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 MACHINE LEARNING SYSTEM FOR PERSONALITY CLASSIFICATION BASED ON STYLOMETRIC FEATURES

SERGIO LÓPEZ LÓPEZ
DICIEMBRE 2019

## TRABAJO DE FIN DE GRADO

| | |
|---|---|
| **Título:** | Diseño y Desarrollo de un sistema de aprendizaje automático para la Clasificación de Personalidad basado en rasgos estilométricos. |
| **Título (inglés):** | Design and Development of a machine learning system for Personality Classification based on stylometric features |
| **Autor:** | Sergio López López |
| **Tutor:** | Carlos A. Iglesias Fernández |
| **Departamento:** | Departamento de 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 MACHINE LEARNING SYSTEM FOR PERSONALITY CLASSIFICATION BASED ON STYLOMETRIC FEATURES

**Sergio López López**

Diciembre 2019

# Resumen

Durante los últimos años, el uso de dispositivos digitales con acceso a Internet, tales como tablets o smartphones, ha ido aumentando considerablemente. Esto ha supuesto un incremento en el uso de Internet y las redes sociales, sobre todo entre la población más joven. Las redes sociales permiten el intercambio de información entre personas y promueven una difusión de datos personales y contenidos que pueden ser utilizados para estudios sociológicos de las empresas. De este modo, la caracterización de la personalidad de los usuarios a través de su actividad puede ser llevada a cabo.

El conocimiento de la personalidad de un individuo es algo complicado de obtener pero su utilidad está más que contrastada. La personalidad determina cómo una persona se puede comportar, qué preferencias tiene y qué actitudes y aptitudes demuestra. En definitiva, determina la conducta y los pensamientos característicos de un individuo. Esto adquiere una importancia relevante para las empresas, por ejemplo, a la hora de reclutar nuevos trabajadores ya que se ha demostrado que la personalidad influye en el desempeño laboral posterior.

Este trabajo de fin de grado surge de la combinación de las redes sociales y la personalidad de los usuarios. De esta manera el objetivo es predecir la personalidad de las personas a partir de datos recogidos de una de las redes sociales más populares, Twitter. Todo esto se llevará a cabo mediante el uso de técnicas de Machine Learning y Big Data.

Para desarrollarlo, se crearán cuatro clasificadores diferentes que, combinados entre sí, proporcionen un perfil de personalidad de un usuario de forma automática y sencilla a partir de su perfil de Twitter. En el proceso se extraerán diferentes características mediante herramientas de aprendizaje automático y técnicas de procesado del lenguaje natural.

En la última fase, se desarrollará una aplicación web que permita conocer la personalidad del usuario. De esta manera, los departamentos de recursos humanos de las empresas podrán mejorar la toma de decisiones sobre el potencial de los candidatos sin la necesidad de realizar encuestas ni emplear más tiempo del necesario.

**Palabras clave:** PLN, Aprendizaje automático, Rasgos de personalidad, Tweepy.

# Abstract

During the last few years, the use of digital devices with Internet access, such as tablets or smartphones, has been increasing considerably. This has led to an increase in the use of the Internet and social networks, especially among the younger population. Social networks are platforms that allow the exchange of information between people and promote the dissemination of personal data and content which can be used for sociological studies of companies. In this way, the characterization of the users' personality through their activity can be carried out.

The knowledge of an user's personality is something complicated to obtain but its usefulness is more than proven. Personality determines how a person can behave, what preferences he has and what attitudes and aptitudes he shows. In short, it determines an individual's characteristic thoughts and behavior. This has a great importance to companies, for example, when recruiting new employees, as personality has been shown to influence subsequent job performance.

This end-of-degree work arises from the combination of social networks and the personality of the users. Thus, the objective is to predict the personality of people based on data collected from one of the most popular social networks, Twitter. This will be achieved by using Machine Learning and Big Data techniques.

To develop it, four different classifiers will be created which, combined with each other, will provide the users' personality profiles automatically and easily from their Twitter profiles. In the process, different features will be extracted using supervised machine learning tools and natural language processing (NLP) techniques.

In the last phase, a web application will be developed which will allow knowing the user's personality thanks to the trained models which have been developed. In this way, the human resources departments of the companies will be able to improve their decision making on the potential of the applicants without the need to carry out surveys or spend more time than necessary.

**Keywords:** NLP, Machine Learning, Personality Traits, Tweepy.

# Agradecimientos

En primer lugar, me gustaría dar las gracias a mis padres, Javier y Sonia, y mi hermana Patricia, por su amor, trabajo y sacrificio en todos estos años. Sin ellos no habría podido llegar hasta aquí. Para mi es un orgullo y privilegio tenerlos como familia.

Por otro lado me gustaría agradecer también a todos esos amigos y compañeros dentro y fuera de la universidad que me han acompañado todos estos años. Ellos han sido de gran ayuda para hacer este duro proceso más ameno.

Por útlimo, gracias al Grupo de Sistemas Inteligentes por haberme dado la oportunidad de realizar este trabajo y, en especial, a mi tutor Carlos Ángel Iglesias por haberme orientado y ayudado a lo largo de todo el desarrollo.

# Contents

# List of Figures

CHAPTER 1

# Introduction

## 1.1 Context

Nowadays, almost everyone has at least one device connected to the net and, especially young people use them to share personal information in social networks. This information resulting from the interactions between the users and their activity on the net is huge and very useful. Thus, from this data, characteristics and studies can be extracted. For this reason, companies are very interested in Machine Learning or Big Data since using this information can improve their respective businesses.

From social networks, features as tastes, pictures, locations, opinions can be extracted and predicted. Even, users' personality could be estimated in order to have an alternative to some traditional methods like surveys in companies. In this way, companies could know the personality of their employees or applicants easily, quickly and cheaply.

This project will focus in implementing a system which can predict the personality from Twitter. This social network is distinguished as good source of information about the way of thinking of users. Each user shares posts limited to 280 characters with the possibility of including links, images and videos.

The personality will be predicted based on the Myers–Briggs Type Indicator (MBTI). This widely examined theory indicates differing psychological preferences in how people perceive the world and make decisions. It is composed of four dimensions with two possibilities: Introversion-Extraversion, Intuitive-Sensing, Thinking-Feeling and Judging-Perceiving.

To summarize, this work will be able to implement a system to predict personality traits from Twitter thanks for the use of Machine Learning and Big Data techniques.

## 1.2 Project goals

The main goal of this project is the design and deployment of a classifier which is able to predict the personality in Spanish. For that purpose, data will be collected, analyzed, preprocessed and used to train the system. Specifically, our dataset will be given by Twisty [4].

Some linguistic and paralinguistic features will be extracted from data. In this way, the classifier will be trained with these features and will try to predict the MBTI users' profile. Another goal is the feature selection in order to know which techniques and features achieve the best results.

The previous goals culminate in the development of a tool for obtaining the personality traits ready to be used by companies. This tool will be integrated in a web application.

## 1.3 Project tasks

The tasks to develop this project will be presented below:

- Learning the tools and techniques available for developing the project.
- Researching of related works.
- Analysis and preprocessing of the data.
- Features extraction.
- Study of automatic learning algorithms which best fit this case.
- Software development, experimentation and feature selection.
- Analysis of the results obtained.
- Development of the web application.

## 1.4 Structure of this document

In this section, a brief explanation of the chapters included in this thesis will be provided below:

- ***Chapter 1:*** It is the introduction of the project. A description of the context where the project is developed and the main goals are presented.
- ***Chapter 2:*** An analysis of the state of the art surrounding this project.
- ***Chapter 3:*** Description of the available technologies.
- ***Chapter 4:*** It describes the architecture of the classification model and its design to achieve the objectives needed.
- ***Chapter 5:*** Ir provides the development of the web application.
- ***Chapter 6:*** It discusses the conclusions, the achieved goals and future work.

CHAPTER 2

# State of the art

This chapter includes an introduction of the main concepts and the state of the art concerning the project.

Psychological studies have provided several typologies of personality traits. Above all two models stand out: The Big Five (Goldberg, 1990) and Myers-Briggs Type Indicator (Briggs Myers and Myers, 2010).

The Big Five model uses descriptors of common language and therefore suggests five broad dimensions used by some psychologists to describe the human psyche and personality. The five factors have been defined as: Openness to experience, Conscientiousness, Extraversion, Agreeableness and Neuroticism.

On the other hand, the Myers-Briggs Type Indicator (MBTI) is an introspective self-report questionnaire indicating differing psychological preferences in how people perceive the world and make decisions. People are classified along four dimensions:

- Introversion-Extraversion (I-E): How one directs and receives energy.
- Intuitive-Sensing (N-S): How one processes information.
- Thinking-Feeling (T-F): How one makes decisions and comes to conclusions.

- Judging-Perceiving (J-P): How one presents herself or himself to the outside world.

In order to determine personality profiles, linguistic or para-linguistic features can be used. Para-linguistic features refer to the elements that accompany properly linguistic emissions, signals and indicators. These can contextualize particular interpretations of properly linguistic information.

In the next sections, some datasets and studies are presented in this context.

## 2.1 Twisty

The corpus presented by Twisty [4] provides a data-set which contains different Twitter users' gender and personality profile. It contains data in six languages (Dutch, German, French, Italian, Portuguese and Spanish). Meanwhile several personality typologies exist, MBTI is particularly well-known in the non-scientific community and therefore is the chosen one in this data-set. Thus, the structure of the corpus is composed of the user ids with their corresponding confirmed tweet ids, other tweet IDs, MBTI profile and gender as shown in as shown in Table 2.1.

The corpus creation task was divided in two steps. First, mining for user profiles that self-report their MBTI. Then these profiles were manually checked and further annotated for gender. When users were identified, their tweets were downloaded and preprocessed using the Twitter API. For each user, all recent tweets were retrieved giving rise to 2,000 tweets per user on average.

| **other_tweet_ids** | **mbti** | **user_id** | **confirmed_tweet_ids** | **gender** |
|---|---|---|---|---|
| 566773199212666880, 5... | ENTJ | 54056853 | 676146060193079296, 4... | M |
| 648930148654415872, 6... | INFP | 2573243225 | 646509854648049664, 6... | F |
| 631138007282749441, 6... | INTP | 399751937 | 672629376589111297, 6... | M |
| 659517076869144576, 6... | INFJ | 541982091 | 672139144819249152, 6... | F |

Table 2.1: Twisty Corpus

In order to achieve the highest accuracy possible, language detection tools as ldig (Nakatani, 2012), langid (Lui and Baldwin, 2012) and langdetect (Nakatani, 2010) were

used. Around 73% of tweets were confirmed as being in Spanish. For this reason the corpus release have two columns with tweet ids.

Experimental train models to predict the Myers-Briggs personality were carried out. They used a LinearSVC model as implemented in sklearn with standard parameters. They also tested LogisticRegression, which gave comparably results. Preproccessing were divided in two steps: (1) normalizing URLs, hashtags and usernames, (2) tokenization with happiertokenizer. With regard to features, they used binary ones for word n-grams (uni-grams and bi-grams) and character n-grams(tri-grams and tetra-grams). Once a 10-fold cross-validation was performed, precision, recall and f-score were reported as shown in Table 2.2.

| **Task** | **P** | **R** | **F** |
|---|---|---|---|
| I–E | 61.09 | 61.09 | 61.09 |
| S–N | 60.23 | 62.91 | 61.54 |
| T–F | 59.35 | 60.12 | 59.73 |
| J–P | 55.60 | 56.56 | 56.08 |

Table 2.2: Results by Twisty

In the next subsections, other studies and projects related with Twisty will be presented.

### 2.1.1 Language-independent Gender Prediction on Twitter

In this paper [1], a set of experiments and analyses on predicting the gender of Twitter were carried out. Twisty corpus was used as dataset in the six languages previously mentioned.

The authors extracted language-independent features from the text or the metadata of users' tweets. In Fig. 2.1, the most important features are shown. Each feature in each language is quantified by the effect of the gender-conditioned distributions. It is quite interesting the use of emojis is preferred by female gender. Also, they tend to produce more tweets per day, tweets of varying length, favorite more tweets and use more of the red color component in the tweet text. Meanwhile, male users tend to use emoticons, more question marks and hashtags and share their location.

| Feature | Avg rank | DE | IT | NL | FR | PT | ES |
|---|---|---|---|---|---|---|---|
| perc_emoji | 1.17 | 0.63 | 0.21 | 0.45 | 0.49 | 0.41 | 0.5 |
| mean_retweet_count | 11.5 | 0.09 | 0.03 | 0.09 | 0.38 | 0.27 | 0.22 |
| red_back | 12.0 | 0.24 | 0.09 | 0.13 | 0.23 | 0.38 | 0.42 |
| perc_http | 13.5 | -0.21 | -0.24 | -0.25 | -0.15 | -0.27 | -0.17 |
| perc_ios | 14.0 | -0.23 | -0.22 | -0.09 | -0.19 | -0.09 | -0.13 |
| var_retweet_count | 15.17 | -0.1 | 0.05 | 0.1 | 0.11 | 0.03 | 0.04 |
| perc_retweeted | 15.33 | -0.01 | 0.2 | -0.2 | 0.2 | 0.26 | 0.17 |
| perc_question | 16.0 | -0.35 | -0.13 | -0.1 | -0.29 | -0.14 | -0.11 |
| user_tweet_per_day | 17.0 | 0.08 | 0.19 | 0.01 | 0.31 | 0.15 | 0.12 |
| perc_emoticon | 18.17 | -0.23 | -0.25 | -0.17 | -0.18 | -0.24 | -0.1 |
| user_location | 18.67 | -0.17 | -0.2 | -0.21 | -0.11 | -0.17 | -0.12 |
| mean_hour | 19.33 | 0.08 | 0.23 | 0.18 | 0.22 | -0.1 | -0.02 |
| var_len_text | 20.0 | 0.25 | 0.24 | 0.2 | 0.24 | 0.01 | 0.08 |
| user_favour_count | 20.33 | 0.06 | 0.09 | 0.02 | 0.1 | 0.02 | 0.06 |
| user_tweet_count | 20.33 | 0.03 | 0.2 | -0.01 | 0.23 | 0.13 | 0.09 |
| user_follow_friend_rat | 21.5 | -0.13 | 0.12 | -0.05 | -0.08 | -0.04 | -0.03 |
| mean_favorite_count | 21.5 | 0.16 | 0.09 | 0.02 | -0.07 | -0.02 | -0.03 |
| med_hour | 22.0 | 0.13 | 0.23 | 0.17 | 0.2 | -0.01 | -0.07 |
| green_back | 22.17 | 0.2 | 0.04 | -0.04 | 0.12 | 0.26 | 0.25 |
| blue_back | 22.33 | 0.27 | 0.06 | -0.01 | 0.11 | 0.29 | 0.33 |
| perc_is_quote | 22.83 | -0.04 | 0.17 | -0.21 | 0.18 | 0.17 | 0.03 |
| perc_favorited | 23.33 | 0.31 | 0.16 | -0.02 | 0.14 | 0.05 | 0.01 |
| med_retweet_count | 24.17 | -0.08 | -0.06 | -0.09 | 0.16 | 0.06 | 0.04 |
| var_favorite_count | 24.17 | -0.09 | 0.07 | 0.07 | -0.12 | -0.02 | -0.03 |
| var_hour | 25.17 | -0.11 | -0.01 | -0.1 | -0.14 | 0.21 | 0.05 |
| user_red_text | 25.83 | 0.15 | 0.05 | 0.09 | 0.22 | 0.13 | 0.16 |
| user_listed_count | 28.33 | -0.12 | -0.09 | -0.09 | -0.17 | -0.0 | -0.07 |
| perc_exclamation | 28.83 | 0.26 | 0.09 | 0.49 | -0.04 | -0.04 | 0.14 |
| var_day | 29.17 | 0.09 | 0.1 | -0.0 | 0.14 | 0.12 | 0.12 |
| perc_hash | 29.67 | -0.11 | -0.05 | -0.03 | -0.16 | -0.09 | -0.11 |

Figure 2.1: Representation of 30 features with the highest average rank across languages. Each feature in each language is represented through the difference between feature means of the female and male subsets in a standardized dataset. Red encodes higher female mean, blue male [1].

The results showed that while the prediction model based on language-independent features performs worse than the language-dependent model when training and testing on the same language, it regularly outperforms the language-dependent model when applied to different languages, showing very stable results across various languages.

### 2.1.2 Reddit: A Gold Mine for Personality Prediction

The authors of this article [2] used a large-scale dataset derived from Reddit labeled with Myers-Briggs Type Indicators (MBTI) and other set of features. The dataset, which is in English, was preprocessed, trained and tested in order to predict the MBTI. Macro F1-scores between 67% and 82% on the individual dimensions were achieved.

The set of extracted features was divided into two main groups: linguistic features

and user activity features. The first ones included Term Frequency (TF) and Term Frequency–Inverse Document Frequency (TF-IDF) weighed in character n-grams (lengths 2–3) and word n-grams(lengths 1–3), stemmed with Porter's stemmer; LIWC [5] to extract other personality features; number of psycholinguistic words lists, including perceived happiness, affective norms, imageability, and sensory experience...

As user activity features, the authors extracted the number of comments, number of subreddits commented in, number of posts, time intervals between comment timestamps...

The most relevant features for each MBTI dimension are shown in Fig. 2.2. The main observation is that tf-idf-weighted character n-grams are the most relevant features for all dimensions except for E/I, for which tf-idf-weighted word n-grams are most relevant.

| Feature group | E/I | S/N | T/F | J/P |
|---|---|---|---|---|
| char_tf | 29.03 | 45.16 | 35.48 | 51.61 |
| word_tf | 35.48 | 25.81 | 12.9 | 32.26 |
| liwc | 19.35 | 0.0 | 25.81 | 9.68 |
| lda100 | 6.45 | 0.0 | 9.68 | 3.23 |
| psy | 3.23 | 0.0 | 12.9 | 0.0 |
| word | 3.23 | 9.68 | 0.0 | 0.0 |
| char | 0.0 | 12.9 | 0.0 | 0.0 |
| posts | 0.0 | 6.45 | 0.0 | 3.23 |

Figure 2.2: Percentage of each feature group in top-30relevant features for each dimension [2].

| | Dimensions | | | | |
|---|---|---|---|---|---|
| Model | E/I | S/N | T/F | J/P | Type |
| LR all | 81.6 | 77.0 | **67.2** | **74.8** | 40.8 |
| MLP all | **82.8** | **79.2** | 64.4 | 74.0 | **41.7** |
| SVM all | 79.6 | 75.6 | 64.8 | 72.6 | 37.0 |
| LR w_ng | 81.0 | 73.6 | 66.4 | 71.8 | 38.0 |
| LR chr_ng | 62.2 | 64.0 | 66.4 | 65.8 | 26.5 |
| LR liwc | 55.0 | 49.8 | 65.0 | 57.4 | 14.2 |
| LR psych | 52.0 | 48.2 | 64.0 | 57.0 | 12.5 |
| LR lda100 | 50.0 | 48.2 | 62.4 | 56.2 | 13.9 |
| LR posts | 49.4 | 53.2 | 48.0 | 51.8 | 9.5 |
| LR subtf | 49.6 | 49.6 | 50.4 | 50.2 | 13.2 |
| MCC | 50.04 | 50.04 | 50.0 | 50.02 | 25.2 |

Figure 2.3: Macro F1-scores for per-dimension prediction and accuracy of type-level prediction for models with all features, LR models with a single feature group, and the MCC baseline [2].

The authors experimented with three different classifiers: SVM, LR and MLP combined with cross-validation. The prediction results for each dimension in terms of the macro F1-score are shown in Fig. 2.3.

### 2.1.3 TECLA

TECLA is a temperament and psychological type prediction framework from Twitter English data [3]. The database used was downloaded from the paper [6], in which the Twitter users are classified according to the psychological types of Myers-Briggs. The dataset contains MBTI, tweets, gender, number of followers, number of tweets, number of favorites and number of listings.

The information extracted from the database can be divided into two categories: grammatical and behavioral. The first category considers information from LIWC [5], MRC [7], sTagger [8] or oNLP [9], extracted from the user's set of messages, similarly to what was proposed in the Polarity Analysis Framework introduced by the authors [10]. The behavioral category includes the number of tweets, number of followers, followed, favorites, number of listings and number of times the user was favorited.

For the classification of the MBTI model, the system was designed with four classifiers that receive the same data, but it is trained to identify the opposing pairs of attitudes and functions. The model algorithms were TiMBL, NB, SVM,LR, LinearSVC and Random Forest. All tests were performed with 10 runs of a k-fold cross-validation (k = 5). The results in Fig. 2.4 showed that Random Forests achieved the better performance.

| **Algorithm** | **Features** | **Measure** | **I/E** | **N/S** | **T/F** | **J/P** |
|---|---|---|---|---|---|---|
| TiMBL | MBSP, n-gram, Lexical features | F-Measure | 65.38% | 61.81% | 49.09% | 51.67% |
| NB, SVM | n-gram, LIWC | F-measure | I: 9.00% E: 50.00% | N: 75.88% S: 78.42% | F: 75.00% T: 73.00% | J: 84.26% P: 72.93% |
| NB, logistic regression and SV classification | n-gram | Accuracy | 63.90% | 74.60% | 60.80% | 58.50% |
| Logistic regression | n-gram | Accuracy | 72.50% | 77.40% | 61.20% | 55.40% |
| LinearSVC | n-gram | F-Measure | 67.87% | 73.01% | 58.45% | 56.06% |
| NB | n-gram, POS-tags | Accuracy | 80.00% | 60.00% | 60.00% | 60.00% |
| Random Forest | LIWC, oNLP | Accuracy | 82.05% | 88.38% | 80.57% | 78.26% |
| Random Forest | LIWC, oNLP | F-measure | I: 87.2% E: 70.38% | N: 92.66% S: 72.13% | F: 84.49% T: 74.01% | J: 81.49% P: 73.66% |

Figure 2.4: MBTI results with different models and features [3].

### 2.1.4 Characterizing the Personality based on their Timeline

In this study [11] the authors analyzed publications of the Portuguese users of the social network Twitter. The aim is predicting the personality using The Big Five psychological model. They used different features and methods such as sentimental analysis for each tweet, lexical complexity, POS, number of friends, publication times, number of followers, locations, etc. to get a more precise picture of a personality.

They also used the Brazilian Portuguese LIWC [5] Dictionary to calculate scores for every psychological category in order to predict each of the Big Five personality traits. The authors remarks the importance of analyzing hashtags because they may carry some emotions, sentiments, interests, the things a user is focused on, ads of brands or other significant information.

With regard to sentimental analysis they compared each word in all tweets with words in two lexicons (SentiLex-PT and NRC Emotion Lexicon). As a result, the probability of each tweet being positive, negative or neutral was calculated.

Part of Speech plays a vital role in understanding the users' personalities. The authors [12] pointed out that in oral language high extraverts tend to use more adverbs, pronouns, verbs, while the usage of nouns, adjectives and prepositions is very low. Other important features were the average number of words per tweet, the number of swear words and the number of positive and negative emoticons.

## 2.2 Personality Prediction based on Mobile-Phone Metrics

Now, some related works which are not based on Twisty will be presented.

In this project [13] a system was designed and implemented in order to predict personality based on Mobile-Phone based metrics such as phone call logs, SMS logs and estimated geolocation. The personality model chosen was The Big Five Personality model.

The feature extraction can be divided in two: features from call and SMS logs and features from location data.

Some interesting features from the first group are the number of interactions that a user have in a period of time; the inter-event time which refers to the time between different calls and SMSs; number of contacts and the number of interactions at night.

From Location data, the author used the radius of gyration in order to discover how far

a user moves from the place he usually is.

The best results when classifying were with the Random Forest and Decision Tree algorithms. The Grid Search tool and the cross-validation were used in the process. Finally, the results obtained were 71% ,77% ,81% ,84% and 65% for Extraversion, Agreeableness, Conscientiousness, Neuroticism and Openness respectively.

## 2.3 Personality Prediction from YouTube

The author of this project [14] tried to predict the MBTI personality in Spanish from a YouTube dataset. This corpus was composed of a collection of behavioral (audiovisual) features, speech transcriptions, and personality impression scores. In order to so, semantic features from the dataset were extracted, sentiments and emotions were included and other features were added.

Some key metrical features were the number of words, words per sentence, capital letters, first and second pronouns, POS, exclamations and questions, paragraphs, long words and punctuation marks. Gender, bag-of-words and tf-idf played a vital role too.

Regarding the sentiment features, the author used Senpy [15], a MeaningCloud plugin which identifies positive/negative polarity in any text, including comments in surveys and social media.

Finally, the extraction of all the features resulted in a dataset containing all the information that could be read by the classifier. The classification algorithms with the best performance were Logistic Regression, K-Nearest Neighbors and Random Forest combined with k-fold. The final results were around 95% in Extraversion and 75% in the rest dimensions.

## 2.4 Comparison between algorithms to predict personality

This project [16] established a comparison between the SVM, Random Forest and Naive Bayes classifiers to predict personality from Twitter data. Some of preprocessing techniques were deleting punctuation marks, HTML, numbers, emojis, urls and capital letters. After that, tokenization, stemming, vectorization and tf-idf were carried out. These were the only techniques used so the final results were not very good. The authors achieved at maximum 65.25% with Random Forest and even worse with SVM and Naive-Bayes.

Some techniques the authors did not use but recommended to test in future researches were linguistic techniques as association between synonyms and antonyms. Also they suggested proving with other classifiers.

## 2.5 Predicting MBTI type using text data

From a English posts dataset, the author of this blog [1] tried to predict MBTI dimensions. In order to achieve this, the first step was basic data cleaning. Instead of doing sixteen imbalanced classifications head-on, the author decided to create four binary classifiers.

Concerning the features extraction the author ensured that video, image and other links could potentially add to the data set, other than their mere count. Thus, video titles from links were extracted. Other extracted features were the number of: mentions, tags, emphasis or action words, emoticons, bracket words and dot usage. Other data available were the word count, character count and number of fully capitalized words.

Also, POS tagging was used for identifying the type of words within a sentence. For each tag, the author took the mean, the median and standard deviation using Sanford NLP version of POS tagging[17].

Once removal of stop words is used, TF-IDF is used separately across the four MBTI dimensions. The author runs the TF-IDF model with n-gram range of 1 to 4 and maximum features of 10,000 words/phrases. Then, the features were reduced to 500 with Truncated Singular Value Decomposition (Truncated SVD). The end result was 1500 truncated columns of various n-gram ranges which clearly helped managing computer memory.

Finally, using the Logistic Regression model, scores around 82% were achieved across the four MBTI dimensions.

## 2.6 Conclusions

Once an exhaustive search has been made for related works, it is time to pool all features in order to note which of them are the most useful. In Table 2.3 all the extracted and useful features are collected referencing the section or subsection where they have been extracted with an “x”. A “Total” column has been added as the sum of the related works where each feature has been extracted.

[1] https://yix90.github.io/blog/2018/02/23/Predicting-Your-MBTI

| | 2.1 | 2.1.1 | 2.1.2 | 2.1.3 | 2.1.4 | 2.2 | 2.3 | 2.4 | 2.5 | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| **Tf-idf** | x | | x | | | | x | x | x | 5 |
| **Nº emoticons/emojis** | | x | | | x | | | x | x | 4 |
| **Nº words per tweet** | | x | | | x | | x | | x | 4 |
| **Bigrams** | x | | x | | | | | | x | 3 |
| **Hashtags** | | x | | | x | | | | x | 3 |
| **LIWC** | | | x | x | x | | | | | 3 |
| **Location** | | x | | | x | x | | | | 3 |
| **Nº followed** | | | | x | x | x | | | | 3 |
| **Nº followers** | | | | x | x | x | | | | 3 |
| **POS** | | | | | x | | x | | x | 3 |
| **Trigrams** | x | | x | | | | | | x | 3 |
| **Capital letters** | | | | | | | x | x | x | 3 |
| **Nº exclamation marks** | | | | | | | x | | x | 2 |
| **Gender** | | | | x | | | x | | | 2 |
| **Nº characters per word** | | x | | | | | | | x | 2 |
| **Nº favorited** | | x | | x | | | | | | 2 |
| **Nº tweets** | | | x | x | | | | | | 2 |
| **Publication time** | | | | | x | x | | | | 2 |
| **Nº question marks** | | x | | | | | x | | | 2 |
| **Sentiment analysis** | | | | | x | | x | | | 2 |
| **Tetragrams** | x | | | | | | | | x | 2 |
| **Nº sentences per tweet** | | x | | | | | x | | | 2 |
| **Nº tweets per day** | | x | | | | x | | | | 2 |
| **Nº links** | | | | | | | | x | x | 2 |
| **Time between tweets** | | | x | | | x | | | | 2 |
| **Cursed words** | | | | | x | | | | | 1 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Dot usage** | | | | | | | | | x | 1 |
| **Lexical complexity** | | | | | x | | | | | 1 |
| **Nº bracket words** | | | | | | | | | x | 1 |
| **Nº images** | | | | | | | | | x | 1 |
| **Nº mentioned** | | | | | | | | | x | 1 |
| **Nº retweeted** | | x | | | | | | | | 1 |
| **Nº mentioning** | | | | | | | | | x | 1 |
| **Nº videos** | | | | | | | | | x | 1 |
| **Profile link color** | | x | | | | | | | | 1 |

Table 2.3: State-of-art features summary

CHAPTER 3

# Enabling Technologies

Technologies used throughout this project will be illustrated in this chapter. It includes several libraries and programming languages which are available in order to carry out this project.

## 3.1 NumPy

NumPy [18] is the fundamental package for scientific computing in Python. It provides an efficient, fast and powerful multidimensional array object; some linear algebra operations and tools for writing and reading data sets to disk; and sophisticated broadcasting functions. Furthermore, integration with other programming languages is supported.

Besides its scientific uses, NumPy can be used as a dimensional container of generic data. Arbitrary data-types can be defined. This makes NumPy one of the most based-on libraries in Python such as Pandas [19].

## 3.2 Pandas

Pandas [19] is an open source, BSD-licensed library written for the Python programming language for data modeling and analysis. The main advantage of this library is that provides facilities for grouping, querying and merging Pandas data structures. Furthermore, high-performance and easy-to-use data structures are provided:

- **Series:** One-dimensional object where each element has an own index. It is similar to an array, a list, a dictionary or a column in a table.
- **DataFrame:** Two-dimensional object for data manipulation with integrated indexing. It is similar to a database table, or a spreadsheet.

Pandas implements useful functionalities as reading and writing data from different formats (CSV, JSON, text files and SQL databases); merging and joining different data sets; intelligent label-based slicing, fancy indexing, and subsetting of large data sets; intelligent data alignment and integrated handling of missing data, etc.

Thus, thanks to the facilities offered by Pandas, tweet messages from Twisty [4] will be imported from JSON files and managed by Pandas.

## 3.3 Matplotlib

Matplotlib [20] is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Thanks to this library plots, histograms, power spectra, bar charts among others can be generated.

All the graphical representations of this project have been generated with this library.

## 3.4 Seaborn

Seaborn [21] is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Some of the functionalities that Seaborn offers are:

- A dataset-oriented API for examining relationships between multiple variables.

- Specialized support for using categorical variables to show observations or aggregate statistics.
- Options for visualizing univariate or bivariate distributions and for comparing them between subsets of data.
- Automatic estimation and plotting of linear regression models for different kinds dependent variables.
- Convenient views onto the overall structure of complex datasets.
- High-level abstractions for structuring multi-plot grids that let you easily build complex visualizations.
- Concise control over Matplotlib figure styling with several built-in themes.
- Tools for choosing color palettes that faithfully reveal patterns in your data.

## 3.5 Scikit-Learn

Scikit-learn [22] is an Open Source library of Python that provides a wealth of machine learning algorithms. It provides simple and efficient tools for data mining and data analysis. This library is built upon other libraries: Numpy, SciPy(Scientific Python) and Matplotlib.

Scikit-learn includes tools such as:

- **Classification:** The aim is to identify to which category an object belongs to. Thus, each input vector is assigned to one finite number of discrete categories.
- **Regression:** If the desired output consists of one or more continuous variables, then the task is called regression. So, in this case the label's format is continuous.
- **Clustering:** Automatic grouping of similar objects into sets. Customer segmentation and grouping experiment outcomes are examples of some applications.
- **Dimensionality reduction:** Reducing the number of random variables to consider. This can be used for visualization or increasing efficiency.
- **Model selection:** Comparing, validating and choosing parameters and models. A possible application is improving accuracy via parameter tuning, like using Grid Search and vectorizers. GridSearchCV is able to tune hyper-parameters of a certain

model. Thus, by defining which parameters to use in a parameter grid, all combinations of parameters are tested. So the selection of the parameters with the best scores can be selected. On the other hand, vectorizers are used to transform sentences or words into vectors in order to, for example, word counting.

- **Preprocessing:** Feature extraction and normalization. This is important to transform input data such as text for use with machine learning.

## 3.6 Natural Language Processing

Natural Language Processing (NLP), is a branch of artificial intelligence that deals with the interactions between humans and computers using the natural language. Its objective is to read, decipher, understand and make sense of the human languages in manner that is valuable. In order to achieve this, some Python libraries has been used in this project regarding the tweets.

In the next sections, some the most popular libraries will be explained.

### 3.6.1 NLTK

Natural Language Toolkit [23], usually shortened as NLTK, is a set of libraries and programs for working in Python with NLP techniques. It provides easy-to-use interfaces to over 50 corpora and lexical resources. Thus, a variety of text processing libraries are provided for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

- **Tokenization:** It consists in taking a text or a set of texts and breaking it up into its individual words, called tokens. These are then used as the input for other types of analysis or tasks. In this process the tokenizer function filters for avoiding undesirable characters and deletes possible mistakes in the transcription of the words.

  As in this project the text is a set of tweets, a special tokenizer is required: TweetTokenizer. This tokenizer takes into account processing special tokens, such as hashtags.

- **Stemming and Lemmatization:** These are linguistic processes which are responsible for reducing the inflectional form of a token into its base or root. The main difference between stemmers and lemmatizers is that stemmers operate in isolated words, while lemmatizers take into account the context.

- **Pos-Tagging:** This is responsible of tagging each word of a text to its grammatical category (i.e. adjectives, verbs, nouns, etc.).

- **Removing Stop Words:** Stop words provide no information about the content of the document, in other words, they are meaningless and are used only for syntactical purposes. Thus, it is possible to remove them thanks fot a corpus provide by NLTK [23]. This corpus contains a list of typical words in Spanish add no meaning.

- **N-gram Similarity:** N-grams [24] are substrings of n elements, in this case words, which let the extraction of characteristics from sequences of words instead of isolated words.

- **Term Frequency Inverse Document Frequency:** Usually shortened as *TF-IDF* [25], it is an efficient an simple algorithm whose aim is reflecting how important a word is to a document. It doesn't take into account relationships between words but increases proportionally the tf-idf value to the number of times a word appears in the document.

### 3.6.2 GSITK

GSITK (GSI Toolkit) [26] is a library on top of scikit-learn that eases the development process on NLP machine learning driven projects. It uses NumPy [18], Pandas [19] and related libraries to easy the development. It manages datasets, features, classifiers and evaluation techniques, so that writing an evaluation pipeline results fast and simple. The main given features are the Word2Vec Features, that implements a generic word vector model, previously loaded a Word Embeddings model. Among others, it allows to transform a text into a numeric vector to work with it.

The *pprocess_twitter* module provided by GSITK let the detection of special characters or words from Twitter text, such as urls, users, hashtags, emotional faces, hearts, elongations of words, and words written in all caps.

## 3.7 Tweepy

As the dataset given by Twisty [4] provides the ids of the tweets, collecting those tweets from theirs ids is mandatory. In order to do so, the API Twitter[1] and the library Tweepy[2]

[1] https://developer.twitter.com/en/docs
[2] http://www.tweepy.org/

are needed.

The API Twitter makes reading and writing Twitter data possible. Twitter users who has been registered using authentication and OAuth authorization [27] can create and collect tweets and read the profile of users. The Twitter REST API responses are in JSON format.

Recently, the number of people using APIs has increased; therefore, numerous wrappers have been implemented by the developer community to be able to use these APIs. Tweepy is an open source library that allows using Python to communicate with the Twitter API. Thus, handling of authentication, connection/disconnection, errors and filtering is simple and intuitive.

CHAPTER 4

# Machine learning model building and evaluation

## 4.1 Introduction

We live in an interconnected world where it is very difficult to find someone who does not have a device with Internet access and any profile in a social network. Thus, during the last few years, users of social media have increased exponentially. These platforms are used to share content with your friends and your environment. This content can be photos, thoughts, experiences, videos...

Among all social media, Twitter excels by its content: a limited number of characters where a user can write his thoughts, daily business, general opinions, personal issues, politics, etc. This is why Twitter is known as a social network of microblogging, that is, a reduced way of blogging. In turn during the last years, Twitter has become one of the most popular communication platforms. Such is its importance, that numerous celebrities, companies and brands have accounts from which they promote their products, tell their problems, thoughts and private lives.

All of this makes Twitter a powerful tool to know the thinking of the users and society.

Even though, it is possible to predict with some certainty the personality profile of an active user. Thus, the aim of this project is trying to predict, through a machine learning model, the personality profiles based on the stylometric features from different users thanks to their tweets.

In this sections, we will explain the implementation of a personality classifier using the Twisty corpus [4]. But first of all, in the next overview, the general vision and the steps taken during this project will be presented. Then we will explain in detail each step and the reasons why they are necessary.

### 4.1.1 Overview

In this section, the global overview of the project will be presented, defining the different steps which have been taken for the creation of the project. As we previously mentioned the aim is the creation of a classifier system capable of predicting personality traits. In order to achieve this, five different steps has been followed as it is shown in Fig. 4.1

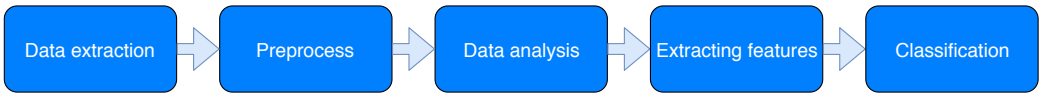


Figure 4.1: Model Development Phases.

First of all, a data extraction using the API Twitter is necessary because the corpus by Twisty [4] does not provide tweets but their ids. Once we have the raw data, a preprocessing step is necessary in order to create a Dataframe with the useful data. Then, data analysis can be carried out, focusing on the distributions, correlations, stats, etc. Afterwards, we can extract features using different techniques which can improve the system. Finally, in the classification step, different models interpret those extracted features, and based on that information, try to predict the personality tries.

## 4.2 Data extraction

The corpus given by Twisty [4] provides a dataset where every user has a list of confirmed tweets ids and a list of other tweets ids, as it is mentioned in Section 2.1. In this project our focus will be in the list of confirmed tweets ids, because we are only interested in tweets which are confirmed as being in Spanish.

In order to download the related tweets, the API Twitter [27] is necessary. For that,

a signing up for a Free RapidAPI User Account must be done. Then, a consumer key, consumer and access keys will be given by Twitter. Lastly, using the library Tweepy 3.7 and the id of a tweet, we can download its JSON file.

The process of downloading tweets can fail in three different cases: if the tweet or the user account has been deleted or if the user account has been made private.

JSON tweet files of are objects which encapsulate core attributes of the tweet, as the author, the message itself, the timestamp, the number of followers and followed, etc. An entity object is generated too with each twee, which are arrays of common Tweet contents such as hashtags, media, links and mentions.

In our case, we have downloaded at most 500 tweets per user from 1000 different random Spanish user accounts. Then, we have created a dataframe as is shown in Fig. 4.2 with these users, a concatenation of their tweets, their number of tweets, their gender and their MBTI profile.

| | user_id | confirmed_tweet_ids | n_tweets | tweets | gender | mbti |
|---|---|---|---|---|---|---|
| 0 | 107114427 | ['111661663295512576', '116114206771785728', '... | 282 | ['Como circo pobre... Ultimo dia en Bogotaaaaa... | M | INTJ |
| 1 | 62512224 | ['558384846570274816', '605747045421490176', '... | 185 | ['@notiven @unidadvenezuela lo que deberia es ... | M | ESFJ |
| 3 | 166027360 | ['120608345668325376', '105834375018586112', '... | 799 | ['@SancadillaNorte que no te gusten es una cos... | F | INTJ |
| 5 | 76869645 | ['254322314571825153', '412593920649203713', '... | 2129 | ['Jerry Rivera en el Aeropuerto... #ñaaaaaau',... | M | ENFP |
| 6 | 76985121 | ['676661434676338688', '687333665651748865', '... | 1545 | ['Nuevos seguidores: 2, unfollowers: 1 (04:35)... | F | INFP |
| ... | ... | ... | ... | ... | ... | ... |
| 994 | 109333777 | ['273276966067642368', '209701982959710208', '... | 1746 | ['Con antojo de algo y no de que... Es desespe... | M | ENFJ |
| 995 | 111464907 | ['684357766660657152', '194262938306097153', '... | 1571 | ['@Elaine523 Mira yo no soy tan viejo, usado s... | M | ESTJ |
| 996 | 4010949399 | ['665181612477440000', '686299105560293376', '... | 1821 | ['// Oye el rechazo lo damos muy por hecho per... | M | ISFP |
| 997 | 67729811 | ['649007197083553792', '322214395977539587', '... | 995 | ['"Voy a despertarte con un beso. Tú has como ... | F | ENFP |
| 998 | 53289729 | ['647521589127352320', '687640587395190784', '... | 2360 | ['@aradenatorix Yo al principio pensé también ... | F | ENFP |

Figure 4.2: Initial Dataframe.

## 4.3 Preprocessing

Once the data extraction has been carried out, a preprocessing is needed. This is a crucial step since complexity of the data under analysis is reduced. In this phase, the raw text from tweets is processed in order to make it easier to analyze. This includes a cleaning achieved by eliminating the stop words, punctuation marks, urls, emoticons, urls, etc. However, is it possible that some of these steps will not be taken, this is, maybe some features can be extracted from punctuation marks or emoticons, for example.

This short of cleaning can be done thanks to NLTK [23] as we already explained in Section 3.6.1. Thus, tokenization, stemming and removing stop words have been applied to

the tweets.

On the other hand, information like gender and MBTI profile in the dataframe are given as text. So, encoding these categorical values must be done. The best option is not encoding features as continuous variables, since Scikit-learn [22] estimators would expect continuous inputs, and the categories could be interpreted as being ordered, which is not the case. So we have applied binary variables to gender and hot encoding to MBTI values. So the chosen encoding values are:

- **Gender:** The character "F" refers to female users and has been encoded as "1". Meanwhile, the character "M" refers to male users and has been encoded as "0".
- **MBTI:** In this column, we have sixteen different strings corresponding to the sixteen types of personalities. So, the MBTI value of "INTJ" refers to an Introvert, iNtuitive, Thinking and Judging user. In order to encode this, we have applied one hot encoding that is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction. Thus, four different columns have been created, each of them containing the value of each MBTI dimension as "1" or "0".
  - The column "I_E" contains a "1" if the user is introvert and a "0" if he is extravert.
  - The column "S_N" contains a "1" if the user is sensitive and a "0" if he is intuitive.
  - The column "T_F" contains a "1" if the way the user makes decisions is thinking and a "0" if the way is feeling.
  - The column "J_P" contains a "1" if the user has the attribute "Judging" and a "0" if he has "Perceiving".

The Fig. 4.3 is the result of this one hot encoding.

Finally, the last preprocessing step has been deleting users with less than 200 collected tweets in order to have enough tweets to work with.

| | user_id | n_tweets | tweets | gender | mbti | I_E | S_N | T_F | J_P |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 107114427 | 282 | ['Como circo pobre... Ultimo dia en Bogotaaaaa... | 0 | INTJ | 1 | 0 | 1 | 1 |
| 1 | 62512224 | 185 | ['@notiven @unidadvenezuela lo que deberia es ... | 0 | ESFJ | 0 | 1 | 0 | 1 |
| 2 | 166027360 | 799 | ['@SancadillaNorte que no te gusten es una cos... | 1 | INTJ | 1 | 0 | 1 | 1 |
| 3 | 76869645 | 2129 | ['Jerry Rivera en el Aeropuerto... #ñaaaaaau',... | 0 | ENFP | 0 | 0 | 0 | 0 |
| 4 | 76985121 | 1545 | ['Nuevos seguidores: 2, unfollowers: 1 (04:35)... | 1 | INFP | 1 | 0 | 0 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 838 | 109333777 | 1746 | ['Con antojo de algo y no de que... Es desespe... | 0 | ENFJ | 0 | 0 | 0 | 1 |
| 839 | 111464907 | 1571 | ['@Elaine523 Mira yo no soy tan viejo, usado s... | 0 | ESTJ | 0 | 1 | 1 | 1 |
| 840 | 4010949399 | 1821 | ['// Oye el rechazo lo damos muy por hecho per... | 0 | ISFP | 1 | 1 | 0 | 0 |
| 841 | 67729811 | 995 | ['"Voy a despertarte con un beso. Tú has como ... | 1 | ENFP | 0 | 0 | 0 | 0 |
| 842 | 53289729 | 2360 | ['@aradenatorix Yo al principio pensé también ... | 1 | ENFP | 0 | 0 | 0 | 0 |

Figure 4.3: Preprocessed Dataframe.

## 4.4 Data analysis

In this section, a data analysis will be performed in order to discover the distribution of the data. Thus, we will visualize the most important data aspects. We will show the proportions of each MBTI trait and gender of our dataset as well as their correlations. This can provide us key information to extract features for the classifier.

After the preprocessing step, our dataframe has been reduced to 734 different users. With regard to the gender, our dataset is distributed as it is shown in Fig. 4.4. Thus, we have 411 female and 323 male users. This is important to know because the use of language is different between genders [28].

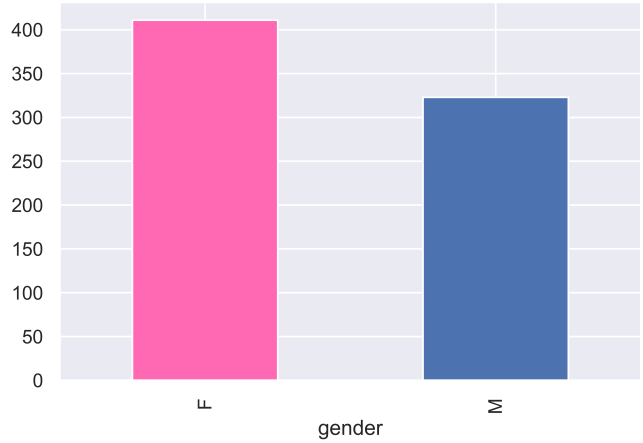


Figure 4.4: Gender distribution.

In Fig. 4.5 we can observe the distribution between the sixteen different personality types. As we can see the distribution is not balanced since we have more than 80 "INFP" users while only 19"ISFP" ones.

Imbalance data refers to situations where the number of observations is very different for all classes in a classification dataset. Machine learning classifiers tend to favor the class with the largest proportion of observations, which may lead to misleading accuracies. So, as we are interested in finding high accuracies, it is important that the inputs to the classifier are balanced.

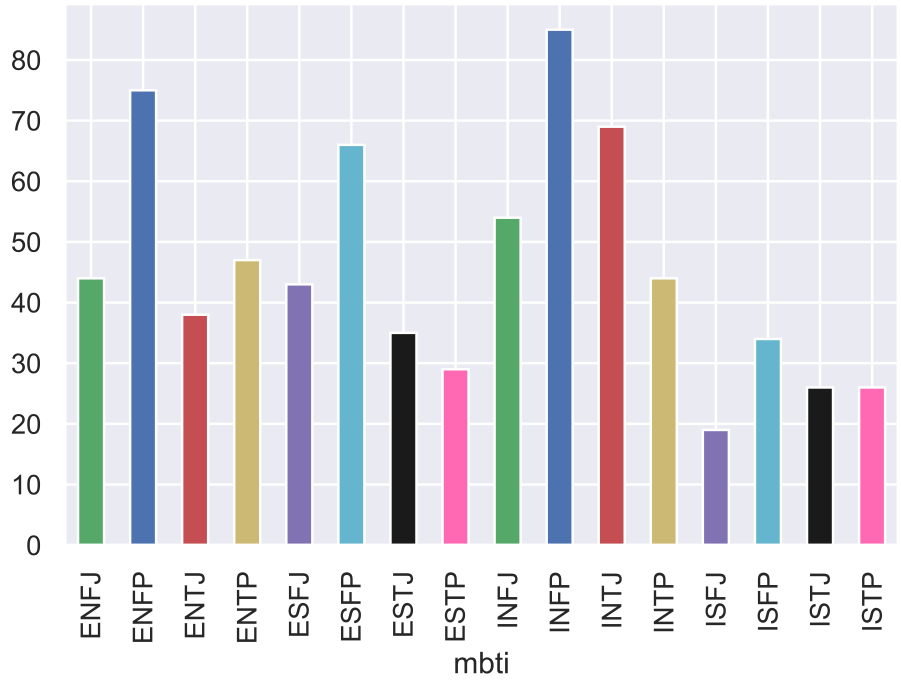


Figure 4.5: MBTI distribution.

However, the distribution between each pair of traits is balanced as it is shown in Fig. 4.6. So, as we will create different classifiers for each trait, we can assume that our dataset is balanced.

Another interesting visualization is the pairwise correlation between gender and MBTI traits. The correlation is usually defined as a measure of the linear relationship between two quantitative variables. In other words, it measures the relationship between two or more variables or factors where dependence between them occurs in a way that cannot be attributed to chance alone. Fig. 4.7 illustrates an overview of the correlation in our dataset. As a result, we can ensure that correlations between these traits are barely high.

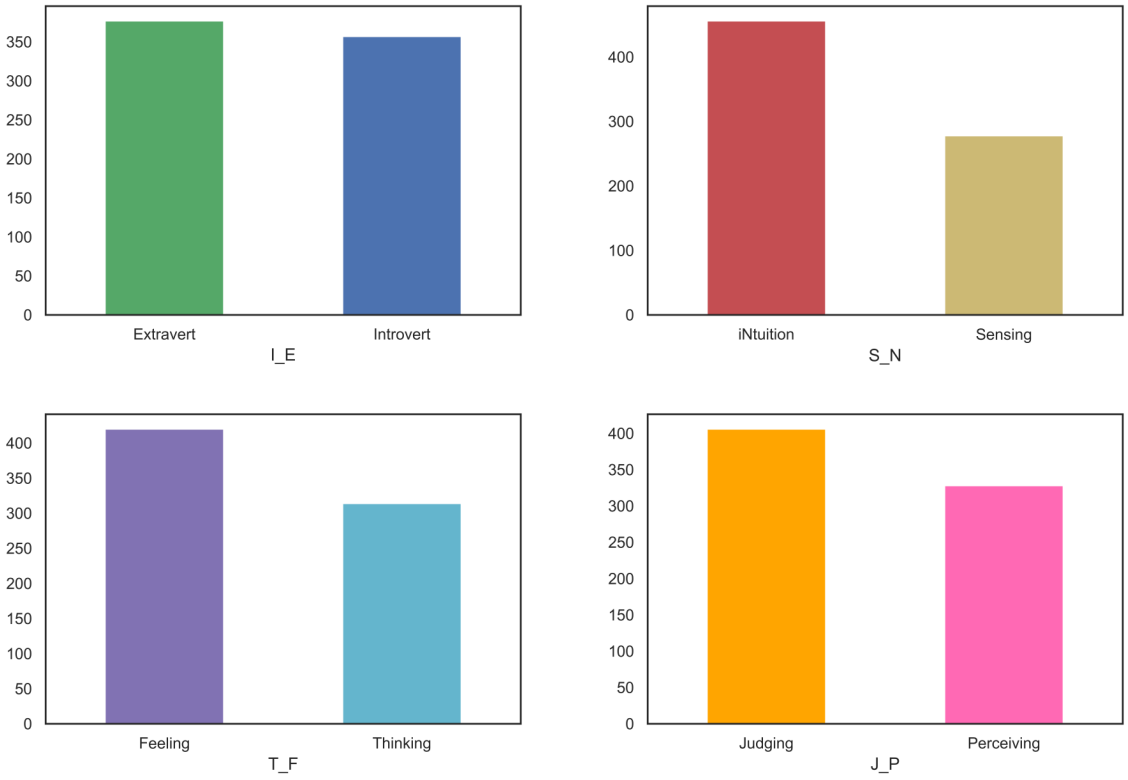


Figure 4.6: Each MBTI trait distribution.

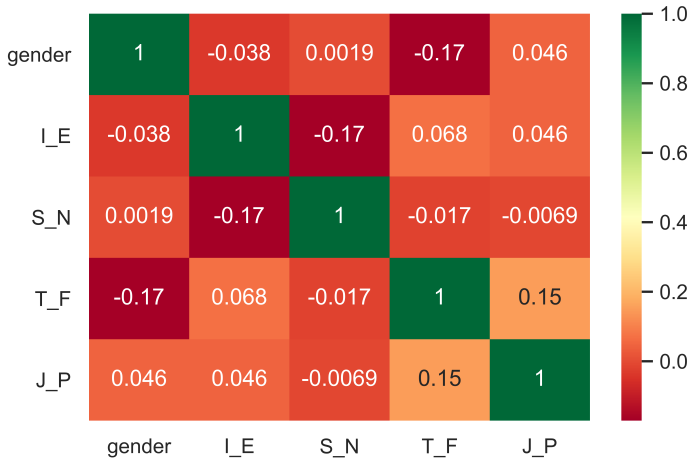


Figure 4.7: Correlations.

## 4.5 Feature extraction

In this section, we will analyze different methods of feature extraction. This features will be used later to train the classifier, hence its importance. We will try to extract similar features to related works as it is shown in Chapter 2. Certain features may provide valuable information to the classifier while others may not. Our aim is to find as many good features as possible, and at the same time, discard the ones which only add confusion to the classifier.

Our feature extraction can be divided in two clearly separate parts:

- **Linguistic features:** The elements which can be taken from the text itself.
- **Para-linguistic features:** These refer to the elements that accompany properly linguistic emissions, signals an indicators.

In the next sections we will widely explain it.

### 4.5.1 Linguistic features

The starting point to extract these features is the raw text given by the tweets. However, from a computer's point of view, raw text does not give information, so there is a need of preprocessing. Once we have applied tokenization, stemming and removal of stop words we can extract these linguistic features.

When creating the Twisty corpus, Twitter users self-reported their MBTI sharing the results of a previous test with a tweet like:

*"Acabo de realizar el Test de Personalidad de Twitter. Mi personalidad es "ENFP". http://www.intelligentelite.com/i/14IcZ3"*

Thus, the MBTI profile of this user is "ENFP" as it appears in the tweet. Consequently, the importance of adding to the stop words dictionary the word "ENFP" (and the other fifteen MBTI contractions) is essential. If this step is skipped, we will achieve a false 100% of accuracy in the classifier because the correlation between the word and the MBTI profile would be 1.

#### 4.5.1.1 POS

Part Of Speech tagging (POS) is the process of marking up a word in a text as corresponding to its grammatical category based on both its definition and its context. The words

to be analyzed are nouns, adjectives, verbs, adverbs, conjunctions, pronouns, adpositions and numerals. In this way, we extract until eight different features corresponding to the eight grammatical categories. NLTK [23] provides an Universal tag set which is shown in Table 4.1. Once, each label is assigned to each word, the total elements of each category is divided by the total elements of all categories. Thus, the weight of each category is extracted. We will use the DictVectorizer function from Scikit-learn to transform the dictionary list from POS into vectors.

| **Tag** | **Meaning** | **English Examples** |
|---|---|---|
| ADJ | adjective | new, good, high, special, big, local |
| ADP | adposition | on, of, at, with, by, into, under |
| ADV | adverb | really, already, still, early, now |
| CONJ | conjunction | and, or, but, if, while, although |
| PRON | pronoun | he, their, her, its, my, I, us |
| VERB | verb | is, say, told, given, playing, would |
| NUM | numeral | twenty-four, fourth, 1991, 14:24 |

Table 4.1: POS tagging examples table

#### 4.5.1.2 TF-IDF

TF-IDF [25] has been used to extract a matrix of vectors representing the relative importance of each word in the dataset. The importance is measured by a numerical value from the normalized Term Frequency (TF) and the Inverse Document Frequency (IDF).

The term frequency of a word in a document is calculated as a simple count of instances the word appears in the document.

The inverse document frequency of the word across a set of documents is calculated by taking the total number of documents, dividing it by the number of documents that contain a word, and calculating the logarithm.

As a result, when a word is very common, the numerical value will be close to “0”. Otherwise, it will approach “1”.

The reason why we have used this feature extractor is under the assumption that each kind of personality tend to use more some words than others.

#### 4.5.1.3 N-grams

These transformers have a behavior similar to TF-IDF. In this case, we measure the importance of the relationships between most likely words. They are basically a set of co-occurring words within a given window.

The implementation is based on the combination between the function CountVectorizer provided by Scikit-learn and TF-IDF. This is, raw data passes through the CountVectorizer which contains as a parameter the N range we want to extract. Thus, we obtain a matrix of vectors with the number of times a N-gram appears in the raw text. Afterwards, TF-IDF is applied to convert that matrix into a TF-IDF matrix.

In this project we have extracted bigrams (N=2), trigrams (N=3) and tetragrams (N=4). Obviously, this would generate disproportionate matrixes which overload our classifier so a selection of the more relevant group of words is needed.

#### 4.5.1.4 Word Embeddings

Word embeddings is the collective name for a set of language modeling and feature learning techniques. Words and phrases are mapped to vectors. Word2Vec is a set of models to produce Word Embeddings. Thus, if Word2Vec has as input a large corpus of text, the output will be a vector space where each word has a corresponding vector.

In this project we have used GSITK library with Word2Vec [26]. In turn, we have used a model of Word Embeddings given by Kaggle[1]. Word vectors are very computationally intensive to train, and the vectors themselves will vary based on the documents or corpora they are trained on. For these reasons, it is often convenient to use word vectors which have been pre-trained rather than training them from scratch for each project. The dataset concerned contain 1,000,653 word embeddings of dimension 300 trained on the Spanish Billion Words Corpus, and it has been trained using Word2Vec.

We have located in a vector space different groups of words that the tool has formed from the corpus. Thus, these words have been organized both syntactically and semantically in a quite optimal way.

[1]https://www.kaggle.com/rtatman/pretrained-word-vectors-for-Spanish

#### 4.5.1.5 Preprocess Twitter features

As mentioned before in Section 3.6.2, this module provided by GSITK [26] let the detection of special characters or words. Thus, we have used it to detect urls, mentions to users, smiles, lolfaces, sadfaces, neutralfaces, hearts, hashtags, repetitions, elongations, words written in capital letters. We have counted the number of detections of each feature and then divided by the number of tweets. Here, we have an example:

Tweet: *"Lo estoy pasandooo GENIAL con @antonio !!! :). Aunque me parece que ÉL NO TANTO :( #viernes"*

Output: *"Lo estoy pasando <elong> genial <allcaps> con <user> ! <repeat><smile>. aunque me parece que él no <allcaps> tanto <allcaps> <sadface> <hastag> viernes"*

#### 4.5.1.6 Lexical features

In addition and following related works, we have added four more features which are lexical:

- **Sentences per tweet:** The average number of sentences per tweet per each user has been added to our collection of features.
- **Words per tweet:** As before, we have calculated the average number of words per tweet per each user.
- **Number of exclamation marks:** We also have calculated the number of exclamation marks that a user writes per tweet.
- **Number of question marks:** Following this path, the number of question marks per tweet has been calculated too.

### 4.5.2 Para-linguistic features

As we mentioned before, these kinds of features refer to the context that accompany the text. Therefore, we will not use the raw text to extract these features. Following related works in Chapter 2 we have introduced to our classifier until four different features:

- **Gender:** Gender of each user has been given by Twisty. This is a key feature in order to predict personality traits because the way of expressing depends on the gender [28].

- **Number of tweets:** We have extracted the number of tweets that users have published throughout his life. We have the hypothesis that extravert people should have more tweets.
- **Followers and followed:** Finally, we have extracted from the tweet JSON files, the number of people the users follow and the number of people who follow the user.

## 4.6 Classification and Evaluation

Finally, the classification and evaluation steps can be carried out. These consist in training several classification models and evaluate them. Scikit-learn provides us multiple automatic learning models based on different algorithms.

At this stage of the proceedings, we have two options: creating a unique classifier to predict the fourth personality traits at the same time or creating four different binary classifiers, one per each personality trait. As other works related in Chapter 2, this second option must be taken in order to achieve the best accuracy. This is so because it is possible that some features are more relevant to know how one processes the information (the pair of intuitive-sensing traits) than how one makes decisions (the pair of thinking-feeling traits), i.e.

The training procedure to follow starts with the split of the data in two sets. On the one hand, we have the training set which will be used to train the model. On the other hand, the testing set will be passed through the trained model. Then, the model's output will be compared to the real output and we will evaluate these results.

However, this procedure makes the model dependent on the training set. Cross Validation emerges as a solution of this problem. It consists of the division of the data into K subsets. Thus, in a first iteration K-1 subsets will train the model, while the remaining other will be used to test. This operation will be repeated K times using different subsets to test. As a result, the model's predictive performance has a better estimation. In our case, generally we have used a K=10 using the toll called K-fold. Fig. 4.8 illustrates an overview of this procedure.

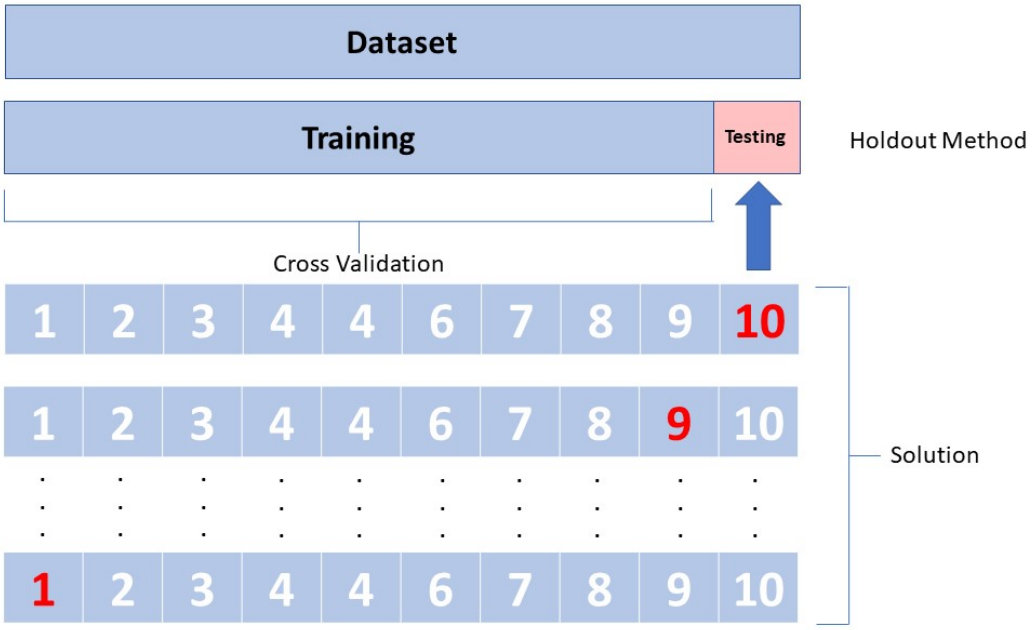


Figure 4.8: Scheme of the Cross Validation method.

As we mentioned before, Scikit-learn provides us several classification models and each of them have different parameters (called hyper-parameters) which can be tuned to obtain the higher accuracy. These parameters cannot be tuned randomly since our success depends on it. GridSearchCV makes an exhaustive search on the values of the parameters specified for an estimator. Basically, this module optimizes the algorithm in order to obtain the maximum success rate.

Now, we will proceed to describe the classifiers tested in this project:

- **Logistic Regression:** This is a supervised learning method which consists in predicting the outcome of a categorical variable based on independent or predictive variables. Thus, it is a statistical method which determines the contribution of various factors to a pair of results.
- **Decision Tree:** In this case, the prediction of the value is made by learning simple decision rules inferred from the extracted features. It uses an if-then rule learned sequentially using the training data one at a time. It can be over-fitted if there are too many branches so the performance would be poor.
- **Random Forest:** It consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction

- **Support Vector Machines:** It creates an optimal plane which categorizes the samples. It is provided by Scikit-learn under the name SVC.

- **K-Nearest Neighbours:** It is a lazy learning algorithm which stores all instances correspond to training data points in n-dimensional space. A sample is classified by the class most common among its k nearest neighbours.

- **Multinomial Naive Bayes:** This classifier implements the naive Bayes algorithm for multinomial models.

- **AdaBoost:** It is a meta-estimator that begins by fitting a classifier on the original dataset. Then it fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases[2].

- **Gradient Boosting:** Group of machine learning algorithms that combine many weak learning models together to create a strong predictive model[3].

In the next section we will see how we can evaluate the performance of our classifiers.

### 4.6.1 Evaluation

In this section we will establish the standard metrics which will let us compare between classifiers. Thus, we will be able to measure the success rate in order to deploy them or continue looking for a better combination of features, classifiers or hyper-parameters. The metrics suggested for this evaluation step are:

- **Confusion matrix:** It is the summary of prediction results on a classification problem. This matrix shows how confused the classification model is. It is composed by:
  - **True Positives (TP):** The classifier correctly predicts with a positive label.
  - **True Negatives (TN):** The classifier correctly predicts with a negative label.
  - **False Positives (FP):** The classifier incorrectly predicts with a positive label.
  - **False Negatives (FN):** The classifier incorrectly predicts with a negative label.

[2]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html
[3]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

- **Accuracy:** It measures the percentage of correct predictions in relation to the total amount of predictions. This will be the most used because its simplicity and utility.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

- **Precision:** It measures the percentage of correct positive predictions in relation to the total of positive predictions.

$$Precision = \frac{TP}{TP + FP}$$

- **Recall:** It measures the percentage of correct positive predictions in relation to the total positive expected values.

$$Recall = \frac{TP}{TP + FN}$$

- **F1 score:** It is the harmonic mean of precision and recall and tries to combine both into a single value.

$$F1Score = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision}$$

### 4.6.2 Feature selection

In this section, we will see which features are the most important and we will try to find which combination between them give us the best success rate. The best results have been achieved with Logistic Regression. This makes sense since the value of the target is categorical in nature, so we have a binary output. Anyway, we will also show the results of other classifers: Random Forest, AdaBoost and Gradient Boosting.

All results shown below have been found using the cross-validation and k-fold. In this way we obtain the best possible results according to the input parameters and an independent model on the training sets.

In order to join the different features which will be implemented we will use a *Pipeline*[4]. This is a chain of independent modules which allows us to group the features we want to add to the model.

As we have developed four different classifiers, we will explain the process of selecting features for one of them. We will see how it can be improved or not with the introduction of the new features. Once this is done, we will present the results of the four classifiers and the final joint result.

[4]https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html

Thus, for the Introvert-Extravert classifier we have started applying TF-IDF as it is widely used in related works. The results are shown in Table 4.2.

| ID Classifier | Features | | Model | Accuracy |
|---|---|---|---|---|
| | Linguistic | Others | | |
| #01 | word (tfidf) | - | Logistic Regression | 0.639 |
| | | | Random Forest | 0.622 |
| | | | AdaBoost | 0.59 |
| | | | Gradient Boosting | 0.634 |

Table 4.2: Results Classifier #01

However, when using procedures which extract many features, using techniques for selecting the best ones is advisable. This is achieved by using the *SelectKBest* function[5]. This is able to select features according to the "k" highest scores and remove the unnecessary, irrelevant and redundant attributes. The results are shown in Table 4.3.

| ID Classifier | Features | | Model | Accuracy |
|---|---|---|---|---|
| | Linguistic | Others | | |
| #02 | word (tfidf) | selectKBest | Logistic Regression | 0.697 |
| | | | Random Forest | 0.670 |
| | | | AdaBoost | 0.651 |
| | | | Gradient Boosting | 0.650 |

Table 4.3: Results Classifier #02

The four models improve with the use of *SelectKBest* so from now on this function will be applied to achieve better results.

POS and bigrams are also widely used in related works. In Table 4.4, the results achieved with these feature extractions are shown. Thus, in general, the models have improved their performance.

[5]https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

| ID Classifier | Features | | Model | Accuracy |
|---|---|---|---|---|
| | **Linguistic** | **Others** | | |
| #03 | word (tfidf)<br>bigrams<br>POS | selectKBest | Logistic Regression | 0.727 |
| | | | Random Forest | 0.674 |
| | | | AdaBoost | 0.650 |
| | | | Gradient Boosting | 0.656 |

Table 4.4: Results Classifier #03

As these results obtained by linguistic features are not yet good enough, we have decided to implement some paralinguistic onSe. Among them we have the number of tweets, the gender and the *pprocess* features explained in Section 4.5.1.5. Among these *pprocess* features we have implemented the number of: urls, mentions to users, smiles, lolfaces, sadfaces, neutralfaces, hashtags, repetitions, elongations, words written in capital letters on average per tweet per user. The obtained results are shown in Table 4.5

| ID Classifier | Features | | Model | Accuracy |
|---|---|---|---|---|
| | **Linguistic** | **Others** | | |
| #04 | word (tfidf)<br>bigrams<br>POS | selectKBest<br>n_tweets<br>gender<br>pprocess | Logistic Regression | 0.780 |
| | | | Random Forest | 0.663 |
| | | | AdaBoost | 0.602 |
| | | | Gradient Boosting | 0.618 |

Table 4.5: Results Classifier #04

Comparing these results with the previous classifier we can conclude that we have improved our Logistic Regression model, while the others have not. We will focus now in trying to improve this Logistic Regression model to obtain a better accuracy by adding some other paralinguistic features: number of followers and followed, sentences per tweet, words per tweet and the number of question and exclamation marks used in average. Results are shown in Table 4.6.

| ID | Features | | | | Model | Accuracy |
|---|---|---|---|---|---|---|
| | **Linguistic** | **Others** | | | | |
| #05 | word (tfidf)<br>bigrams<br>POS | selectKBest<br>n_tweets<br>gender<br>pprocess | followers<br>followed<br>sentXtweet | wordXtweet<br>n_exclamation<br>n_question | Logistic Regression | 0.785 |
| | | | | | Random Forest | 0.664 |
| | | | | | AdaBoost | 0.602 |
| | | | | | Gradient Boosting | 0.618 |

Table 4.6: Results Classifier #05

Now, we have reached a good level of accuracy in the Logistic Regression model. Nevertheless, it can be improved. Some related works have also applied trigrams as well as unigrams and bigrams. Table 4.7 shows the achieved results adding these features.

| ID | Features | | | | Model | Accuracy |
|---|---|---|---|---|---|---|
| | **Linguistic** | **Others** | | | | |
| #06 | word (tfidf)<br>bigrams<br>POS<br>trigrams | selectKBest<br>n_tweets<br>gender<br>pprocess | followers<br>followed<br>sentXtweet | wordXtweet<br>n_exclamation<br>n_question | **Logistic Regression** | **0.791** |
| | | | | | Random Forest | 0.663 |
| | | | | | AdaBoost | 0.624 |
| | | | | | Gradient Boosting | 0.601 |

Table 4.7: Results Classifier #06

The Logistic Regression model has been improved again to an accuracy of 0.791. Finally we have decided to try to implement word embedding features as it is shown in Table 4.8. However, in this case we have not improved the Logistic Regression model. So, we can conclude that our best classifier is the sixth one with Logistic Regression in Table 4.7.

| ID | Features | | | | Model | Accuracy |
|---|---|---|---|---|---|---|
| | **Linguistic** | **Others** | | | | |
| #07 | word (tfidf)<br>bigrams<br>POS<br>trigrams<br>word embedding | selectKBest<br>n_tweets<br>gender<br>pprocess | followers<br>followed<br>sentXtweet | wordXtweet<br>n_exclamation<br>n_question | Logistic Regression | 0.763 |
| | | | | | Random Forest | 0.664 |
| | | | | | AdaBoost | 0.639 |
| | | | | | Gradient Boosting | 0.629 |

Table 4.8: Results Classifier #07

Once, we have reached the maximum level of accuracy, we must follow the same procedure for the other three classifiers, realizing when the model improves with the introduction of the new features and when it does not. Thus, it is possible that some features may be decisive for one classifier while for another one they are totally irrelevant. Table 4.9 shows the best scores and the features implemented of the four classifier: Introvert-Extravert, Sensing-iNtuitive, Thinking-Feeling and Judging-Perceiving.

As a result, we can conclude that by using different combinations of features for each classifier, we achieve the highest accuracy for each of them. All classifiers obtain the best results using the Logistic Regression model. Thus, the best accuracy is achieved when discerning between Judging and Perceiving. Nevertheless, the difference between the worst and the best is only 0.03. Now we can calculate how accurate the four MBTI dimensions are. Thus, these four classifiers have a joint hit probability of 42%. This result is quite good, since if we decided randomly the MBTI dimensions we would have a probability of success of 6.25%. On the other hand, if we allow the system to fail one of the dimensions, the hit probability would be more than 50%.

| **Classifier** | **Features** | | | | **Model** | **Accuracy** |
|---|---|---|---|---|---|---|
| | **Linguistic** | **Others** | | | | |
| **Introvert - Extravert** | word (tfidf) bigrams POS trigrams | selectKBest n_tweets gender pprocess | followers followed sentXtweet | wordXtweet n_exclamation n_question | **Logistic Regression** | **0.791** |
| | | | | | Random Forest | 0.663 |
| | | | | | AdaBoost | 0.624 |
| | | | | | Gradient Boosting | 0.601 |
| **Sensing - iNtuitive** | word (tfidf) bigrams POS trigrams | selectKBest n_tweets gender pprocess | followers followed sentXtweet | wordXtweet n_exclamation n_question | **Logistic Regression** | **0.796** |
| | | | | | Random Forest | 0.666 |
| | | | | | AdaBoost | 0.545 |
| | | | | | Gradient Boosting | 0.639 |
| **Thinking - Feeling** | word (tfidf) bigrams POS | selectKBest n_tweets gender pprocess | - | - | **Logistic Regression** | **0.813** |
| | | | | | Random Forest | 0.646 |
| | | | | | AdaBoost | 0.616 |
| | | | | | Gradient Boosting | 0.611 |
| **Judging - Perceiving** | word (tfidf) bigrams POS | selectKBest n_tweets gender pprocess | - | - | **Logistic Regression** | **0.822** |
| | | | | | Random Forest | 0.637 |
| | | | | | AdaBoost | 0.567 |
| | | | | | Gradient Boosting | 0.628 |

Table 4.9: Results four classifiers

CHAPTER 5

# HR Personality Service

## 5.1 Introduction

In this chapter we will describe the selected use case: A system capable of predicting personality for the Human Resources departments of companies.

Personality tests that assess traits relevant to job performance have been shown to be effective predictors of subsequent job performance [29]. Thus, personality predicts how a person will work: diligently, intelligently, cheerfully, and cooperatively.

In this way, recruiters use personality tests to enhance their decision-making about the potential of applicants. No recruiter wants to spend time on a low potential applicant. The more information available, the more efficient and accurate a recruiter can be with referrals[1]. For that reason the Society for Human Resource Management Foundation (SHRM) recommends applying personality tests to recruit new applicants [29].

However, this process can be complex, cumbersome and costly both in terms of time and money. Also, there is the possibility that the user answers without sincerity trying to give the profile that the company is looking for. For that reason, recruiters also look to

[1] https://www.hoganassessments.co.uk/media/1695/why-is-personality-important-to-recruitment.pdf

applicants on social networks, so they can gather more information.

This is where our system arrives with a solution for recruiters: predicting personality profiles on Twitter automatically and instantaneously thanks for machine learning algorithms. Thus, based on the classification model developed in Chapter 4, our system is able to predict the four MBTI dimensions of an user with a high accuracy.

In the next sections, we will explain the procedure to achieve this.

## 5.2 Web application

In this section we will explain the procedure followed for the creation of our system. Its aim is to develop a platform that allows recruiters to detect the personality of applicants. It consists in a web application based on Flask framework[2].

Flask is a micro web framework written in Python which does not require particular tools or libraries. This web application can come in the form of web pages, blogs, or even an extensive web-based calendar app or a commercial site. It is considered fitter into the Python guidelines because the code is more explicit. This has been the main reason to choose this framework over others.

At the same time, we have used Bootstrap[3]. According to the official website, Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive on the web. Essentially, Bootstrap focuses on the front-end web development and makes developing a modern website easier. It enables developers and designers to quickly build fully responsive websites.

Our web application resides on a networked file server: Heroku. This is a cloud platform as a service (PaaS) that lets companies build, deliver, monitor and scale apps[4]. It supports a widely list of programming languages such as Java, Node.js, Scala, Clojure, Python, PHP, Ruby, etc.

The web application layouts can be described as simple, intuitive and modern. We have used two layouts: the first one, where we introduce the nickname of the Twitter user whose personality we want to know; and the second one, which will provide us with the results.

The system's diagram is shown in Fig. 5.1. Below, each module will be detailed.

[2]https://www.fullstackpython.com/flask.html
[3]https://getbootstrap.com/
[4]https://www.heroku.com/what

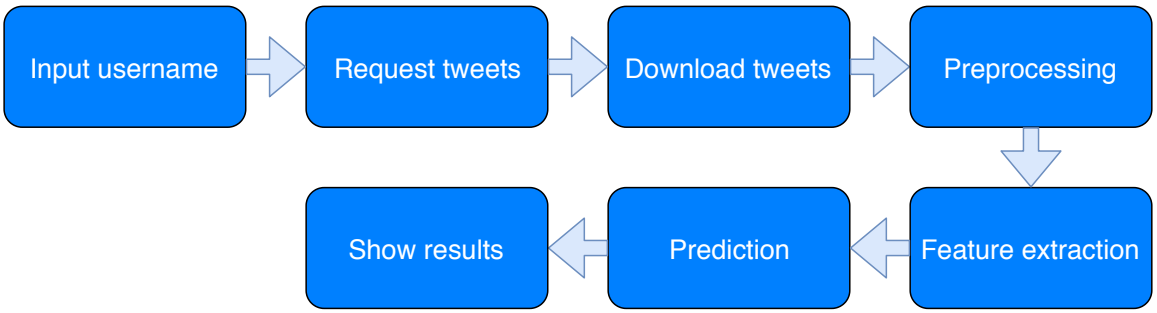


Figure 5.1: Functional diagram of HR Personality System.

First of all, the recruiter must introduce the user's nickname. We must highlight the characteristics which the user must comply with: the account must be public in order to be able to request the corresponding tweets and it must contain at least 200 own tweets in order to ensure a good accuracy. In addition, users must be the only ones that control their accounts, this is, some important people contract agencies to tweet in their place. In these cases, the predicted personality would not be in line with the reality.

Then, through the API Twitter, until 600 tweets from the user account will be requested and downloaded in a single CSV document. The preprocessing step will be carried out. With regard to the feature extraction, we must distinguish between paralinguistic and linguistic features. The first of them are taken from JSON information downloaded from Twitter and must be extracted separately. The linguistic ones are taken from the timeline text through the four different classification models. *Pickle* module is used in order to load these models[5].

As a result, we obtain four outputs with the predictions of the four MBTI dimensions. Finally, the results' layout shows the calculated personality and additional information such as the suggested careers and the description of the user's personality. Fig. 5.2 and Fig. 5.3 show an example of the interfaces for one user.

[5]https://docs.python.org/3/library/pickle.html

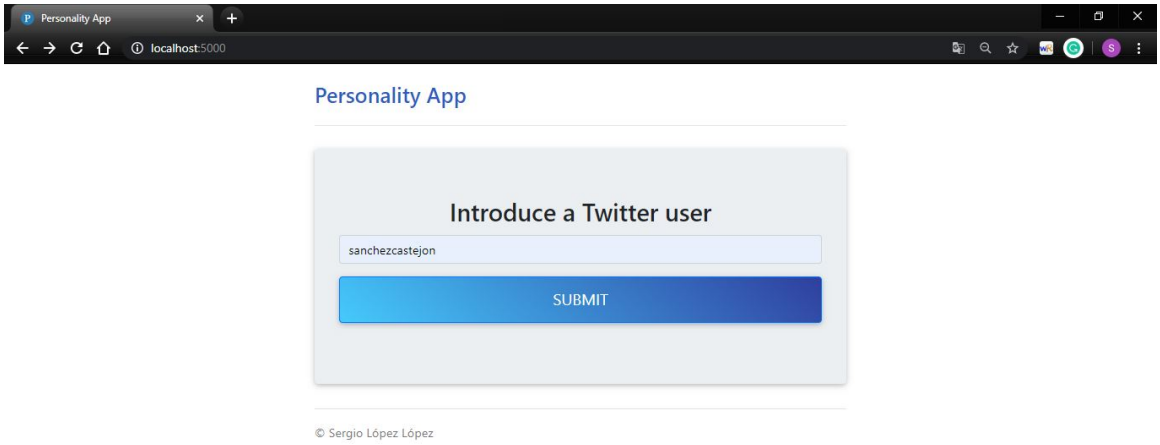


Figure 5.2: HR Personality System main interface.

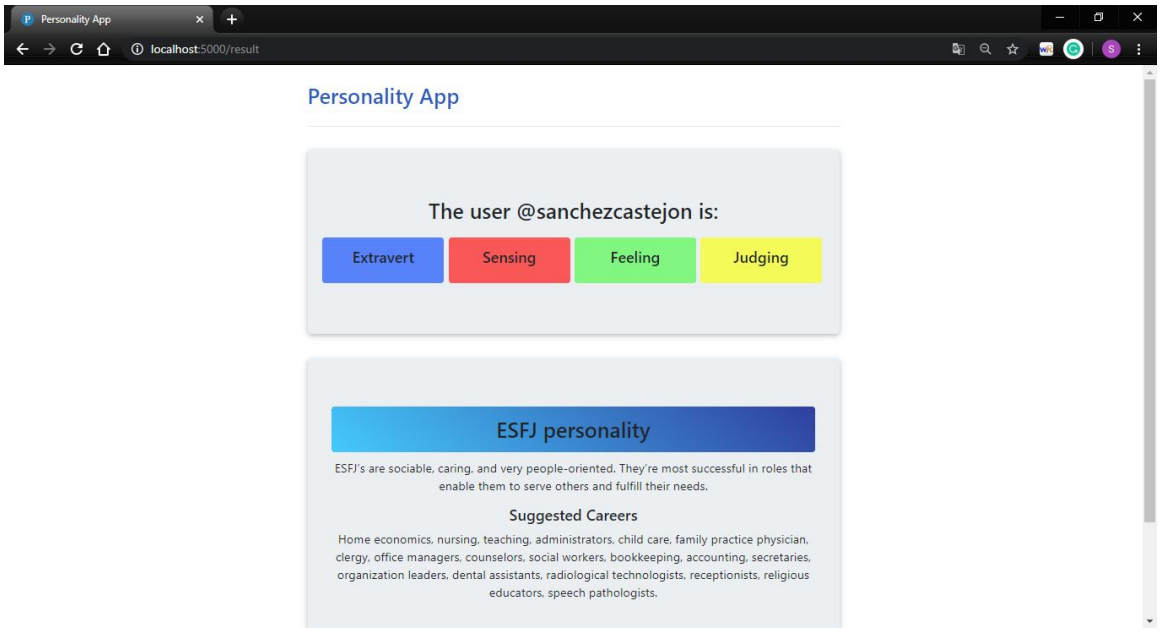


Figure 5.3: HR Personality System results interface.

CHAPTER 6

# Conclusions and future work

## 6.1 Introduction

In this chapter we will describe the conclusions extracted from this project. We will also talk about the goals achieved and the problems faced in achieving these goals as well as the solutions taken to solve them. Finally, we will present some suggestions about future work.

## 6.2 Conclusions

In this section we will explain the conclusions we have reached at the end of this project.

The main goal of this project was to build a machine learning algorithm able to predict personality using stylometric features. To achieve this, we started from the Twisty dataset. As we previously mentioned in Section 2.1, the authors achieved 61.09%, 60.23%, 59.35% and 55.60% of accuracy for the four MBTI dimensions. We have been able to improve substantially these predictions through the use of more extracted features. This has been achieved thanks for an exhaustive search in related works.

Table 6.1 shows a comparison between the related works mentioned in Chapter 2, which

try to predict MBTI dimensions, and our final results. In general, we can conclude that our classifier works far better when Spanish text is analyzed. We can also conclude that in general it is easier to analyze English texts than Spanish ones. This, maybe, is due to the structure of the language itself.

| | **I-E** | **S-N** | **T-F** | **J-P** | **Language** | **From** |
|---|---|---|---|---|---|---|
| 2.1 | 61.09% | 60.23% | 59.35% | 55.60% | Spanish | Tweets |
| 2.1.2 | 82.8% | 79.2% | 67.2% | 74.8% | English | Blogs |
| 2.1.3 | 82.05% | 88.38% | 80.57% | 78.26% | English | Tweets |
| 2.3 | 95% | 75% | 75% | 75% | Spanish | YouTube |
| 2.5 | 82% | 82% | 84% | 79% | English | Blogs |
| **This work** | **79.1%** | **79.6%** | **81.3%** | **82.2%** | **Spanish** | **Tweets** |

Table 6.1: Comparison of results with related work in Chapter 2

Between the main features which are useful to determine the personality, we find the n-grams and bigrams which play a vital role. Furthermore, *pprocess_features* and the number of tweets provide relevant improvements to our classifier. The number of followers, followed, sentences per tweet, words per tweet, exclamation marks and question marks play more important roles at the hour of predicting between Introvert-Extravert and Sensitive-iNtuitive than between Thinking-Feeling and Judging-Perceiving. We also want to underline the importance of using *SelectKBest* in order to obtain better results when working with so many features.

With regard to classifier models we must highlight the performance of Logistic Regression, becoming by far, the best model in our system.

Finally, we must undersocre the classifier's performance developed in this project. This classifier has performed outstandingly well with a dataset which personality traits has not been extracted from text itself. This is, MBTI user's values were taken from self-reporting while tweets from the user account.

## 6.3 Achieved goals

In this section we will explain the goals that we have accomplished in this project.

- **Development of a personality user classifier.** Thanks for the use of NLP techniques and other features extraction techniques, we have been able to develop a personality classifier. We have created a model which extracts a huge amount of features from the information given by Twitter user and classify the user depending of the four MBTI dimensions.
- **Overcome the previous related work.** As we mentioned in the previous section, we have improved the classification procedure followed by Twisty with around a 20% of accuracy. Likewise, we have achieved similar or even better results than the other works in 2.
- **Implemented system as a service.** In order to show the system functionalities we have deployed a HR personality service which can be used by any Human Resources department interested in recruiting. Thus, recruiters can predict the personality traits of the applicants through an application web.

## 6.4 Problems faced

While carrying out the different steps included in this project we have faced some problems, and overcame the difficulties to achieve the goals mentioned before.

- **Unfamiliarity with the technologies used in this project.** One of the first problems encountered was the ignorance about most of technologies used in this project. In this way, once the state of art was realized and the technologies to use were decided, we had to learn how to implement them.
- **Features did not improve our model as they should.** Compared to other related works, once we applied any feature extraction, our model did not improve as much. With less features, some related works achieved higher scores. The solution was to extract more features than usual.

## 6.5 Future work

Finally, in this section we will explain some improvements which could be implemented in this project.

- **Adding more features.** Although we have applied many features, it would be

worthwhile to try the extraction of other features, as for example, sentimental analysis. Maybe, the system could improve a bit.

- **Adding other social networks as a source of information.** There are other social networks such as Instagram or Facebook which are widely used in Spain too. In fact, Instagram is one of the most used social networks nowadays between almost all groups of population. The idea would be to complement our classifier with data from these social networks in order to improve the model.

- **Adding new languages.** As in this project we have only used a Spanish dataset, it would be interesting to extend the project whit other languages, such as English, German, Italian, etc.

APPENDIX A

# Impact of this project

In this appendix we will analyze the impacts related to the realization of this project from a social, economic, environmental and ethical point of view.

## A.1 Social impact

Nowadays, social networks are the most used online platforms on the Internet. The great amount of content, which is shared, is analyzed by our project in order to draw conclusions about the population personality traits.

The purpose of this project is to provide an useful tool to human resources departments which want to know the personality of the new applicants. It makes this procedure simpler and quicker than actual surveys. Therefore, the main target is mostly companies which want to recruit more employees.

The tool and the study developed in this project will give researchers on this subject a start from the resources created and from the conclusions made.

## A.2 Economic impact

In this section, the possible economic impacts that companies may experience when using our system will be described. The use of this system can improve efficiency in recruiting new workers. Currently, recruiters develop surveys and study them to get an idea about the personality of the applicants. This leads to a waste of time, effort and, therefore, money. Thanks to our system, companies will be able to know with a greater certainty the personality of their applicants easily. In this way, a considerable reduction of the costs can be achieved.

Researchers can rely on the tool developed and the conclusions made. This would reduce research time and costs for them.

## A.3 Environmental impact

This section aims to define the main environmental impact of the development of our system.

To make possible the development of our system, it is necessary to have the required equipment. This includes computers, servers, and other computer materials which need energy to run. This could produce a huge charge on our electricity networks and contributing to greenhouse gas emission. To this consumption must be added the energy needed for the cooling system.

In addition, our system is deployed in a Heroku server, the high consumption of the server must be taken into account too.

For these reasons, technologies as Machine Learning or Big Data require a transition to the use of renewable energies to reduce the environmental impact.

## A.4 Ethical Implications

In this section we will describe the possible ethical impacts which the usage of our project may produce.

The first ethical problem involves the automation of a part of recruitment. Thanks to our system, companies reduce costs, by reducing the human resources needed to perform the tasks that our tool completes. The implementation of this system could eliminate some human resource jobs.

The second ethical problem we face is the impact related with the privacy. As we use Twitter data, we have to be careful to comply with the privacy policies of this company. Thus, tweets from private accounts have not been used in this project.

APPENDIX B

# Economic budget

In this appendix we are going to make an adequate economic budget for the realization of this project. The main parts of this budget will be explained in the following sections.

## B.1 Project structure

First of all, the project structure will be presented. The hours spent developing the tasks presented in Section 1.3 are shown in Table B.1. Thus, the total duration of this project has been about 570 hours. We have taken into account 5 hours worked per day and six days per week. In this way, this work has been split in 19 weeks. The project has been carried by mainly one person with high knowledge in programming.

| **Activity** | **Duration (days)** |
|---|---|
| Learning the tools and techniques available | 15 |
| Researching of related works. | 8 |
| Analysis and preprocessing of the data | 7 |
| Features extraction | 20 |
| Study of automatic learning algorithms which best fit this case | 5 |
| Software development, experimentation and feature selection | 20 |
| Analysis of the results obtained | 5 |
| Development of the web application | 10 |
| Report writing | 25 |
| **Total** | **115** |

Table B.1: Project structure by activity

## B.2 Physical resources

This section presents the physical resources needed to develop this project. These resources can be split in two types: software and hardware.

- **Software.** This part includes licences of the software tools necessary for the development of the system carried out in this project. However, this one has been accomplished with Open Source Software (OSS), which means that there is no need to pay for the use of any of the technologies utilized.
- **Hardware.** The hardware necessary for the development of this project has been mainly a computer and a big data cluster.

  The main computer's characteristics are:

  - *RAM:* 16 GB
  - *CPU:* Intel Core i7, 2,5 Ghz x 4
  - *Hard disk:* 500 GB

Regarding the big data cluster, we must highlight:

- DELL PowerEdge R320
- ntel® Xeon® E5-2430 v2 (2,50GHz, 6 cores with hyperthreading, 15M cache, QPI 7,2GT/s, Turbo) 80W
- 4x32GB RDIMM, 1333 MHz
- 3x3TB, SATA, 7,2k rpm, 3,5" hot-swappable. RAID+LVM. About 3 TB are currently shared by all nodes using glusterfs

The approximate cost of a computer with these characteristics is about 1,000€, while the approximate cost of the big data cluster is 2,600€.

In this way **the total physical resource cost is 3,600€.** Amortization has not been calculated so it is possible that the final cost will be lower.

## B.3 Human Resources

In this section we will cover the part of the budget that consists of the cost of the human resources. As previously mentioned, the project has been carried out by one person with an estimated salary of 8€ per hour since the average salary of a person with knowledge of NLP and ML is 24,000€ per year.

As up to 570 hours have been spent on this work, the cost in human resources is 4,560€. However, the Spanish legislation must be taken into account and it establishes that a company must pay an extra 32.6% of the employee's salary. **So the final cost in human resources is around 6,045€.**

## B.4 Taxes

If the final product were sold to a company which is interested in its acquisition, the sale must be subject to a tax of 15% of the product's price.

## B.5 Conclusion

This project has a **duration of 570 hours** with a **total cost of 9,645€**.

# Bibliography


[1] Nikola Ljubešić, Darja Fišer, and Tomaž Erjavec. Language-independent gender prediction on twitter. In *Proceedings of the Second Workshop on NLP and Computational Social Science*, pages 1–6, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[2] Matej Gjurković and Jan Šnajder. Reddit: A gold mine for personality prediction. In *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*, pages 87–97, New Orleans, Louisiana, USA, June 2018. Association for Computational Linguistics.

[3] Ana Carolina E. S. Lima and Leandro Nunes de Castro. Tecla: A temperament and psychological type prediction framework from twitter data. *PLOS ONE*, 14(3):1–18, 03 2019.

[4] Ben Verhoeven, Walter Daelemans, and Barbara Plank. Twisty: a multilingual twitter stylometry corpus for gender and personality profiling. In *Proceedings of the 10th Annual Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia, 05/2016 2016. ELRA, ELRA.

[5] James Pennebaker, Ryan Boyd, Kayla Jordan, and Kate Blackburn. The development and psychometric properties of liwc2015. *The University of Texas at Austin*, 09 2015.

[6] Barbara Plank and Dirk Hovy. Personality traits on twitter-or-how to get 1,500 personality tests in a week. In *The 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA), EMNLP 2015*, 2015.

[7] Wilson M. The mrc psycholinguistic database: Machine readable dictionary. behavioural research methods, instruments and computers. In *Springer*, 1988.

[8] Manning CD Toutanova K. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. proceedings of the 2000 joint sigdat conference on empirical methods in natural language processing and very large corpora: Held in conjunction with the 38th annual meeting of the association for computational linguistics—volume 13 (pp. 63–70). In *Stagger*, Hong Kong: Association for Computational Linguistics, 2000.

[9] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA, 1998.

[10] Ana Carolina E.S. Lima, Leandro Nunes de Castro, and Juan M. Corchado. A polarity analysis framework for twitter messages. *Applied Mathematics and Computation*, 270:756 – 767, 2015.

[11] Anzhela Zhusupova. Characterizing the personality of twitter users based on their timeline information. *ISTCE*, 2016.

[12] J Oberlander and AJ Gill. Individual differences and implicit language: personality, parts-of-speech and pervasiveness. In *26th Annual Conference of the Cognitive-Science-Society*, pages 1035 – 1040, January 2004.

[13] Carlos Alonso Aguilar. Design and development of a personality prediction system based on mobile-phone based metrics. Trabajo fin de titulación (tfg), Universidad Politécnica de Madrid, ETSI Telecomunicación, 2017.

[14] Diego Benito-Sánchez. Design and Development of a Personality Traits Classifier based on Machine Learning Techniques. Tfg, ETSI Telecomunicación, Universidad Politécnica de Madrid, June 2017.

[15] J. Fernando Sánchez-Rada, Carlos A. Iglesias, Ignacio Corcuera-Platas, and Oscar Araque. Senpy: A Pragmatic Linked Sentiment Analysis Framework. In *Proceedings DSAA 2016 Special Track on Emotion and Sentiment in Intelligent Systems and Big Social Data Analysis (SentISData)*, pages 735–742, Montreal, Canada, October 2016. IEEE.

[16] Adriana Mansilla and Fausto Jacques-García. Un estudio comparativo entre algoritmos de aprendizaje automático orientados a la clasificación de personalidad para selección de personal en un contexto hispano. *Revista Electrónica de Divulgación de la Investigación Vol. 16*, 12 2018.

[17] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[18] Travis Oliphant. NumPy: A guide to NumPy. USA: Trelgol Publishing, 2006–.

[19] W. McKinney. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython.* O'Reilly Media, 2012.

[20] J.D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science and Engineering*, 9:90–95, 06 2007.

[21] Michael Waskom, Olga Botvinnik, drewokane, Paul Hobson, David, Yaroslav Halchenko, Saulius Lukauskas, John B. Cole, Jordi Warmenhoven, Julian de Ruiter, Stephan Hoyer, Jake Vanderplas, Santi Villalba, Gero Kunter, Eric Quintero, Marcel Martin, Alistair Miles, Kyle Meyer, Tom Augspurger, Tal Yarkoni, Pete Bachant, Mike Williams, Constantine Evans, Clark Fitzgerald, Brian, Daniel Wehner, Gregory Hitz, Erik Ziegler, Adel Qalieh, and Antony Lee. *seaborn: v0.7.1 (June 2016).* Zenodo, June 2016.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[23] Steven Bird and Edward Loper. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[24] Grzegorz Kondrak. N-gram similarity and distance. In *Proceedings of the 12th International Conference on String Processing and Information Retrieval*, SPIRE'05, pages 115–126, Berlin, Heidelberg, 2005. Springer-Verlag.

[25] Juan Ramos. Using tf-idf to determine word relevance in document queries, 1999.

[26] Oscar Araque, Ignacio Corcuera-Platas, J. Fernando Sánchez-Rada, and Carlos A. Iglesias. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, 77:236 – 246, 2017.

[27] Kevin Makice. *Twitter API: Up and Running Learn How to Build Applications with the Twitter API.* O'Reilly Media, Inc., 1st edition, 2009.

[28] Alicia Skinner Cook, Janet J. Fritz, Barbara L. McCornack, and Cris Visperas. Early gender differences in the functional usage of language. *Sex Roles*, 12(9):909–915, 1985.

[29] E.D. Pulakos and SHRM Foundation. *Selection Assessment Methods: A Guide to Implementing Formal Assessments to Build a High-quality Workforce.* SHRM Foundations, 2005.

[30] Anthony Ma. *Neural Networks in Predicting Myers Brigg Personality Type From Writing Style.* Stanford, 2017.