PROYECTO FIN DE CARRERA

Título:	Caracterización de la Reputación mediante Análisis de Redes Sociales
Título (inglés):	Reputation characterisation using Social Network Analysis
Autor:	Antonio José Prada Blanco
Tutor:	Jose Ignacio Fernández Villamor
Departamento:	Ingeniería de Sistemas Telemáticos

MIEMBROS DEL TRIBUNAL CALIFICADOR

Presidente:	Mercedes Garijo Ayestarán
Vocal:	Tomás Robles Valladares
Secretario:	Carlos Ángel Iglesias Fernández
Suplente:	Francisco Gonzalez Vidal

FECHA DE LECTURA:

CALIFICACIÓN:

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

Departamento de Ingeniería de Sistemas Telemáticos
Grupo de Sistemas Inteligentes

PROYECTO FIN DE CARRERA

REPUTATION CHARACTERISATION USING SOCIAL NETWORK ANALYSIS

Antonio José Prada Blanco

Junio de 2016

Resumen

La aparición de la *Economía Colaborativa* posibilita nuevas transacciones entre desconocidos: desde compartir coche hasta micro-créditos pasando por alquileres de corta estancia. Estas transacciones suponen una serie de riesgos inherentes para ambas partes por lo que para que ocurran es necesario que ambas *confíen* entre ellas. Para facilitar que aparezca esta confianza, las plataformas de Economía Colaborativa muestran información sobre sus usuarios y sus transacciones pasadas, lo que forma su *reputación*.

Mientras que esta es una solución efectiva, supone un problema para los nuevos usuarios: como no pueden demostrar antiguas transacciones no tienen reputación. Proponemos una solución en la que usamos datos de sus *Redes Sociales Online* para predecir como será su reputación en el futuro: de esta forma es posible mejorar la experiencia para los buenos usuarios y prevenir problemas con usuarios malos o fraudulentos.

El presente proyecto tiene como objetivo *minar fuentes sociales para determinar la reputación de sus usuarios*. Con este fin, se ha realizado la extracción de información de dichos usuarios en dos plataformas sociales, la Red Social Online (Twitter) y la plataforma de Economía Colaborativa (Wallapop). Para ello se han empleado técnicas avanzadas de extracción de información publicada en dichas plataformas que cruzan información entre ambas. Después se han empleado estos datos para entrenar un sistema de aprendizaje automático (*Machine Learning*) que usando los datos de Twitter clasifica entre buenos y malos usuarios en Wallapop. Para acabar se presentan y analizan que características de Twitter han sido importantes para estos algoritmos a la hora de diferenciar entre usuarios en función de su reputación.

Keywords: Confianza, Reputación, Economía Colaborativa, Análisis de datos, Análisis de Redes Sociales, Extracción de datos, Minería de datos, Aprendizaje automático

Abstract

The rise of the *Sharing Economy* has enabled new types of transactions between strangers: from car sharing to micro-lending or short-term house renting. These peer-to-peer transactions come with inherent *risks*, so for they to happen both participants have to *trust* each other. To facilitate building trust, Sharing Economy platforms display feedback from past transactions forming the users *reputation*.

While this solution is very effective, it makes harder for new users to make their first transactions as they are not trusted by others users. We propose a solution to know more about how these new users will behave by analysing their *Online Social Networks*. By identifying key traits from social network accounts we can infer beliefs about the user future reputation on the platform and act accordingly to this information to avoid future problems (as fraud) or allow an easier on-boarding process for good willing users.

The goal of this project is to mine Online Social Networks to predict users behaviour at Sharing Economy Platforms. To make it possible it was necessary to extract users information from both platforms and match them using advanced user matching techniques. This data was used to train machine learning algorithms for the task of classifying Wallapop users according to their reputation just by looking at their Twitter accounts. To finish, the features that were important for the machine learning classifier to make the predictions are presented and analyzed.

Keywords: Trust, Reputation, Sharing Economy, Data Analysis, Social Networks Analysis, Scraping, Data Mining, Machine Learning

Agradecimientos

A mi familia y en especial a mis padres y a mi hermana, porque sin ellos esto no hubiera sido posible.

Al Grupo de Sistemas Inteligentes, y en especial a Carlos por la confianza y el apoyo.

A Traity por todo lo aprendido.

Y a Mónica, por haberme ayudado y apoyado hasta el final.

Gracias.

Contents

List of Figures

Tables Index

CHAPTER 1

Introduction

This chapter provides an introduction to the objectives of this master thesis: looking for user characteristics at Online Social Networks that correlate with their behaviours when participating in peer-to-peer transactions. Later on, it explains how the document is organized.

1.1 Context

Peer-to -peer transactions have always been an important part of the economy: people selling at marketplaces, renting houses, working for others; all of them worked without intermediaries for a long time as long as they happened inside closed communities. But with the rise of information technology the so called *Sharing Economy* appeared, where new technologies (as the *Internet*) enable more complex relationships that could not happen before.

These new transactions came with a problem of *trust*: while in the past people knew and trusted each other, in the Internet users usually don't know nothing about each others which can be problematic.

The new platforms (as *eBay*, *Airbnb* or *Uber*) that appeared tried to fix it by using *reputation*: publicly showing historic data about each user so everybody could decide who was good and who was not worth to transact with, just like when in small communities people knew about others past actions. This reputation profile was mainly made of *past transactions data* and feedback about these transactions.

This solution helped to mitigate the trust problem but it created a new one: while *good users* could be identified, *new users* could not be differentiated from fraudulent ones that just created a new account. To solve it, platforms allow users to also input additional data (identification documents, social network accounts, etc) to show that they are who they say they are and to avoid fraudulent users with multiple accounts.

This master thesis proposes a method to use this social network data to also *predict about users future behaviours* on the platform. By identifying key traits from social network accounts we can infer beliefs about the user future reputation on the platform and act accordingly to this information to avoid future problems (as *fraud*) or allow an easier *on-boarding process* for good willing users.

1.2 Project description

The main purpose of this master thesis is to develop a process to predict future users behaviours on economy sharing platforms just by looking at their Online Social Networks' (OSNs from now on) data.

By inferring users reputation from an alternative source, we facilitate new users experiences and at the same time avoid problems. Users with good intentions should have an

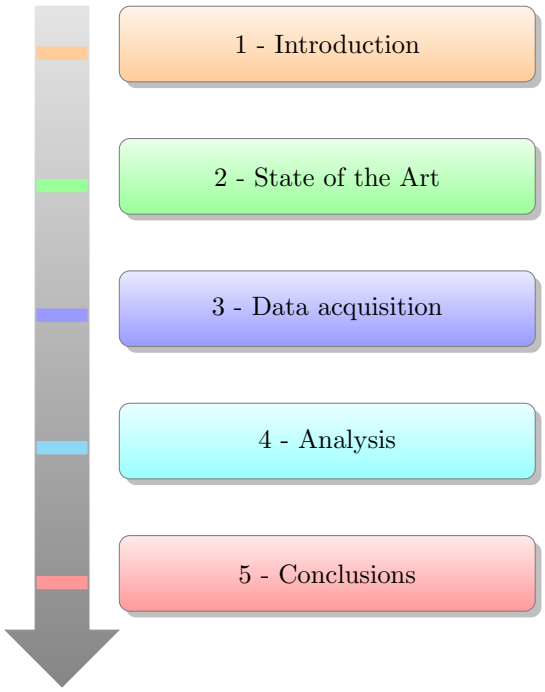
easier on-boarding process and be more trusted while users with bad intentions should be flagged to avoid fraud and discontent among the platform. This process brings a great value by optimising the economies of peer-to-peer transactions, avoiding possible problems and bringing more users to the platform as the trust problem is reduced.

To reach the project goals, two phases have been defined:

- A first phase where we acquire acquire a proper dataset for the later analysis. This dataset will contain pairs of *reputation profiles* and *Online Social Networks profiles*.
- Later on, the analysis will be done looking for user traits in the OSN data that correlate with the reputation of the user at the peer-to-peer marketplace .

1.3 Structure of this Master Thesis

This document has been structured as follows:



- **Chapter 1** provides an introduction to master thesis and its content.
- **Chapter 2** describes the problem and the state of the art of the current solutions.

- **Chapter 3** describes the *data acquisition* process.
- **Chapter 4** describes the *data analysis* process and explains the results.
- **Chapter 5** sums up the findings and conclusions found throughout the document and gives a hint about future development to continue this master thesis work.

CHAPTER 2

State of the art

In this chapter we present the concept of Sharing Economy as an enabler for new types of peer-to-peer transactions. Later on, we identify the risks that come with them and how trust is needed for them to happen. We continue describing what reputation is and why is the approach choosen by the majority of Sharing Economy platforms to address the trust problem. To finish we present our approach to improve current reputation management implementations.

2.1 Overview

Since the beginning of mankind, humans have interchanged goods and services with each others: tools, food, resources, time... These transactions, commonly known as *peer-to-peer* transactions (as they work without the necessity of a third party) were an important factor on the rise of civilisation.

But trading with other people involved some risks: products could be of an unexpected quality, deals could be broken, goods could be stolen... nevertheless it was not much of a problem thanks to people living in small communities where they would knew about everybody past actions: *everyone of them had a reputation.*

With the rise of capitalism and modern societies, communities started to expand and transactions started to happen between strangers. Also, as the reach of the transactions expanded, the inherent risks increased too due to the added uncertainty of not knowing about other's reputation, which made people more vulnerable. Such problems were fixed through *corporate and personal reputations*: vendors provided references, brands were known for their quality and gossip would told you on whom you could rely and whom you could not.

Later on, the rise of the information technologies brought another evolution to the peer-to-peer economy: communities expanded even more and become worldwide. Peer-to-peer transactions started to happen between individuals that not only didn't know anything about each other, but they were not even from the same continent. This phenomenon was called the *Sharing economy* as it enables a set of possible transactions that couldn't happen before, e.g. a lender in the US can now give micro-loans to local farmers in poor villages from Africa or strangers can share a car if they have to travel to the same place at the same time. But *these new transactions brought new risks too.*

In this chapter we present the concept of Sharing Economy as an enabler for new types of peer-to-peer transactions. Later on, we identify the risks that come with such transactions and how trust is needed for them to happen. We continue describing what reputation is and why this approach has been chosen by the majority of Sharing Economy platforms to address the trust problem. To finish we present our approach to improve current reputation management implementations.

2.2 Sharing economy

Sharing economy definition is *the peer-to-peer-based activity of obtaining, giving, or sharing the access to goods and services, coordinated through community-based online services [1].*

In the last decade, along the birth of the so called *web 2.0*, new technological possibilities appeared: better Internet connections, smartphones, online collaboration tools... These technological developments simplified the exchange of both physical and not physical goods and services, enabling new collaborative platforms such as *Github* (an open source code repository) or *Wikipedia* (a collaborative encyclopaedia where everybody can participate).

These multi-sided platforms technologies were the basis for the proliferation of the Sharing Economy platforms. These platforms use technology to allow the use of under-utilised inventory, creating new possibilities and disrupting conventional business. In fact it is expected that the Sharing economy will help to alleviate societal problems as hyper-consumption, pollution and poverty by lowering the cost of economic coordination between communities.

Some of the fields were the Sharing Economy is currently gaining great popularity are:

- **House renting:** while peer-to-peer renting has been always a big part of economy, it's in *short-them rentals* where most of the efforts to disrupt the market have been done. One of the better examples of the new *sharing* culture was the *couchsurfing* movement, where individuals all around the world offered accommodation to strangers for free in exchange of the possibility of doing the same by themselves in the future.

  Later on, *Airbnb* was born as a platform to stay in strangers' properties in exchange for money. Nowadays, it has been estimated that Airbnb's impact on hotels' earnings at some locations surpass the 10% [2]. while its holds a valuation of multiple billions of dollars.

- **Mobility:** in the last years, the notion of sharing bikes, cars or even rides on an on-demand basis have gain popularity. It is interesting not only looking at individuals' economics but also in modern cities context; when population growth, pollution and traffic congestion are everyday problems and these tendencies could be part of the solution. Some examples are: *Uber*, where users drive other users around the city like normal taxis; *Blablacar* for long distance car sharing and *RelayRides* for short term car renting.

- **Marketplaces:** new platforms allow optimise the peer-to-peer selling and buying

market by offering a real-time searchable showcase of products and allowing transactions between strangers in opposites part of the world. An example of the success and popularity of these platforms is *eBay*, funded in 1995 and with more than 160M of users.

- **Others:** as money lending (*Kiwa, Lending club*), pet caring (*DogVacay, DogBuddy*), freelancing (*TaskRabbit, Zaarly*), WiFi sharing (*Fon*)...

What each one of these peer-to-peer platforms have in common is that their transactions have inherent risks that they have to attack from their beginning.

2.3 Risk

The Internet offers us new interaction opportunities: to make money, to access to information, to socialise, etc. But due to anonymity these activities also involve big risks: risk to be scammed, to be given false information, to put ourselves in a dangerous position, etc.

For a sharing economy platform the risks are bigger as the relation is more intimate and complex: to allow a stranger to sleep in your house or to enter in a stranger's car are just two of the multiple possibilities. For these platforms to succeed is necessary that their users feel safe when using it, because without that safety feeling is unlike that people would want to expose themselves to such danger.

A typical example of risks involved in peer-to-peer transactions, specifically the information asymmetry, is the one that Akerlof called *market for lemons*. [3]. Akerlof identified two different types of cars in the used cars market related to a car being an asset with an inevitable wear and tear: *lemons* and *cherries*. A *cherry* is a car that has been well maintained by its owner and a *lemon* is a car that has not been well maintained and will give problems to the owner in the near future.

The information asymmetry problem appears when buying and used car from another person: the buyer don't know what type of car (cherry or lemon) is the one that he is looking at because the quality of the car depends of a set of variables that can not be observed by inspection and depend on the owner: driving style, quality and frequency of maintenance and accident history. So the buyer, facing the uncertainty of the car's quality, will pay an average price between the two qualities. This means that *cherries* owners will be paid below the actual value of the car and *lemons* owners will be paid above its value. In the long run this situation will fill the market with *lemons* instead of *cherries*, lowering

the average price paid for a car again and again, and driving out more and more cherries: *the bad driving out the good.*

When there is uncertainty in a bilateral exchange there is a certain *risk* of at least one of the parties having a negative payoff. This risk can be economical (as the one explained before), fraud, or a matter of personal safety. The quantity of risk present it is not the same for all transactions and depends on multiple variables: face-to-face exchanges tend to be more secure [4] as the ones when both sides participate in the transaction at the same time [5]).

An example of risk can be observed at peer-to-peer online transactions, where a buyer must move first and pay for the good before it is shipped. In other words, the buyer must initially trust that the seller will deliver the good as promised, and in doing it to allocate a payment to that seller before receiving the good. We can model it as a case of the *Prisoner's Dilemma* where the options are *to commit fraud, to transact normally or to do nothing at all.* Both sides have a positive payoff if they transact normally, but when one of them commits fraud and the other transacts normally, the one that committed fraud will have a bigger positive payoff and the one that didn't will have a negative one. As both of the sides have a self-interested perspective, the *Nash equilibrium* will consist of the exchange not happening: it will only happen if there is *trust* involved. If a seller's *trustworthiness* is too low the exchange will not occur.

2.4 Trust

Trust plays an important role not only on internet transactions but almost in all human relationships: family, friendships, economical relations... It is part of or nature and it even has a biological basis: there is strong evidence that it is not just a special case of risk-taking but it is based on important forms of social preferences [6] [7].

The majority of researches about trust make use of the so called *trust games*, usually a derivation of the *Dictator's Game.* In the Dictator's game one player, called *the dictator*, has to split an endowment (such as a cash prize) between himself and the second player, who simply receives the remainder left by *the dictator.* Results of players splitting the endowment offer evidence against the rationally self-interested individual concept of economic behaviour.

Trust games are repetitions of modified versions of the Dictator's Game [8]. For example, the experiment ran by Bapna et al [9] starts by giving $10 to the Dictator (here called the

Sender) who can send them to the other player. Any sent amoint is tripled and upon receipt of this investment the Receiver decides how much to return to the Sender. This single shot game concludes after the Sender learns how much of the investment has been returned, then it can be repeated again.

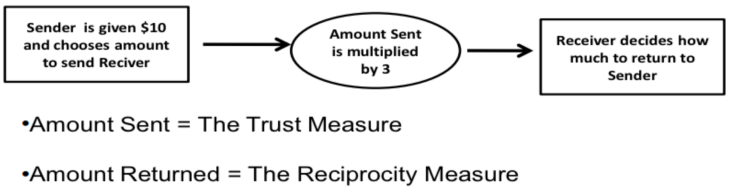


Figure 2.1: Trust game.

The existing infrastructure to create trust is vast and includes such elements as credit card companies, credit rating services, public accounting firms, and – if the exchange goes bad – such services as collection agencies or the court system. But what happens when there is not a third party available, such as when we want to perform a peer-to-peer transaction? Is in this moment, when we need to trust another peer, when we have to take a look at his *reputation*.

This peer does not have to be a person, in the case of modern society most times is actually a *company*. We trust that Visa will not get all our money, that McDonalds food will not make us sick, that an Apple product will be durable, etc. Companies have a *reputation*, but when doing a peer-to-peer transactions there is not more company behind that the platform where the transaction is happening.

In these peer-to-peer transactions we need to know about others *reputation* to feel safe and trust each other.

2.5 Reputation

Reputation, as defined by Wilson [10], is *a characteristic or attribute ascribed to one person (firm, industry by another (e.g., "A has a reputation for courtesy").* Most interesting for our context is that *[...] operationally, this is usually represented as a prediction about likely future behaviour (e.g., "A is likely to be courteous"). It is, however, primarily an empirical statement (e.g., "A has been observed in the past to be courteous").* These statements means

that:

- Reputation is based in past experiences.
- Reputation has predictive capabilities.

Of course the best source of information for somebody's reputation is our own experience, but when transacting for the first time such experience does not exist. In the past small communities used gossip to leverage each others experiences: when something bad happened all community would know about it, but as we started living in bigger communities that effect disappeared.

Sharing Economy platforms try the emulate the same behaviour: after transacting with a stranger, one of both parties are offered to leave feedback about each other, feedback that will be publicly available to other users. This way when transacting with strangers users can know about their reputation, avoiding possible risks.

To see how these platforms display reputation and try to increase trust between their users we decide to study *Airbnb*, one of the though leaders for trust and reputation.

2.6 Case study: Airbnb

Airbnb is one of the most popular Sharing Economy platforms and it is also considered one of the most advanced in terms of trust management. In fact its creators explain how the platform has been designed with trust in mind from the beginning: *[...] We bet our whole company on the hope that, with the right design, people would be willing to overcome the stranger-danger bias* [11].

As we can observe at figure 2.2 Airbnb profiles contain:

- *Personal data*, including the name, photo and city of the user, with the intention of increasing trust and empathy.
- *Text reviews* about past transactions left by other users.
- *Verifications* of the user's identity, both online (email, Online Social Networks) and offline (phone number, identification card). Usually for new users this is the only source of reputation available.

This is now the standard for all Sharing Economy platforms: Blablacar, Etsy, etc; all of them use the same concepts to build their reputation profiles. But Airbnb is also known for

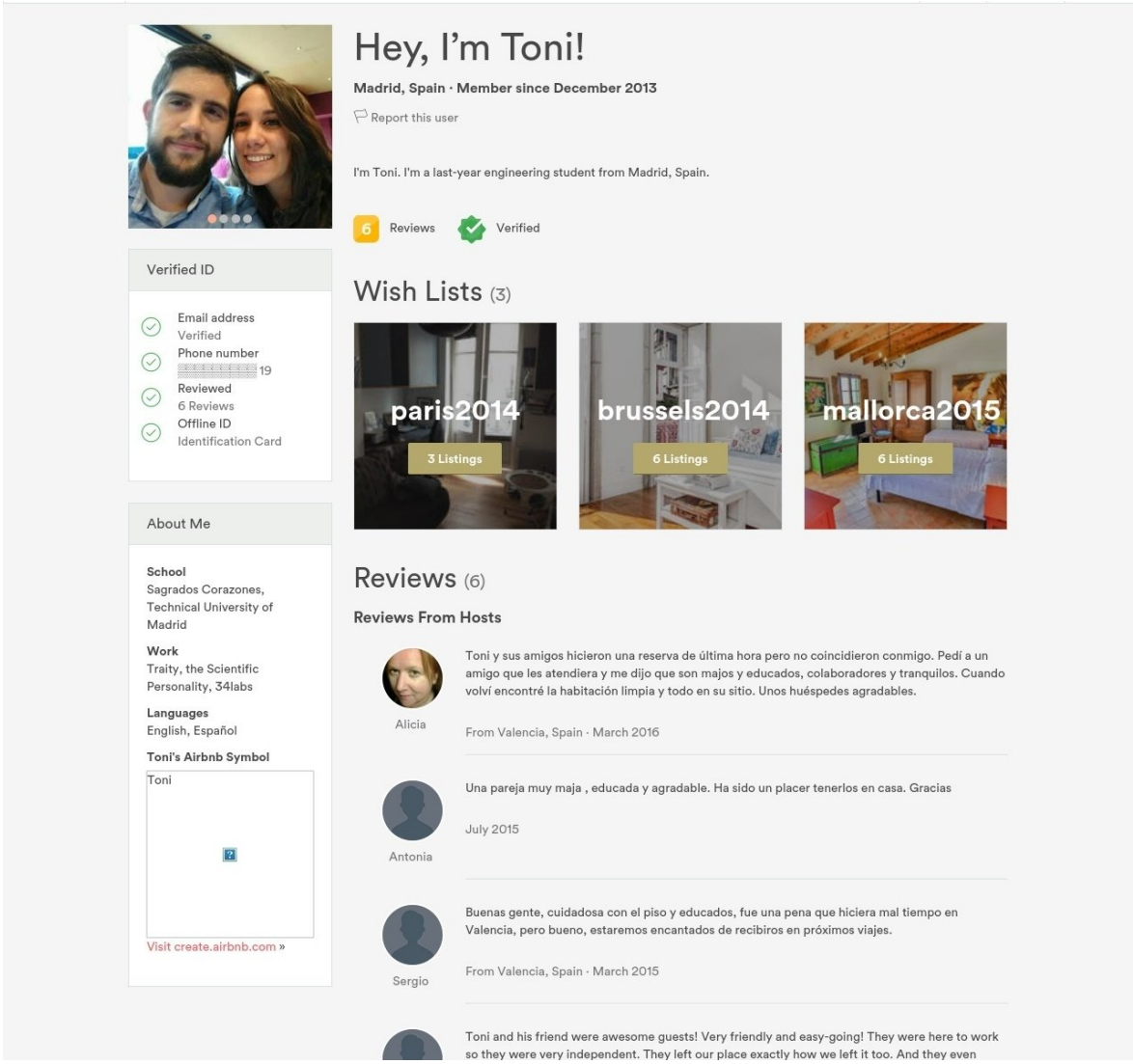


Figure 2.2: Airbnb profile example

its innovations and though leadership on the matter. Some of the innovations that AirBnb has implemented are:

- They introduced a system of blind reviews after discovering that user were not leaving bad reviews due to fears of retaliation: after one user leaving a bad review to other, he could do the same back. After implementing these systems users can not see other participants reviews until both are written, then they can not be altered. The result was a 7% increase in review rates and a 2% increase in negative reviews [12].
- They tweak elements of the user interface to match the optimal values to increase trust: for example after proving that short introductions would case acceptance rates to go down they made User Interface elements to input text bigger than before. [11].

While reputation systems work very well for existing users at the platform, new users find that because they start from scratch other users don't trust them. To attach this problem we have designed a solution that leverages existing data to predict new users behaviour.

2.7 Improving reputation management with data

Reputation in the Sharing Economy is a function of the *past transactions at the platform*, which implies that new users (that have not transact with anybody yet) will not have any reputation and therefore will be difficult for them to prove that they are trustworthy. An example of this behaviour appears at Airbnb, when hosts reject guests without reputation because they do not want to risk their homes and safety.

Sharing Economy platforms try to overcome this problem in two different ways:

- Asking for *references* to users that are already part of the platform. These references work in the same way that the reviews but without being attached to a known transaction.
- By *verifying offline and online identities*: by asking new users for identification documents and Online Social Networks. The platform intention is to scare new users with possible real-life (and even legal) consecuences and at the same time to avoid fraudulent users creating multiple accounts.

The second solution is very popular in not only Airbnb but also other platforms (e.g. Blablacar or Wallapop) because users have a clear incentive to input all sorts of data into the system.

We propose to leverage the popularity of this solution to predict user behaviour: data coming from social networks capture all sorts of user traits, as economical status [13] or personality [14]. Our intuition is that it is possible to have a previous knowledge of the outcome of the user's future transactions by analysing OSN data.

Applying this concept to Sharing Economy platforms: after new users introduce their OSN accounts as an identity verification (as Twitter or Facebook) the platform could have a previous knowledge of the possible outcome of future transactions and act accordingly; **for example by asking possible *bad users* for more data or by promoting and helping the ones that have been predicted as *good users***.

Such system would work seamlessly with current implementations of Sharing Economy platforms, but it could even be applied to other everyday life risky situations: asking for a credit at a bank or buying insurance, **as a sort of *credit scoring* that captures the reputation capital [15]**. This could help to reduce frictions in markets where OSN are very present but there are few data about their inhabitants: India is the second country with more Facebook users [16] but it doesn't have a proper *credit scoring* infrastructure.

To summarise: we propose as a solution to reduce the pains of new users at peer-to-peer platforms by predicting their future behaviours from the Online Social Networks data they already provide.

Next chapter we will start the process of gathering the data needed to make this process possible.

CHAPTER 3

Data acquisition

To be able to find correlations between data coming from Online Social Networks *and users reputation at* peer-to-peer platforms *we need data from both places. This chapter describes the data gathering process, from researching the available data sources to the actual process of building the datasets.*

3.1 Overview

The objective of this master thesis is to find correlations between *users' data avaliable at Online Social Networks* (*OSNs* from now on) and their *real world behaviours*. To reach this goal the first step of the process is to gather the data to analyse.

This chapter is structured as follows:

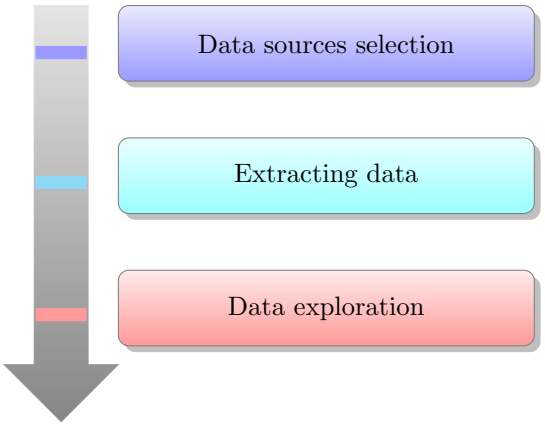


Figure 3.1: Steps of the data acquisition process.

- **Data sources selection**: there are multiple OSNs and peer-to-peer platforms to extract data from. The first step is to identify the available options and select the ones that will be used for the analysis.
- **Extracting data**: after selecting the data sources, the next step is to actually extract the data from them and save it in a suitable manner to analyse it.
- **Data exploration**: we need to explore and understand the data to be able to analyse it effectively.

3.2 Data sources selection

For this project we need to gather data from both an *OSN* and a *peer-to-peer platform* with reputation information. It is necessary to match users information from both sites so we have to choose them in a way that we can relate accounts from one to accounts from the other.

3.2.1 Online Social Networks data

There are multiple options to choose as the OSN where to extract users data. But it's important to have in mind that most of them take privacy very seriously and therefore it is not possible to download user data without the explicit permission of the user. While some of them, as Twitter, are designed to be used publicly others, as Facebook, are designed to be be more private and connected to the users' personal life.

This is the reason for Twitter being one of the most used OSNs for researching purposes. It is oriented to sharing content publicly and it provides an API (Application Programming Interface) to facilitate working with its data. We will use Twitter to leverage that API for extracting users information and for matching Twitter users to peer-to-peer platforms users.

3.2.2 Reputation data

We need to find peer-to-peer transactions platforms that provide information about users reputation.

Thanks to the rise of the *sharing economy* platforms we can easily obtain online data about how users behave in the real world. In these platforms it is needed a high level of trust between their participants because the inherent risks of their transactions. To overcome this problem these platforms are designed to show theirs users reputation, including reports of their behaviours at past transactions. Because they display lots of reputation information, they are the perfect source of information for this project.

The objective is to build a dataset that complies with the following guidelines:

- To come from a platform where users perform peer-to-peer transactions (such as buying, selling, lending..) that need a high component of trust for them to happen because of the inherent risks: e.g. giving your house keys to a stranger or buying a product you don't know nothing about.
- To include rich data about such transactions and its participants, including reputation data, so we are able to understand what happens in each transaction.
- To enable the possibility of classifying such transactions between successful and unsuccessful without the need of human-supervised classification, which can be painful and prone to errors.

To decide which one to select we did a research of the most popular sharing economy platforms at the moment:

- **Lyft**: peer-to-peer ride-sharing by connecting passengers who need a ride with drivers who have a car. Lyft now operates in over 200 U.S. cities, including San Francisco, Los Angeles, and New York City, and is valued at $5.5 billion [17].

- **Uber**: it allows consumers with smartphones to submit a trip request which is then routed to Uber drivers who use their own cars (similar to Lyft). As of May 28, 2016, the service is available in over 66 countries and 449 cities worldwide [18].

- **Airbnb**: platform for people to list, find, and rent lodging. It has over 1,500,000 listings in 34,000 cities and 190 countries [19].

- **Couchsurfing**: hospitality exchange and social networking website. The website provides a platform for members to "surf" on couches by staying as a guest at a host's home, host travelers, or join an event [20].

- **Etsy**: peer-to-peer e-commerce website focused on handmade or vintage items and supplies, as well as unique factory-manufactured items. The site follows in the tradition of open craft fairs, giving sellers personal storefronts [21].

- **Ebay**: founded in 1995, Ebay is an e-commerce company, providing consumer-to-consumer and business-to-consumer sales services via the internet. Today it is a multibillion-dollar business with operations localized in over 30 countries. [22].

- **Wallapop**: hyper-local mobile marketplace for buying and selling secondhand goods with a high penetration in Spain. [23].

- **TaskRabbit**: online and mobile marketplace that matches freelance labor with local demand, allowing consumers to find immediate help with everyday tasks, including cleaning, moving, delivery and handyman work. [24].

From these options we rejected the ones that didn't fit our purposes because the transactions are highly asymmetrical: in the case of Uber, Lyft and Taskrabbit one of the participants is more of a worker for the platform, while at Ebay a minority of (professional) users perform most of the sales. Therefore the selected options are Airbnb, Couchsourfing, Wallapop and Etsy.

3.2.3 Matching users

We need a way to match users from an OSN and a peer-to-peer platform, but for privacy reasons these sites don't facilitate any way to link one to each other.

The problem of matching users between multiple platform has been an subject of research in the last years under the term *matchmaking*. In *matchmaking* finding an exact match of different profiles is not the objective; instead the objective is to find the best possible match. In fact, such a match is very unlikely to be found and in all cases where an exact it does not exist a solution to matchmaking must provide one or more best possible matches to be explored [25]. This is not a good solution for this project as we need exact matches; if not we would lose performance and introduce bias into the task, so we can not make use of the existing knowledge about the field.

Instead we propose another solution: **to track Twitter for users that have shared *links* to their peer-to-peer platform profiles, and to select the *links* from which we can confirm that the owner of the Twitter account and the peer-to-peer platform account *are the same person***. Thanks to selecting Twitter as the OSN to use for this project we can make use of their public API to easily to track content for certain keywords or links.

With these requirements (finding content from Twitter users that allow us to match them with accounts from Airbnb, Wallapop, Etsy or Couchsurfing) we designed two solutions:

- **To track for invitations that Airbnb users share at Twitter**: Airbnb users can benefit from inviting other users to the platform by getting credit for future purchases. As they want to maximise their probability of their invitations being accepted some of them try to share them in every channel they know. Airbnb offers a *share in Twitter* button that comes with a default text that we can use to confirm that the user posting the invitation is the owner of the Airbnb account, an that account includes a link to the profile, therefore enabling the matching of both accounts.

- **To track for Wallapop items shared at Twitter**: when users post items at Wallapop they are offered to share these item listings to multiple social networks (figure 3.2). Users have a high incentive to do it as it can make easier to sell the product, aligning interests from both the user and the platform (that wants to bring new users to it). This implies that is possible to find a high count of shared links. When sharing a link using the default buttons, the shared text starts with a default text that Wallapop has chosen (figure 3.3). This text is usually not modified by the

user, in the case of Wallapop has been the same for a long time and it is different from other sharing default texts at the platform. Thanks to it, we can have a high certainty that users that share this text are the owners of the items that are being sold. The links shared can be followed and processed until the seller user profile is reached. Due to the fact that some of these links are being posted at social networks, it enables the matching between social network profiles and Wallapop profiles.

By setting a Twitter search for both options and tracking the number of tweets valid for the matching we checked that items with links to Wallapop items are much more common that links to Airbnb invitations. As machine learning algorithms as the ones that we will user at chapter 4 will benefit from having lots of data, we chose Wallapop as the source of reputation data.

Figure 3.2: Wallapop share dialog after uploading a product.

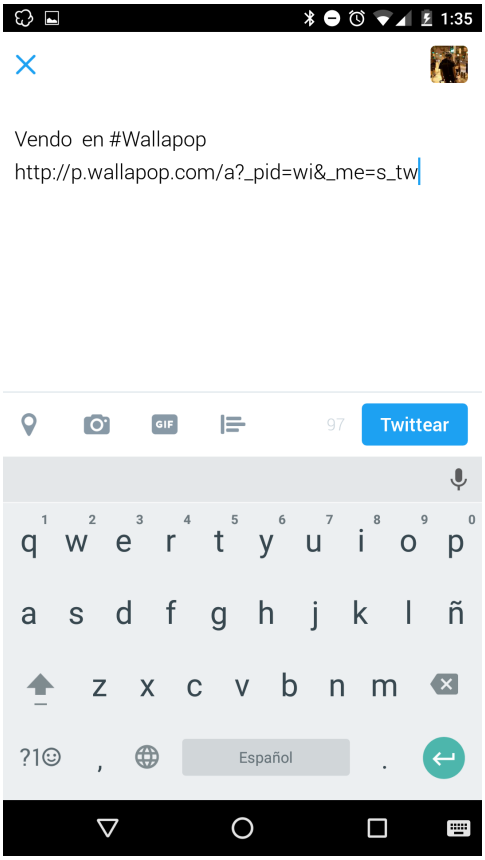


Figure 3.3: Twitter default sharing text after uploading a product.

3.3 Extracting data

It is needed to extract data from both platforms: Twitter and Wallapop.

3.3.1 Twitter data

Downloading data from Twitter is, in principle, a straightforward task because they expose an public API for that purpose. Using this API we can access most of the information available at the official website and applications. This data is structured as JSON files which facilitates its handling.

Twitter offers different endpoints for each one of the actions that can be performed at the platform. For the objective of finding links of Wallapop items at Twitter we will make use of *search tweets* endpoint, that returns a collection of relevant Tweets matching a specified query. By searching for the *default text* that Wallapop sets for new tweets shared through its mobile applications, we started to build a database of tweets with links to Wallapop items. As the number of tweets shared per day is not very high, to gather a dataset big enough it was needed to keep running this process for months.

Each saved tweet comes with a variety of metadata, including the Twitter account of the creator of the Tweet. We want to download all the data we can for each one of this users:

- **The profile**: available under the *users/show* endpoint, it returns a variety of information about the requested user: name, picture, description, number of followers, number of friends...

- **The list of tweets**: available under the *statuses/user_timeline* endpoint, it returns a list of tweets posted by the user. For most of the users is necessary to query the API multiple times to get the full list of tweets as each query returns a limited number of them each time, with a limit of the last 3200.

- **The list of followers**: available under the *followers/list*, it also need to be queried multiple times if the number of followers is high. To keep the size of the final files under a manageable size, only the last 2000 followers were gathered.

- **The list of friends**: The same as the followers but under a different enpoint: *followers/list* and also limited to the last 2000 results.

One problem we found when querying Twitter data is that, while it is free, it is also has some limits to access it: it allows 15 requests each 15 minutes. This limit is a huge obstacle for this process as we are managing high counts of users and each user needs multiple queries to be fully downloaded.

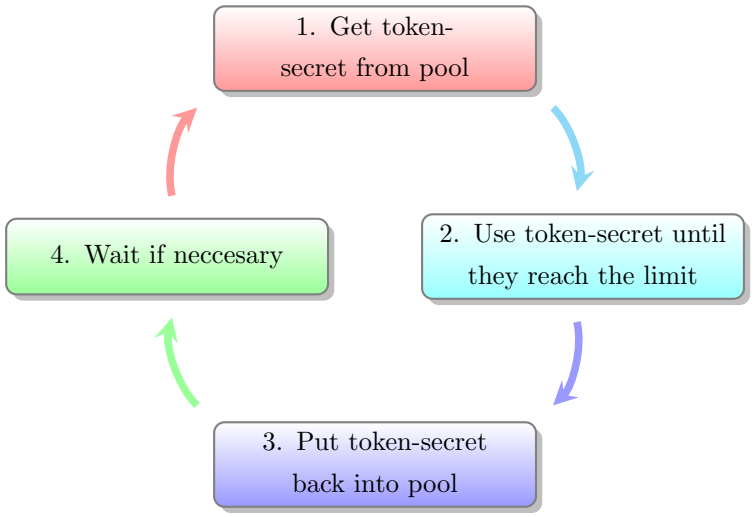
To overcome this problem we built a custom solution: a Twitter extractor that uses a pool of different accounts to overpass the limit.

3.3.1.1 The Twitter extractor

To access to this API is necessary to register as a developer under the Twitter Developers platform and create a new Twitter application which gives access to a token and a secret to identify the application. Then, for each user that joins that application another pair token-secret is generated, and by using the two pairs of token-secrets (one for the application and another for the user) the API can be queried.

Usually what developers do is to join themselves their own Twitter apps so they can obtain a user token and secret pair and perform queries to the API, always under the limit of 15 queries each 15 minutes. But this limit is not per application but per user, so for every user that joins the app the developer can make another additional 15 queries each 15 minutes.

This brings the possibility of creating a system that automatically manages a pool of users tokens and secrets to download big quantities of data in short periods of time. The system that we created works as follows:



1. A token-secret pair is extracted from a pool. This pool is implement in *redis*, a database stored in memory, so it is as fast as possible.
2. The token is used to query the Twitter API.
3. When the token reaches the Twitter limit, it returns to the pool of token-secrets and it is deactivated for 15 minutes.
4. If another token is available, the process starts again, if not it will wait until one of them is.

Thanks to the development of this Twitter extractor I was able to download all the Twitter data in hours instead of weeks. After the whole process a dataset of more than 30000 Twitter accounts with links to Wallapop items was created.

3.3.2 Wallapop data

Wallapop is an online peer-to-peer marketplace. If we use one of their official application we can navigate to other user profiles where we can observe their name, gender, photo, location... in addition of data related to the users reputation: a list of verifications, a list of reviews left by other users and another list of reviews given by them (figure 3.5). These reviews come with a numeric score which allow us to to get a deeper understanding of the reputation of each user without the need of using text mining tools that can introduce bias to the analysis due to an imperfect accuracy.

While Twitter offers a public and free API, this is note the case for Wallapop. Instead we researched the different platforms that make use of Wallapop data:

- **The website**: it allows the search of products and the visualisation of other users' information, but it doesn't have full functionality. The data is send as HTML content to be rendered by a browser.
- **Mobile apps**: the prefered option to use Wallapop because they have full functionality. Using the applications it is possible to make transactions, communicate with other users... The data is not available in a straighforward way: it is sent to the application from the official servers for them to render it.

The information that we can obtain from a website is mean to be rendered by a browser so it can be read by humans. Because of this, the information sent by Wallapop is buried in human-readable semantics and browser-rendering metadata. To work around this problem

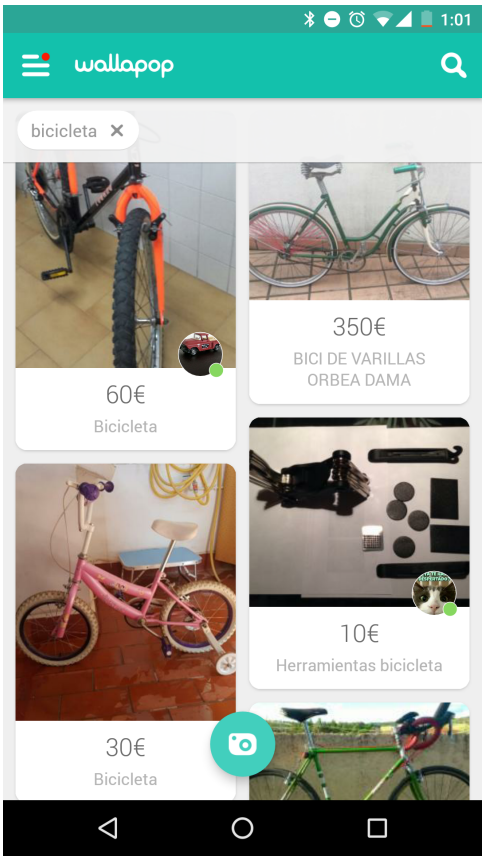


Figure 3.4: Wallapop items list

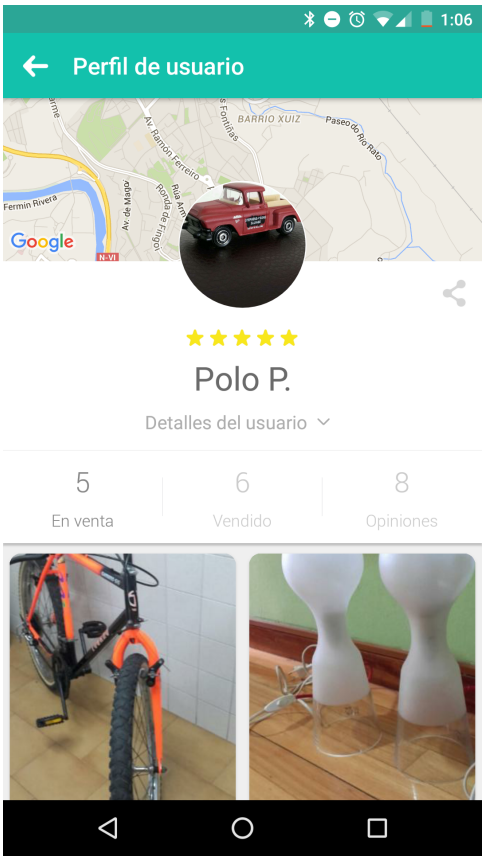


Figure 3.5: Wallapop user profile

we can make use of a set of techniques under the definition of *Web scraping*. They allow to overcome the problem of finding the information through the understanding of text-based mark-up languages (such the ones that the browser understand as HTML), with the downside that the way data is displayed in webpages changes frequently, breaking a possible data gathering process based on scraping HTML data.

The other option is to access the data the same way the official applications do by using the private API: it has the advantage of returning structured data, and that like at any other API its format does not change very much. Nevertheless it needs a previous phase of finding how to access it this way.

After researching both options, we discovered that the information sent by the Wallapop servers to the mobile applications was in fact structured in a JSON structure and decided to use it to assure the stability of the data gathering project over time.

3.3.2.1 Using the Wallapop private API

Wallapop mobile applications follow a typical pattern of server-client communication: a set of servers that perform most part of the business logic feed the applications with structured data for it to be rendered to the user. Both parties connect through the use of a defined API (Application Programming Interface) that is usually stable along time.

To download data from the internal API the first step was to identify to which endpoints the mobile apps were connecting to and how they exchanged the information. The process started by configuring and deploying a transparent web proxy (Charles Proxy) with a bundled HTTP interceptor that logged all the data that sent to it. Then we configured an Android smartphone to connect to it so the proxy could log all the data sent and received by the phone. Then we installed the Wallapop application on the device and used the application like a normal user could do.

The output of the process was the full set of requests exchanged between the server and the application, including what they contained and where were they headed to. By studying these logs it was possible to get all the endpoints available at the Wallapop internal API, including the most interesting for our own purporses:

- **Search**: a list of items available near the physical location of the user or a custom location. It enables to filter them by multiple variables as price and item category.
- **Item**: an item characteristics as price, location or description but also data about the seller of the item.

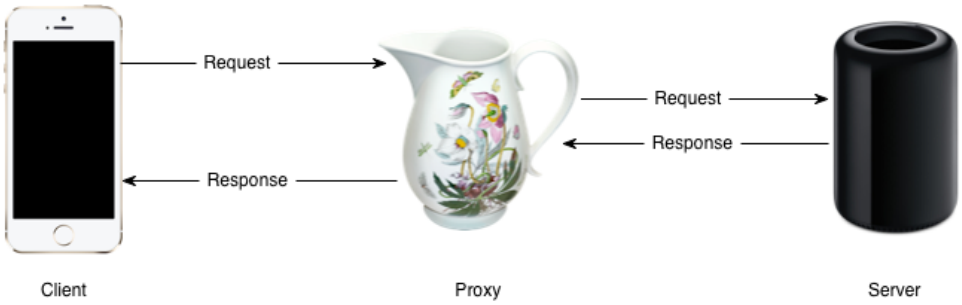


Figure 3.6: App with proxy architecture

- **User**: a user characteristics, from general data as the gender of current location to a set of reputation related data as reviews, scores and verifications.
- **User reviews**: full set of reviews given or received by a user.

At section 3.3.1 we downloaded a set of twitter accounts that posted tweets with links to their Wallapop items. We used these endpoints to download the data of the users that posted these links and then merged it with the Twitter data that we got before. The output of this process is a dataset of users that include Wallapop and Twitter data for each one of them.

While this dataset is enough to perform our objective of looking for relations between Twitter user traits and Wallapop behaviours, it do not represent a general population of Wallapop users but a population of users that share their items at Twitter. To understand the problems of trust and reputation at Wallapop it is better to have a general population, so we designed another process to download as much Wallapop users as possible.

3.3.2.2 Extracting a general Wallapop population

With all the data needed to perform our own requests to the Wallapop internal API as we were one of the official applications, we defined a strategy to try to get the maximum number of users from the platform. There is no index of registered users so its necessary a strategy to find these users and download them all.

IWEdeveloped a *spider crawl* strategy: it starts with an initial state of a set of searches on some of the most populated cities at Spain. As explained before, these searches return a list of items around a location, and each one of these items are posted by a user. By querying all the items we obtain a set of users that are selling items at that moment. Also, each one

of these user can have reviews about past transactions, which have information about the other peer in the transactions. By repeating this process again and again (download a user, go to check each one of the users he or she has transacted with, and repeat until it runs out of users to download) we can get the majority of users that have used the platform, and the network of transactions.

Wallapop has a high count of users from Spain and it is starting to rise in the USA and other countries: in Spain is a popular platform with a heterogeneous user base thanks to an aggressive marketing campaign including ads on national TV. This means that network of transactions between users will be heterogeneous and highly connected.

To implement this solution we have a web crawling framework called *Scrapy* because of its popularity and it being a open source project. A *web crawler* is a bot that browses the World Wide Web, usually by following links between different pages, and therefore typically by using a spider strategy too. Their typical structure is:

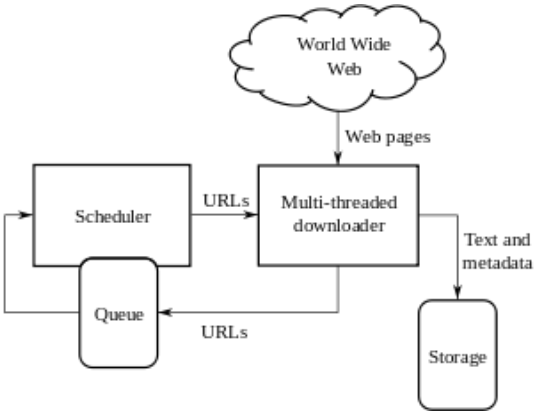


Figure 3.7: Crawler architecture

- A queue of items to store the links that have to be visited.
- A scheduler that manages what links will be visited and when.
- A muti-threaded asyncronous downloader that fetches the content and adds new found links to scheduler if necessary.
- A storage for the downloaded content.

Scrapy implements this architecture and thanks to it complies with all the requirements for the data gathering process: it makes easy to follow links, to not follow multiple times

the same link to avoid circular references, it implements pipelines to effectively process and store the downloaded data, etc.

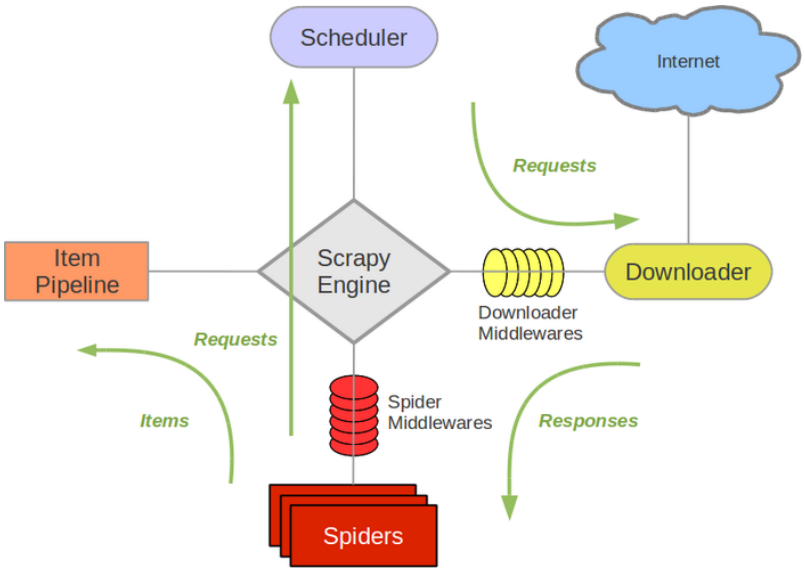


Figure 3.8: Scrapy architecture

We implement this strategy using Scrapy and it successfully downloaded a dataset of more than 200000 users with all their available information (profile, reviews and items).

3.4 Data exploration

After the last processes two datasets were gathered:

- A dataset containing the most part of Wallapop users structured as a network of transactions. This dataset will be used to analyse the Wallapop platform and understand how they manage trust and reputation.
- Another dataset containing matched Wallapop and Twitter profiles. With this dataset we will try to find correlations between Twitter users characteristics and their behaviours at Wallapop.

We can take a first look to both datasets to know what possibilities they offer for the data analysis that will be developed at chapter 4.

To perform a data analysis like this one the usual tools to use depends on the complexity of the task. While it is possible to use software as *Microsoft Excel*, *Tableau*, or even just database queries to extract insights from data, they are not powerful enough for complex tasks and therefore it is necessary to develop a custom analysis.

The programming languages most used for data analysis are *R* and *Python*. Despite that the former has been designed from the beginning as a statistical tool and is a great tool for the task, because I have experience with the later and that it has a vibrant community of data science developers I decided to chose Python before R. Some of the open source libraries that are currently being maintained by the open source community and that I have used on this project are:

- *numpy and scipy*: packages for scientific computing, including N-dimensional arrays and sophisticated indexing.

- *pandas*: data structures and data analysis tools.

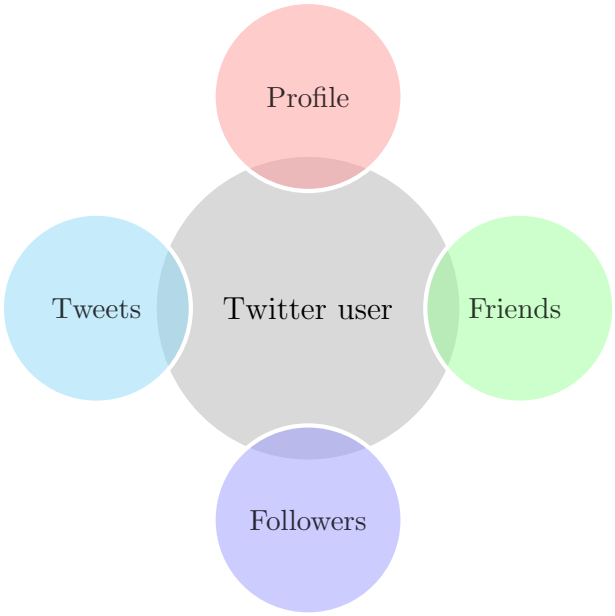
- *matplotlib*: visualization library.

- *scikit-learn*: machine learning package, oriented to data mining and data analysis.

- *networkx*: package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

With these tools we can extract take a look at the data we extracted.

3.4.1 Exploring Twitter data



For each one of the users that we have downloaded we have available four fields of data:

- **Profile**: it contains from data related to the user (as the name, picture or description) to summarising statistics (as the count of friends or followers). Here we display an anonymised example of what is available at the profile:

```
{
   "id": "----------------",
   "name": "----------------",
   "screen_name": "----------------",
   "location": "St. Francis, Mn.",
   "profile_location": null,
   "description": "----------------",
   "url": "----------------",
   "entities": {
     "url": {
       "urls": [
         {
           "url": "----------------",
           "expanded_url": "----------------",
           "display_url": "----------------",
           "indices": [
             0,
```

```
            23
          ]
        }
      ]
    },
    "description": {
      "urls": [
        {
          "url": "----------------",
          "expanded_url": "----------------",
          "display_url": "----------------",
          "indices": [
            127,
            150
          ]
        }
      ]
    }
  },
  "protected": false,
  "followers_count": 128,
  "friends_count": 19840,
  "listed_count": 1,
  "created_at": "Mon Dec 21 19:04:42 +0000 2009",
  "favourites_count": 3,
  "utc_offset": -18000,
  "time_zone": "Central Time (US & Canada)",
  "geo_enabled": false,
  "verified": false,
  "statuses_count": 7374,
  "lang": "en",
  "contributors_enabled": false,
  "is_translator": false,
  "is_translation_enabled": false,
  "profile_background_color": "C0DEED",
  "profile_background_image_url": "http://pbs.twimg.com/------.jpeg"
    ,
  "profile_background_image_url_https": "https://pbs.twimg.com
    /------.jpeg",
  "profile_background_tile": true,
  "profile_image_url": "http://pbs.twimg.com/-------.jpeg",
  "profile_image_url_https": "https://pbs.twimg.com/-------.jpeg",
  "profile_banner_url": "https://pbs.twimg.com/------",
  "profile_link_color": "0084B4",
  "profile_sidebar_border_color": "FFFFFF",
```

```
    "profile_sidebar_fill_color": "DDEEF6",
    "profile_text_color": "333333",
    "profile_use_background_image": true,
    "has_extended_profile": false,
    "default_profile": false,
    "default_profile_image": false
  }
```


- **Tweets**: each one of the provided tweets contains not only the proper tweet text but also a variety of metadata as the creation data, the times it has been retweeted, if it is marked as sensitive content, etc. An anonymised tweet looks as follows:


```
    {
      "created_at": "Mon May 12 21:01:02 +0000 2016",
      "id": ------------,
      "text": "-------------------------------------",
      "truncated": false,
      "entities": {
        "hashtags": [],
        "symbols": [],
        "user_mentions": [],
        "urls": [
          {
            "url": "---------",
            "expanded_url": "---------",
            "display_url": "---------",
            "indices": [
              92,
              115
            ]
          },
          {
            "url": "---------",
            "expanded_url": "---------",
            "display_url": "---------",
            "indices": [
              116,
              139
            ]
          }
        ]
      },
```

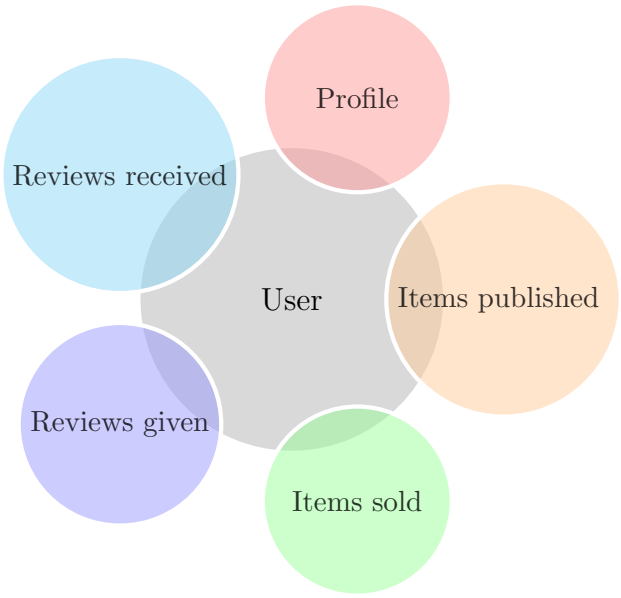
```
31      "source": "<a href=\"http://twitter.com\" rel=\"nofollow\">
            Twitter Web Client</a>",
32      "user": "---------",
33      "is_quote_status": false,
34      "retweet_count": 0,
35      "favorite_count": 0,
36      "possibly_sensitive": false,
37      "lang": "en"
38    }
```

- **Friends and followers**: for the user connections we have available a summary of their most basic statistics: the count of followers, friends and statuses, apart from their names, locations and descriptions.

```
•
1  {
2    "name": "----",
3      "screen_name": "----",
4      "location": "",
5      "description": "i like sports. listenning to music, shopping , and
           of course having new friends",
6      "followers_count": 7417,
7      "friends_count": 552,
8      "statuses_count": 4236
9  }
```

3.4.2 Exploring Wallapop data

Each one of the users from Wallapop contains the following fields:



- **Profile**: the Wallapop profile contains lots of personal data: name, gender and approximate location. It also contains platform metadata, verifications and basic statistics.


```
{
    "gender": "F",
    "image": {
      "averageHexColor": "FFFFFF",
      "pictureId": -------,
      "originalHeight": 100,
      "mediumURL": "http://cdn.wallapop.com/-----",
      "bigURL": "http://cdn.wallapop.com/-----",
      "originalWitdh": 100,
      "xlargeURL": "http://cdn.wallapop.com/-----",
      "smallURL": "http://cdn.wallapop.com/-----",
      "type": "jpg"
    },
    "userId": ------,
    "userVerification": {
      "genderVerifiedStatus": 30,
      "locationVerifiedStatus": 30,
      "verificationLevel": 1,
      "pictureVerifiedStatus": 30,
      "googlePlusVerifiedStatus": 30,
      "scoringStars": 84.0
    },
    "microName": "-----",
```

```
    "location": {
      "city": "Alcantarilla",
      "countryCode": "ES",
      "zip": "30820",
      "title": "30820, Alcantarilla",
      "approximatedLongitude": ----------,
      "kmError": 0.9,
      "approximatedLatitude":  ----------,
      "locationId": ---------,
      "active": true,
      "regionName": "Murcia"
    },
    "userUUID": "--------",
    "statsUser": {
      "receivedReviewsCount": 9,
      "sendReviewsCount": 0,
      "selledCount": 70
    },
    "responseRate": "Responde el mismo dia"
  }
```


- **Reviews given and received**: each one of the reviews come with information about the other participants in addition to the review text, score, creation date, etc.


```
    {
      "itemId": -------,
      "itemCategoryId": -------,
      "createDate": 1451923426000,
      "reviewId": -------,
      "comments": "Todo en orden. Amable y sin ningun problema. Mas
          que recomendable.",
      "toUserId": -------,
      "score": 100,
      "toUser": {
        "gender": "F",
        "image": {
          "averageHexColor": "FFFFFF",
          "pictureId": -------,
          "originalHeight": 100,
          "mediumURL": "http://cdn.wallapop.com/-------",
          "bigURL": "http://cdn.wallapop.com/-------",
          "originalWitdh": 100,
```

```
          "xlargeURL": "http://cdn.wallapop.com/-------",
          "smallURL": "http://cdn.wallapop.com/-------",
          "type": "jpg"
        },
        "userId": 10000274,
        "microName": "------.",
        "location": {
          "city": "Alcantarilla",
          "countryCode": "ES",
          "zip": "30820",
          "title": "30820, Alcantarilla",
          "approximatedLongitude": -------,
          "kmError": 0.9,
          "approximatedLatitude": -------,
          "locationId": -------,
          "active": true,
          "regionName": "Murcia"
        },
        "userUUID": "-------",
        "statsUser": {
          "receivedReviewsCount": 5,
          "sendReviewsCount": 5,
          "selledCount": 10
        },
        "responseRate": "Responde el mismo dia"
      }
```


- **Items sold and published**: for each one of the items the data includes title, description and other useful metadata as the creation data of the category:


```
{
      "reserved": true,
      "itemActionsAllowed": {
        "allowReport": true,
        "allowFavorite": true,
        "allowVisualization": true,
        "allowCheckProfile": true,
        "allowShare": true
      },
      "currency": {
        "defaultFractionDigits": 2,
        "currencyCode": "EUR",
```

```
        "symbol": "e"
      },
      "images": [
        {
          "averageHexColor": "7c8a92",
          "pictureId": 110785588,
          "originalHeight": 768,
          "mediumURL": "http://cdn.wallapop.com/-------",
          "bigURL": "http://cdn.wallapop.com/shnm-portlet/-------",
          "originalWitdh": 1024,
          "xlargeURL": "http://cdn.wallapop.com/shnm-portlet/-------",
          "smallURL": "http://cdn.wallapop.com/shnm-portlet/-------",
          "type": "jpg"
        }
      ],
      "shippingAllowed": true,
      "title": "Bq Aquaris E6 ",
      "originalSalePrice": 200,
      "fixPrice": true,
      "categories": [
        {
          "color": "#9b9b9b",
          "categoryId": 12545,
          "name": "Electronica"
        }
      ],
      "itemUUID": "-------",
      "banned": false,
      "description": "Vendo BQ AQUARIS E6 con un mes de vida. Se
          compro en Agosto y no se ha usado mas que ese verano ya que
          pronto me regalaron otro.",
      "sold": true,
      "publishDate": 1451389296000,
      "sellerUser": {
        "gender": "F",
        "image": {
          "averageHexColor": "FFFFFF",
          "pictureId": 35082377,
          "originalHeight": 100,
          "mediumURL": "http://cdn.wallapop.com/shnm-portlet/-------",
          "bigURL": "http://cdn.wallapop.com/shnm-portlet/-------",
          "originalWitdh": 100,
          "xlargeURL": "http://cdn.wallapop.com/shnm-portlet/-------",
          "smallURL": "http://cdn.wallapop.com/shnm-portlet/-------",
          "type": "jpg"
```

```
        },
        "userId": 10000274,
        "microName": "-------",
        "location": {
          "city": "Alcantarilla",
          "countryCode": "ES",
          "zip": "30820",
          "title": "30820, Alcantarilla",
          "approximatedLongitude": -------,
          "kmError": 0.9,
          "approximatedLatitude": -------,
          "locationId": -------,
          "active": true,
          "regionName": "Murcia"
        },
        "userUUID": "-------",
        "statsUser": {
          "receivedReviewsCount": 5,
          "sendReviewsCount": 5,
          "selledCount": 10
        },
        "responseRate": "Responde el mismo dia"
      },
      "modifiedDate": 1451423426000,
      "removed": false,
      "itemCounters": {
        "conversations": 11,
        "replyConversations": 55,
        "shares": 2,
        "favorites": 1,
        "views": 206
      },
      "itemId": -------,
      "itemFlags": {
        "sold": true,
        "reserved": true
      },
      "itemURL": "http://p.wallapop.com/i/-------",
      "soldDate": 1451904179000,
      "salePrice": 200
    }
```

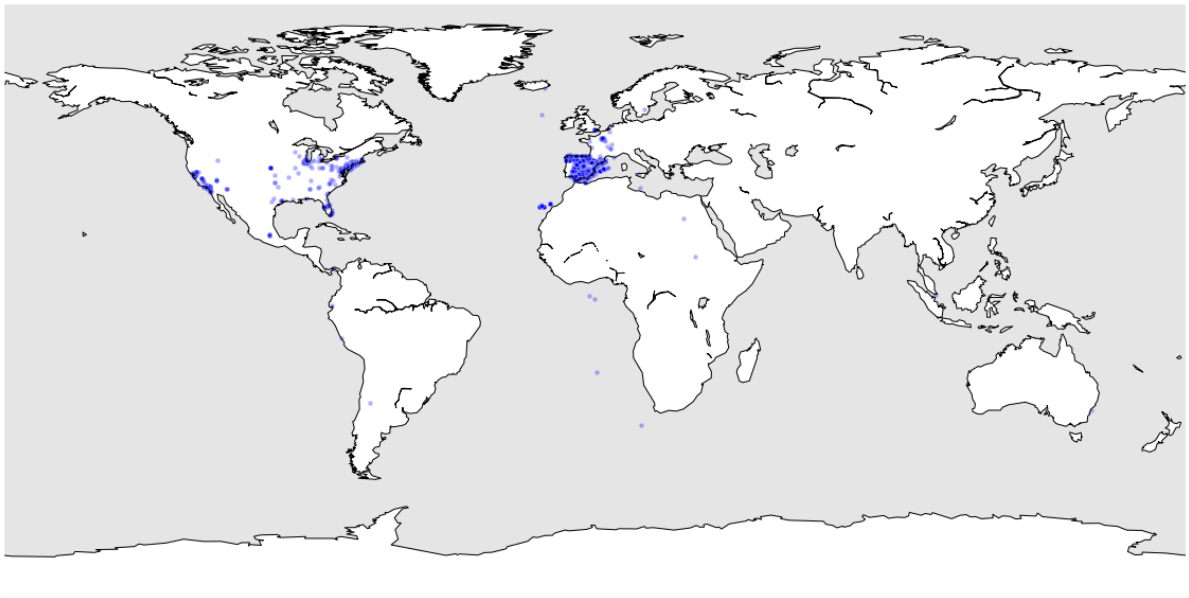


Figure 3.9: Worlwide map of users location

To have a better understanding of Wallapop users we can take a deeper look into some of their characteristics. For example, Wallapop makes available the location of their users, and by putting them in a map (figure 3.9) we can observe that the highest quantity of users come from Spain and the USA. It's Spain where I want to center the analysis, as being a Spanish myself facilitates relating the information with social, economical and other factors. Using the location additional data we can create rankings of the users location by country, and city.

When taking a look at the distribution of users from Spain (figure 3.10) we can observe that, as it was expected, the majority of users comes from big cities. In fact, if we check Spain's demographic data we can observe that the percentage of users coming from the biggest cities is higher than the percentage of habitants from these cities, meaning that Wallapop is more popular on locations with high density of population and there are network effects affecting the user penetration.

By visualising the percentage of the total population that a region holds against the percentage of the Wallapop userbase we can see that regions with a low number of habitants will not have very much penetration but that the most densely populated cities (Madrid and Barcelona) have higher user percentage that their population percentage (figure 3.11)

The dataset also includes summarising statistics about activity on the platform, more specifically about the items sold, purchased and currently listed by the user; the reviews sent and received; and the message notifications pending to be read. At next chapter we

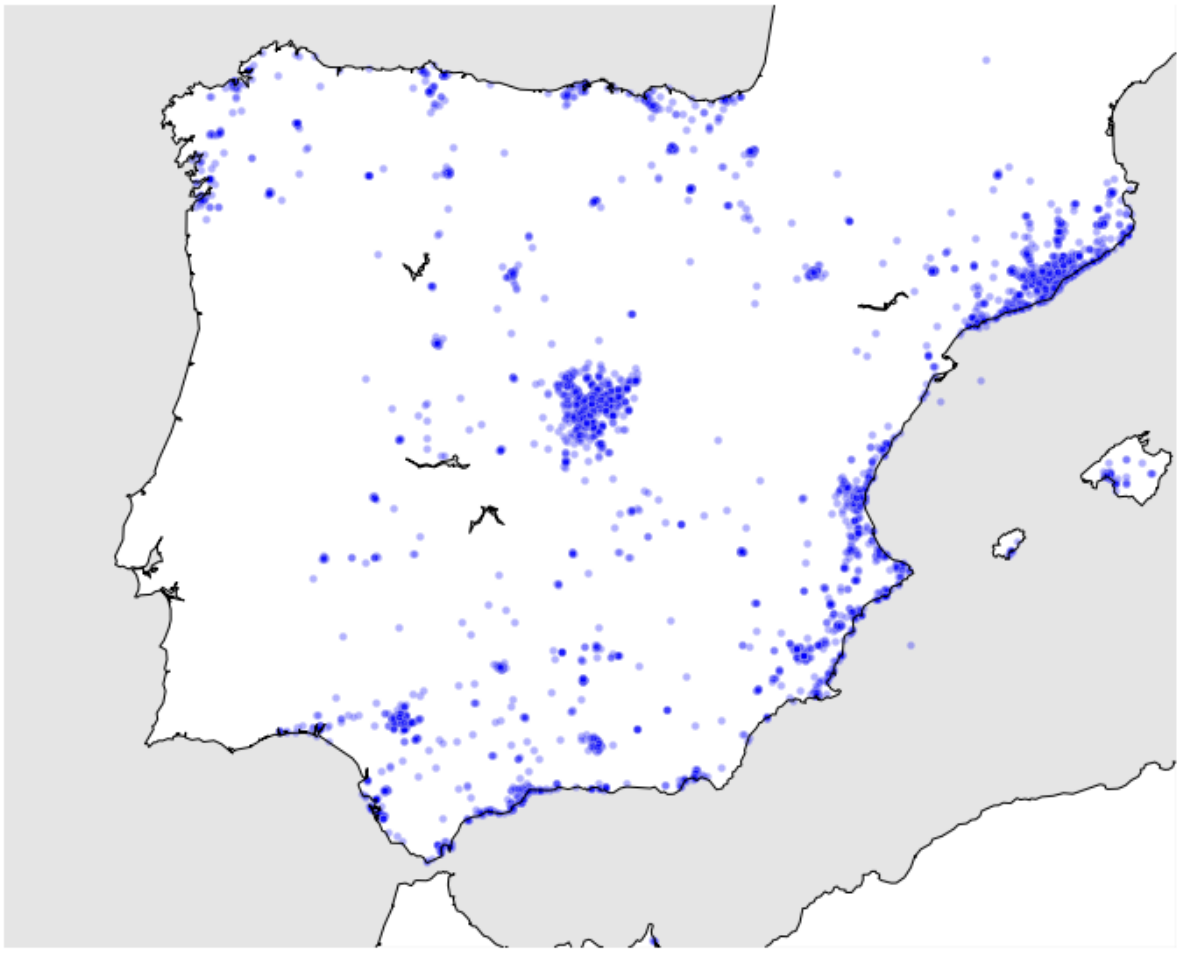


Figure 3.10: Spain map of users location

country	percentage
Spain	67.8%
USA	19.68%
France	0.86%
Andorra	0.09%
Mexico	0.06%

Table 3.1: Top 5 Wallapop users locations by country

city	percentage
Barcelona	10.64%
Madrid	10.51%
Valencia	2.87%
Sevilla	1.63%
Málaga	1.56%

Table 3.2: Top 5 Wallapop users locations by city

will start using the ones related with the reputation of Wallapop users classify according to their behaviours.

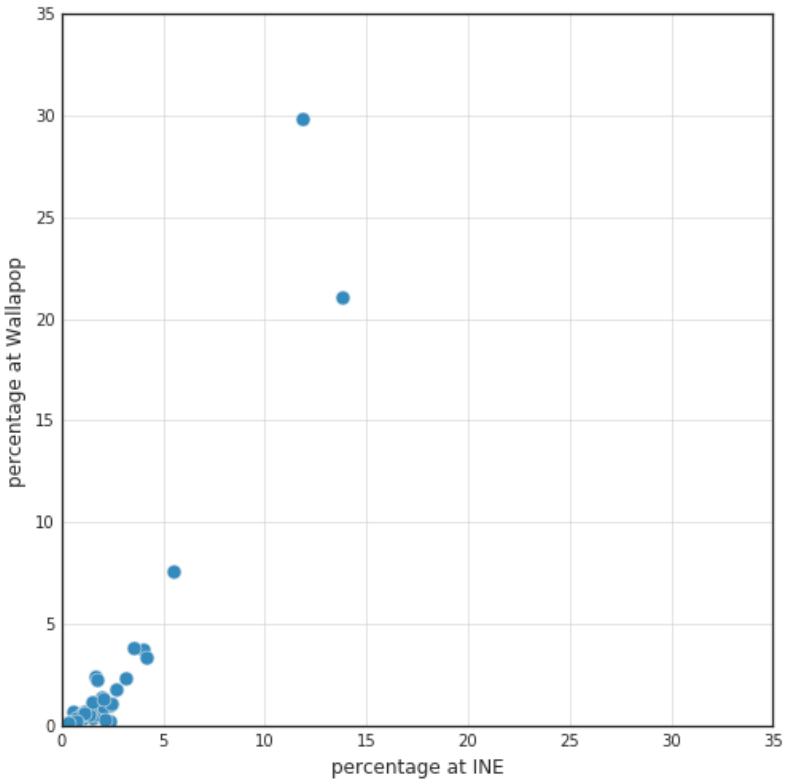


Figure 3.11: Percentage of user-base against percentage of population by region. The outliers are Madrid and Barcelona.

CHAPTER 4

Analysis

This chapter contains the actual development of the analysis to find relations between OSNs data and reputation at peer-to-peer online marketplaces. At the end of the process we present such relations.

4.1 Overview

The objective of this chapter is to unveil possible relations between what we can observe about users at *Online Social Networks* (*OSNs* from now on) and their behaviours at real world transactions, specifically at the marketplace platform named *Wallapop*.

For it we will create a machine learning classification task designed to predict who are the bad and good users at Wallapop from *OSNs* data and then we will leverage it to find what data is the machine learning algorithm using to take its decisions.

The process is structured as follows:

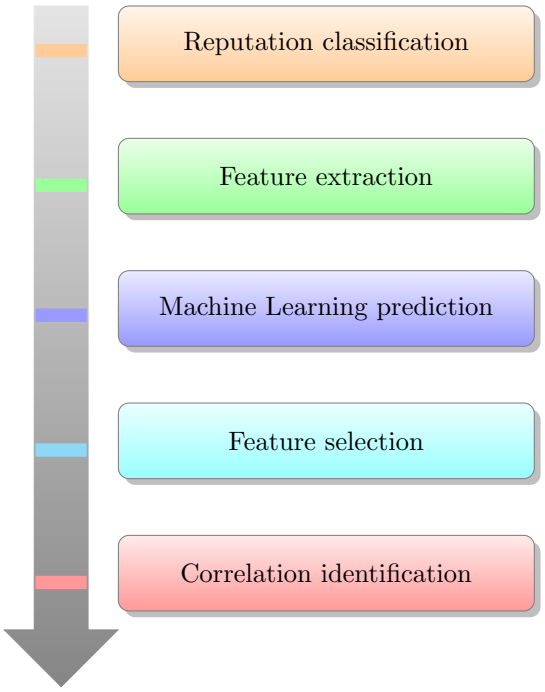


Figure 4.1: Steps of the analysis process.

- **Reputation classification**: to create our classification problem the first thing is to classify Wallapop users depending on their reputation. For it we will need a *reputation metric* that captures users past behaviours and enables comparing them. The output of this phase is a formula to classify users between *good users*, *bad users* and others.

- **Feature extraction**: to be able to feed Twitter data to a machine learning algorithm it is needed to build a set of numerical features suitable for the task. In this

process we will look into the different parts of the Twitter dataset (profile, tweets and connections) and generate a vector of numerical features.

- **Machine Learning prediction**: with the input of the Twitter users features (last section) labelled depending on their reputation at Wallapop (first section) it is possible to feed a Machine Learning Classification algorithm to predict how new users will behave at Wallapop depending on their Twitter data. In this section we will set up and benchmark different machine learning algorithms to see which one performs better for this task.

- **Feature selection**: after training a machine learning algorithm is possible to reduce the dimensionality of the input features by removing the redundant ones and the ones without prediction power. By selecting the important features we will improve the performance of the classifier and have a better knowledge of which features are the important for the task.

- **Conclusions**: Taking the most important Twitter features from the last section we will study the relations between Twitter data and Wallapop reputation, presenting and interpreting the results.

The first step is to classify users between good ones and bad ones by looking at their reputation.

4.2 Reputation metric

As we are designing a classification task the first step is to define which ones are the classes we want to predict. There are two questions that we want to answer:

*It is possible to detect **good** users (**high reputation**) just looking at their OSN profiles?*

*It is possible to detect **bad** users (**low reputation**) just looking at their OSN profiles?*

For it we will need to create two different classes: good users and bad users. That is the objective of this section that has the following structure:

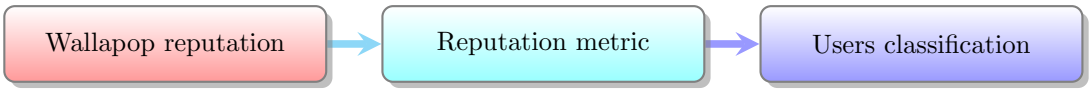


Figure 4.2: Steps of the analysis process.

- **Wallapop reputation**: at this step we will extract the users reputation from Wallapop data.

- **Reputation metric**: from the Wallapop reputation data we will develop a numerical reputation metric that allow to compare different users.

- **User classification**: based on the reputation metric this step defines a way of classifying users depending on their reputation.

4.2.1 Wallapop reputation

As explained on chapter 2, platforms as Wallapop suffer from a *trust problem*: their transactions imply risk for one of both parties participating on it and such participants are usually total strangers. These platforms use *reputation* as a tool to overcome this problem, displaying personal and past transactions data so the participants can know more about each other and check that in the past they behaved well. Thanks to the displaying of this data we can develop a metric that captures the reputation of each user.

At Wallapop, when two users finish a transaction they are offered to fill a review about each other. These reviews include a mandatory score that goes from *1 to 5* (internally from *20 to 100*) and an optional text field where they can write about their experience with the other user (figure 4.3).

Figure 4.3: Wallapop screen to leave reviews. This screen is shown after finishing a transaction and display a field to value the transaction from 1 start to 5 stars.

By extracting such scores is possible to gather reputation information from each one of Wallapop users.

4.2.1.1 Extracting reviews scores

After the chapter 3 we have two different datasets: one that contains a general population of Wallapop profiles and other that contains a subset of the Wallapop population that have shared their items at Twitter. we will use the first of them because as a source of reputation data because it is a bigger dataset and it contains a general population, while the second dataset is biased by the *shared at Twitter* action.

By gathering all the reviews from all the users of our dataset and checking the scores given on these reviews we can observe the distribution of these scores. This distribution (figure 4.4) shows how the majority of reviews come with a perfect score of *100 of 100*, but

there is also a tail of reviews that come with low scores, even with some of them being *20 or 100* (a very strong discontent with the transaction). The percentage of these very negative reviews is low but not negligible at all.

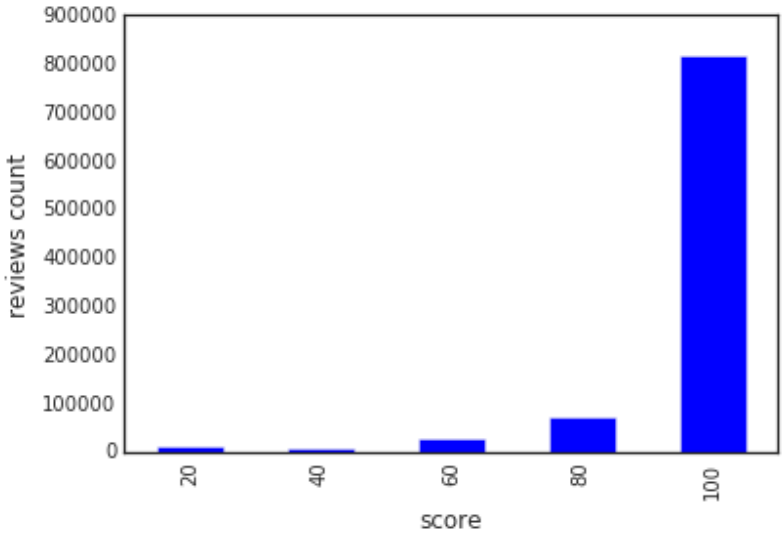


Figure 4.4: Histogram of reviews scores. Most of them are very positive, but there are also very negative ones.

Instead of aggregating all the reviews together, we have extract them user by user and generated for each user a list of reviews scores. These scores can be used next to build a simple reputation metric.

4.2.2 Building a reputation metric

Next step after gathering the reviews scores for each user is to generate a simple numerical metric that we can use to compare different users based on their reputation. The most simple of the options that can be designed is to just use the average of these scores as the final metric.

4.2.2.1 Using the average score

We can easily calculate the average reviews score for each one of the users and create a metric whose basic statistics are shown at table 4.1. As we can tell, the score is very high for almost all users because most of the user give very positive reviews, with a score if *100 over 100* becoming the usual.

	average score
count	222265
mean	95.85
std	9.53
min	20
25%	95.56
50%	100
75%	100
max	100

Table 4.1: Average score for each user received reviews.

	reviews	reviews w/ text
count	223855	223855
mean	4.17	1.8
std	4.62	2.4
min	0	0
25%	1	0
50%	3	1
75%	5	2
max	50	50

Table 4.2: Average number of reviews for each user.

But what about how many reviews are generating this average score? The statistics for the count of reviews each user has received are shown at table 4.2. They show that for a high number of users this metric is generated by looking just at a few reviews. This is a problem as reviews are crowdsourced information and with a low count of reviews per person the metric can be inconsistent: a user with an undeserved bad review will look very bad. Also, not everybody has the same mental framework when leaving reviews and what for one user can be a *80 of 100* rating for another can be a 100 of 100 rating.

Because we don't trust the reviews score we could also take a look at the comment that comes with the review and hand pick the ones that we think are *fair*. But the truth is that most of the reviews don't come with any comment so even a human supervised process it is not enough (table 4.2)

While capturing information about how each user behaves this metric is too unstable and noisy because the scores come from crow-sourced information. We need a metric that handles better inputs that we don't completely trust.

4.2.2.2 Using the true bayesian estimate

Crowdsource information can be trusted better when multiple participants give their opinion for the same person, mitigating the effects of a badly given review.

A solution implemented by IMBD to mitigate this problem is to use the *true Bayesian estimate* of the reviews scores. They use it for the same reason: to avoid films with only a few ratings to appear at the top of their famous *Top 250* of films. It assumes that the score of a new movie is the average score across all the movies (C) and then each new review gives new information about its real score by weighting them by a predefine value (m).

In our case we assume that a user will have the average reviews score across all the network (C) and new reviews will give new information about good or bad behaviour. This way we avoid the effects of undeserved reviews on people with low reviews at the same time that notice users that separate from the average behaviour.

We can define the function as:

$$W = \frac{v}{v+m} * R + \frac{m}{v+m} * C$$

C = *average review score across all users*

R = *average review score for this user*

v = *number of total reviews received by this user*

m = *value to control the number of reviews needed to trust the score. we empirically chose the average number of reviews received across all users*

Checking empirically the profiles of the worst and best users we noticed that this metric is better that the average score because it handles gracefully the problems of crowdsourced ratings and is better at creating a ranking of users based on their past behaviours. The statistics for this metrics are displayed at table 4.3.

	bayesian estimate
count	222265
mean	95.87
std	1.93
min	66.51
25%	95.7
50%	96.26
75%	96.89
max	99.37

Table 4.3: Bayesian estimate of reviews scores as reputation score.

4.2.3 User classification

After choosing the true bayesian estimate as the metric for reputation the only thing left is to choose how to categorise the users between different groups: bad, good and others.

This process gives a meaning to the reputation metric, then it has to be based on the empirical knowledge of what is a good transaction and a bad transaction at an online marketplace:

- For bad transactions, we ordered the users by their reputation score and checked their reviews scores and comments. Approximately the **lower 5% of them (reputation score below 91.886)** had multiple low scores and text comments explaining bad experiences, while above this threshold users did not look as bad.
- For good transactions, we did the same process but ordering from high to low reputation scores. A high percentage of reviews have very high scores, but not all them have text comments. By assuming that when enjoying exceptionally good transactions users tend to leave additional text comments we noticed that the **top 2.5% of users (reputation score above 98.03)** had good and long comments where reviewers seem exceptionally thankful.

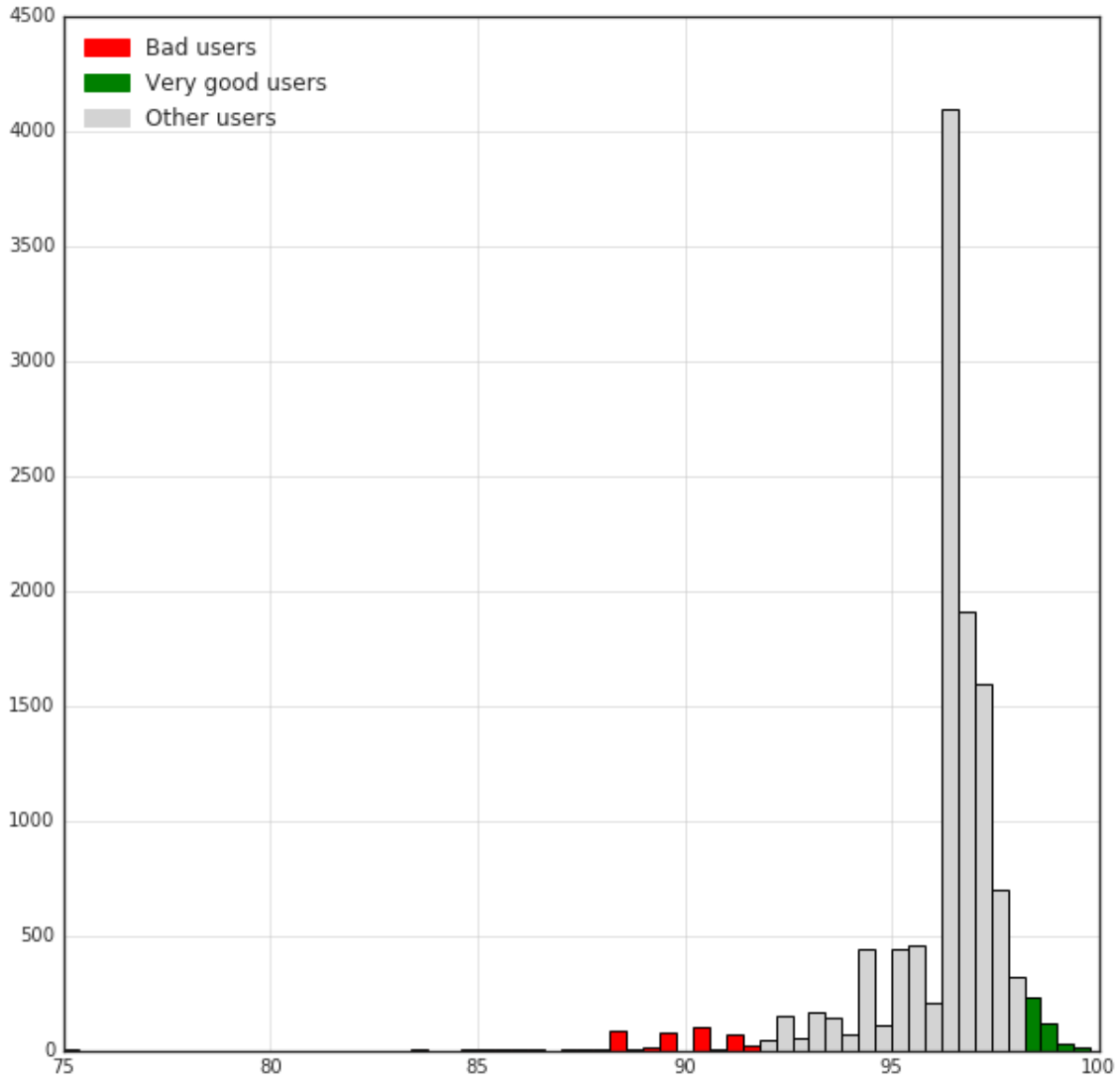
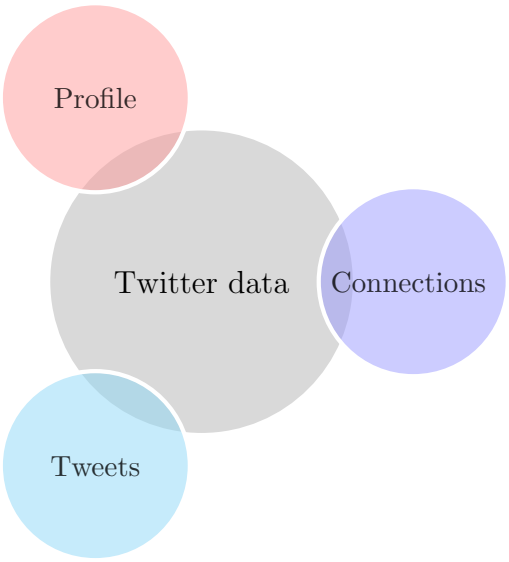


Figure 4.5: Different reputation labeled grouped by bayesian score.

Nowwe will explain how to extract features from Twitter data so we can use them together with this users classes to create a classification problem.

4.3 Features extraction

The dataset of Twitter data is formed by a list of Wallapop users from which we have all information publicly available (reviews, items, reputation, etc) crossreferenced with a list of Twitter users from which we also have all public information. These Twitter accounts structure look as follows:



- **Profile**: a set of user data as the name or the picture of the user, plus metadata like the account creation date or the number of followers.
- **Tweets**: List of the user last tweets, including apart from the tweet text also a set of metadata as the creation date, the language, the location, etc.
- **Connections**: basic information (name, picture, count of followers and friends, etc) about the friends and followers.

We want to transform all this data to features list that a machine learning algorithm could understand. We can start this process by analysing the twitter profile as it contains the most information about the user.

4.3.1 The Twitter profile

The twitter profile contains all sorts of information about the users but the actual content of the tweets and the list of connections. It contains summarising statistics (as the count of friends and followers), user data (name, picture, etc) and platform metadata (as the account creation date).

We can start by taking a look at the most basic metrics, like the ones related with the engagement of the user with the platform:

- **Count of tweets**: interesting metric that not only tell us about how much users uses the platform, but also their predisposition to *share his thoughts* and *engage in*

conversations versus being just passive readers. We have not a previous assumption of how this psychological trait can influence how a user will behave, but it's clear that it can capture a glimpse of users personality. By looking at able 4.5 we can check a set of basic statistics (having in mind that the distribution is skewed because of some extreme outliers).

- **Favourites count**: another metric directly related to the user engagement is the *count of times he has favourited* another tweet: the higher the time he spends on the platform reading other people tweets the higher the count of times he will favourite some of them. But we can also think about other interpretation: because there is not a significant reward from favouriting other users tweets (at least as retweets that have the purpose of resharing content) it could be that users that favourite a lot of tweets do it as a form of rewarding and thanking their creators.

The profile also includes the **account's creation date** which allows us to calculate how old accounts are. It is interesting to notice as at our dataset the distribution of account ages follow the typical distribution of user adoption (figures 4.6 and 4.7).These metrics allow to distinguish between *early adopters* (as users that tend to be the first to join a platform and have a more tech-savvy background) and *late majority adopters*. We have the intuition that early adopters are more tech savvy and understand better the mechanics of new platforms and therefore they will perform better at Wallapop and will have better reviews on their profiles.

Other interesting metrics are the ones related to the network of connections, as the **followers count** (or *in-degree*) and the **friends count** (or *out-degree*). These numbers have been subject of study multiple times, in fact in the past it has been defined an optimal number for the size of human groups [9], and later researchs [26] have confirmed the effectivity of this metrics in the online world. Nevertheless here at twitter interactions can happen with not only people but also companies, entities, etc; so this optimal number is not as straightforward as it would be at other OSNs as Facebook.

This in-degree metric is important because it implies that the user generates content interesting to other users and that it has some influence over them; but we should also check the out-degree because some users just follow others as a way of getting their attention (as by default they will be notified by Twitter). Nevertheless a ratio of followers to friends bigger than one is a good signal of influence.

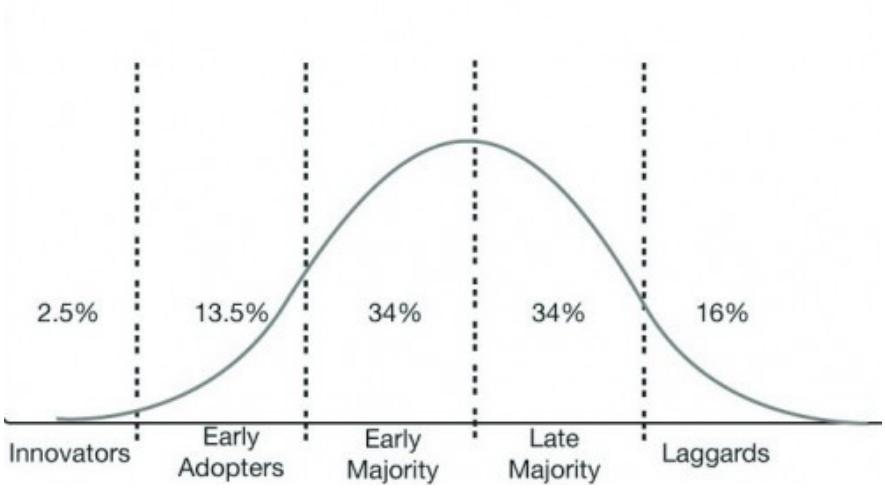


Figure 4.6: Typical adoption curve

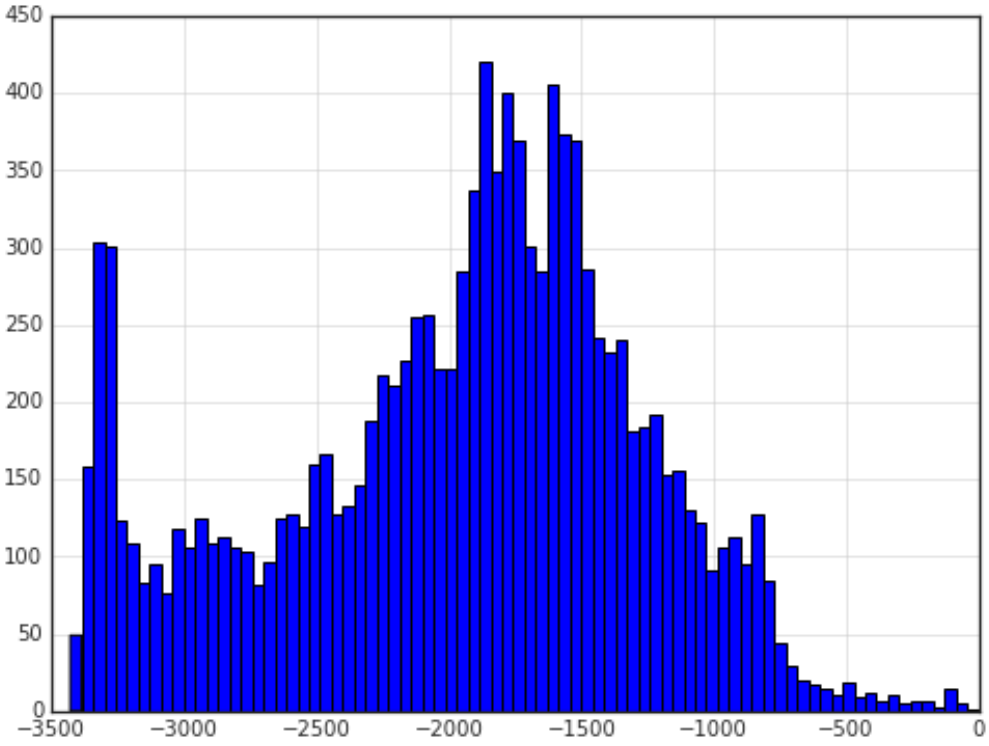


Figure 4.7: Adoption curve at our dataset

	followers count	friends count	ratio
count	12022	12022	12022
mean	472.46	439.97	2.09
std	3228.87	860.04	56
min	0	0	0
25%	57	114	0.34
50%	158	246	0.6
75%	380	487	1.06
max	264788	35637	5892

Table 4.4: Followers and friends count stats

Another interesting metric of influence is the **count of times that a users has been included into a Twitter user list**. Twitter lists are intended for tracking and grouping users under the same topic and being included in one of them imply some sort of *status* and knowledge. It is not a popular feature and most of the time this metric will be very low or even zero, but for high values it can be a good signal of reputation and the intuition tell us that these users can have better behaviours. Also, like when we were checking the friends count, it can be used as a spam tool and therefore probably low values will not have any correlation with the Wallapop reputation.

We will extract some features that are binary, and because the machine learning algorithms will need numbers as inputs it is needed to convert the negatives to 0 and the positives to 1. These features, which distributions are shown at figure 4.9, are:

- **Protected profile**: users can choose to hide their profiles from other users until they are given explicit permission, while the rest of other users can not see any tweets. While a minority, protected profiles could have implications on users personality and their behaviours at Wallapop.

- **Verified profile**: if the user profile has been verified by the Twitter team. They only do it for celebrities so it is a very rare trait and in fact there are not verified profiles at our dataset.

	statuses count
count	12022
mean	5957.73
std	13260.69
min	0
25%	225
50%	1381
75%	5570
max	207167

Table 4.5: Listed count stats

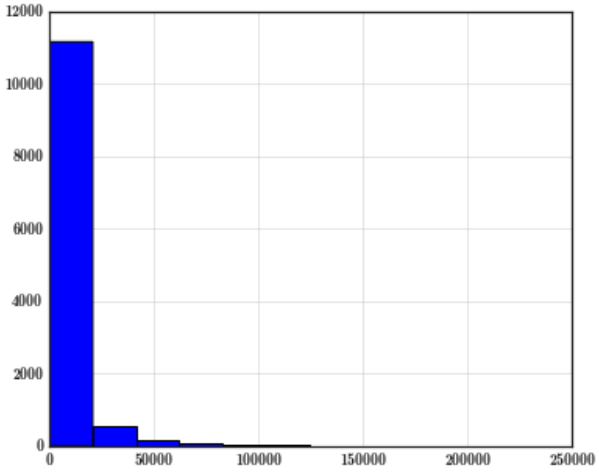


Figure 4.8: Histogram of user statuses count

	favourites count
count	12022
mean	1837.5
std	7488.98
min	0
25%	27
50%	192
75%	1007
max	331134

Table 4.6: Favourites count stats

	listed count
count	12022
mean	8.07
std	39.4
min	0
25%	0
50%	1
75%	6
max	2688

Table 4.7: Listed count stats

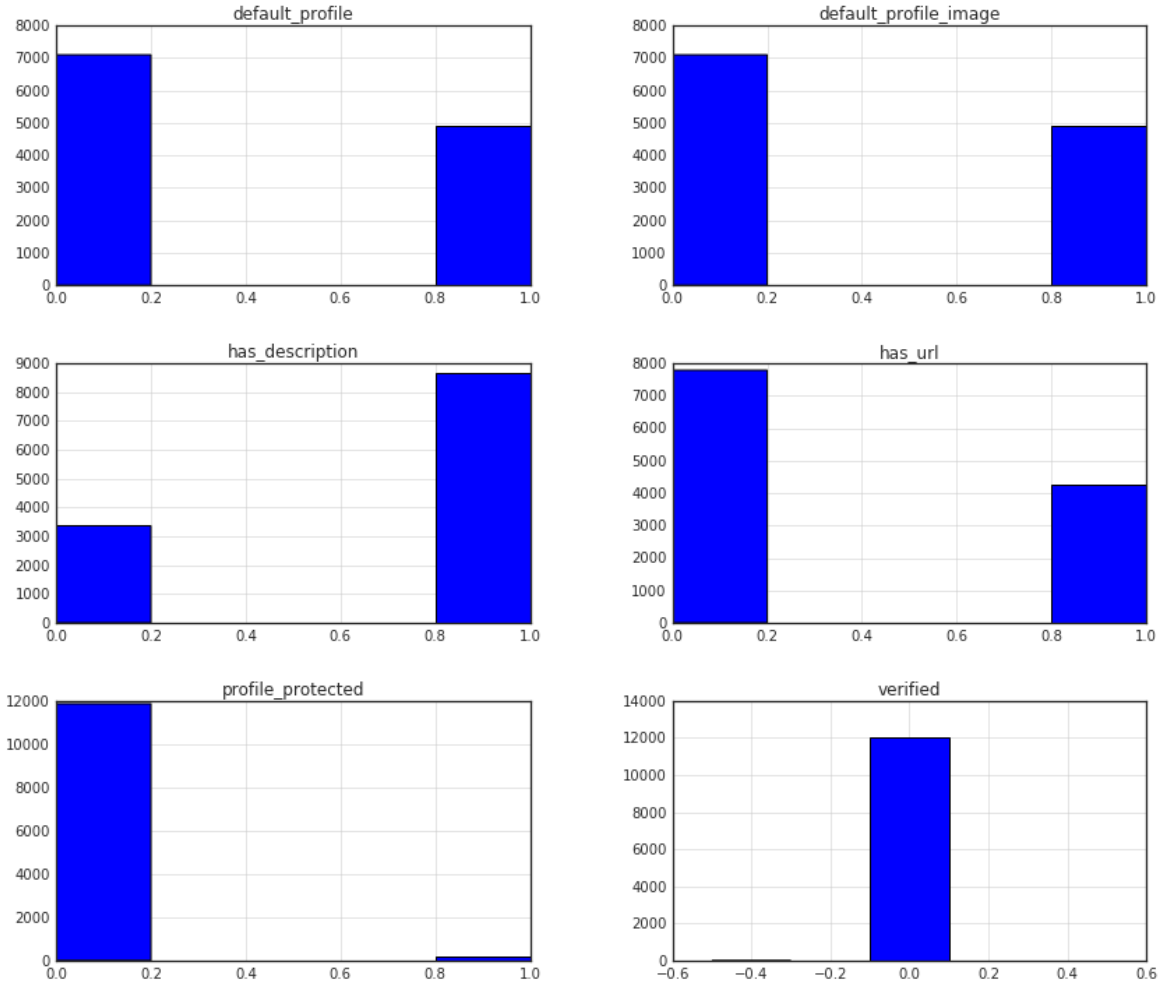


Figure 4.9: Histograms of binary profile characteristics (0 is negative, 1 is positive): user profile is protected, verified, still has the default profile, still has the default image, added a description, added a url on the description.

- **Default profile** and **default profile image**: if the user profile and/or image are the default ones because the user has not modify them. This is the case for users that are not very tech-savvy or don't care about *individualism* and customisation.
- **Available description**: if the user have written a description about himself or not. It expresses a desire of opening themselves to the world.
- **URL in description**: if the user has added a URL to the description, something that usually happens with companies, hobbyists, owners of businesses... as a way of promotion.

The last set of metrics are focused on **user's names**. An easy assumption to do is

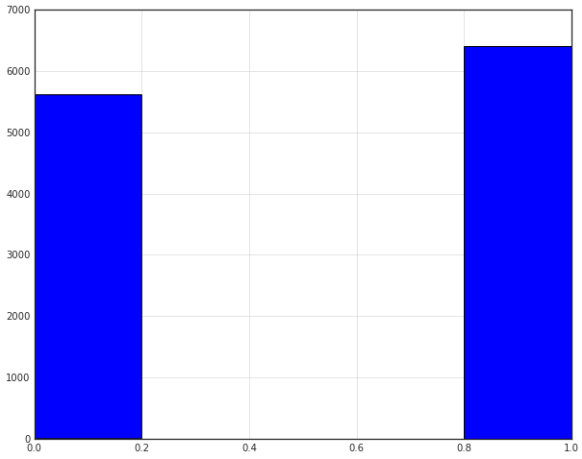


Figure 4.10: Distribution of Twitter names being part of a list of common real names (0 is negative, 1 is positive).

that users that expose themselves be sharing their names don't have anything to hide and therefore will have a better behaviour, for that we propose to label users that display real names instead of aliases.

It is impossible to know just looking at a Twitter profile if a user is showing his real name or not but at least we can know if it is a real name as it is not pretty common to use the name from another person but just use an alias. To be able to create this metric we had downloaded the last census data from both USA and Spain as they are the source for the majority of the users we are analysing, and then we combined them to create a list of possible real names. By checking for each one of the users we can know if they are using a real name as their Twitter names. We empirically checked the performance of this algorithm and it works pretty well for looking for real names. We also created features that contain the presence of numbers or non alphanumeric characters at Twitter names.

4.3.2 The list of tweets

The next set of Twitter data we can analyse is the actual list of tweets. Each one of these tweets come with a variety of metadata fields as the creation date, if it is a retweet or not, language of the tweet text... Our intention again is to try to find some metrics about how users tweet that have predicting capabilities of user behaviour. Some of them have more to do with the action of tweeting (as the date and time) and others generated by myself have more to do with the proper content of the tweet (as the count of grammatical errors per tweet).

We will start by taking a look at the non content-related metrics. Taking a look at figure 4.11 we can observe that the **most common tweeting hours** are that around the night, followed by a huge drop when people go to sleep. This temporal pattern can help us to find people that tweets outside of the general pattern, for example very late at nights, and maybe find a correlation of this metric with their reputation.

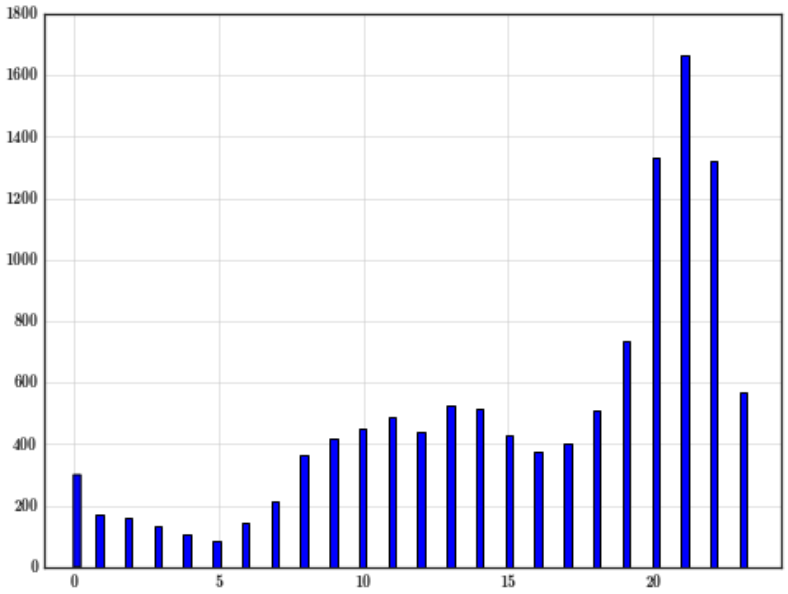


Figure 4.11: Most common tweeting hour histogram.

Another characteristic about how users tweet is the **average length** of each tweet. As we know tweets are limited to 140 characters but they could be shorter. We also care about the average count of **hashtags per tweet** as it relates with the skills of the user at Twitter as they know the feature and exploit it to have a bigger audience.

We can also extract the ratio between **replies to other users agains the total tweets count** and the ratio between **retweets agains the total tweets count**. This metrics are interesting because while users that frequently reply others could be labeled as more extroverted the ones that only tend to retweet content from others may be more introverted. accounts are the opposite as they share less their own content and more content from others.

There are two metrics that are related with tweets popularity and reactions: the average number of times **each tweet has been favourited** and the average number o times **each tweet has been retweeted**, with higher numbers signalling popularity and influence.

To finish we have performed a content analysis for each one of the tweets. Extract-

	tweet average length	hashtags ratio per tweet
count	12022	12022
mean	74.17	0.42
std	20.33	0.52
min	0	0
25%	61.12	0.14
50%	74.74	0.3
75%	87.71	0.56
max	135.81	8.26

Table 4.8: Tweet characteristics: average length and average hashtags per tweet

	replies ratio	retweets ratio
count	12022	12022
mean	0.14	0.34
std	0.14	0.25
min	0	0
25%	0.03	0.12
50%	0.1	0.31
75%	0.2	0.51
max	1	1

Table 4.9: Statistics about ratio of replies and ratio of retweets.

	retweets per tweet	favourites per tweet
count	12022	12022
mean	0.25	0.4
std	9.43	8.67
min	0	0
25%	0.02	0.05
50%	0.06	0.14
75%	0.14	0.28
max	933	756

Table 4.10: Average count of retweets and favourites per tweet.

ing features from text can be harder that doing it from structured data but it also have broader possibilities. For this analysis we will not use complex Natural Language Processing methodologies to understand the content of each tweet (for example for extracting the meaning of each tweet and building a list of users' interests, something that I'll do later using the connections network) but instead we will focus on grammar and vocabulary.

There are characteristics that correlate with characteristics as the user social and economic status [13] so they could be good indicators of user behaviour. The features extracted are:

- **Bad words per tweet**: we obtained a metric that indicates the average count of bad words by tweet by building a list of known bad words in Spanish, Catalan and English (as the three majority languages in the dataset) and checking the presence of every tweeted word on this list. This list was built by merging different datasets of banned words at internet forums. It's easy to make the assumption of that users that swear more than others can be more aggressive or less easy going than others and this can affect the reputation scores.

- **Misspellings per tweet**: the task of extracting this metric is more complex than the former one as it implies an understanding of the language. The process we have developed consists of:

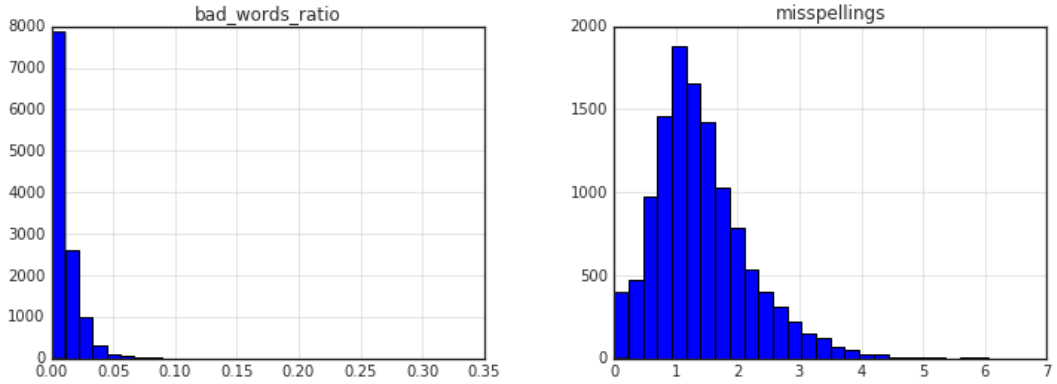


Figure 4.12: Bad words and misspellings per tweet distribution.

- Cleaning each tweet only leaving words with alphabetic characters.
- Using *Enchant* as a spellchecker to check every tweet. Enchant is a free software project developed as part of the AbiWord word processor with the aim of unifying access to the various existing spell-checker software. It wraps a common set of functionality present in a variety of existing products/libraries, and exposes a stable API for doing so [**?**]. By using Enchant to check the presence of every word in dictionaries of the three major languages of the dataset (Spanish, English and Catalan) we can obtain the ratio of misspellings per word tweeted.

All these features form a comprehensive set of features that extract meaningful value from the list of user tweets. Next section analyses the twitter connections network.

4.3.3 The connections

The Twitter network is unweighted and directed, with edges formed through the *follow action*. We can analyse these edges separating them between inner and outer edges:

- **Inner edges (followers)**: most of the users that follow the ones in our dataset are just normal users that follow the distribution of the three variables that we looked at: followers count, friends count and statuses (figure 4.11). Users that are followed by users that have a lot of followers themselves probably have a status a high network centrality too.
- **Outer edges (friends)**: these users are different: they follow less people, tweet a bit more and have way more followers (figure 4.12). Twitter is a platform where there is a minority or very influential users that has high followers counts, and these metrics

capture for each user what they prefer, it they prefer to follow celebrities or other users not as popular as them.

Another feature extracted from the network of users followers is a summary of each twitter interests. The idea is to look at which users are following and then check what they tweet about. To build this dataset we made use of a webpage called *Wefollow* that describes itself as a *A directory of prominent people organized by interests* (as shown at 4.14 and 4.15). As we said before, there are a minority of Twitter accounts that have very high counts of followers, in fact part of this accounts are part of the Twitter first time tutorial and they are suggested to be followed by the platform to every new user.

While this webpage not longer exists (as it was bought and later on closed by *about.me*) it contained a directory of popular Twitter accounts organised by topic. Using a custom web crawler we managed to download the full list of popular users and the topics they talk about, including the influence score they got at these topics

Later on, we iterated through all users and checked how many of their friends were in the database. After that, we added the scores of each one of their friends by topic, generating a multidimensional vector that contains the scores of interests for each one of the topics. An example of the information extracted can be found at table 4.13. Then we built a ranking of the top 100 most common topics and the users that are influential on them.

For example a user following lots of football players will have a high score of football interests, as well as sports, celebrities, and other interests categories. This multidimensional array can be fed to the machine learning algorithm to try to find correlations between it and the reputation score.

After extracting all these features we have a list of users and their numerical features. This dataset can be fed into a machine learning algorithm to perform the classification task we want to.

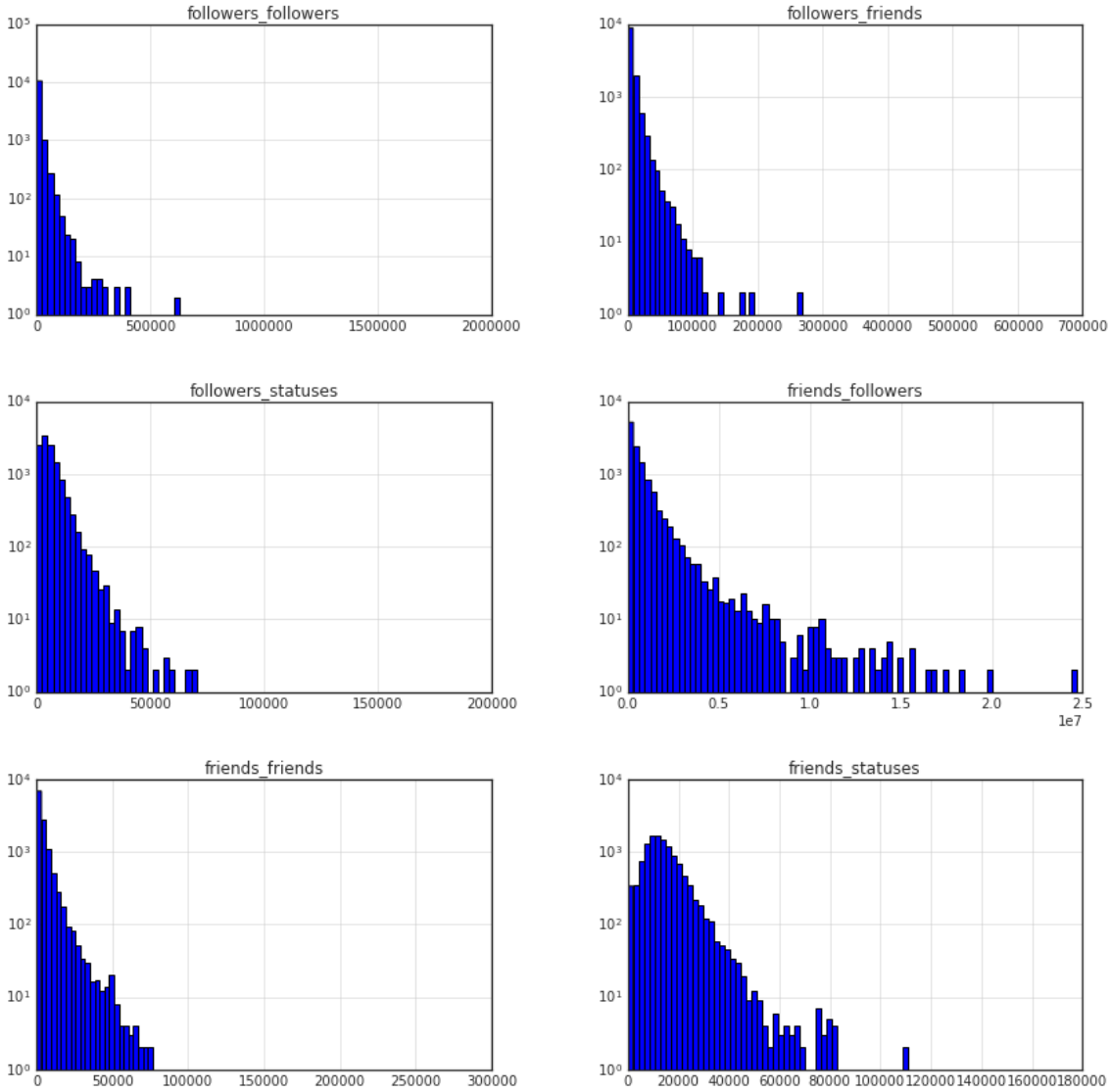


Figure 4.13: Distribution of average followers, friends and statuses count for each user followers and friends.

	followers followers	followers friends	followers statuses
count	12022	12022	12022
mean	13119.33	7853.23	6298.1
std	32053.74	14766.14	6031.57
min	0	0	0
25%	2317.27	1741.40	2804.75
50%	6208.99	4024.15	4936.78
75%	14521.05	8748.1	8045.99
max	1927392.21	649954.29	194083

Table 4.11: Statistics of average followers, friends and statuses count for each user followers.

	friends followers	friends friends	friends statuses
count	12022	12022	12022
mean	843107.34	4736.83	14224.66
std	1582655.83	7274.37	8777.48
min	0	0	0
25%	135535.95	1301.85	8884.92
50%	395966.81	2684.42	12713.94
75%	908769.12	5373.85	17750.2
max	24698661.2	255221.33	170259

Table 4.12: Statistics of average followers, friends and statuses count for each user friends.

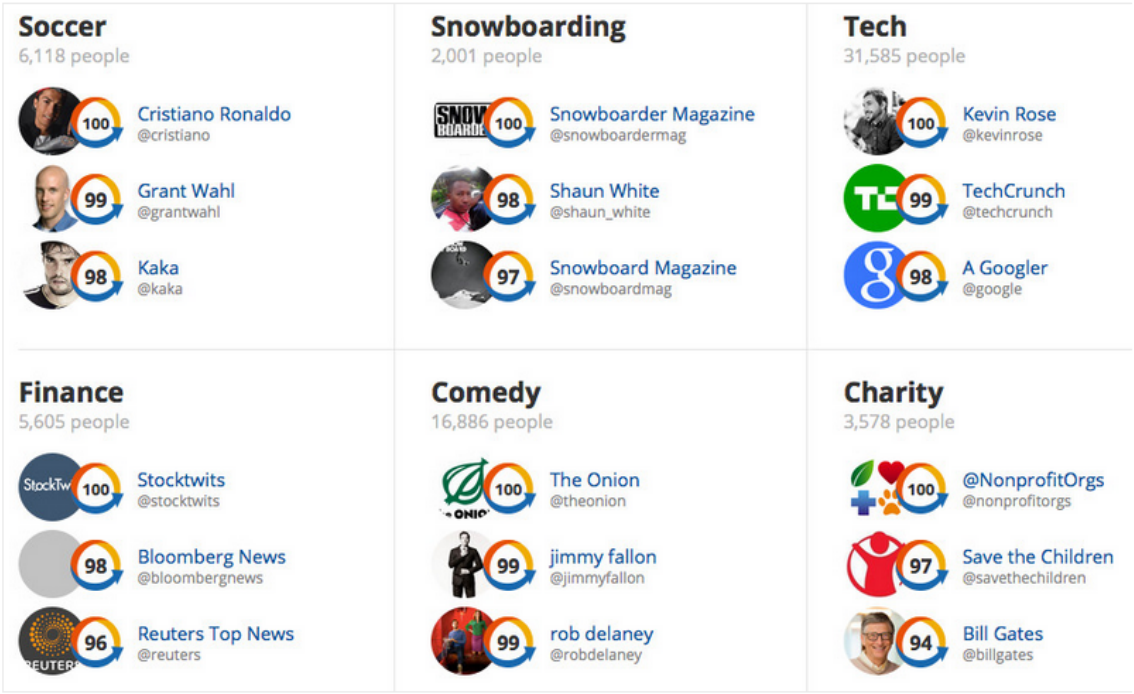


Figure 4.14: An example of Wefollow, a directory of Twitter users organised by topic.

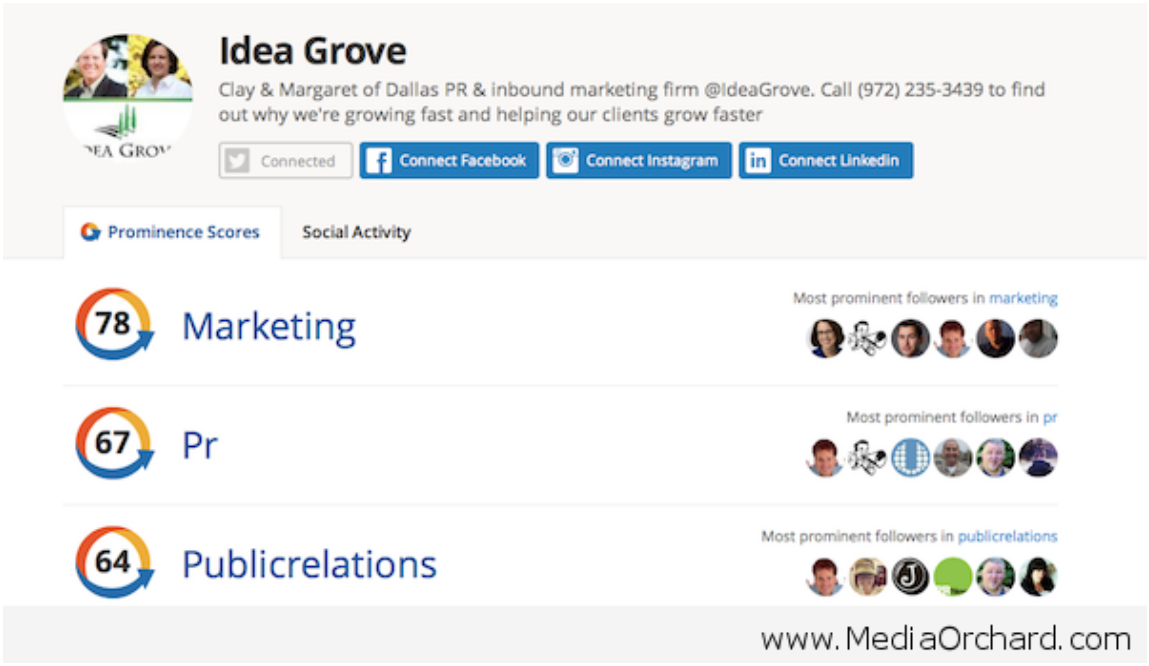


Figure 4.15: An example of a Wefollow profile with a list of topics the user writes about sorted by an influence score.

```
[{"username":"@ladygaga","interests":[{"score":"100","name":"Music"}]},{"username":"@katyperry","interests":
[{"score":"93","name":"Celebrity"},{"score":"99","name":"Music"}]},{"username":"@kanyewest","interests":
[{"score":"93","name":"Celebrity"},{"score":"100","name":"Music"}]},{"username":"@drake","interests":
[{"score":"91","name":"Celebrity"},{"score":"99","name":"Music"}]},{"username":"@justinbieber","interests":
[{"score":"98","name":"Celebrity"},{"score":"99","name":"Music"}]},{"username":"@rihanna","interests":
[{"score":"96","name":"Celebrity"},{"score":"98","name":"Music"}]},{"username":"@nickiminaj","interests":
[{"score":"95","name":"Celebrity"},{"score":"98","name":"Music"}]},{"username":"@taylorswift13","interests":
[{"score":"90","name":"Celebrity"},{"score":"97","name":"Music"}]},{"username":"@pitchforkmedia","interests":
[{"score":"100","name":"Indie"},{"score":"84","name":"Media"},{"score":"97","name":"Music"}]},
{"username":"@ladygaga","interests":[{"score":"100","name":"Music"}]}]
```


Figure 4.16: Example of influence scores for Twitter celebrities in JSON format.

ranking	category
1	Blogger
2	Socialmedia
3	Music
4	Marketing
5	News
6	Entrepreneur
7	Travel
8	Photography
9	Politics
10	Tech

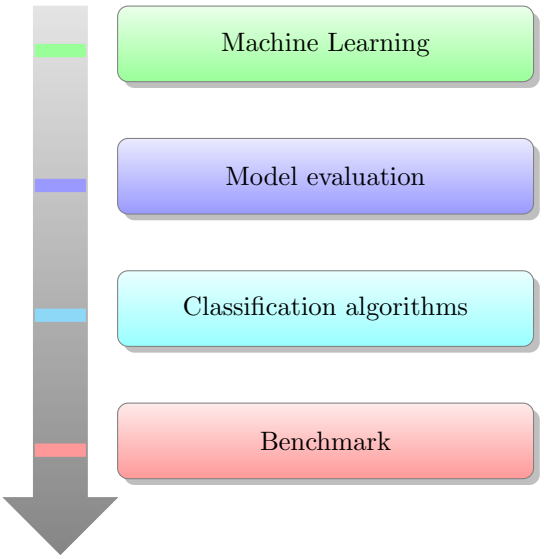
Table 4.13: Ranking of most common categories at Wefollow dataset.

4.4 Classification

After choosing a particular metric to measure the reputation of Wallapop users and dividing users on different groups, we will to study the relations that exist between reputation and user traits.

We are going to handle this problem as a classification task: we have two different classes that we want to predict (*good users vs normal users*, or *bad users vs normal users*) and a set of features that input the classification task. The objective is *to predict to what group the user will belong just looking at these features*. We will not treat the task as a simple correlation calculation as the majority of the variables don't have a lineal correlation with the reputation but this doesn't mean that such correlations doesn't exist: different groups of users will have different characteristics that differentiate them and by looking at this characteristics we can have a previous knowledge of how they will behave.

The structure of this project goes as follows:



- **Machine Learning**: explanation of the basics of Machine Learning techniques.
- **Model evaluation**: description of the possibilities for evaluating a Machine Learning model and selection of the ones that better suit our problem.
- **Classification algorithms**: summary of the most common Machine Learning classification algorithms.

- **Benchmark**: comparison of the performance of the machine learning algorithms explained before at our task and selection of the best for the next sections.

4.4.1 Machine learning

Machine learning is a subfield of computer science that evolved from the study of pattern recognition and computational learning theory in artificial intelligence. Machine learning explores the study and construction of algorithms that can learn from and make predictions on data. Such algorithms operate by building a model from example inputs in order to make data-driven predictions or decisions expressed as outputs, rather than following strictly static program instructions.

Within the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction. These analytical models allow to and uncover "hidden insights" through learning from historical relationships and trends in the data. Tom M. Mitchell provided a widely quoted, more formal definition: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E".

There are two types of learning: supervised and unsupervised. When the learning is supervised the computer is presented with a set of example inputs and their desired outputs with the goal of learning a general rule that maps new inputs to the predicted outputs. This is the typical case of a classification task: first we input the algorithm with a set of labeled examples hoping that it learns how to generalize the differences between different labels, and then we feed it new inputs to be classified

When the learning is unsupervised there are no labels or desired outputs for the example inputs, leaving it on its own to find a structure and discovering hidden patters in data. This is the case of clustering tasks where it is needed to group inputs for being near in a space of features, for example clustering news articles by a multidimensional space of topics.

In this case the task is a supervised learning task, specifically a classification task. The *inputs* are the set of features extracted at the last section of the chapter, and the *ouputs* are the classes we defined by looking at the Wallapop's bayesian estimate reputation score. We defined three groups: bad users, normal users and good users because we are interested on finding features that relate to both: users that do it really well vs the rest of users and others who do it really bad against the rest of users too.

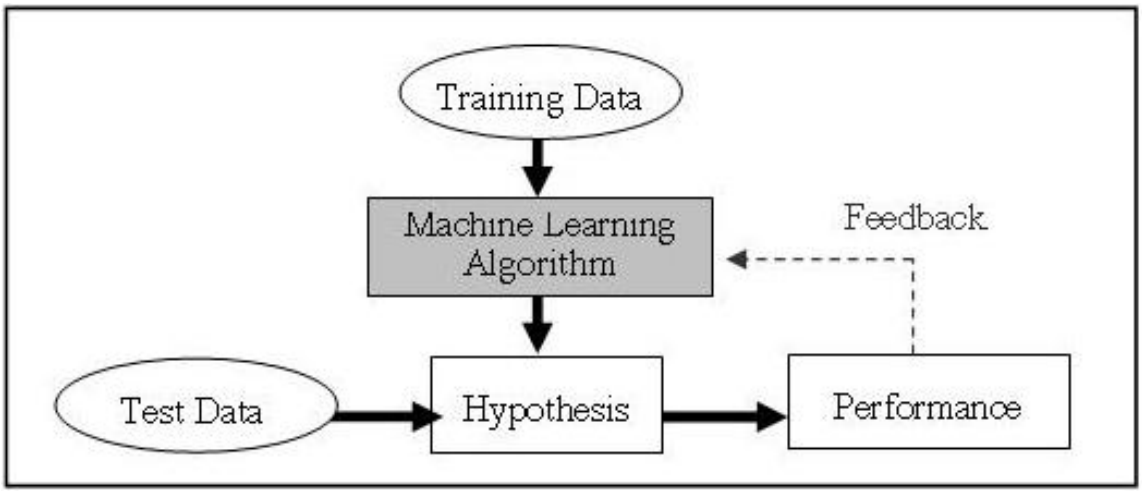


Figure 4.17: Machine learning diagram.

4.4.2 Model evaluation

The objective of this section is to run a bechnmark of different machine learning classification algorithms to be able to measure how this set of features can perform for a classification task and to choose the best machine learning algorithm for it.

To be able to compare them first we have to introduce which metrics will we be using to measure their performance. The most basic metric used to measure the performance of a machine learning algorithm is *accuracy*, a statistical measure of how well a binary classification test correctly identifies or excludes a condition, or in other words the proportion of true results (true negatives and true positives) among the total number of cases examined. But accuracy comes with a problem: it assumes equal cost for both kinds of errors, and given the implications of marking a good user as bad user or vice versa it is not a good fit for our task.

A better option is to use precision an recall. We call precision to the proportion of the true positives against all the positive results (both true positives and false positives), while recall is the fraction of relevant instances that are retrieved (true positives over the sum of true positives and false negatives). It captures better the performance of the positive class which fits better our task. Precision and recall are also part of the F-score, a weighted average of the precision and recall.

Additionally, we want to avoid the effects of the partitioning of the dataset between the training data and the test data (usually a 70-30 division). To avoid problems as overfitting and getting a more accurate estimate of the model prediction performance we will use

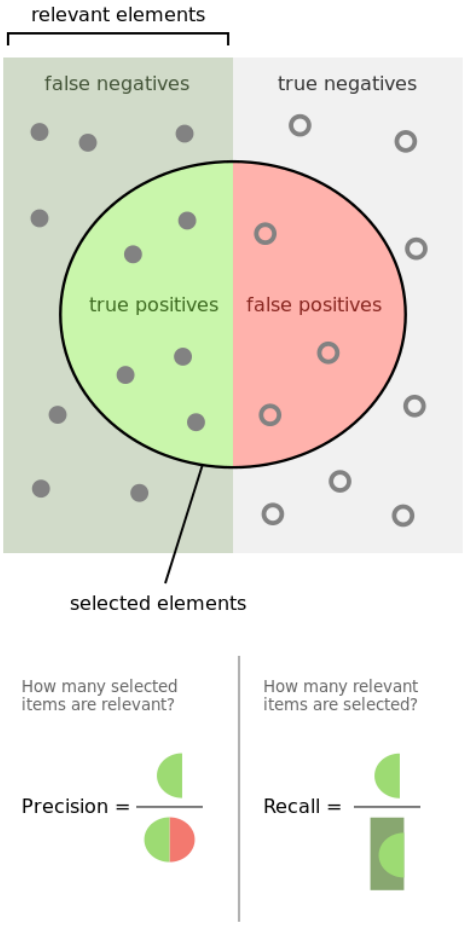


Figure 4.18: Precision and recall explanation (Wikipedia).

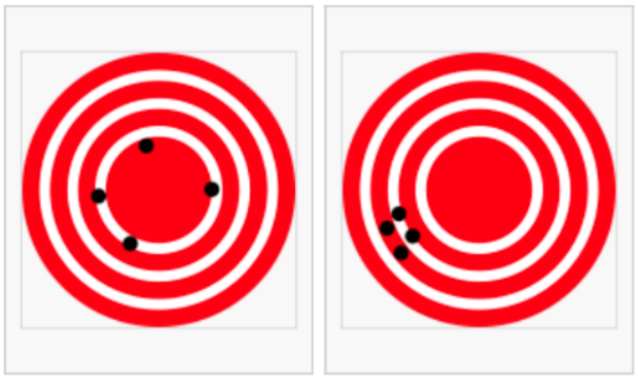


Figure 4.19: Low accuracy, poor precision and good trueness (left) vs low accuracy, good precision and poor trueness (right) (Wikipedia).

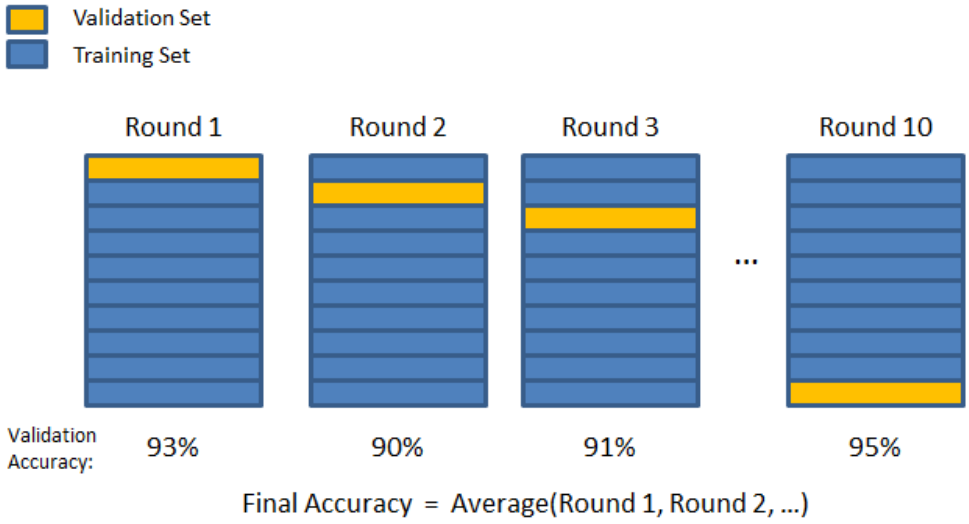


Figure 4.20: Diagram of a 10-fold cross validation.

cross-validation, a model validation technique for assessing how the results of a statistical analysis will generalise to an independent data set. One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the validation set or testing set). To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds. Usually the number of rounds this process is executed is 10, with training sets with a size of 90% the dataset (called 10-fold cross validation as displayed at figure 4.20)

4.4.3 Classification algorithms

We will introduce the classification algorithms that will be benchmarked:

- **K-Nearest Neighbours**: a non-parametric method used for classification and regression.The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. In the classification phase, k is a user-defined constant, and an unlabelled vector (a query or test point) is classified by assigning the label which is most frequent among the k training samples nearest to that query point. A commonly used distance metric for continuous variables is Euclidean distance. For discrete variables, such as for text classification, another metric can be used, such as the overlap metric (or Hamming distance).

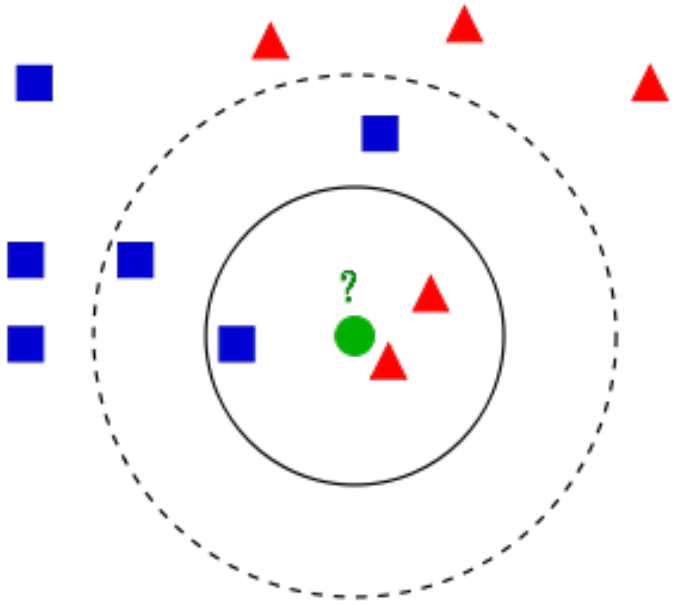


Figure 4.21: K-Nearest Neighbours example: one input classification in a two dimensional space.

- **Support Vector Machine**: a support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks by looking at the hyperplanes that separate the different classes. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalisation error of the classifier.

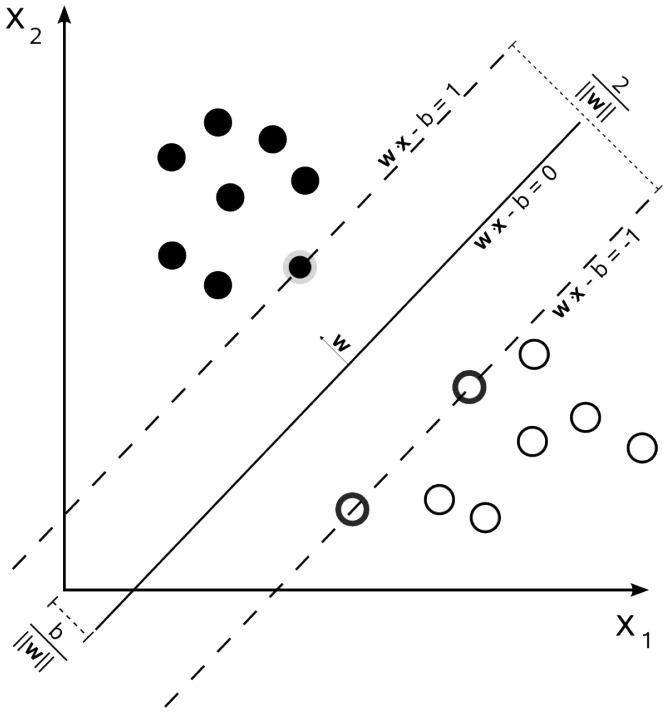


Figure 4.22: Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors (Wikipedia).

- **Decision Tree Learning**: it uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value. A tree can learn by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node has all the same value of the target variable, or when splitting no longer adds value to the predictions. This process of top-down induction of decision trees (TDIDT) is an example of a greedy algorithm, and it is by far the most common strategy for learning decision trees from data.

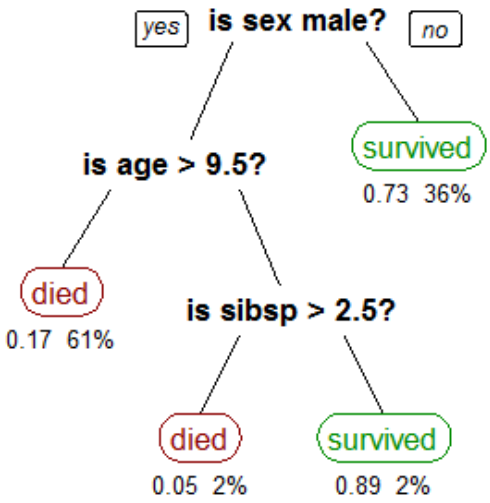


Figure 4.23: A tree showing survival of passengers on the Titanic ("sibsp" is the number of spouses or siblings aboard). The figures under the leaves show the probability of survival and the percentage of observations in the leaf (Wikipedia).

- **Random Forest**: technique that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set at the same time that growth accuracy. It has the same advantages that decision trees as the easiness to understand and interpret, the little need for data preparation, robustness, etc while keeping the advantages of ensembles techniques as avoidance of overfitting or better performance.

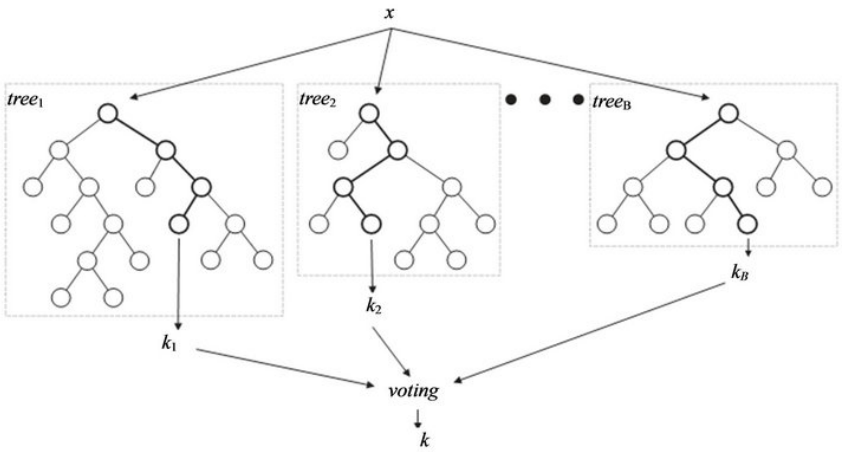


Figure 4.24: How Random Forests work by aggregating votes of randomised Decision Trees.

- **Adaptive Boosting**: also know as *AdaBoost*, is a machine learning meta-algorithm that makes use of other types of learning algorithms to improve their performance. While ensemble algorithms as Random Forests group different *strong learners* to improve their performance, boosting algorithms make use of *weak learners* that as long as the performance of each one is slightly better than random guessing (e.g., their error rate is smaller than 0.5 for binary classification), the final model can be proven to converge to a strong learner. AdaBoost (with decision trees as the weak learners) is often referred to as the best out-of-the-box classifier.

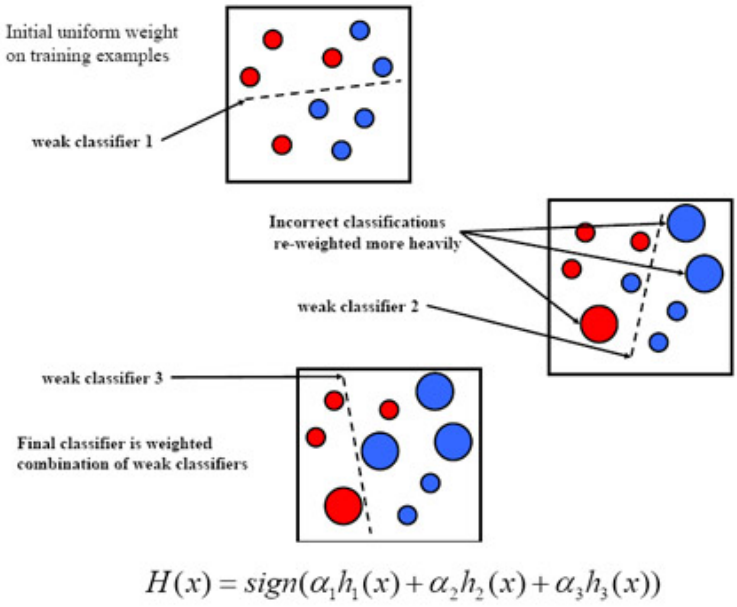


Figure 4.25: Description of Adaptative Boosting.

- **Linear discriminant analysis**: is a generalization of Fisher's linear discriminant, a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. LDA is also closely related to principal component analysis (PCA) and factor analysis in that they both look for linear combinations of variables which best explain the data.

- **Quadratic discriminant analysis**: closely related to linear discriminant analysis, where it is assumed that the measurements from each class are normally distributed. Unlike LDA however, in QDA there is no assumption that the covariance of each of the classes is identical

- **Naive Bayes classifier**: a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. An advantage of naive Bayes is that it only requires a small amount of training data to estimate the parameters necessary for classification.

4.4.4 Benchmark

Now we will use the Python library *scikit-learn* to run each one of these algorithms and compare their performances. While all of them can be tweaked for this approach we will use their default configurations.

Before feeding the features to the classification algorithms is important to notice that as we chose 85% percentile for the positive class the dataset is highly unbalanced. If we feed this data directly to the algorithms we would obtain a classifier that only predicts one of the classes, as that would mean an 85% accuracy. To overcome this problem there are some techniques available:

- **Oversampling**: repeating elements of the smaller class until it reaches the size of the bigger one.

- **Undersampling**: randomly selecting elements of the bigger class until it reaches the

size of the smaller one.

- **Synthetic samples generation**: oversampling method that generates new samples of the smaller class that look like real ones. An implementation of this technique is *SMOTE*.

- **Weighting**: passing weights to the algorithm so it tries to compensate the less populated class with higher weights.

After experimenting with these options we discovered that the best options for this task is undersampling, as the rest of them were lowering the task performance. The downside is that, because we are losing elements from the bigger class, we will have a smaller dataset. Due to a smaller datasets two things can happen: a decrease of the classifier performance, and an increased likelihood of incurring into overfitting (a common problem of machine learning algorithms when they learn to deeply the features of the training set and they are unable to generalise for later runs). To try to overcome a overfitting problem we use cross-validation to evaluate the models.

For each one of the algorithms we have trained a classifier and obtained four metrics (precision, recall, f1-score and accuracy) for the two classification tasks: good users vs the rest of users and bad users vs the rest of users. The results are available at table 4.14.

	Bad users classification				Good users classification			
algorithm	prec.	recall	f1	accur.	prec.	recall	f1	accur.
K-Neighbours	0.537	0.549	0.542	0.535	0.520	0.532	0.525	0.521
SVC	0.538	0.440	0.473	0.528	0.529	0.473	0.475	0.518
Decision Tree	0.528	0.564	0.544	0.530	0.542	0.552	0.546	0.543
Random Forest	0.580	0.557	0.568	0.577	0.599	0.603	0.600	0.599
AdaBoost	0.509	0.489	0.494	0.514	0.533	0.562	0.543	0.534
Naive Bayes	0.501	0.817	0.617	0.501	0.499	0.624	0.527	0.510
LDA	0.542	0.511	0.522	0.539	0.585	0.483	0.527	0.569
QDA	0.510	0.851	0.636	0.515	0.539	0.344	0.353	0.517

Table 4.14: Machine learning classifiers benchmark for our two classification tasks.

As we can see the results show some weak prediction capabilities from most part of the algorithms. To interpret the results we have to have in mind that for these two tasks we are more interested in having a high precision and accuracy than in having a high recall (In fact, it is possible to have perfect recall but with low precision by just labelling all the elements as the positive task).

Between the rest of classifiers the one that performed the best is the Random Forest classifier. This is good news as Random Forests are easily interpretable and it allows to understand which features are more important for the prediction. The results also show how the predictions are stronger for finding good users that finding bad users.

In the next section we will user Feature selection to improve the performance of a Random Forest classifier.

4.5 Feature selection

It's possible to improve the performance of the selected classifier and to remove the unnecessary features so we can discern which ones have predicting power. It can be done making use of the *feature selection* techniques.

Feature selection is the process of selecting subset of relevant features (variables, predictors) for use in model construction. Feature selection techniques are used for three reasons:

- Simplification of models to make them easier to interpret.
- Shorter training times.
- Enhanced generalisation by reducing overfitting.

The central premise when using a feature selection technique is that the data contains many features that are either redundant or irrelevant, and can thus be removed without incurring much loss of information. Redundant or irrelevant features are two distinct notions, since one relevant feature may be redundant in the presence of another relevant feature with which it is strongly correlated.

We have designed a pipeline with two feature selection techniques. Thanks to this pipeline we can effectively reduce the dimensionality by removing redundant and irrelevant features at the same time:

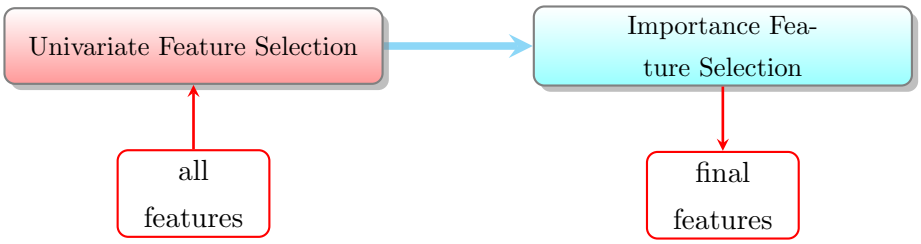


Figure 4.26: Feature selection process. At the end of the process there are less features than at the beginning

- **Univariate feature selection**: removes all *irrelevant* features whose variance doesn't meet some threshold, as features with the same value in all samples.
- **Univariate feature selection**: removes all *irrelevant* features whose variance doesn't meet some threshold, as features with the same value in all samples.

4.5.1 Univariate Feature Selection

With univariate feature selection we can remove the low-variance features, for example the ones that have the same almost the same value across all dataset. This feature selection

algorithm looks only at the features, not the desired outputs, and thus is independent of the classification algorithm be used for unsupervised learning.

The machine learning library that I'm using for the analysis (scikit-learn) come with a bundled univariate feature selector, so the implementation is straighforward.

After running it on the feature matrix (with a size of 131 columns), we obtain a features matrix of 129 columns: the two features removed are *if the twitter profile is verified* and *if the twitter profile is protected* as their never appear as positive or almost never respectively.

4.5.2 Importance Feature Selection

Thanks to using a decision tree based algorithm it is possible to compute the importance of each one of the features. We can leverage it to discard unimportant features: just by setting a threshold for the feature importance weights and discarding those below that threshold we are removing features that add little to no value to the prediction.

By setting such threshold at the mean value of all the importance weights we reduce the size of the features matrix from 129 columns to only 33. The majority of features removed are user interests (100 of the features are interests related) but also others like *if the user still has the default profile picture* or *the presence of numbers characters on the name.*

4.5.3 Results: classifier performance

After the feature selection process in theory the classifier should have improved its performance. To test this hypothesis we have performed a 10-fold cross validation of the Random Forest algorithm inputing the new features dataset and looking for the precision, recall, accuracy and f-score of the classification task. The model evaluation is available at table 4.15.

	precision	recall	f1-score	accuracy
good users	0.604	0.613	0.609	0.606

Table 4.15: Model evaluation for both classification tasks after the feature selection.

The results show a little increase of the classifier performance with not real significance, but a reduce of the features dataset size from 133 features to 33 (24% of the original size).

4.6 Correlations identification

In this chapter we have just developed a process to train a classifier of Twitter users according to their reputation at Wallapop: first we classified them looking at their reputation (section 4.2), then we extracted their Twitter features (section 4.3) and after that a Machine Learning classifier was trained (section 4.4) and improved (section 4.5).

Now it is possible to perform the actual objective of the project: to find what correlations appear between the users OSN data and their reputation. These feature importances can be studied for both classification tasks:

- A classification task to classify users between *exceptional users* (high reputation) and the rest of users. The list of features importances is available at figure 4.27.

- A classification task to differentiate between *bad users* and normal users. The list of features importances is available at figure 4.28.

We trained classifiers with prediction capabilities for both tasks that while they don't reach very high accuracy metrics they are statistically meaningful. By plotting the feature importances for both classifiers (figure 4.27 and figure 4.28) we can check that the important features are the same for both, therefore we can perform a joint analysis and compare the statistics between both user classes.

It's important to notice that because of the nondeterministic nature of the classifier used (Random Forest) the ranking of features can look different in different runs of the training process but the important features are the same between different training processes.

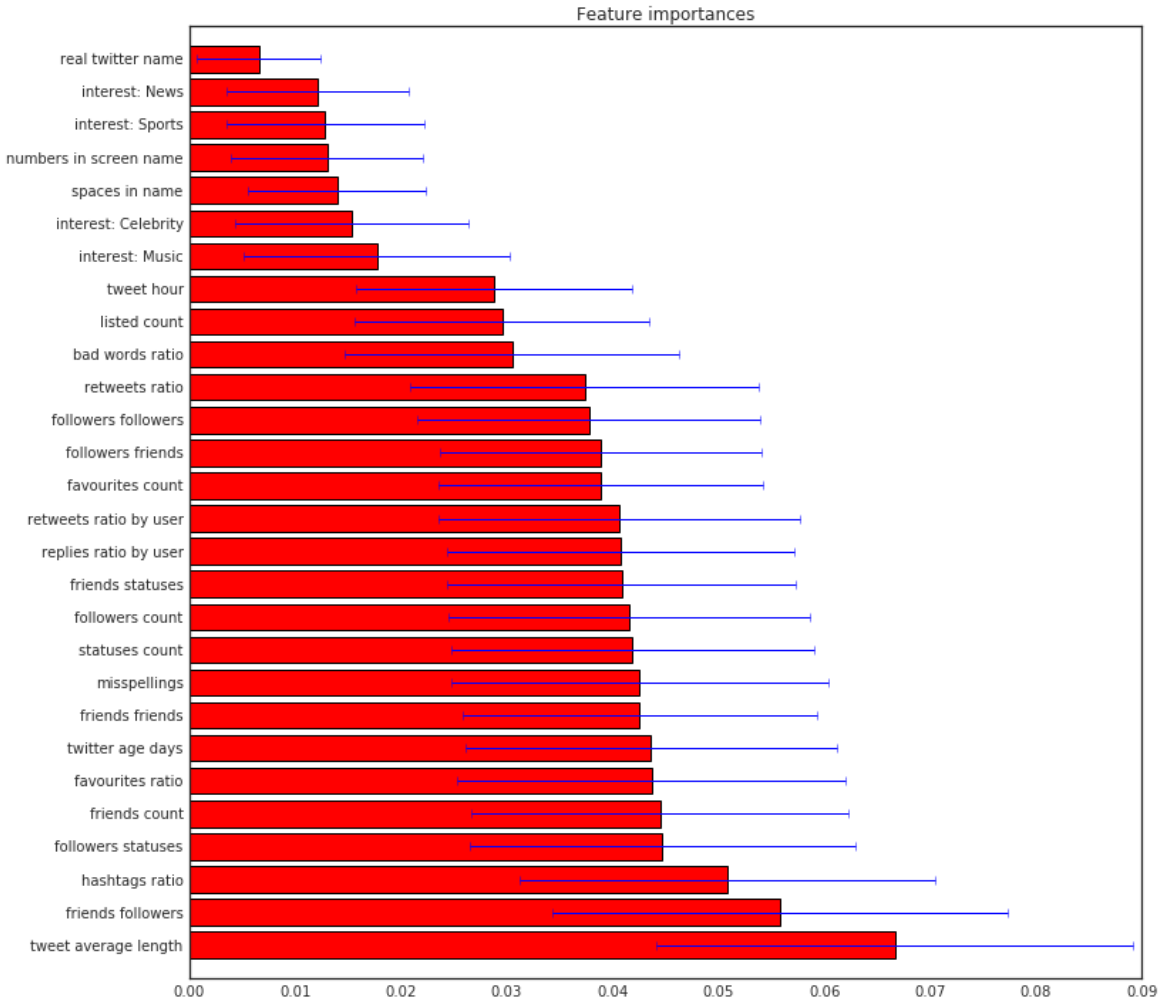


Figure 4.27: Most important Twitter features for good behaviour classifier.

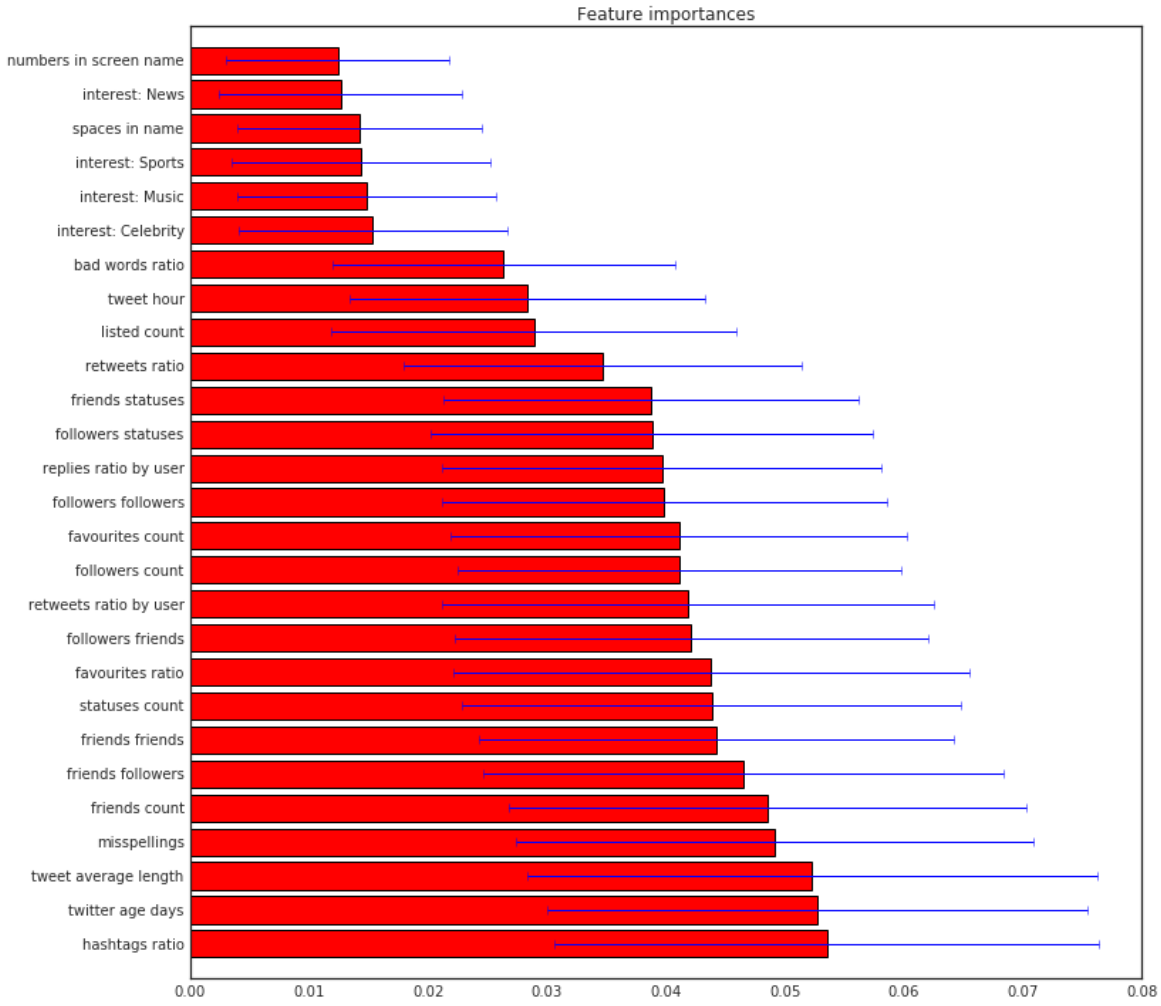


Figure 4.28: Most important Twitter features for bad behaviour classifier.

Between all the features related to user reputation we can group the by four different topics: *user engagement*, *tweeting behaviours*, *network structure* and *influence*.

4.6.1 User engagement

Some of the features we extracted and that showed some relation with user behaviour are related with how engaged the user is with the Twitter platform. We don't know the inherent cause of some users spending more time at Twitter than others, but we can make assumptions: maybe these users are more tech-savvy, maybe the have more content to share (as at Twitter the content you share has more weight in how many followers you have, while at Facebook is more a matter of number of friends).

The set of important features that can be grouped under the *engagement* label are three:

[H]	bad users	normal users	good users
count	470	10925	627
mean	5608.19	5882.34	7533.44
std	15621.04	12925.36	16610.05
min	0	0	0
25%	111	226	326
50%	1128	1363	1935
75%	4509	5545	6812
max	207167	197393	194849

Table 4.16: Comparison of statuses count between different reputation groups.

- **The count of statuses shared by the user**: at table 4.16 we can observe how the group of users with exceptional reputation has higher statistics (e.g. the median) than users with low reputation. This metric also captures the *openness* and preference to share content of users.

- **The count of tweets favourited by the user**: the group with good reputation also has marked more tweets as favourite (table 4.17) pointing again the relation between user behaviour and engagement at Twitter. Nevertheless, while the count of statuses is related with time spent *sharing content* this one is more related to time spen *reading content* as the most time users spend on the platform the higher probability of they seeing something they like.

- **The age of the account in days**: results indicate that *early adopters* seem to have better behaviour that *late majority adopters*. At figure 4.29 we can see a normalised histogram of the Twitter account age (in days) grouped by reputation class: bad users concentrate more in the lower part of it (new accounts) while good users tend to have older accounts. There is also a spike of new accounts for bad users.

	bad users	normal users	good users
count	470	10925	627
mean	1344.33	1867.92	1677.17
std	4256.22	7741.35	4172.56
min	0	0	0
25%	12	27	35
50%	169.5	190	234
75%	855.25	997	1254
max	57829	331134	41883

Table 4.17: Comparison of favourites count between reputation groups.

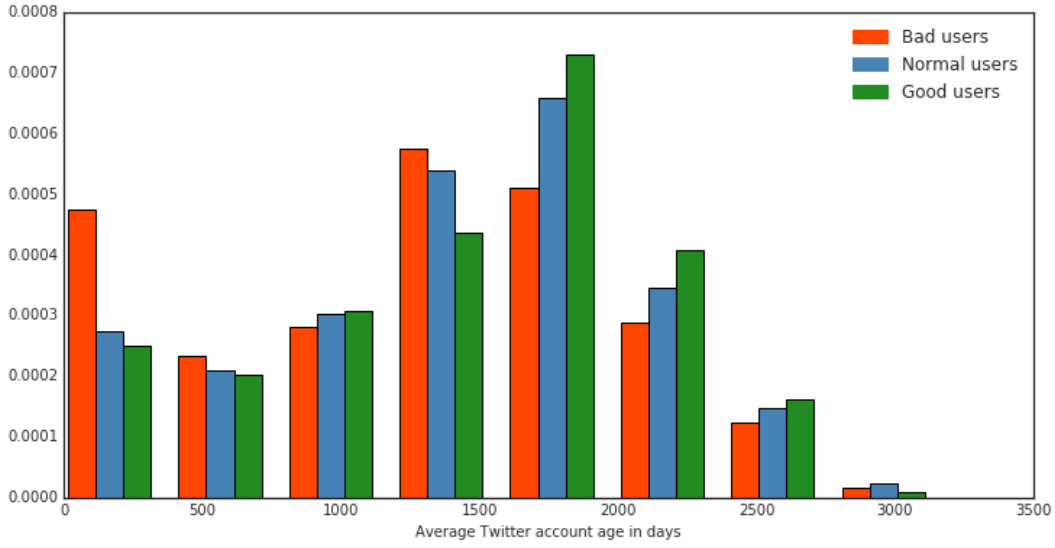


Figure 4.29: Normalised distribution of Twitter account age in days grouped by reputation group.

4.6.2 Tweeting behaviours

Other features relate more with *how users tweet*.

- **Tweet average length**: the results say that *better users share longer tweets*. This is an interesting phenomenon feature length because there can be multiple reasons for some users sharing shorter tweets than others: maybe they are less communicative, maybe they have worse writting skills. Whatever the reason behind it, figure 4.30 shows how users with very short tweets are worse while users that write long tweets are better.

- **Misspellings count**: this feature was extracted because we made the assumption of *poor writing skills* being related to user behaviour because it could capture some characteristics as the educational level. To measure it we counted the ratio of misspellings per word, considering a misspelling anything that is not part of a dictionary: from serious misspellings that affect the readability of the text to others not as serious as *non capitalised city names* or *missing accent marks*. Figure 4.31 confirms this assumption, showing how the distribution of the ratio of misspellings per word has a longer tail for the *bad users* group. This feature is very powerful because **while some of the features are only related to Twitter this can be applied for other platforms**, e.g. the texts that users write about themselves at Sharing Economy platforms.

- **Bad words ratio**: again, this metric is related too to the writing skills and educational level of the users, but it also captures psychological traits as *aggressiveness*. Here our assumptions are confirmed too: while there is not a huge difference in the distribution of the ratio of bad words between the two user groups (figure 4.32), *bad users* swear more often than *exceptional users*.

- **Hashtags ratio per tweet**: we decided to extract the ratio of hashtags shared by tweet because we think that it captures users characteristics: using hashtags at Twitter increases the audience while it doesn't have any cost, so there are no reasons for not using it in a public OSN apart for not, for example, not being very tech-savvy. Figure 4.33 shows how the better users use more hashtags than others.

- **Most common tweeting hour**: another interesting metric involves the most common tweeting hour. As we can observe at figure 4.34 while the behaviour between groups is the same there are two differences:

  - *Bad users* tweet more late in the night (0-1 AM).

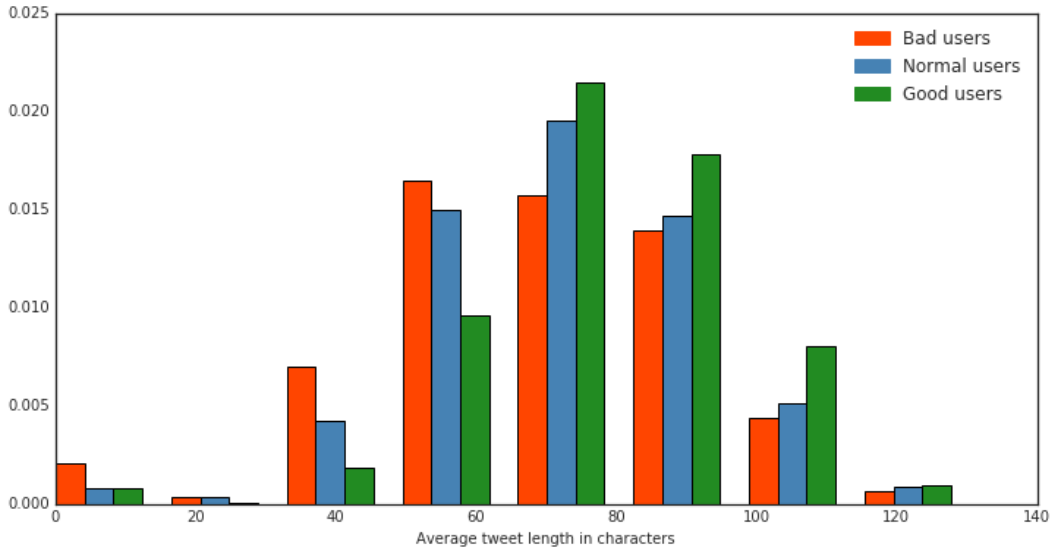


Figure 4.30: Normalised distribution of tweets average length in characters.

– *Exceptional users* tweet early in the morning (7-8 AM).

- **Ratio of shared retweets** and Ratio of shared replies: the last of the important metrics related to *how users tweet* are the ratio of tweets that are replies to others and the ratio of tweets that are retweets of other users content. The only difference between different groups is when the ratios are very low, more related to the *bad users* group (figure 4.35 and figure 4.36).s

4.6.3 Network

When extracting features about the users network we did not only extracted first level metrics but also information about the network one step ahead (the network of the followers and the network of the friends). All the features related to the network appear as important in our classifiers.

The users network consists in a set of directed links, incoming links coming from followers and outgoing links going to friends. The degree of incoming links (followers count) is shown at table 4.18 and the degree of outgoing links (friends count) is shown at table 4.18. Both of them have a weak relation with reputation: higher counts for better users. This could be a side effect of user engagement: spend more time in the platform and you will have more connections.

Nevertheless we have two different sets of features for the network one step away from

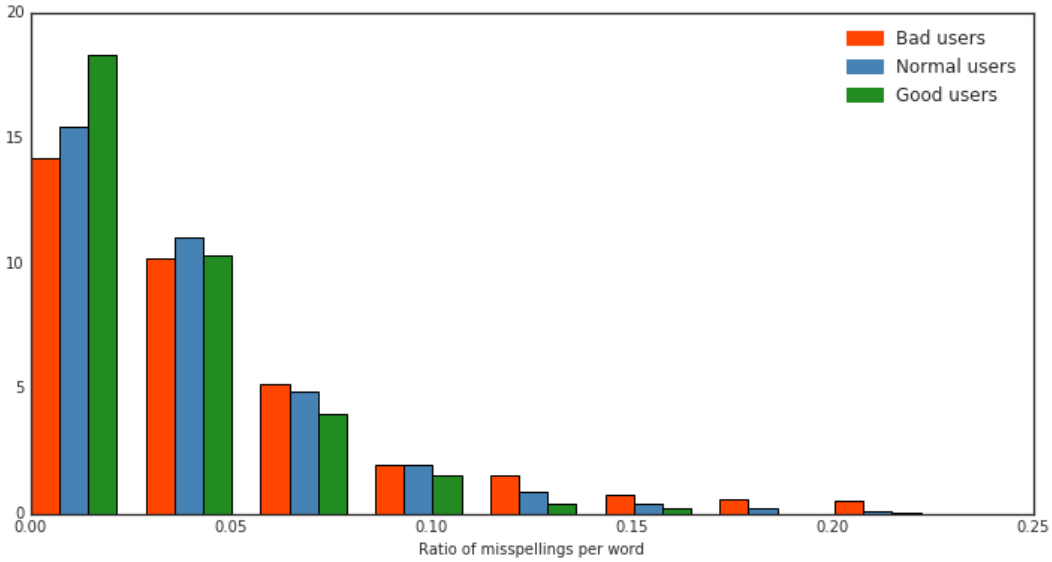


Figure 4.31: Normalised distribution of misspellings per word tweeted.

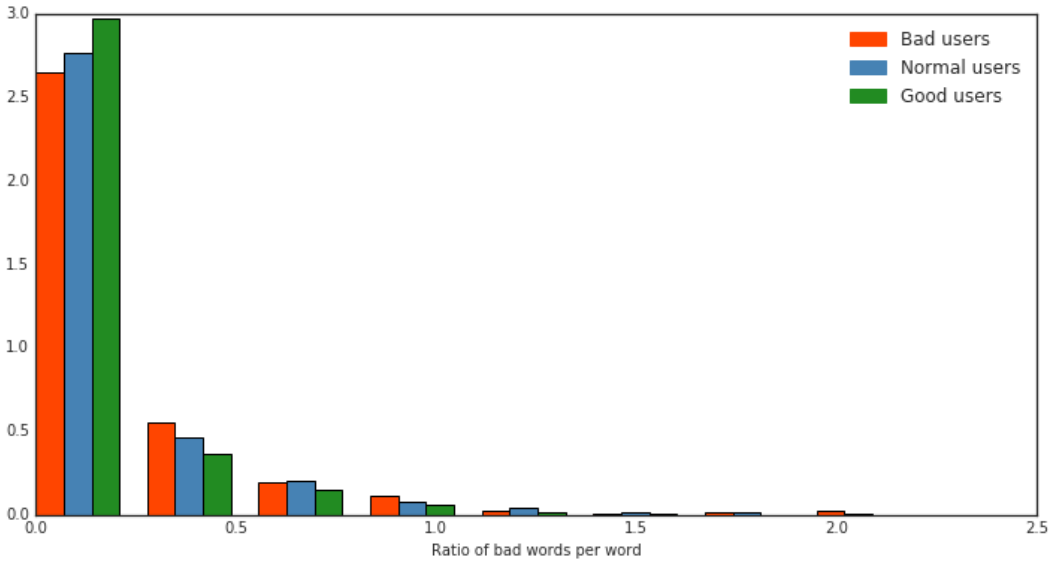


Figure 4.32: Normalised distribution of the ratio of bad words per word.

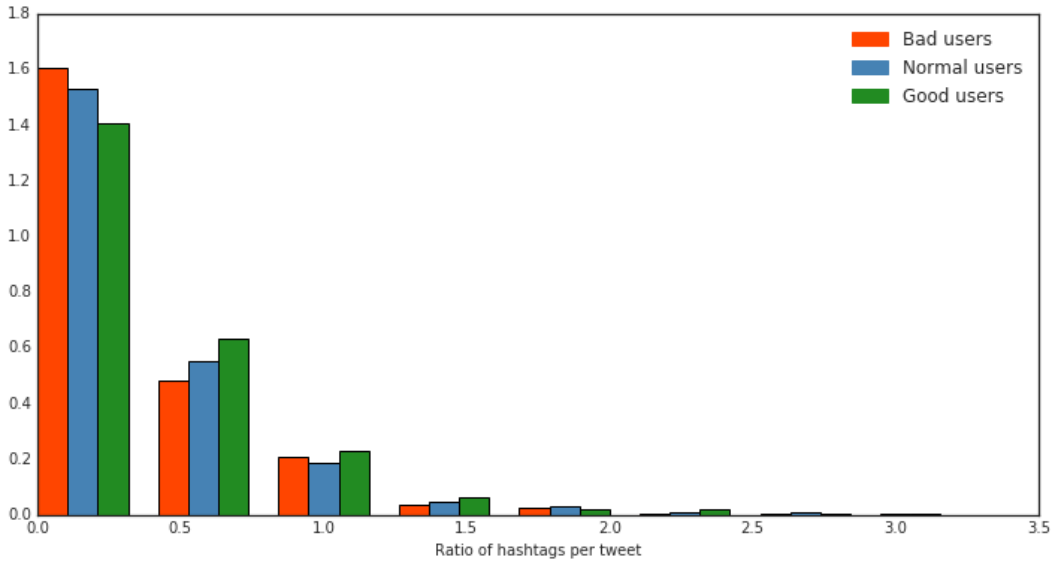


Figure 4.33: Normalised distribution of the ratio of hashtags per tweet.

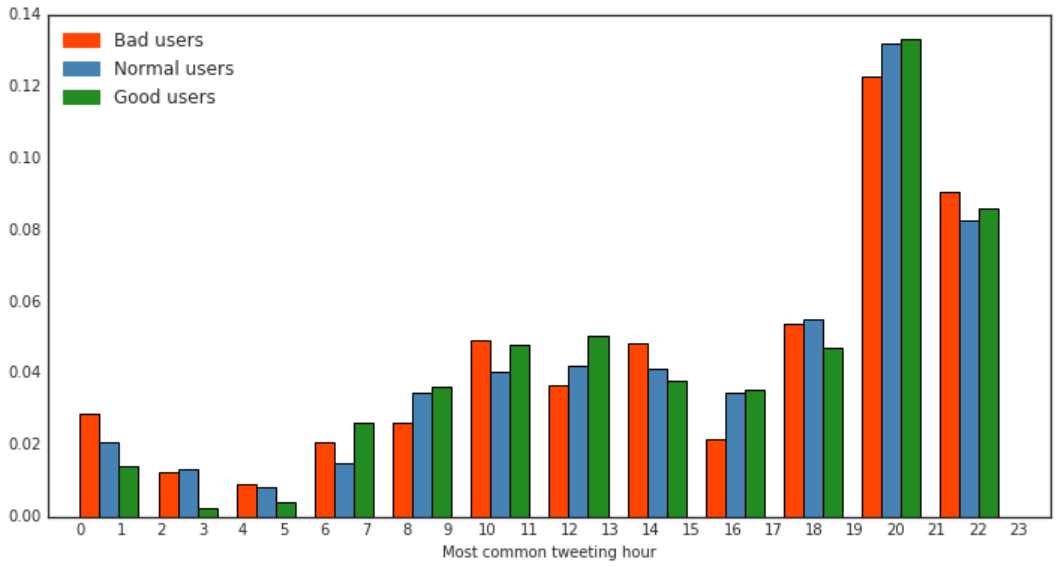


Figure 4.34: Normalised distribution of the most common tweeting hour.

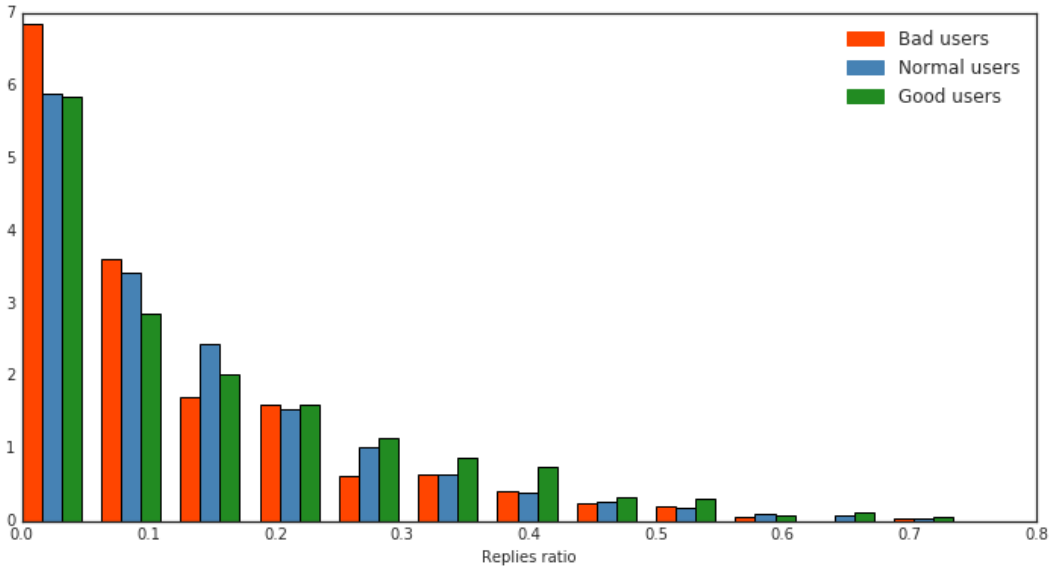


Figure 4.35: Normalised distribution of the ratio of tweets that are replies.

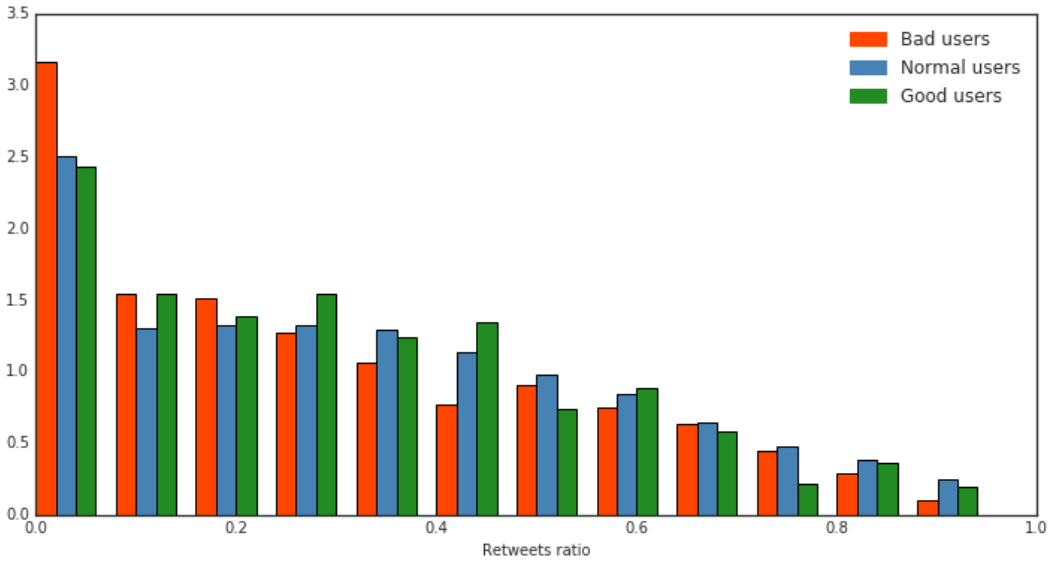


Figure 4.36: Normalised distribution of the ratio of tweets that are retweets.

index	bad users	normal users	good users
count	470	10925	627
mean	382.59	475.81	481.54
std	929.06	3357.23	1692.11
min	0	0	0
25%	48.5	58	58
50%	148.5	158	163
75%	373.5	378	432.5
max	10197	264788	33596

Table 4.18: Followers count by group.

the analysed user. It's important to notice the presence of outliers and therefore that the median and quartiles are better metrics that the mean:

- **Friends network** as the network of who the users are following at. An unexpected result is that the friends of the good users are less popular that the friends of bad users in average (figure 4.20), indicating that bad users are more prone to follow *Twitter celebrities*. Again, when looking at their friends count the number is also high for *bad users* friends. The last of the important features that is related to the friends network is the average statuses count for such friends, but the summarising statistics for this feature don't show any difference between the groups: it looks like the classifier is finding a pattern that involves more than one feature at the same time (figure 4.22).

- **Followers network** as the network of who is following the users. We were expecting to find that the bad users had followers that had less followers themselves (as a measure of influence) but the results suggests that there is almost not difference. Here again we can't find a direct relation between the metric and the result which suggests that the relation is more complex (figure 4.23). The same for the average statuses count, where there is not a direct relation. Nevertheless what it is directly related is the average count friends for the followers: is actually lower for good users, which suggests that followers from good users are more *selective*.

index	bad users	normal users	good users
count	470	10925	627
mean	480.12	434.02	513.69
std	840.27	842.69	1129.35
min	0	0	0
25%	100	115	107.5
50%	254	246	236
75%	522.75	484	510.5
max	9981	35637	21665

Table 4.19: Friends count by group.

index	bad users	normal users	good users
count	470	10925	627
mean	1090747.95	851528.76	510738.8
std	2193094.03	1584584.09	722146.22
min	0	0	0
25%	129142.75	140712.32	82244.55
50%	443780.69	402857.94	272298.76
75%	1097320.68	918482.73	665246.67
max	20607461.38	24698661.2	6676935.45

Table 4.20: Average followers count for the user friends.

index	bad users	normal users	good users
count	470	10925	627
mean	6164.68	4716.67	4017.85
std	10826.6	7173.35	5322.11
min	0	0	0
25%	1234.36	1312.12	1158.5
50%	3017.75	2687.32	2424.85
75%	6593.37	5364.23	4442.58
max	139465.5	255221.33	46294.2

Table 4.21: Average friends count for the user friends.

index	bad users	normal users	good users
count	470	10925	627
mean	14176.86	14229.93	14168.81
std	9586.18	8798.83	7720.69
min	0	0	0
25%	8171.38	8878.59	9342.48
50%	12262.76	12714.82	12957.5
75%	17809.27	17757.24	17423.6
max	69017.57	170259	58454.12

Table 4.22: Average statuses count for the user friends.

index	bad users	normal users	good users
count	470	10925	627
mean	13563.5	13250.47	10501.53
std	20748.49	33175.38	13925.03
min	0	0	0
25%	1723.67	2323.71	2411.02
50%	5823.26	6224.33	6154.95
75%	16240.24	14513.17	13233.14
max	165355.83	1927392.21	135395.85

Table 4.23: Average followers count for the user followers.

index	bad users	normal users	good users
count	470	10925	627
mean	9235.06	7854.64	6792.7
std	15248.74	14965.75	10087.56
min	0	0	0
25%	1502.85	1753	1769.95
50%	4053.84	4048.02	3631.51
75%	10024.89	8792.07	7517.35
max	138486.53	649954.29	100522.02

Table 4.24: Average friends count for the user followers.

index	bad users	normal users	good users
count	470	10925	627
mean	6121.43	6279.23	6759.31
std	5674.37	6080.23	5394.45
min	0	0	0
25%	2271.07	2804.57	3190.93
50%	4682.89	4910.33	5437.55
75%	8138.2	7985.22	8869.48
max	36112.08	194083	42075.33

Table 4.25: Average statuses count for the user followers.

4.6.4 Influence

The last features are related with the *influence* users have at Twitter. We measure influence by the response users gets from their actions at the platform, as the times that each time is favourited or retweeted, or how many times an user has been added to a Twitter list. We have to be careful here too because some outliers can skew the distribution, so it is better to look at the quartiles values that the mean. The features are:

- **Favourites ratio** and retweets ratio: both of them show a weak relation of *good* users receiving better responses from others.

- **Number of times added to a user list**: related with the *good behaviour*, users with high reputation are included more times in Twitter lists than the rest of users. As we said at chapter4.3, Twitter lists are intended for tracking and grouping users under the same topic and being included in one of them imply some sort of *status* and influence.

These are the metrics that our classifier used to gain prediction power for the user reputation classification task, but there are others did not show any importance.

index	bad users	normal users	good users
count	470	10925	627
mean	0.25	0.41	0.28
std	0.83	9.1	0.38
min	0	0	0
25%	0.04	0.05	0.07
50%	0.13	0.13	0.16
75%	0.24	0.28	0.33
max	17.13	756.56	4.22

Table 4.26: Average number of times each tweet is favourited.

index	bad users	normal users	good users
count	470	10925	627
mean	0.12	0.26	0.14
std	0.21	9.89	0.32
min	0	0	0
25%	0.01	0.02	0.03
50%	0.06	0.07	0.08
75%	0.13	0.14	0.17
max	2.04	933.52	6.41

Table 4.27: Average number of times each tweet is retweeted.

index	bad users	normal users	good users
count	470	10925	627
mean	6.15	7.8	14.17
std	36.83	31.04	109.32
min	0	0	0
25%	0	0	0
50%	1	1	3
75%	3	6	9
max	703	1311	2688

Table 4.28

4.6.5 Unimportant features

We can gain more knowledge of the process by also looking at what *it is not* important when trying to predict a user reputation.

The first set of unimportant features contains all the profile features. Despite the fact that some of the important features seems to be relate to the technical knowledge of users, contrary to our intuition the fact that users still have the *default profile image* or have not yet personalised their profiles are not signals for bad behaviours. They are neither the presence of a description about themselves of the inclusion of web links into such descriptions (that we thought could be related with having personal projects, business, etc).

Other features, as the *verified* of *protected* state of the profiles did not pass the feature selection process because they had a very low variance: with most of the users having the same value the information gain of adding to the classifier is very low.

We also had the intuition that using real names as Twitter names could influence the behaviours of users at Wallapop. Nevertheless the importance of this feature is too low to be considered, so again our assumptions **using a real name it is not a signal of good behaviour**. Neither they are using number or spaces in the names.

The user interests show correlations that are too weak. The reasons for this are:

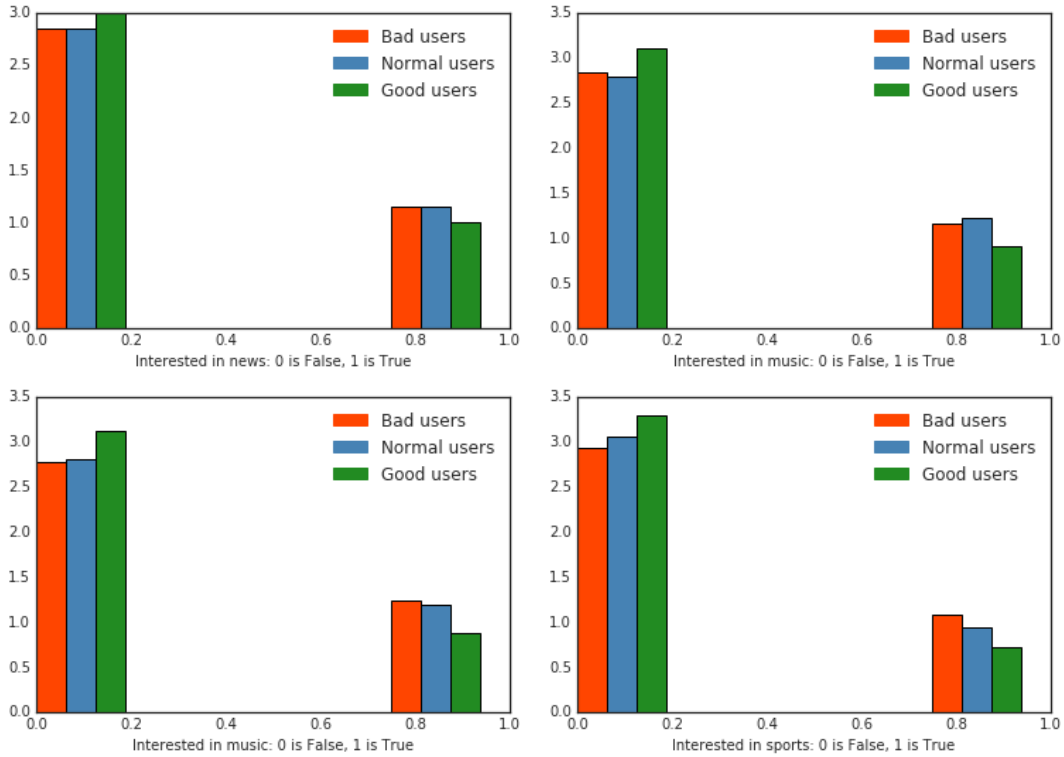


Figure 4.37: Interests by group: news, music, celebrities and sports.

- Low variance: for an interest the majority of users has that interest feature with value zero as they don't follow anybody from that field.
- Weak correlation: in addition to having low variance, none of them showed having a strong relation with the reputation (despite we assumpt that some of them as *Christianity* would do).

The ones that showed a weak importance on the classifier is because of them being one of the most popular: *Sports*, *Music*, *News* and *Celebrities* (figure 4.37).

In the next chapter we will summarise this findings and their potential for the future.

CHAPTER 5

Conclusions and future work

In this chapter we present the conclusions of this master thesis. We analyse how the proposed solution improves the current situation for new users at Sharing Economy platforms and propose new and more powerful solution for improving the lives of underserved people. To finish we propose what steps should happen next to continue this master thesis work.

5.1 Conclusions

In this project we have identified relations between users behaviours at *a peer-to-peer marketplace (Wallapop)* and at an *Online Social Network (Twitter)*.

Such process is intended to mitigate the problem that new users face at Sharing Economy platforms: existing users do not trust them because they can not prove their reputation. By leveraging the Online Social Networks data that new users currently provide when joining Sharing Economy platforms it is possible to predict their future behaviours and therefore act accordingly to these predictions: increasing trust for possible new users and taking preventive actions against possible bad users.

By reducing risks and increasing trust on these platforms Sharing Economy platforms would increase their user base, allowing users that do not fit nowadays (non tech-savvy users, minorities, etc) to also benefit from the economic, social and even environmental benefits enabled by these new transactions, and doing it by leveraging an asset that they already own: their Online Social Networks accounts.

But such benefits are not only tied to the Sharing Economy: there are many people that do not have access to some benefits of our current society because they are not trusted: e.g immigrants being rejected by landlords or banks refusing credits to local business owners at third world countries.

India is the second country with more Facebook users in the world, but at the same time part of their population is underserved by banks and other institutions because the lack of identification, credit scoring, and in definitive reputation information about their inhabitants. If we can prove who the trustworthy people are it is possible to empower them to improve their lives, for example by having access to micro-credits to start new business.

Another example of underserved people is immigrants being rejected by landlords when trying to rent a property. This is another situation when an alternative reputation metric that can be generated with existing data such as the available at Online Social Networks (that is closely related to the live of their owners) would improve their lives. By increasing trust for underserved people we can empower them to have access to assets that they can not get today.

To summarise: while this project is an introduction to the topic it has reached its goal of proving the existence of correlations between users behaviours and Online Social Network data. Given the implications that it could bring to underserved people, the environment and in general to modern societies it should be researched more deeply.

5.2 Achieved goals

This project faced multiple challenges. The first one appeared when gathering the need data as it was necessary to match users between a Sharing Economy platform and a Online Social Network. After researching the options to build such dataset of matched users and finding that there are not any public directory that exposes such information, we developed a process that uses the Twitter API capabilities and the characteristics of the *Sharing Economy* platforms' user interfaces to effectively match the users.

Later on it was needed to download data from a Sharing Economy platform (Wallapop). It was found that they do not expose any official way to do it, so we analysed how the official smartphone applications internally connect to Wallapop servers and replicated that behaviour to build a data scraping solution.

After downloading the data and starting to analyse it we found that some interesting user characteristics were not directly available on the data. To extract these user features we had to design custom solutions. The first of these user features is the count of times a user writes a bad word. To build a dataset of bad words we searched for lists of banned words at online forums and merged them all together. By the presence of every word on the list it is possible to extract this metric.

Later on we started to generate a metric to capture users writing skills by counting their number of misspellings. For it we leveraged existing spell checker libraries and designed a system that checked every user generated word agains a list of dictionaries in multiple languages.

For last one of these features we wanted to extract the interests of Twitter users. To reach this goal we scraped a website for categorised information about popular Twitter accounts and then analysed the connections of each one of the users at our dataset to check if they were following their content.

We also faced problems with the Machine Learning algorithms overfitting the user features. To solve this problems we made use of cross validation technologies and algorithms tuning.

At the end of the project we successfully reached the goal of finding the existence and what features relate with user behaviour.

5.3 Future work

Next developments should be focused on generalise these findings for both: more peer-to-peer platforms and more Online Social Networks.

We have already started by analysing if the same effects happen also at Airbnb. By analysing a dataset of 200 million Twitter profiles and searching for Airbnb invitation links (similar to the data acquisition process of this master thesis) we have built a dataset of more than 10000 matches between the two platforms. If we can prove that the same relations can be found also at Airbnb then the result would me more powerful.

Finding matches with other Online Social Networks is more difficult: as explained at chapter 3 data from platforms as Facebook is private and has to be extracted with the explicit users consent. Nevertheless Sharing Economy platforms have access to this private data therefore they could perform such analysis using data from their users.

This project has been developed at Traity, a reputation management platform where users can export their reputation from closed platforms to be used anywhere. At Traity users connect their accounts from both Sharing Economy and Online Social Network platforms. Empowering underserved users is one of the main values of Traity's company values, which together with the access to the right data is the perfect scenario for future work on the field.

Bibliography

[1] J. Hamari, M. Sjöklint, and A. Ukkonen, "The sharing economy: Why people participate in collaborative consumption," *Journal of the Association for Information Science and Technology*, 2015.

[2] G. Zervas, D. Proserpio, and J. Byers, "The rise of the sharing economy: Estimating the impact of airbnb on the hotel industry," *Boston U. School of Management Research Paper*, vol. 16, p. 2014, 2013.

[3] G. Akerlof, *The market for "lemons": Quality uncertainty and the market mechanism*. Springer, 1995.

[4] R. Botsman and R. Rogers, *What's mine is yours: how collaborative consumption is changing the way we live*. Collins London, 2011.

[5] P. Kollock, "The production of trust in online markets," *Advances in group processes*, vol. 16, no. 1, pp. 99–123, 1999.

[6] E. Fehr, "On the economics and biology of trust," *Journal of the European Economic Association*, vol. 7, no. 2-3, pp. 235–266, 2009.

[7] B. King-Casas, D. Tomlin, C. Anen, C. F. Camerer, S. R. Quartz, and P. R. Montague, "Getting to know you: reputation and trust in a two-person economic exchange," *Science*, vol. 308, no. 5718, pp. 78–83, 2005.

[8] I. Bohnet and R. Zeckhauser, "Trust, risk and betrayal," *Journal of Economic Behavior & Organization*, vol. 55, no. 4, pp. 467–484, 2004.

[9] R. Bapna, A. Gupta, S. Rice, and A. Sundararajan, "Trust, reciprocity and the strength of social ties: An online social network based field experiment," in *Conference on Information Systems and Technology (CIST)*, 2011.

[10] R. Wilson, "Reputations in games and markets," *Game-theoretic models of bargaining*, pp. 27–62, 1985.

[11] J. Gebbia, "How airbnb designs for trust," 2015. [Online; accessed 8-June-2016].

[12] A. engineering, "Building for trust," 2015. [Online; accessed 8-June-2016].

[13] A. Llorente, M. Garcia-Herranz, M. Cebrian, and E. Moro, "Social media fingerprints of unemployment," *arXiv preprint arXiv:1411.3140*, 2014.

[14] S. D. Gosling, A. A. Augustine, S. Vazire, N. Holtzman, and S. Gaddis, "Manifestations of personality in online social networks: Self-reported facebook-related behaviors and observable profile information," *Cyberpsychology, Behavior, and Social Networking*, vol. 14, no. 9, pp. 483–488, 2011.

[15] R. Botsman, "The currency of the new economy is trust.," 2012. [Online; accessed 8-June-2016].

[16] statista, "Facebook statistics.," 2016. [Online; accessed 8-June-2016].

[17] Wikipedia, "Lyft — wikipedia, the free encyclopedia," 2016. [Online; accessed 8-June-2016].

[18] Wikipedia, "Uber (company) — wikipedia, the free encyclopedia," 2016. [Online; accessed 8-June-2016].

[19] Wikipedia, "Airbnb — wikipedia, the free encyclopedia," 2016. [Online; accessed 8-June-2016].

[20] Wikipedia, "Couchsurfing — wikipedia, the free encyclopedia," 2016. [Online; accessed 8-June-2016].

[21] Wikipedia, "Etsy — wikipedia, the free encyclopedia," 2016. [Online; accessed 8-June-2016].

[22] Wikipedia, "Ebay — wikipedia, the free encyclopedia," 2016. [Online; accessed 8-June-2016].

[23] Crunchabse, "Wallapop - crunchbase," 2016. [Online; accessed 8-June-2016].

[24] Wikipedia, "Taskrabbit — wikipedia, the free encyclopedia," 2016. [Online; accessed 8-June-2016].

[25] A. Calì, D. Calvanese, S. Colucci, T. Di Noia, and F. Donini, "A logic-based approach for matching user profiles," in *Knowledge-Based Intelligent Information and Engineering Systems*, pp. 187–195, Springer, 2004.

[26] V. Arnaboldi, A. Passarella, M. Conti, and R. I. Dunbar, *Online Social Networks: Human Cognitive Constraints in Facebook and Twitter Personal Graphs*. Elsevier, 2015.