEEE VAST (2014).