UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

DESIGN AND DEVELOPMENT OF A PERSONALITY PREDICTION SYSTEM BASED ON MOBILE-PHONE BASED METRICS

CARLOS ALONSO AGUILAR

2017

TRABAJO FIN DE GRADO

Título:	Diseño y desarrollo de un sistema capaz de predecir la personalidad usando datos obtenidos de teléfonos móviles
Título (inglés):	Design and development of a Personality Prediction System based on Mobile-Phone based Metrics
Autor:	Carlos Alonso Aguilar
Tutor:	Carlos A. Iglesias Fernández
Departamento:	Ingeniería de Sistemas Telemáticos

MIEMBROS DEL TRIBUNAL CALIFICADOR

Presidente:

Vocal:

Secretario:

Suplente:

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

TRABAJO FIN DE GRADO

DESIGN AND DEVELOPMENT OF A PERSONALITY PREDICTION SYSTEM BASED ON MOBILE-PHONE BASED METRICS

Carlos Alonso Aguilar

Junio de 2017

Resumen

La necesidad de comunicación entre las personas es algo que que va impregnado en nuestra naturaleza como seres humanos, pero la forma en la que esto se realiza ha cambiado en los últimos años desde la llegada de los teléfonos móviles y posteriormente con la llegada de Internet. Por otro lado, conocer la personalidad de un individuo es algo complicado que se consigue a base de tener cada vez una mayor comunicación con esa persona. A raíz de la relación existente entre estos dos aspectos, surge este trabajo de fin de grado, cuyo objetivo es predecir la personalidad de las personas a partir de datos recogidos a través de los teléfonos móviles, usando técnicas de Big Data y Machine Learning.

En primer lugar, serán descritas las diferentes tecnologías usadas durante el desarrollo del proyecto para el tratamiento de grandes cantidades de datos y para el posterior proceso de aprendizaje automático supervisado. Tras esto, se procede al caso de estudio en el a partir de datos de llamadas, mensajes y localización, se extraen las características con las que luego se entrenarán algoritmos de Machine Learning. Se diseñarán tres sistemas diferentes basándose en características extraídas usando: llamadas, mensajes y una combinación de ambos.

Por otra parte, desde la polarización de los teléfonos inteligentes, los hábitos de uso a la hora de comunicarnos han cambiado mucho. Cada vez hacemos menos llamadas, ya no enviamos SMSs y nuestra principal vía de comunicación son las aplicaciones de mensajería instantánea como WhatsApp. Por ello y con la idea de poder aplicar los modelos construidos en el mundo actual, se utilizará el sistema de aprendizaje creado a partir de SMSs para predecir la personalidad usando, como entrada del sistema, las conversaciones de WhatsApp.

Como conclusión, en este proyecto se han desarrollado por tanto tres sistemas de aprendizaje automático y uno de ellos ha sido aplicado a WhatsApp con una tasa de acierto mayor de lo esperada teniendo en cuenta que no existían precedentes en este ámbito.

Palabras clave: Machine Learning, Big Data, WhatsApp, Personalidad, Telecomunicaciones.

Abstract

The need of communication between people is something that is associated in our nature as human beings, but the way people do it has completely changed since the smartphone and Internet appeared. Otherwise, knowing human personality of someone is something really difficult that we gain after working communication skills with others. Based on this two principal points in my TFG election, whose aim is predict human personality by recollecting information of smartphones, using Big Data and Machine Learning techniques.

Firstly, we will proceed with a description of the diverse technologies utilized during the project's development for the manage of large quantities of data, and for the future automated process of supervised learning. After this, we will proceed with the study of the cases using data compiled from calls, messages, and geolocation, from which we will obtain the diverse characteristics that will later be compiled into the algorithms of the Machine Learning program. Three different systems will be designed using the features obtained from: calls, messages, and a combination of them both.

On the other hand, communicational habits have shown a tremendous change since the polarization of smartphones. The use of calls has seen a steady decrease, while we progressively tend to ignore SMS's and communicate largely using instant messaging apps, like WhatsApp. For this reason, and with the objective of applying our models to the real world, the learning system created from SMS's to predict the user's personality will focus and use data entirely from WhatsApp conversations.

In conclusion, this project has seen the development of three automated learning systems, with one of those previously mentioned applying data extracted from WhatsApp conversations. This has in turn lead to a greater percentage of success from the expected value, considering that there was no precedence in this area.

Keywords: Machine Learning, Big Data, WhatsApp, Personality, Telecommunications.

Agradecimientos

Gracias a todas aquellas personas que, en general, han confiado en mi.

Contents

List of Figures

CHAPTER 1

Introduction

1.1 Context

We live in a society in which smartphones have become undoubtedly an extremity of our body. These cell phones collect a great amount of data that can be taken advantage of. Nowadays, companies are very interested in techniques like Big Data and Machine Learning in order to use these data to improve their business. However, there are a lot of them that are not leveraging the potential of the data collected, like telecommunications companies.

We are aware that companies are able to predict our tastes, analysing the navigation data (history, likes in social media, etc.) and offer us customized advertising based on this; but the society does not think it could be possible to estimate our personality through data collected by the smartphone. More particularly, using information accessible by the telecommunications companies like location, phone calls or messages.

If this prediction is made correctly, the data collected by the phones could be especially valuable for the companies and would be a great alternative to some traditional methods like surveys that have a higher cost.

All this is possible thanks to the advances in Machine Learning and Big Data, giving rise to other a lot of studies in different areas that, for example, relating the mood of a person with the way in which he uses social media.

1.2 Project goals

The main goal of this work is to design and implement a system that is able to estimate our personality from certain data. In general terms, to analyse the data collected by the telecommunications company and extract some useful information about the clients. More particularly, the information will be the personality for each of them.

On one hand, the data is obtained through the phone communication activity and these are the ones that we are going to use in this project: Phone call logs, SMS logs and the estimated location of the users.

On the other hand, the personality is predicted based on the Big Five personality traits. This widely examined theory suggests five broad dimensions used by some psychologists to describe the human personality and psyche. The five factors have been defined as: Openness to experience, Conscientiousness, Extraversion, Agreeableness and Neuroticism.

This project will specifically focus on classification techniques, which try to identify which class an object belongs to. Besides obtaining the best possible accuracy results, it is also important to analyse in each case which features determine better if a user belongs to a class or another in order to get conclusions from them.

1.3 Project Phases

The tasks to be performed are divided into the following phases:

- Phase 1: Review of literature about the topic.
- Phase 2: Learning techniques and tools for developing the work.
- Phase 3: Analysis of the data obtained for detecting patterns that allow us to find the difference between some personality types or others.
- Phase 4: Study and search of automatic learning algorithms that best fit this case
- Phase 5: Software development and experimentation
- Phase 6 : Results analysis for obtaining resemblance with reality

CHAPTER 2

Enabling Technologies

2.1 Introduction

In this chapter, we are going to give an insight into techniques used in this project. First of all, we are going to talk about Machine Learning and Big Data. Then we are going to explain Scikit-Learn, the Python library in charge of the machine learning algorithms. And finally, we are going to take a look at Pandas, a library that provides data analysis tools for Python.

2.2 Big Data

Nowadays, everyone is talking about Big Data [8], but not all of them really now what it means, because is a concept and not a technology itself. It encloses a group of new techniques for data processing, which are able to process data sets that are so large that the traditional methods could not do it. The goal is to extract value from the amount of data that is collected. The data sets are growing rapidly due to the fact that the number of devices connected to Internet are increasing rapidly. With the uptake of Internet of Things, this new techniques has become strictly necessary. Studies form 2016 forecast that in 2020 the number of connected devices will be around 30 billion [6], a pretty staggering quantity, but lower than 50 billion, predicted in 2010 by Cisco [7].

2.3 Machine Learning

Machine Learning [2] is a field of computer science that gives computers the ability to do something for what they have not been programmed to. That field shows the progress in pattern recognition and computational learning theory in artificial intelligence, developing algorithms that can learn from a data set and make predictions about it. In our case, machine learning is very useful because developing a specific algorithm with a good performance would be very difficult. Depending on the data that we use to train our system, we can cassifield machine learning into three categories:

- *Supervised learning*: The data set includes the correct output, and the algorithm looks for a function that relates both. In turn, supervised learning is divided into:
  - *Regression*: In that kind of problem, we are trying to predict results of a function with a continuous output.
  - *Classification*: In this case, the output is a set of discrete values. In other words, the goal is to map input variables into different categories through algorithms, known as classifiers. In the next chapter, we are going to deepen the most popular classification techniques.

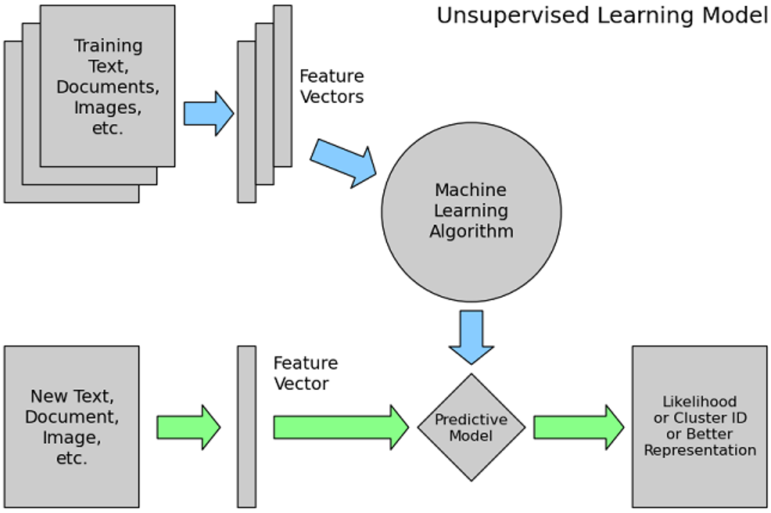

Unsupervised Learning Model
Training Text, Documents, Images, etc.
Feature Vectors
Machine Learning Algorithm
New Text, Document, Image, etc.
Feature Vector
Predictive Model
Likelihood or Cluster ID or Better Representation

Figure 2.1: Unsupervised Learning Model

- *Unsupervised learning*: The dataset given is not labeled, so the algorithm tries to find structure, models, patterns, and then classified it. In this problems we do not know what our results should look like, so it is impossible to value, in a numeric way, how good the system works. Unsupervised learning problems are categorized into Clustering and Collaborative Filtering problems.

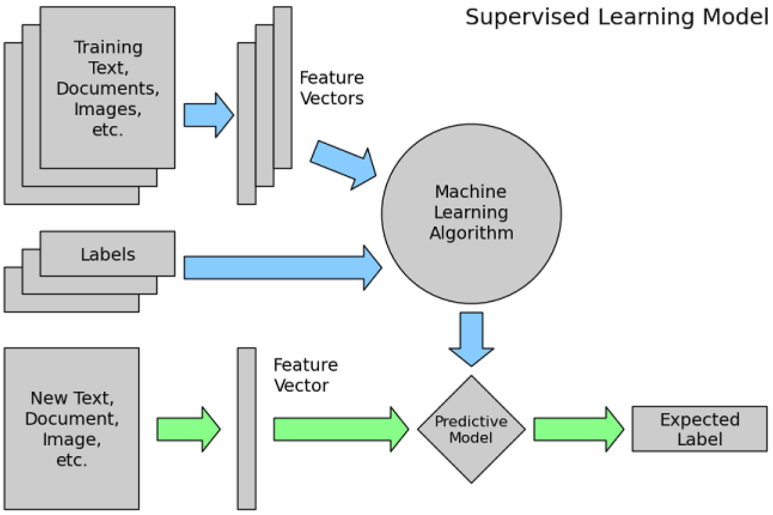

Supervised Learning Model
Training Text, Documents, Images, etc.
Feature Vectors
Machine Learning Algorithm
Labels
New Text, Document, Image, etc.
Feature Vector
Predictive Model
Expected Label

Figure 2.2: Supervised Learning Model

- *Reinforcement learning*: Inspired by behaviorist psychology, this area studies how a software can make decisions in a dynamic environment and this is why is called approximate dynamic programming. A clear example are the autonomous car systems [9] that allow to avoid obstacles based on the data collected by sensors.

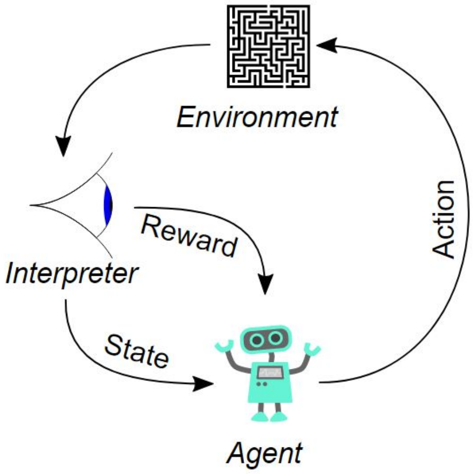


Figure 2.3: Reinforcement learning Model

If we want to achieve a good learning, we need an input with a lot of data and this is why Big Data and Machine Learning are very related with each other. The data set has to be preprocessed using Big Data techniques before putting it into the algorithm.

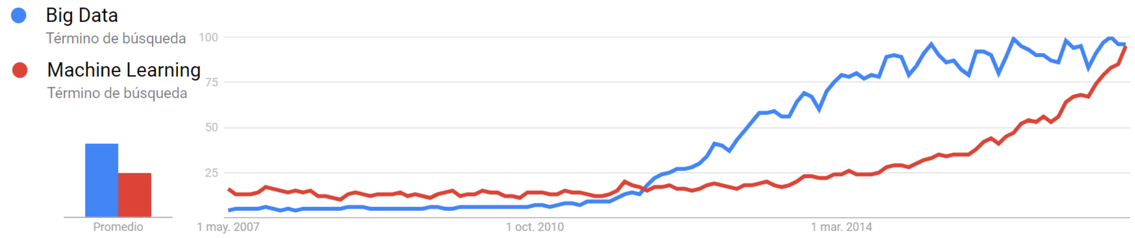


Figure 2.4: Popularity of Machine Learning against Big Data

As we can observe in the figure 2.4, the interest on Machine Learning started when Big Data was already popular. These is because the potential of artificial intelligence has increased enormously thanks to the advantages of new techniques for managing big files. Nowadays, both go hand in hand and are one of the highest interest for companies [4].

2.4 Scikit-Learn

Scikit-Learn [3] is a Python library that provides a wealth of machine learning algorithms. The library is open source and it is built upon SciPy (Scientific Python). It contains various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN and also has some tools for converting objects to Numpy or SciPy representations, which are used by the estimators.

The library is focused on modelling data. It is not focused on the preprocessing phase, like loading, manipulating and summarizing data. For these features, we will use NumPy and Pandas, libraries we will describe later.

When the data set is ready, it is time to choose what is the best algorithm that you can use. As shown in Fig 2.5 we have a good decision tree which helps us to understand which could be the best choice in each case.

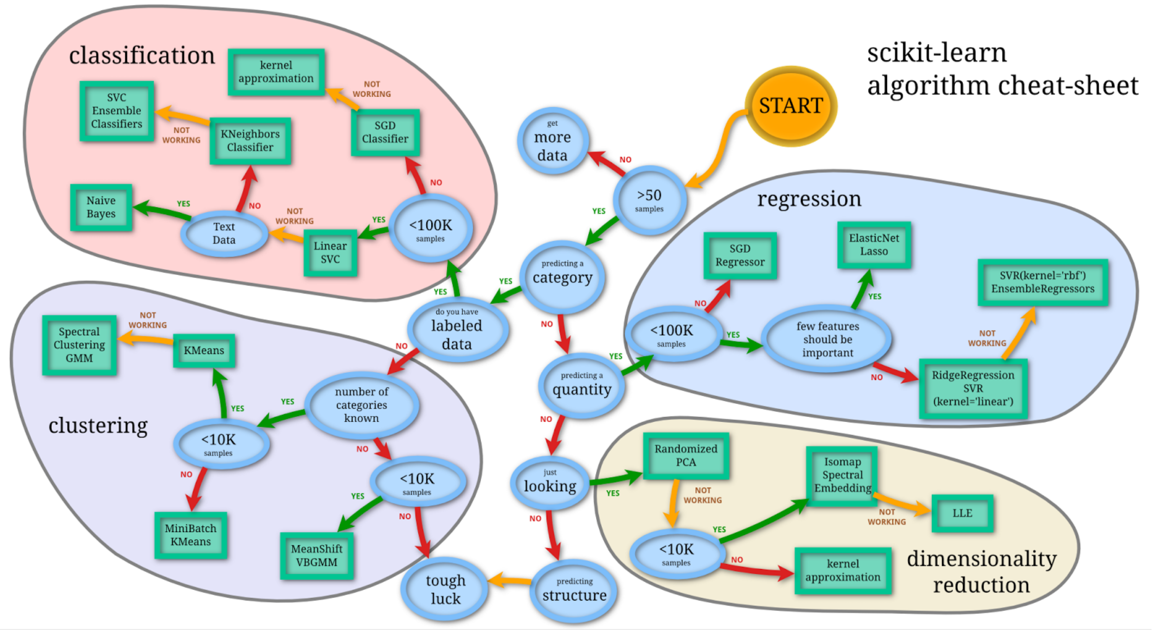


Figure 2.5: Scikit-learn algorithm cheat-shet

2.5 NumPy

Numpy[10] is the foundational python library for scientific computing. Proof of this is the fact that most of the libraries are based on this one, for example pandas. It provides a fast and efficient multidimensional array object, tools for reading and writing data sets to disk, some linear algebra operations and it supports the integration with other languages such as C, C++ or Fortran. Thanks to this library, we have arrays that are much more efficient in storing and manipulation data than the other built-in Python.

2.6 Pandas

Pandas [1] is a library written for the Python language for data modeling and analysis. Python has long been great for data munging and preparation so it allows us to continue with the data analysis in Python, without changing to other lanaguage more specific like R. What makes it special is that offers several features for import data, make operations in numerical tables and time series. These features are:

- DataFrame object for data manipulation with integrated indexing.
- Tools for reading and writing data between in-memory data structures and different

formats: CSV and text files, Microsoft Excel, SQL databases, and the fast HDF5 format.

- Intelligent data alignment and integrated handling of missing data: gain automatic label-based alignment in computations and easily manipulate messy data into an orderly form.
- Reshaping and pivoting of data sets.
- Label-based slicing, fancy indexing, and subsetting of large data sets.
- Columns can be inserted and deleted from data structures for size mutability.
- Group by engine allowing split-apply-combine operations on data sets.
- Data set merging and joining.
- Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.
- Time series-functionality: date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging. Even create domain-specific time offsets and join time series without losing data.

CHAPTER 3

Case Study

3.1 Introduction

The “Friends and Family” living laboratory study was conducted over a period of 15 months between March 2010 and June 2011, with a subject pool of 140 individuals. The dataset contains 20 million Wi-Fi scans (243 million scanned devices), 5 million Bluetooth proximity scans (16 million scanned devices), over 200,000 phone calls, 100,000 text messages, and more. The study also collected self-reported personal information on each participant, such as age, gender, religion, origin, current and previous income status, ethnicity, and marital status.

In this chapter, we will explain the development of this project, including the design phase and implementation details. First of all, we will analyse related works that have been published. Secondly, we will present a global vision about the project data, identifying the different files and the most useful information. Thirdly, we will focus on how new features have been extracted from each dataset, explaining the algorithm used. Finally, we will present the analysis of the classification model.

3.2 Data

The Friends and Family experiment was designed to study how people make decisions, with emphasis on the social aspects involved, and also to see how we can empower people to make better decisions using personal and social tools. The subjects were members of a young-family residential living community adjacent to a major research university in North America. All members of the community were couples, and at least one of the members was affiliated with the university. The community was composed of over 400 residents, approximately half whom had children. Among the 130 adult members or 65 couples included in the experiment, 55 were added in the experiment during the first phase, since the spring of 2010, and the rest since the fall of 2010. Their behaviour and interactions were closed tracked with their personal Android-based mobile phones. Their behaviour, personality and interactions were also tracked with monthly, weekly and daily surveys.

We are now going to describe the data that was collected by the smartphones, analysing the structure and the quality of the ones that we are going to use in this project. These are the following ones:

- **Call log:** Every call that a user received, make or miss is recorded in this file. The logs has a lot of information that can be used and have the following columns:
  - ***ParticipantID.A:*** the participant ID associated with the data-collecting mobile phone
  - ***ParticipantID.B:*** the participant ID of the other end of the call, if the other end is associated with a known participant
  - ***Local-time:*** the time of the call
  - ***Type:*** the type of the call (incoming, outgoing or missed)
  - ***Duration:*** the duration of the call
  - ***Hash:*** the hashed phone number shown on the data-collecting phone

  The file contains 164905 logs from all the users, and 60518 of these calls were between people from the experiment. Something remarkable is the fact that there are only 59124 non-null values in the column duration. In the table 3.1, there are a few rows of the data set and we can see that outgoing and incoming calls have a null value in the duration column, not only in the missed, which would have more sense. As we will see later, it makes very difficult the extraction of a featured related with this value.

	participantID.A	**participantID.B**	**local_time**	**type**	**number.hash**
0	sp10-01-53	NaN	2010-07-14 14:56:29	incoming	d4498b
1	sp10-01-53	NaN	2010-07-14 14:56:26	incoming	e8b7ee
2	sp10-01-53	NaN	2010-07-14 14:56:13	incoming	d4498b

Table 3.1: Call Logs

- **SMS log:** Every SMS that a user sends or receives is recorded in this file. The logs have lots of information that can be used and have the following columns:

  - ***ParticipantID.A:*** the participant ID associated with the data-collecting mobile phone
  - ***ParticipantID.B:*** the participant ID of the other end of the short message, if the other end is associated with a known participant
  - ***Local-time:*** the time of the call
  - ***Type:*** the type of the sms (incoming or outgoing)
  - ***Hash:*** the hashed phone number shown on the data-collecting phone

  The SMS logs have a similar structure as the call logs. In this case 88655 SMSs were recorded and 33757 were between members of the experiment, almost half of them. It is remarkable that 5 participants did not send or receive any message.

- **Location:** every 30 minutes, the location of the user is recorded with a certain accuracy, which is include in the data set with the geographic coordinates.

  - ***ParticipantID:*** the participant ID associated with the data-collecting mobile phone
  - ***Date:*** the date-time when the location is recorded
  - ***Accuracy:*** the 68 per cent confidence offset of the estimation from true location in meters
  - ***x, y:*** the estimated location affine-transformed

  The file include almost 1 million logs with 96 per cent of non-null values.

- **Survey results:** The survey results in this data release include the following items:

	participantID.A	**date**	**accuracy**	**x**	**y**
0	sp10-01-53	2010-07-12 18:02:44	182	3813.0	3920.0
1	sp10-01-53	2010-07-14 06:02:19	1056	1744.0	2314.0
2	sp10-01-53	2010-07-16 00:01:49	28	-20.0	-70.0
3	sp10-01-53	2010-07-16 14:59:12	60	3876.0	3700.0
1	sp10-01-53	2010-07-14 06:50:19	106	174.0	214.0

Table 3.2: Location Logs

- **Monthly surveys:** It includes the initial surveys, conducted on March 2010 and November 2010, and continuing monthly surveys, conducted from April 2010 to April 2011.
- **Big-5 personality survey:** made on April 2010, includes 44 questions that had to be answered in a range from 1 to 5, depending on your grade of level of agreement of the sentence. With the results of this survey, we can calculate each factor of the personality. Following the instructions to recode these factors, we subtract the score for all reverse-scored items from 6. For example, if you gave yourself a 5, compute 6 minus 5 and your recoded score is 1. That is, a score of 1 becomes 5, 2 becomes 4, 3 remains 3, 4 becomes 2, and 5 becomes 1. Next, we create scale scores by averaging the following items for each Big Five Factor domain (where R indicates using the reverse-scored item).
  - **Extraversion**: 1, 6R 11, 16, 21R, 26, 31R, 36
  - **Agreeableness**: 2R, 7, 12R, 17, 22, 27R, 32, 37R, 42
  - **Conscientiousness**: 3, 8R, 13, 18R, 23R, 28, 33, 38, 43R
  - **Neuroticism**: 4, 9R, 14, 19, 24R, 29, 34R, 39
  - **Openness**: 5, 10, 15, 20, 25, 30, 35R, 40, 41R, 44
- **Weekly surveys:** we get information about the sleep quality and dynamic social network (dining, watch TV, providing babysitting service, etc)
- **Social network structure:** who and whom are couples inside the experiment and friendship surveys, conducted on 09/01/2010, 12/01/2010, 03/01/2011, 05/01/2011.
- **Funfit related surveys:** about daily activity stats, daily weight report, reward based on daily weight report.

These are the data sets that we are going to use in this project but it is also interesting how the Bluetooth was used in the experiment to know the time that a participant meet with another participant. The details are:

- **Bluetooth:** the devices scan every 5 minutes the devices that were near. Thanks to that, we can know if one participant was in the same place with another participant. The data set includes the following information:
  - ***ParticipantID:*** the participant ID associated with the scanning mobile phone
  - ***Date:*** date and time of the scan
  - ***ParticipantID.B:*** the Participant ID associated with the scanned device if such association is known
  - ***Address:*** the anonymized MAC address of the scanned device. This information would be very useful is everybody would have the Bluetooth turn on, because we could know with how many people a user meet, but this not the common case.

In this experiment we can also find data, which have been used in related works, about the apps installed in the phone, the apps that are being used, information of the accelerometer and battery status.

3.3 Related Work

All this data has been used to investigate questions like how things spread in the community, such as ideas, decisions, mood, or the seasonal flu. There are a lot of researches using this dataset with different points of view. We are going to analysed a few of them in order to get a better idea about what kind of conclusions can be obtained. We detect that the majority have some features in common, so the analysis is going to be struct following this pattern:

- Goal of the study
- Technology used
- Data taken from the “Family and Friends” data set.
- Conclusions

There are two big types of goals that we have identified in the works that we found using this data set. On one hand there are some of them that try to predict something from certain data using Machine Learning or other techniques. These are the ones that are more related to this project so where are going to inquire more on these ones. On the other hand, we found studies that try to analyze how people relate to others using in most cases network-level features obtained from the "Friends and Family" dataset.

As we do in our project, some try to predict the people's personality, using The Big Five personality traits, also known as the five-factor model (FFM). It is a model based on common language descriptors of personality (lexical hypothesis). This widely examined theory suggests five broad dimensions used by some psychologists to describe the human personality and psyche. The five factors have been defined as openness to experience (inventive/curious vs. consistent/cautious), conscientiousness (efficient/organized vs. easy-going/careless), extraversion (outgoing/energetic vs. solitary/reserved), agreeableness (friendly/compassionate vs. analytical/detached), and neuroticism (sensitive/nervous vs. secure/confident), often listed under the acronyms OCEAN or CANOE. The "Friends and Family" data set contains the result of surveys from which we can calculate these five factors. Next we review the main works related to this dataset.

- **Predicting Personality Using Novel Mobile Phone-Based Metrics:** The goal of the Yves-Alexandre de Montjoye research is that users' personalities can be predicted from basic information that can be obtained by any service provider. Specifically, they calculate five sets of psychology-informed metrics: Basic phone use, Active user behaviors, Mobility, Regularity, and Diversity. These can be easily extracted from standard phone logs to predict how extroverted, agreeable, conscientious, open to experience, and emotionally stable a user is.

  Machine learning is the base of this research. Due to the fact that relationship between personality traits and numerous behavioral and psychological factors can often be nonlinear, SVM with a 10-fold cross validation is the classifier that have been used, over the more traditional GLM, which is the basis of many machine-learning algorithms.

  The data taken from the "Family and Friends" data set is: Call Logs, SMS Logs, Location and the survey results of the Big Five.

  The research provides the first evidence that personality can be predicted from standard carriers' mobile phone logs. Using a set of novel indicators based on personality research, it is predicted whether users were low, average or high on each of the big five from 29% to 56% better than random.

- **Incremental Learning with Accuracy Prediction of Social and Individual Properties from Mobile-Phone Data:** The study examines the way learning and prediction process evolves in time, as the amount of data available to the learning algorithm increases. So the goal, rather than evaluate the specific models and how they generalize, is to investigate the learning process over time.

The modeled of this process of learning is done using the Gompertz curve, which is a sigmoid function. It is a type of mathematical model for a time series, where growth is slowest at the start and end of a time period. The right-hand or future value asymptote of the function is approached much more gradually by the curve than the left-hand or lower valued asymptote, in contrast to the simple logistic function in which both asymptotes are approached by the curve symmetrically. It is a special case of the generalised logistic function. There are a lot of examples where it is used, like modeling of growth of tumors[], modeling the population in a confined space[], etc.

For the implementation of machine learning algorithms, they used WEKA, which provides a graphical user interfaces for easy access to a collection of machine learning algorithms for data mining tasks. The algorithms can be applied directly to a dataset and contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

The data taken from the "Family and Friends" data set is: Call Logs, SMS Logs, Location, App Running Logs, Apps Installed, Alarm Logs and Internet Usage.

The findings of these research can be used as a method for advance prediction of the maximal learning accuracy using just the first few measurements, by extrapolating the learned Gompertz. As a result, we could know when it is time to stop collecting data as we have reached a state of saturation. Also, this study can be useful to compare different algorithms of learning processes to analysis which is the best depending on the data obtained.

- **How Many Makes a Crowd? On the Evolution of Learning as a Factor of Community Coverage:** This work is very similar to the previous one because it tries to analyze the evolution of the learning process of personal features and behavioral properties with a constant increase in sampling group size. It is concerned in use it as a benchmark for the learning process. As it happens with the work above, it is important to this project the conclusions extracted because it provides a clue for knowing if the amount of data that we have in this data set is enough for the learning process.

  It uses the Louvain Method for extracting communities from large networks, in this case, using the SMS Logs and their friends personal information. With this, they look at the community which the participant belongs to and predict that one attribute would be equal to the most common attribute in the community. Then, in order to model the evolution of this process, as before, they use the Gompertz function.

The data taken from the “Family and Friends” data set is: SMS Logs and Survey Friendship

The accuracy obtained applying the Gompertz function for modeling progress of behavior patterns prediction of mobile and social user is very high, taken into account the regression result in the study of the ethnicity, religion, children, origin and age.

- **Composite Social Network for Predicting Mobile Apps Installation:** The goal is to develop a simple computational model to predict the app that a user can install by using data collected by the phone and surveys(“Friends and Family data set”). It shows the importance of considering all these factors in predicting app installations, and demonstrates that app installation is indeed predictable.

  The data taken from the “Family and Friends” data set is: Call Logs and Survey Friendship

  After all the analysis, we can conclude from the study that there are strong network effects in app installation patterns even with big uncertainty in app installation behavior, with a prediction accuracy four times better than random guess in predicting future installations.

Conclusions

After reviewing all of them, we get the conclusion that there is only one previous work who tries to predict the personality using only data that a telecommunications carrier can collect. Most of them use other data that is not usually possible to get, like the information from the surveys or the proximity to other people using the Bluetooth Logs. The result of this work are resumed in the next table and in this study we will try to improve these result and also we will analyse if the classifier developed can be applied in other contexts like WhatsApp.

Factor	**Accuracy**	**Top feaures**
Neurotism	63%	Daily distance traveled
		Places entropy
		AR Coefficients
Extraversion	61%	Entropy of contacts (text
		Entropy of contacts (c&t)
		Variance of inter-event time (text)
Conscientiousness	51%	Variance of inter-event time (call)
		Text inter-time average
		Average inter-time (call)
Agreeableness	51%	Entropy of contacts (text)
		AR Coefficients
		Variance of inter-event time (text)
Openness	49%	Variance of inter-event time (text)
		Percent initied (text)
		Average inter-event time (call & text)

Table 3.3: Results obtained from Montjoye research

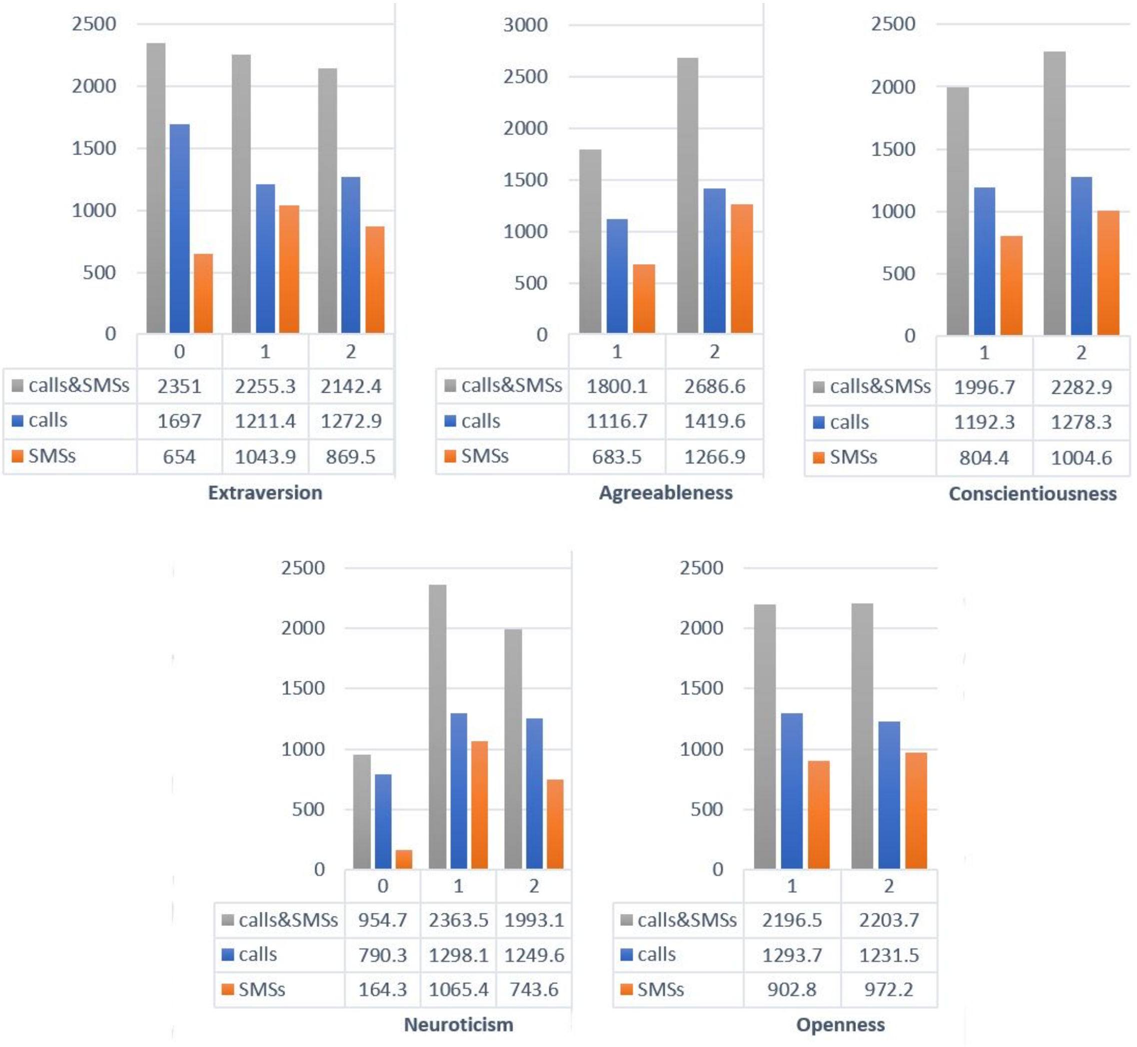
3.4 Feature extraction

In machine learning, we usually can not use the raw data to train the classifier, so we need to preprocess the data set in order to get a more valuable information. From the 3 data sets that we previously explain, we are going to extract as much features as we could and then they will be evaluated. With all of these, we will be able to train a classifier to predict each factor of the personality of the users.

3.4.1 Features from Calls and SMS logs

3.4.1.1 Number of interactions

The number of interactions that a user have in a period of time is probably the first featured that we can think in when we see the logs of calls and SMSs. In other words, we are going to get the number of calls and SMSs that each user does in this period of time. To get that, we group the DataFrame by the parameter “participantID.A” and count how many times each user appears.



Extraversion

	0	1	2
■ calls&SMSs	2351	2255.3	2142.4
🟦 calls	1697	1211.4	1272.9
🟧 SMSs	654	1043.9	869.5

Agreeableness

	1	2
■ calls&SMSs	1800.1	2686.6
🟦 calls	1116.7	1419.6
🟧 SMSs	683.5	1266.9

Conscientiousness

	1	2
■ calls&SMSs	1996.7	2282.9
🟦 calls	1192.3	1278.3
🟧 SMSs	804.4	1004.6

Neuroticism

	0	1	2
■ calls&SMSs	954.7	2363.5	1993.1
🟦 calls	790.3	1298.1	1249.6
🟧 SMSs	164.3	1065.4	743.6

Openness

	1	2
■ calls&SMSs	2196.5	2203.7
🟦 calls	1293.7	1231.5
🟧 SMSs	902.8	972.2

Figure 3.1: Number of interaction for each personality factor

We can observe in the figure 3.1 that there is a strong relation between the number of interactions and the personality in the Agreeableness and Neuroticism factors. A low score is directly related, in each of them, with the total number of interaction. In the first case, it is easier because there is only two targets, but in the second one we observe that the numbers are very similar between medium and high score. For the rest, there is no pattern that suggests us the existence of any relation.

3.4.1.2 Inter-event time

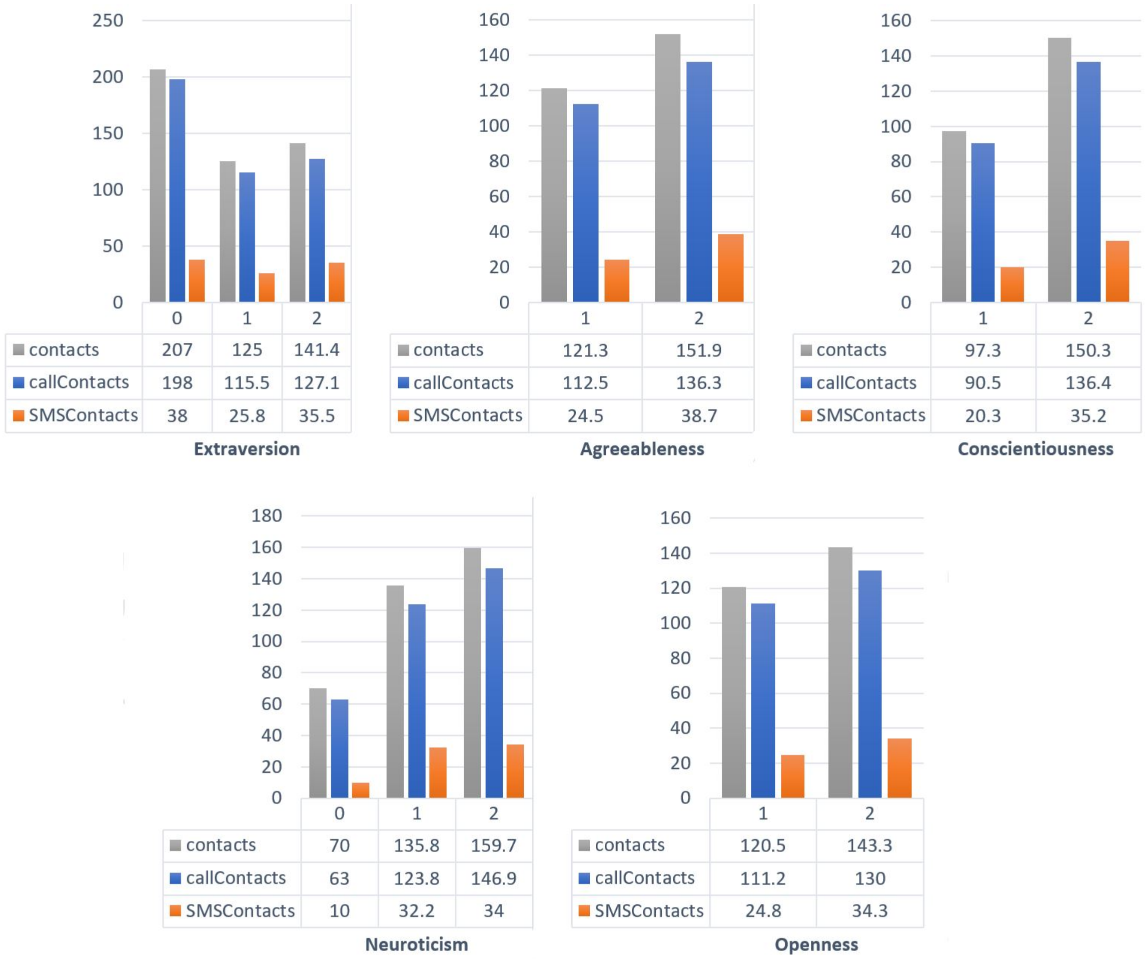
It is interesting to use the date of each interaction to know the mean and the variance of the time between each of them. To achieve that, we have some tools in Pandas to parse a column to a datetime format. Thanks to this tool we can now calculate the time difference in seconds between two dates. First of all, we have to sort the DataFrame by "participantID.A" and date. Then, we would like to take the date of an interaction and compare it with the previous one. For this, we could think about iterating the DataFrame but this is not very efficient. So, instead of that, using the pandas function "shift", which let you move up or down a column a number of positions, we copy the date column in and new column shifted one position and then we calculate the time difference between the two columns. We should also consider that both events have to belongs to the same user. Finally, grouping by participant id, we can get the mean and the variance for each user, both for calls and SMSs.

People with high values in the time variance between interaction can be related with changes in the personal mood. This is a indicator of people who score high in neuroticism.

3.4.1.3 Number of contacts

The number of interactions is a relevant feature, but we probably agree that it is not the same, someone who only talks with four people compared with someone who does it with one hundred. In this case, we are not going to take into account if the contacts are in the experiment or not, so we are going to use the hashed number from the data set. For getting that, we are going to group by the "participantID.A" and count the number of different "number.hash" that each user has.

Observing the figure 3.2, we can see that, for Agreeableness, Conscientiousness, Neuroticism and Openness, the value of the factor increase with the number of contacts. This fact occurs both with calls and SMSs. In the case of Extraversion, it exists the same relation but only with medium and high values. Low values has more contacts which is a little bit confusing.



	0	1	2
■ contacts	207	125	141.4
■ callContacts	198	115.5	127.1
■ SMSContacts	38	25.8	35.5

Extraversion

	1	2
■ contacts	121.3	151.9
■ callContacts	112.5	136.3
■ SMSContacts	24.5	38.7

Agreeableness

	1	2
■ contacts	97.3	150.3
■ callContacts	90.5	136.4
■ SMSContacts	20.3	35.2

Conscientiousness

	0	1	2
■ contacts	70	135.8	159.7
■ callContacts	63	123.8	146.9
■ SMSContacts	10	32.2	34

Neuroticism

	1	2
■ contacts	120.5	143.3
■ callContacts	111.2	130
■ SMSContacts	24.8	34.3

Openness

Figure 3.2: Number of contacts for each personality factor

3.4.1.4 Percent initiated (calls and SMSs)

We know that a messaging conversation is started by someone if an interaction is the first in a day with a certain contact. So we can calculate these in a similar way as it is done in the “inter-event time” feature, but comparing the day instead of calculating the time difference. Something remarkable is that we count the conversation initiated by the user (type equal to outgoing) and also the initiated by others (type equals to incoming). With all of these, we now can measure the percent of the conversations initiated by a participant, comparing it with the total of conversations. Even though this feature makes more sense for SMSs, we have also calculated it for the calls.

3.4.1.5 Contacts to interactions ratio

Having the number of contacts and interactions from each user, it is interesting to know the ratio between these parameters, both on calls and SMSs. Some user could talk with a few contacts, but a lot with each of them or vice versa.

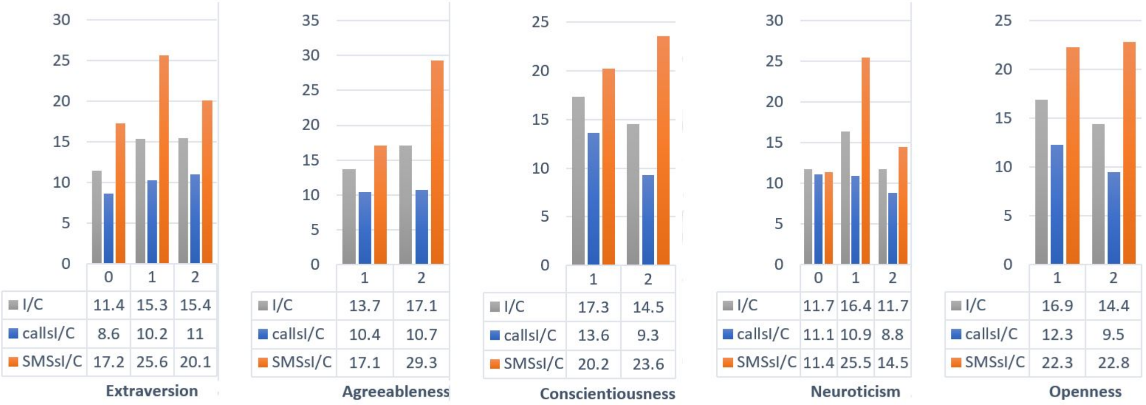


Figure 3.3: Interactions per contact for each personality factor

For this feature, it is difficult to find relations between the variables. We can see in the figure 3.3 that for the Extraversion and Agreeableness factor, the interactions per contact are directly proportional to the value of these factors. In Conscientiousness and Openness occurs the reverse.

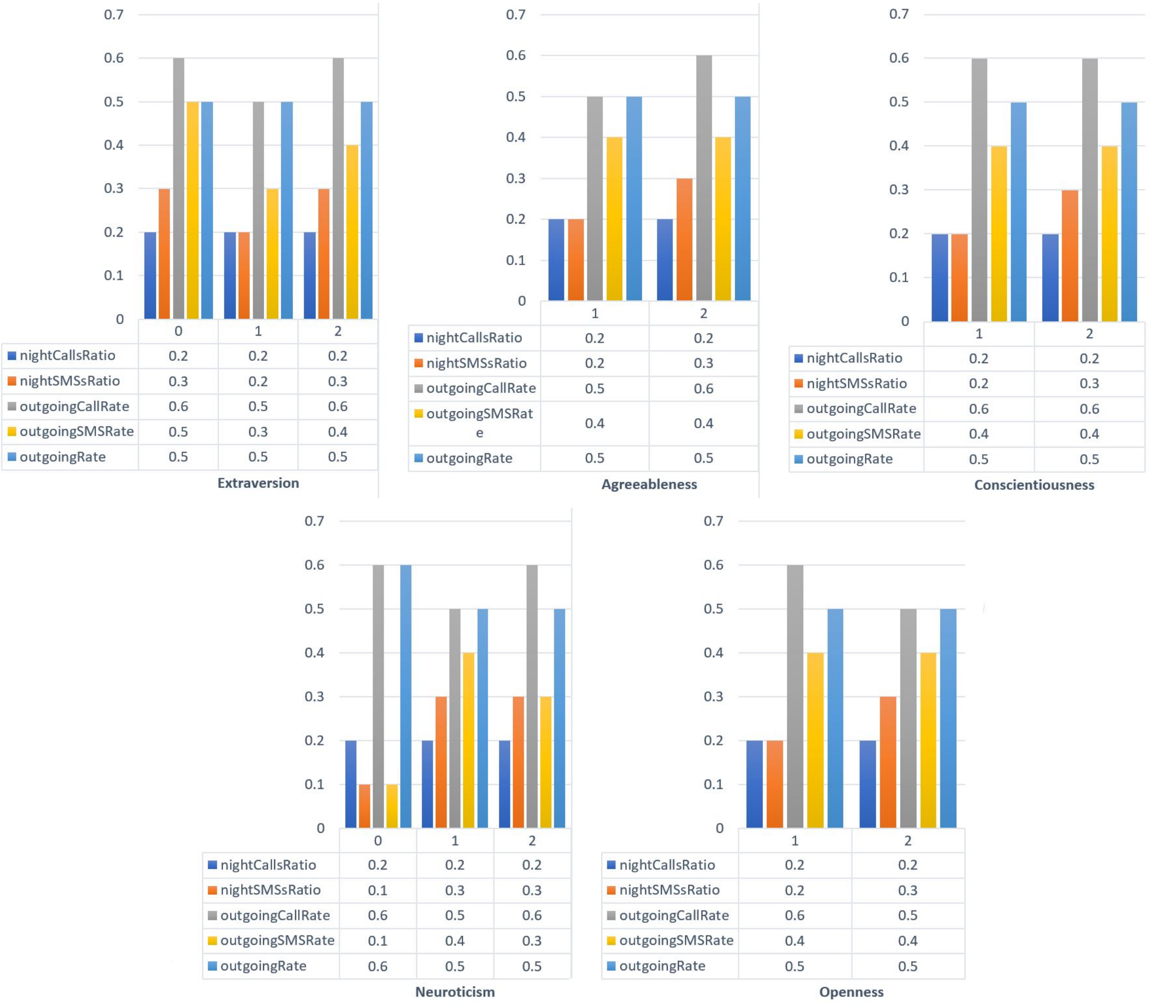
3.4.1.6 Missed rate

Being this feature related with the "outgoing rate" one, we will try to analyse the situation of some people who always have the phone in silence mode and never take you a call or people who see it but they just do not want to answer. These behaviours may give us some clues about the personality of a user. For this reason, we calculate, similar to the outgoing case, the number of missed calls and the ratio between this number and the total of calls. In this case, this feature can be only applied to calls because the concept missed SMS makes no sense.

3.4.1.7 Outgoing and night rate

One aspect that may describe your personality is if you are someone who waits until a friends calls you or if you are the one who always call them. Using the parameter "type", we extract the number of outgoing interactions, counting the times that the word “outgoing” appears in the column “type” and grouping by “participantID.A”.

Other relevant aspect is your night activity, so using the date we can know when a call or a SMS is happening at night or not.



	0	1	2
🟦 nightCallsRatio	0.2	0.2	0.2
🟧 nightSMSsRatio	0.3	0.2	0.3
⬜ outgoingCallRate	0.6	0.5	0.6
🟨 outgoingSMSRate	0.5	0.3	0.4
🟦 outgoingRate	0.5	0.5	0.5

Extraversion

	1	2
🟦 nightCallsRatio	0.2	0.2
🟧 nightSMSsRatio	0.2	0.3
⬜ outgoingCallRate	0.5	0.6
🟨 outgoingSMSRate	0.4	0.4
🟦 outgoingRate	0.5	0.5

Agreeableness

	1	2
🟦 nightCallsRatio	0.2	0.2
🟧 nightSMSsRatio	0.2	0.3
⬜ outgoingCallRate	0.6	0.6
🟨 outgoingSMSRate	0.4	0.4
🟦 outgoingRate	0.5	0.5

Conscientiousness

	0	1	2
🟦 nightCallsRatio	0.2	0.2	0.2
🟧 nightSMSsRatio	0.1	0.3	0.3
⬜ outgoingCallRate	0.6	0.5	0.6
🟨 outgoingSMSRate	0.1	0.4	0.3
🟦 outgoingRate	0.6	0.5	0.5

Neuroticism

	1	2
🟦 nightCallsRatio	0.2	0.2
🟧 nightSMSsRatio	0.2	0.3
⬜ outgoingCallRate	0.6	0.5
🟨 outgoingSMSRate	0.4	0.4
🟦 outgoingRate	0.5	0.5

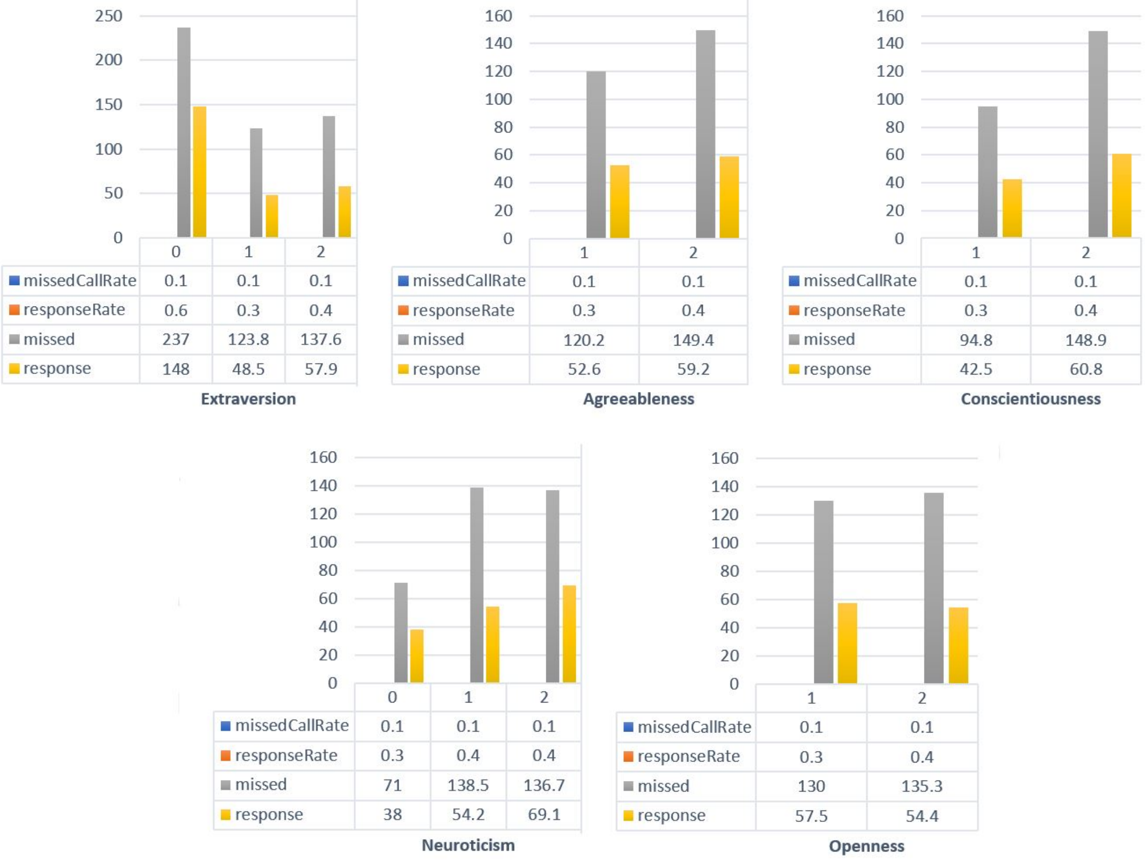
Openness

Figure 3.4: Outgoing and night rate for each personality factor

The unique relation that we can see analysing these graphics is that the rate of SMS sent at night is directly proportional to the Agreeableness, Conscientiousness and Openness. The problem is that this difference is not very high (0.2 to 0.3) so it would be necessary to see the variance, not only the mean.

3.4.1.8 Response rate

Knowing the amount of missed calls that a user received, it would be relevant to see how many of them have been responded. In order to calculate this feature, we assumed that an interaction is a response if the the first interaction after a missed call from a user A to a user B is an outgoing call from user B to user A. Considering this assumption, we sort our DataFrame by “participantID.A”, “number.hash” and date, and count how many times the conditions that we mentioned before are fully met. The ratio is calculated comparing the score with the number of missed calls of each user.



	0	1	2
■ missedCallRate	0.1	0.1	0.1
■ responseRate	0.6	0.3	0.4
■ missed	237	123.8	137.6
■ response	148	48.5	57.9

Extraversion

	1	2
■ missedCallRate	0.1	0.1
■ responseRate	0.3	0.4
■ missed	120.2	149.4
■ response	52.6	59.2

Agreeableness

	1	2
■ missedCallRate	0.1	0.1
■ responseRate	0.3	0.4
■ missed	94.8	148.9
■ response	42.5	60.8

Conscientiousness

	0	1	2
■ missedCallRate	0.1	0.1	0.1
■ responseRate	0.3	0.4	0.4
■ missed	71	138.5	136.7
■ response	38	54.2	69.1

Neuroticism

	1	2
■ missedCallRate	0.1	0.1
■ responseRate	0.3	0.4
■ missed	130	135.3
■ response	57.5	54.4

Openness

Figure 3.5: Missed and response rate for each personality factor

3.4.2 Feature from Location data: Radius of gyration

We are interested in knowing how far a user moves from the place where he usually is. A good way to do it is looking how many meters would be the radius of a circle that would include all the locations that he visited. This is a mathematical problem called “smallest-circle" that tries to look for the radius and the center of this circle that includes all the points. Known as Bomb Problem [5] in a military context, taking the points as objectives to be destroyed, it resolves the problem of dropping the bomb in the right position to hit all the targets using the least amount of explosive. There are a lot of studies in this area and, in spite of some algorithms suggest a complexity O(n4), we conclude from the literature that it can be resolved in linear time. The algorithm that we have chosen, of Raimund Seidel, is expected to run in 0(N) time. It is recursive and has two sets of points, initially one with all the points and the other empty. As it calls recursively, it will go including the boundary points of the circle until it includes all the boundary points of the circle. Having that function developed, we have to group by the parameter participant calling the smallest-circle function to each of them. We are going to use the figure 3.6 to explain the process of the algorithm used in 4 steps:

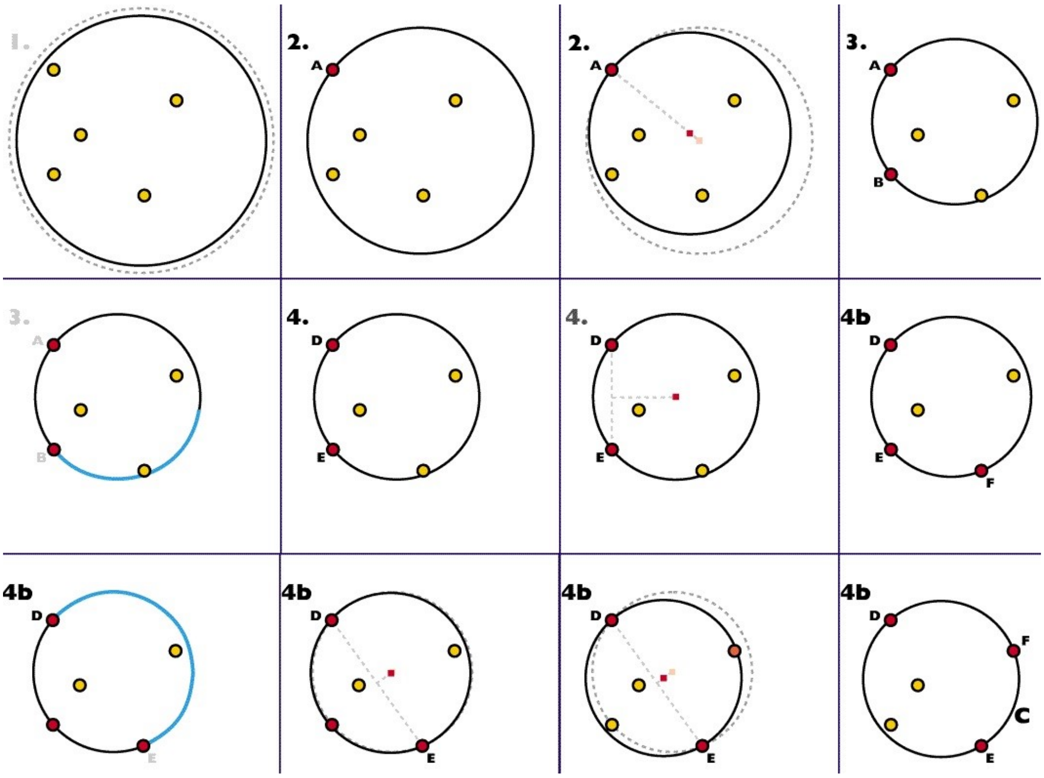


Figure 3.6: Algorithm process for the radius of gyration

1. We create a circle big enough to include all the points.
2. We reduce the radio until one point, which we will called A, touches the edge of the circle. This point will be the farthest from the center.
3. We make the circle smaller by moving its center towards points A until the circle

makes contacts with other point, which we will call B. Now we know that our circle passes by two or more points, but we can not assume that this is the smallest one.

4. We have two cases inside this step. If the distance between the point A and B is equal to the diameter, then we find the smallest one. Otherwise, we reduce the radius of the circle until we find another point on the edge, moving the center in the direction of the perpendicular bisector of the segment that joins the points A and B. Being that new point C, we need to test if the circle contains an interval of arc greater than half the circle's circumference on which no points lie. If there is no arc that verifies that condition, then we have finished. Otherwise, if the arc between A and C is the one that is bigger, we have to reduce the radius and to move the center in the direction of the perpendicular bisector of the segment that joins the points A and C until the distance between A and C is equal to the diameter or we find a new point on the edge that verifies the condition that there is no interval of arc greater than half the circle's circumference on which no points lie.

3.5 Classification model

The feature selection process has a problem which is that you can not know if the features that we are extracting are going to be relevant for the classification model. What we have done is to get features that, after analysing it, we conclude that have some relation with the personality. Now, using this data we are going to prove different classification model using some features. Another problem occurred when we are training the machine and it is called over-fitting. This happens when we use too many features and therefore, the model is too much adapted to the training data set. Additionally, selecting the correct classifier is essential so we are going to review and to try a few of them.

Inside the supervised learning algorithms, there are so many, so were are going so were are only going to explain the one that have been used in this project:

- **Support Vector Machines**: based on the concept of decision hyperplanes, it creates an optimal plane which categorizes new samples. Works right with linear classification but also it performs well with non-linear classification using high-dimensional feature spaces.

- **Decision Tree Classifier**: it creates a model that predicts the value of a target variable by learning some simple decision rules inferred from the data features. Decision trees are easy to understand and interpret, can be visualised and do not require techniques like normalisation, etc, as other algorithms do. It supports multi-output and the explanation for the condition is easily explained by boolean logic.

- **Random Forests and Extra Trees**: Decisions trees have some disadvantages that these try to solve. Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This problem is mitigated by using decision trees within an ensemble like Random Forests and Extra Trees do. It divides the training data set in groups and creates a decision tree for each group. Then, when you want to predict something, each tree predict the output and each prediction is like a vote. The class with more votes will be the one selected.

- **K-Nearest Neighbours**: It is a type of instance-based learning and is among the simplest machine learning algorithms. An object is classified by the class most common among its k nearest neighbours.

- **Gaussian Naive Bayes**: algorithm that applies Bayes theorem assuming that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a car may be considered a car if it has 4 wheels, it is less than 4 meters long and less than 2 meters high. A naive Bayes classifier considers each of these features to contribute independently to the probability that this vehicle is a car, regardless of any possible correlations between the number of wheel and the size features.

3.5.1 Classification algorithm selection

We have started seeking algorithms that look like they are going to work well with our data set, but we cannot know until the classifier is tested. We are going to split our data set in three: features extracted from calls logs, extracted from SMSs logs and a combination of both adding the location feature.

We are going to train with these three different data models the classification algorithms selection that we have previously mentioned and to show the results in the following table.

The model validation has been done using the cross-validation technique. This is used to estimate how accurately a predictive model is and to avoid problems like over-fitting. The data is divided in a certain number of groups. For example, if this number is ten, the cross-validation tool is going to train the algorithm with 9 groups and test the result with the remaining one and repeating these 9 times more with the rest of the groups, we get the mean and the variance of the model accuracy.

Scikit-learn has a cross-validation tool called K-fold, and to use it, we have to put as parameter the number of groups or folds that we want to get in our data set. Our dataset is not too big, so we are going to use a value of 5 instead of 10, that is the most common.

Each Five Factor is predicted individually, being all the classifiers multi label. The label can be 0, 1 or 2, depending if the personality factor is low, medium or high, respectively. All of them have been set using the default parameters, in other words, without tuning the classification model. The values in bold show the best score for each personality factor and the variance of the Calls and SMSs models have not been include because all of them are less than 0.1.

Factor	**Algorithm**	**Accuracy**		
		Call SMS Location	**Calls**	**SMSs**
Extraversion	Linear SVC	**0.630 (+/- 0.087)**	**0.675**	0.505
	Decision Tree	0.595 (+/- 0.060)	0.440	0.530
	Random Forest	0.555 (+/- 0.058)	0.555	0.470
	Extra Trees Classifier	0.475 (+/- 0.090)	0.610	**0.585**
	GaussianNB	0.515 (+/- 0.070)	0.490	0.525
	K Neighbors	0.365 (+/- 0.084)	0.620	0.525
Agreeableness	Linear SVC	0.525 (+/- 0.060)	**0.595**	0.385
	Decision Tree	0.540 (+/- 0.074)	0.510	**0.550**
	Random Forest	**0.580 (+/- 0.077)**	0.550	0.505
	Extra Trees Classifier	0.545 (+/- 0.075)	0.555	0.480
	GaussianNB	0.495 (+/- 0.048)	0.585	0.550
	K Neighbors	0.435 (+/- 0.064)	0.600	0.395
Conscientiousness	Linear SVC	**0.715 (+/- 0.071)**	0.665	**0.715**
	Decision Tree	0.650 (+/- 0.057)	0.670	0.605
	Random Forest	0.600 (+/- 0.068)	0.645	0.570
	Extra Trees Classifier	0.595 (+/- 0.091)	**0.700**	0.545

Factor	**Algorithm**	**Accuracy**		
		Call SMS Location	**Calls**	**SMSs**
	GaussianNB	0.595 (+/- 0.064)	0.555	0.590
	K Neighbors	0.645 (+/- 0.074)	0.640	0.660
Neuroticism	Linear SVC	**0.755 (+/- 0.053)**	**0.730**	**0.755**
	Decision Tree	0.570 (+/- 0.072)	0.545	0.560
	Random Forest	0.630 (+/- 0.094)	0.680	0.660
	Extra Trees Classifier	0.680 (+/- 0.076)	0.730	0.705
	GaussianNB	0.500 (+/- 0.076)	0.495	0.430
	K Neighbors	0.730 (+/- 0.070)	0.730	0.730
Openness	Linear SVC	0.570 (+/- 0.078)	0.570	0.565
	Decision Tree	0.555 (+/- 0.058)	0.600	**0.575**
	Random Forest	**0.645 (+/- 0.079)**	0.600	0.570
	Extra Trees Classifier	0.480 (+/- 0.081)	0.545	0.415
	GaussianNB	0.550 (+/- 0.067)	0.580	0.590
	K Neighbors	0.570 (+/- 0.069)	**0.645**	0.530

Table 3.4: Accuracy for algorithms without tuning

Analysing the result of the table 3.4 we can get some conclusions:

- The scores are between 80% and 130% better than random, which is a good value taking account that we have not modified any parameter.
- The classifiers that obtained better scores are SVC and the group of Decision Trees
- In general, SMSs features performs a little bit worse, less than 10%, but this may change after optimizing the parameters of the classifier.
- The model train with only call features works better predicting the Neurotism and

Extraversion. This may suggest us that adding some features makes the classifier worse.

3.5.2 Feature selection

With the previous values, it would be interesting to know which features are the most important. This is very useful because we can study in which moment, adding a feature to the classifier does not make it better.

We are going to start using the Random Forest for feature selection in each type of model.

3.5.2.1 Calls features

Each Five Factor, as we can see in the table 3.5, has different features that are important for the classification model.

- Starting form the extraversion factor, outgoing call rate and the number of nigh calls are the ones that have more weight. Probably, someone who make more calls than he received is going to be extrovert. Also the night calls may be a good indicator of the social life of someone, more important than the total of calls because some people makes a lot of calls al work and this is not directly related with the personality.
- Moreover, agreeableness factor has missed calls and response rate as the main 2 features. These two are very related in between and also with the personality trait, because agreeableness looks how compassionate and cooperative a person is, rather than suspicious and antagonistic towards others. If people do not respond any missed call, it can be seen as suspicious.
- Observing the column of conscientiousness trait, the number of contacts and the night calls are the main features. This may suggest if you are organized or if you are careless, but it is usually a difficult factor to predict considering that we do not know the content of the conversation.
- Neuroticism is the factor that we better predict with an accuracy of 76% and the most relevant features are the number of contacts and the rate between the outgoing calls and incoming calls. It make sense that these two are important because if we talk with a few people and do not call them a lot, may be an indicator of depression.

	E	A	C	N	O
callContacts	0.046	0.031	0.151	0.114	0.092
calls	0.025	0.085	0.052	0.099	0.064
callsI/C	0.109	0.083	0.061	0.026	0.09
iniciateCall	0.12	0.075	0.064	0.091	0.08
meanDiffCalls	0.025	0.079	0.095	0.05	0.103
missed	0.039	0.102	0.034	0.046	0.084
missedCallRate	0.046	0.031	0.092	0.048	0.09
nightCalls	0.135	0.069	0.124	0.059	0.045
nightCallsRatio	0.054	0.104	0.035	0.064	0.024
outgoingCallRate	0.159	0.027	0.074	0.102	0.025
outgoingCalls	0.049	0.052	0.068	0.076	0.072
response	0.051	0.075	0.009	0.069	0.046
responseRate	0.073	0.119	0.09	0.067	0.152
varDiffCalls	0.069	0.067	0.052	0.089	0.034

Table 3.5: Call feature importances

- Finally, the openness factor we can highlight the response rate of the missed calls, which was also a relevant feature for agreeableness. This factor shows if you are more curious or more cautious. Predict it just looking the calls may look impossible but probably, curious people are always going to answer the missed calls, and on the other hand, a cautious person would not do it so frequently.

3.5.2.2 SMS features

Looking at the table 3.6 and comparing it with the previous one, it is remarkable that the number of contacts is not relevant in any feature, as it was in the other. We are going to analyse which are important now and the main differences.

	E	**A**	**C**	**N**	**O**
SMSContacts	0.067	0.015	0.09	0.061	0.061
SMSs	0.115	0.088	0.09	0.088	0.072
SMSsI/C	0.125	0.041	0.14	0.028	0.078
iniciateSMS	0.168	0.115	0.075	0.144	0.082
meanDiffSMSs	0.112	0.158	0.12	0.062	0.18
nightSMSs	0.048	0.227	0.115	0.049	0.045
nightSMSsRatio	0.058	0.088	0.104	0.151	0.115
outgoingSMSRate	0.059	0.122	0.117	0.131	0.244
outgoingSMSs	0.095	0.102	0.057	0.133	0.049
varDiffSMSs	0.152	0.044	0.093	0.153	0.074

Table 3.6: SMS features importance

- The number of SMSs per contact, the time variance between each event and the percentage of initiated conversations are the most relevant features in the extraversion factor. The iniciated conversation feature is a clear indicator of sociability and the tendency to seek stimulation in the relation with other people. As the outgoing call rate was the most important in the call features, in this case, the number of initiated conversations is the one that we would have thought in the first place.

- Analysing the agreeableness factor, the conversation initiate are at the top of the list. This feature has a lot in common with the number of calls responded after a missed call, we could say that it shows if, on one hand, a person is friendly or if, on the hand, that person is more detached. For this personality trait, the number of SMS sent at night and the time mean difference between each message is important.

- Furthermore, to predict whether someone is too much organized or too much careless, the mean number of SMSs for each contact and ratio of night calls are the ones that have obtained a higher weight. These are almost the same as it was in the previous case so we are not going to explain it again.

- Neuroticism is mostly predicted using the number of messages sent, the time variance between each of them and the percentage of these sent at night. The fact that the time variance is present in the top is remarkable. It could suggest that people who change a lot their habits are more unstable psychologically and that the ones which the time difference is always around the mean are more confident.

- Finally, the openness classification model has the number of SMSs sent and the number of them that are at nigh as the most relevant features. People who like to seek intense and euphoric experiences are more likely to talk at night and to send a lot of messages.

3.5.2.3 Call, SMS and Location features

In the first part of this section we have seen that the result obtained combining all the features was even better that the rest separately. So, it makes sense to also analyse how relevant each feature is in this data set. It is normal to think that the features that were important in the previous cases, will be now too, but this does not always happen. For example, the number of interaction in mean with other contacts appears as one of the one with more weight in each personality factor, but this feature was only relevant for predicting the extraversion factor using SMS logs. We are going now to see in more detail for each classifier, observing the table 3.7:

- Extraversion: The feature ”initiated calls” is the most relevant which is very surprising considering the fact that this was a featured with not too much sense. As we explain in section of the feature extraction, this parameter comes from the idea of conversation initiated through SMSs. So it suggests us that there is a relation between the sociability and the number of times that the participant is the first to call other person each day. It is the same concept as the one present in the SMS model and very related with outgoing call rate. We can conclude that, instead of using these two features, the classifier has found a bigger relation between this and the output.

- Agreeableness: In this case, the number of night calls has a high weight compared with the other. Looking the results of the previous models, the fact that this feature is more important in this model than in the one with only information about calls is conspicuous. But it demonstrates that the activity during the night has direct influence in this personality trait.

- Conscientiousness: On one hand, Night SMS ratio, as it was in SMS model, is one of the main features for the classifier. On the other hand, the number of contacts has a high weight, which was expected considering that was also important in the Call model. In this classifier, all of the features have a low value so it is more difficult to know which one stands over the rest.

- Neuroticism: For this personality factor, the response rate of calls feature obtains a high value, which is normal considering that it was also important in the call model. Moreover, the time variance between interactions has a value of 0.007, which is very low bearing in mind that was very relevant in the SMS model.
- Openness: The feature outgoing call rate is most relevant while in the call model had a lower value. It is very related with the outgoing SMS rate that we got in the SMS model but it looks that there is a stronger relationship between the call than with the messages.

In conclusion, as we said at the beginning, the ratio between the number of contacts and the number of interactions is very important in each category. Furthermore, from the location information we extract the radius of the circle that include all the places where a user has been. This feature has not got too much weight, except in the extraversion classifier which is the second most relevant.

	E	**A**	**C**	**N**	**O**
I/C	0.025	0.101	0.045	0.078	0.078
SMSContacts	0.026	0.068	0.0	0.056	0.018
SMSs	0.015	0.031	0.018	0.019	0.045
SMSsI/C	0.051	0.017	0.046	0.05	0.016
callContacts	0.02	0.006	0.035	0.0	0.026
calls	0.007	0.0	0.034	0.0	0.002
calls&SMSs	0.0	0.034	0.02	0.036	0.034
callsI/C	0.011	0.0	0.01	0.036	0.069
contacts	0.0	0.036	0.097	0.006	0.0
iniciate	0.05	0.022	0.0	0.022	0.07
iniciateCall	0.22	0.017	0.056	0.011	0.016
iniciateSMS	0.064	0.06	0.047	0.043	0.017

	E	**A**	**C**	**N**	**O**
meanDiffCalls	0.043	0.008	0.0	0.004	0.022
meanDiffSMSs	0.001	0.04	0.034	0.024	0.005
missed	0.014	0.023	0.048	0.074	0.037
missedCallRate	0.062	0.0	0.011	0.009	0.018
nightCalls	0.0	0.043	0.033	0.049	0.0
nightCallsRatio	0.034	0.1	0.018	0.005	0.027
nightSMSs	0.038	0.071	0.052	0.01	0.034
nightSMSsRatio	0.044	0.028	0.071	0.05	0.028
outgoing	0.034	0.015	0.037	0.045	0.053
outgoingCallRate	0.026	0.007	0.021	0.028	0.083
outgoingCalls	0.037	0.012	0.045	0.066	0.077
outgoingRate	0.035	0.086	0.034	0.027	0.0
outgoingSMSRate	0.0	0.0	0.013	0.058	0.027
outgoingSMSs	0.014	0.023	0.042	0.036	0.019
radius	0.095	0.022	0.038	0.02	0.048
response	0.0	0.082	0.016	0.006	0.019
responseRate	0.008	0.009	0.033	0.098	0.02
varDiffCalls	0.009	0.013	0.028	0.025	0.047
varDiffSMSs	0.017	0.029	0.018	0.007	0.048

Table 3.7: Features importance

3.5.3 Classifier parameter tuning

As we mentioned before, all these results have been obtained using the default parameters for each Scikit-learn classification algorithm. The best scores were got using SVM or Decisions trees so we are going to look into these two and try to optimize the parameters in order to get the best classifier. As each algorithm has different parameters, we are going to analyse them separately, getting all the information from the official website of Scikit-learn . This task would be very difficult without the help of the Grid Search tool that Scikit-learn provides us. It give you the possibility of having a grid of parameters that you want to try, being able to pass more than one parameter. This is specially valuable because this tool returns the combination of several parameters that gets the best accuracy for the prediction, using cross validation with k equal to 3.

3.5.3.1 SVM

The Support Vector Machines has tree parameters for tuning the algorithm:

- C: With this, you indicate to the classifier how much you want to avoid misclassifying. If the parameter has a high value, the algorithm will try to find the smallest margin hyperplane that classify better all the training data set. In the other case, if the value is low, it chooses a hyperplane that separate the values but without worrying about some errors in classification for the training data set. We have a small data set so, we will probably get better results using a low value for the parameter C. In the figure 3.7, it is visible that the election in very difficult without any tool because it depends a lot in the data.

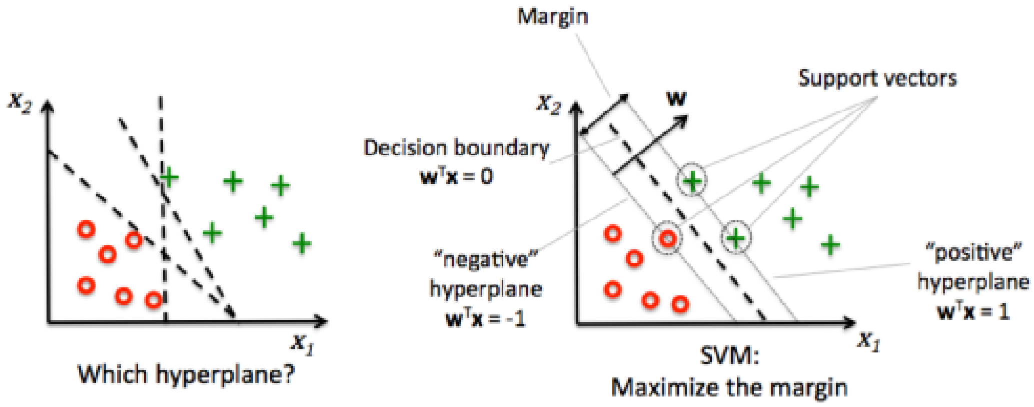


Figure 3.7: SVM 'C' parameter

- Loss: It is a function that penalises *misclassifications* in the samples and in the model parameters. We are not going to tune this parameter because our data set is small and penalising we would get an over-fitting function.
- Omega: As the previous one but only penalising the model parameters.

Using Grid Search, we use 3 type of kernels: linear, polynomial and exponential(rbf); a set of values for the parameter C from 0.001 to 10, with steps of 0.01. The results are shown in the table 3.8.

	kernel	**C**	**Accuracy**
Extraversion	linear	1.4	0.65
Agreeableness	linear	0.01	0.61
Conscientiousness	linear	0.01	0.74
Neuroticism	linear	0.01	0.81
Openness	rbf	0.01	0.61

Table 3.8: Accuracy SVM with parameter tuning

The values of the parameter C are very low, which means that the data is easily divisible and we can have a hyperplane with a big margin. The only kernel that is not linear is the openness estimator but the improvement is not too high, from 0.57 to 0.61. All the results are better than with the default parameters, with an accuracy between 3% and 5% better than before.

3.5.3.2 Random Forest and Extra Trees

Parameters in Random Forest and in Extra Trees classifiers have two goals: Increase the accuracy of the prediction or to make it easier to train the model. We are going to put our effort in the first goal and there are 3 parameters which can be tuned to improve the predictive power of the model:

- Maximum number of features: Random Forest and Extra Trees models create several trees and these parameters set the maximum number of features that can be tried in each individual tree. In general, if we increase the number of features, each node will have more options to consider, but it does not always happen. There are a lot of options available and we are going to use only a few of them:

  - Auto/None : This will simply take all the features in each tree without any restrictions
  - Square root : This option will take square root of the total number of features in individual run
  - Percentage : This option allows the random forest to a certain percentage of variables in individual run, assigning this in a format “0.x” where we want x% of features to be considered.

- Number of estimator: it sets the number of trees that are going to be created. If we increase the number of trees, the result are going to be better until we get to a number of trees which if we add new trees, the accuracy keeps constant.
- Minimum sample leaf: These parameters sets the minimum number of nodes that a sample has to pass to be the end of a branch, or in other word, to be a leaf.

In this case, we are going to apply Grid Search for both classifiers, with the same tuning parameter: a maximum number of features of 25%, 50%, 75% or 100%; a number of estimators between 1 and 300. The result are show in the table 3.9.

	Random Forest			**Extra Trees**		
	n	**m**	**Acc**	**n**	**m**	**Acc**
Extraversion	6	0.25	0.71	7	0.25	0.71
Agreeableness	5	0.5	0.71	16	0.5	0.77
Conscientiousness	17	1	0.77	20	0.75	0.81
Neuroticism	4	1	0.84	44	1	0.81
Openness	3	1	0.65	4	0.25	0.65

Table 3.9: Accuracy Random Forest and Extra Trees with parameter tuning

The improvements in this case are much bigger compared with the ones seen in SVM. On one hand, the Extra Trees classifiers gets results between 15% and 60% better than with default parameters. On the other hand, Random Forest achieves an accuracy between 18% and 37% better. The scores are better in each personality factor using Random Forest, except for agreeableness factor, which gets a better result using Extra Trees. As we can see, the differences between these two are very low, so the similarity was the expected. When the Extra Trees algorithm chooses features at a split, samples are drawn from the entire training set instead of a bootstrap sample of the training set like Random Forest does and being these splits chosen completely at random at each split.

The maximum number of features are very different between them and the number of trees is always bigger in the Extra Trees model. That means that the time process for the prediction is going to larger that the time spend with the Random Forest model.

3.5.4 Conclusions

A good choice in the selection of the classifier is essential in the process of Machine Learning. As we have seen, there is a lot of algorithms and studying each of them and finding the one that best fit with your data is a difficult task. At the beginning, the SVM algorithm got, in general, the best results, but, after the optimization process, Random Forest and Extra Trees improved much more and overtake the SVM accuracy.

Extraversion	**Agreeableness**	**Conscientiousness**	**Neuroticism**	**Openness**
71%	77%	81%	84%	65%

Table 3.10: Final results

Looking at the results in the table 3.10, we can say that using these models, we can predict the each Five Factor of the personality with an accuracy of 74.84% in mean. Comparing this with previous works, the accuracy obtained, using the same data sets, were 61%, 51%, 51%, 63% and 49% for Extraversion, Agreeableness, Conscientiousness, Neuroticism and Openness respectively. Our result are in mean 36% better and comparing with random are 127% higher.

The improvement of the results from previous works may be related with the deep process of tuning parameters for each algorithm. Using the Grid Search tool, we stablish big sets of values to be tested and, using a lot of computational time, the best parameters were obtained.

CHAPTER 4

Personality Prediction using WhatsApp

4.1 Introduction

WhatsApp is an instant messaging application with more than 1 billion users in over 180 countries. Founded in 2009, grew very fast achieving 600 million active users in 2014, the same year that Facebook announced it was acquiring WhatsApp for US$19 billion, its largest acquisition to date. This purchase had to have more interests than money, because the WhatsApp service is totally free: there is not advertisement and you can create your account for free. The added interest is the information about people that WhatsApp collects, but people think that they can not get too much information because the communications are encrypted.

Doing this project, we see that the people personality can be predicted analysing how they communicate with each others, and without knowing the content of the messages or calls. Therefore, the fact that messages are encrypted is not a problem for Facebook, because they know the origin, destination and date of all the messages. This is exactly the input data of the models that we previously trained, so they would be able to predict the personality using the same method.

Nowadays, in Spain and in many other countries, the SMSs are not very used because they have been replaced by more sophisticate internet application like WhatsApp. For this reason, getting a good prediction using that data would be much more interesting than using SMS logs.

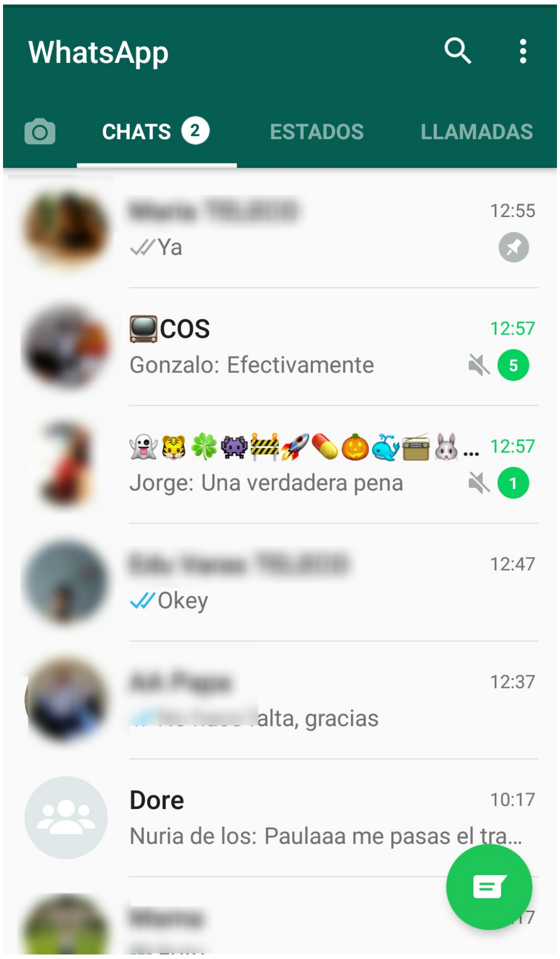
In this section we are going to see if it is possible to apply a model trained with SMS logs of the "Friends and Family" data set and then to obtain good results for WhatsApp information as the input.

4.2 Data and Feature Extraction

For building this part of the project, it is necessary to get the WhatsApp conversations of people and also to know the personality of them.

On one hand, getting the information from the application is harder than it was in the past. The encryption feature was added a few months ago, and before that, you could export all the conversations in one single csv file. Now, this option does not exist but we have found another way to get it. If we open a conversation, there is an option button where there is an action called "Send conversation by email" which generates a text file with all the messages of this conversation. You have to do it one at a time so it is a slow task,but unless, we obtain what we need.

On the other hand, in order to analyse the accuracy of the prediction, we need to know the personality of the people who have sent their conversation. To do that, we send them the same survey that was used to predict the Big Five Factors in the "Family and Friends" experiment.



[redacted] 12:55
✓✓Ya

📺COS 12:57
Gonzalo: Efectivamente 5

👻🐯🍀👾🚧🚀💊🎃🐳📻🐰... 12:57
Jorge: Una verdadera pena 1

[redacted] 12:47
✓✓Okey

[redacted] 12:37
[redacted]alta, gracias

Dore 10:17
Nuria de los: Paulaaa me pasas el tra...

[redacted]

Figure 4.1: WhatsApp conversations

The survey has been added in the Annexed A. It includes 44 questions that had to be answered in a range from 1 to 5, depending on your grade of level of agreement of the sentence. With the results of these survey, we can calculate each factor of the personality, which has several questions associated. The number of these answers and the factors linked are shown in the following list:

- **Extraversion**: 1, 6R 11, 16, 21R, 26, 31R, 36
- **Agreeableness**: 2R, 7, 12R, 17, 22, 27R, 32, 37R, 42
- **Conscientiousness**: 3, 8R, 13, 18R, 23R, 28, 33, 38, 43R
- **Neuroticism**: 4, 9R, 14, 19, 24R, 29, 34R, 39
- **Openness**: 5, 10, 15, 20, 25, 30, 35R, 40, 41R, 44

There are some questions whose answer is inversely proportional to the factor, so we subtract the score for all these reverse-scored items from 6. Finally, we create scale scores by averaging the following items for each Big Five Factor domain.

4.2.1 Preprocessing of the data

The file generated by the application is not a csv file so we need to adapt it to the format of the files that we have used in the previous section.

The format of the lines in these file are the following:

- Date, Time - Name of the sender: Message content

  For example: 27/11/16, 16:44 - Carlos Alonso: Lorem ipsum dolor sit amet, consectetur adipiscing elit

And what we need is to transform it into this format:

- participantID.A, participantID.B, date and time, type (outgoing or incoming), hashed number

To complete this task, we iterate the list of files of each person has sent us. Using python tools to read files and manage strings, we read line by line each file. Over each line, we apply pattern recognition to extract in several variables the useful information of each message. Then, this information is added to a list of arrays and when all the lines have been read, the list is added in a DataFrame. Doing it for all the conversations, we can build a Dataframe with all the messages and with the format that we need. To extract the variables, we have had the following considerations:

- The column participantID.B is going to be empty because this field was used to put the id of a participant if the number belongs to someone inside the experiment. We did not use this parameter to extract any feature so it is not relevant
- The participantID.A is going to be the name of the person who have sent us the list of conversations.
- The hashed number is going to the name found in each file not equal to the name of the person who sent us the list of conversations.
- To know if a message is outgoing or incoming, we have to compare the name field in the message with the name of the person who sent us the conversation: if it is equal, the message is outgoing, otherwise, it is incoming.

Extracting these parameters we found some troubles so we are going to explain them and also the way of fixing them.

- In the date the year appears with only 2 characters, so 2017 is represented as 17. The problem is that the library of pandas does not recognized this pattern when we try to convert to date-time format. For this reason, we need to add the number 20 before the last two characters of the date string.
- If the content of a message has more than one paragraph, the next paragraph appears as a new line in the file, so we can not consider every line as a message. To avoid this problem, what we do is to test if the first characters have the format of a date (xx/xx/xx or x/x/xx or x/xx/xx or xx/x/xx, being x a digit).
- When the encryption feature was added to WhatsApp, in every conversation there is a message alerting the users that the communications are encrypted. Also, when the user has a new smartphone with the same number, a new message appears warming the user that the secure code has change. Therefore, we can not assume again that each line is a message so we test if the line contains that alert sentences before extracting the information

Considering all of these situations, we create a python script that generates a Dataframe and we need to run this script with each list of conversation that a user send us.

When we have the complete dataframe with the data of all the users, we have to extract the features. We consider the messages as SMSs so we are going to apply the same methods that we use in the previous chapter to get the features. These are going to be the followings:

- Number of messages
- Mean time between messages
- Variance time between messages
- Number of contacts
- Messages per contact
- Outgoing messages rate
- Conversations initiated

We ask people close, like family and friends, to participate in this experiment and we got 5 participants. It is difficult to get people because the text file contains the conversation in plain text, so people are usually scare of sharing personal information. Also, the files generated are sent by email, which is not a secure method of communication considering that it is not encrypted.

4.3 Classification model

When we test the different classification algorithms using only SMS features, we obtained the best result using SVM, Random Forest and Extra Trees. We are going to test the accuracy of these models with the WhatsApp data and then we are going to tune the parameters to see if there is any improvement in the result.

4.3.1 Classifier selection without tuning

We have the data from 5 participants and after the feature extraction process, we have proceed to test the accuracy of the 3 algorithms with the default parameters. The results are shown in the table 4.1.

As we can see, the best accuracy for each personality factor is achieved using the Extra Trees Classifiers. The worst prediction is the Conscientiousness one, whose score is 0.2, very distant from the score of 0.7 got using SMSs. It is remarkable the good performance of

	SVM	Random Forest	Extra Trees
Extraversion	0.6	0.6	0.6
Agreeableness	0.6	0.6	0.8
Conscientiousness	0.2	0.2	0.2
Neuroticism	0.6	0.8	0.6
Openness	0.4	0.4	0.8

Table 4.1: WhatsApp accuracy results

the Extraversion and Openness model, with an accuracy of 80%, considering that the score using SMSs was 58%.

4.3.2 Classifier parameter tuning

Now, in order to see if we can get better results by improving the optimization of the classifier, we are going to tune some parameters, get a better accuracy and finally, see if these improvements have a positive impact for the prediction using WhatsApp data.

	SVC			Random Forest			Extra Trees		
	kernel	**C**	**Acc**	**n**	**m**	**Acc**	**n**	**m**	**Acc**
Extraversion	linear	1.4	0.68	36	1	0.74	40	1	0.84
Agreeableness	linear	0.6	0.68	9	1	0.74	8	0.5	0.71
Conscientiousness	linear	0.01	0.74	1	0.25	0.77	3	1	0.84
Neuroticism	linear	0.01	0.81	37	0.75	0.81	13	1	0.81
Openness	rbf	1.5	0.74	9	1	0.65	3	0.75	0.74

Table 4.2: SMS results with parameter tuning

Using SVM, we set a grid search with the following parameters: gamma (1,2,3), C (from 0.01 to 10 by steps of 0.01) and kernel (linear, polynomial, exponential). For all the five classifiers, the best value of gamma was 3. The rest of parameters and the new accuracy obtained are describe in the table.

Using Random Forest and Extra Trees, we set a grid search with the following parameters: the number of estimators (from 1 to 60) and the maximum number of features (25%, 50%, 75% and 100%).

The result shown in the table 4.2 are better than before, especially the Extra Trees classifier which outperforms the other two models. The next step is seeing if the prediction using the WhatsApp data improves or not. For this task, we are going to use the Extra Trees classifier with the parameters calculated using Grid Search.

	E	**A**	**C**	**N**	**O**
Extra Tress Tuned	0.4	0.8	0	0.6	0.6

Table 4.3: My caption

The results, as we can see in the figure 4.3, are worse than before in 3 factors (Extraversion, Conscientiousness and Openness) and equal in the rest (Agreeableness and Neuroticsm). After the optimization, the results were better but not if we applied the WhatsApp data as the input. When we tune the parameters, maybe, we are adapting the function too much to the SMS data, which produces worse scores with the new information because the way that WhatsApp is used is different from the SMS usage.

4.3.3 Final results and conclusions

The best result have been obtained on one hand, using the Extra Trees Classifier with the default parameters for Extraversion, Agreeableness, Conscientiousness and Openness factors; and on the other hand, the Random Forest Classifier for Neuroticism factor.

Extraversion	**Agreeableness**	**Conscientiousness**	**Neuroticism**	**Openness**
60%	80%	20%	60%	80%

Table 4.4: WhatsApp final results

There scores are very good, with an accuracy between **82%** and **142%** better than random, with the exception of the Conscientiousness model which is worse than random. Furthermore, the accuracy of the model globally is almost **11 times better than random**, which remarkable considering that there are 243 different combinations of personality.

It is also interesting to analyse each participant separately and to see if there is any difference between them. In the table, the values correctly predicted are shown as a single number, and in the case that the prediction is wrong, it indicates the expected value and the value predicted by the model. The values are the ones predicted by the Extra Trees classifier and, in the case of Neuroticism, from the Random Forest algorithm. The letter from A to E represents each participant of the experiment.

	A	**B**	**C**	**D**	**E**
Extraversion	2 - 1	2	2-1	2	1
Agreeableness	2	2	1 - 2	1	2
Conscientiousness	1 - 2	1-2	2	1-2	1-2
Neuroticism	1	1	2-1	2	1
Openness	2	1	2	1	1 - 2
ACCURACY	**60%**	**80%**	**40%**	**80%**	**60%**

Table 4.5: Prediction for each participant using WhatsApp data

The results shown in the table 4.5 obtained are not too disparate between users and all of them are better than random. The mean is **64%**, and the lowest value is 40%. These are good values but we have to consider a few problems that affect the accuracy of the results. On one hand, we do not have enough samples to test the quality of the experiment. Moreover, the process of data collected is not as accurate as it was in the ”Friends and Family” data set. For example, people sometimes delete some conversation, or they have bought a new phone a few months ago and we only have the messages from that moment. Also, we have only selected the messages from this year, which is a time period smaller compared to the other experiment. On the other hand, we have to consider that the way people communicated through SMSs is not the same as how they do it using instant messages applications like WhatsApp. People send now many more messages than they used to do in the past, and also, the onset of groups is very relevant, because affects straight to the number of messages that people exchange individually with others.

In conclusion, considering all these facts, we have to be prudent and say that these results are not yet certain because the number of participants and the diversity of them, as age or culture, are very low. However, this was the first contact relating the use of WhatsApp with the personality and the beginning has been even better than expected.

CHAPTER 5

Conclusions and future work

The main goal of this projecte was to build a machine learning algorithm able to predict the personality using basic phone data, and the results obtained have been indeed surprising on the two cases studied. One one hand, using Calls, SMSs, Location data, the accuracy for the personality factor predictions are in mean 75%, which is 36% better than the accuracy of 56% obtained in previous works. On the other hand, using as input the WhatsApp data and a model trained with SMS data, the accuracy obtained is 64%. However, this project had another goal going a step further: showing how valuable can be the data for the companies nowadays. What makes this work special is the fact that is straight related with telecommunications, focusing on the way people communicates with each other and not in the content of the conversations. Furthermore, detecting the personality of someone just analysing the logs is impossible using the traditional methods, because the amount of data is huge and classifying personality traits is something subjective.

Regarding the construction of the classifiers for each personality factor, the first step is to analyse the data and extract 31 features, separating these in tree categories: calls, messages and location and then to train the algorithms looking for the best accuracy result.

In the case of WhatsApp, the idea came up for two reasons. One one hand, the good accuracy reached using only SMS features suggests us that we could leave apart the rest of the features and keep getting good result. On the other hand, the need of testing the classifiers in a real environment and adapted to the technologies that we use nowadays. There is not previous works about it so the uncertainty about the results were high. The accuracy of 64% obtained is better than we thought it could in the first place, considering the differences in usage between SMSs and WhatsApp.

For future work, it could be very interesting to replicate in some way the ”Friends and Family” experiment but using the data from instant messages applications like WhatsApp. The process would be to repeat the same action that we did with the 5 participants applying it to more people and using a period of time bigger.

Bibliography

[1] *Python for data analysis.*

[2] *Machine Learning : an Artificial Intelligence Approach.* Springer Berlin Heidelberg, 1983.

[3] Scikit-learn: Machine Learning in Python Gaël Varoquaux. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[4] Saul J. Berman. Digital transformation: opportunities to create new business models. *Strategy & Leadership*, 40(2):16–24, mar 2012.

[5] Kaustav Bose, Ranendu Adhikary, Sruti Gan Chaudhuri, and Buddhadeb Sau. Euclidean 1-center of a set of static and mobile points. *arXiv preprint arXiv:1609.00523*, 2016.

[6] Ericsson. ON THE PULSE OF THE NETWORKED SOCIETY Ericsson Mobility Report. 2015.

[7] Dave Evans. The Internet of Things How the Next Evolution of the Internet Is Changing Everything. 2011.

[8] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Byers. Big data: The next frontier for innovation, competition, and productivity. 2011.

[9] Jeff Michels, Ashutosh Saxena, and Andrew Y. Ng. High speed obstacle avoidance using monocular vision and reinforcement learning. In *Proceedings of the 22nd international conference on Machine learning - ICML '05*, pages 593–600, New York, New York, USA, 2005. ACM Press.

[10] Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.