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 SYSTEM FOR SLEEP DISORDER CHARACTERIZATION USING SOCIAL MEDIA MINING

DANIEL SUÁREZ SOUTO

2018

TRABAJO FIN DE GRADO

Título: Diseño y desarrollo de un sistema de caracterización de trastornos del sueño mediante Social Media Mining.

Título (inglés): Design and development of a system for sleep disorder characterization using Social Media Mining.

Autor: Daniel Suárez Souto

Tutor: Carlos A. Iglesias Fernández

Departamento: Ingeniería de Sistemas Telemáticos

MIEMBROS DEL TRIBUNAL CALIFICADOR

Presidente:

Vocal:

Secretario:

Suplente:

FECHA DE LECTURA:

CALIFICACIÓN:

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

Departamento de Ingeniería de Sistemas Telemáticos
Grupo de Sistemas Inteligentes

TRABAJO FIN DE GRADO

DESIGN AND DEVELOPMENT OF A SYSTEM FOR SLEEP DISORDER CHARACTERIZATION USING SOCIAL MEDIA MINING.

Daniel Suárez Souto

Junio de 2018

Resumen

El número de personas que sufren alguno de los trastornos del sueño ha aumentado en los últimos años. Sin embargo, se estima que la mayoria de estas personas no siguen ningún tipo de tratamiento para intentar solucionarlo. Por otro lado, las redes sociales se han convertido en plataformas utilizadas por millones de usuarios que interactúan unos con otros comunicándose entre sí. Esto hace, que también se consideren como una fuente valiosa de lo que se conoce como *Social Data*, que es toda información que los usuarios de redes sociales comparten públicamente, incluyendo metadatos como la ubicación del usuario. En este proyecto hemos analizado información compartida por usuarios hispano hablantes sobre el insomnio en la red social Twitter.

Nuestro objetivo ha sido desarrollar un clasificador mediante Machine Learning que es capaz de clasificar mensajes relacionados con insomnio y un segundo clasificador que es capaz de catalogar dichos mensajes en 5 temas diferentes según la información que contienen.

Para desarrollar estos clasificadores, construimos un dataset con tweets que contuvieran la palabra ”insomnio” que se publicaran entre los días 14 de Diciembre y 4 de Enero de 2018. A partir de este dataset, realizamos un estudio geográfico del que concluimos que los países hispano hablantes con más tweets sobre el insomnio son Argentina, México y España; concretamente de España hemos podido estimar que aproximadamente el 1.21% de los usuarios de este país han escrito en algún momento sobre el insomnio. Otra conclusión a la que llegamos con este dataset es que la gran mayoría de los usuarios persentan el síntoma *Dificultad al inicio del sueño*, entre los definidos por el ICSD-3.

El algoritmo que mejores resultados nos ha dado a la hora de entrenar el clasificador de insomnio y el clasificador de temas ha sido *Logistic Regression* con una *Accuracy* y una *F1 score* de 0.84, 0.82 y 0.75, 0.72 respectivamente.

Por último desarrollamos un servicio de monitorización sobre el insomnio que permite visualizar el análisis de temas,sentimientos y emociones realizado a través de Senpy de los tweets capturados.

Palabras clave: Insomnio, Machine Learning, Big Data, NLP, Python, Emociones, Sentimientos, Twitter, Análisis

Abstract

The catalogue of different sleep disorders is one of the main problems that medicine faces today. The percentage of people suffering from any of these disorders is 31% in Western Europe, 56% in the USA and 23% in Japan. However, it is estimated that only some these people are following some form of medical treatment. Nowadays, social networks have become platforms used by millions of users who communicate with each other. This also makes them a valuable source of what is known as *Social Data*, which is all information that social network users share publicly, including metadata such as user location, spoken language, biographical data and/or shared links. In this project we have analysed information shared by Spanish-speaking users about insomnia on the social network Twitter.

Our objective has been to develop a machine learning classifier that is capable of classifying messages related to insomnia and a second classifier that is capable of classifying these messages into 5 different themes according to the type of information they contain.

To develop these classifiers, we built a dataset with tweets containing the word ”insomnia” to be published between December 14 and January 4, 2018. From this dataset, we conducted a geographical study of which we concluded that the Spanish-speaking countries with the most tweets on insomnia are Argentina, Mexico and Spain, specifically Spain, with the data collected, we have been able to estimate that approximately 1.21% of users in this country have ever written about insomnia. Another conclusion we came to with this dataset is that there is a big difference in the proportion of users who have the symptom of *Difficulty at the beginning of sleep* compared to the other two symptoms of *Short sleep duration* and *Difficulty sleeping and low energy during the day*, all defined by the ICSD-3.

The algorithm that gave us the best results when training the insomnia classifier and the theme classifier was *Logistic Regression* with a *Accuracy* and a *F1 score* of 0.84, 0.82 and 0.75, 0.72 respectively.

Finally we developed a monitoring service on insomnia that allows you to visualize the analysis of themes, sentiments and emotions made through Senpy of the captured tweets.

Keywords: Insomnia, Machine Learning, Big Data, Python, NLP, Sentiments, Emotions, Twitter, Analysis

Agradecimientos

Quiero dar las gracias a todas la personas que me han ayudado a lo largo de estos últimos años, en especial a mis padres, Rubén y Pilar, y mis hermanos, Jorge y Ana, por su apoyo incondicional durante toda mi vida.

También quiero agradecer a mi tutor Carlos Ángel Iglesias y a mis compañeros del GSI por orientarme y ayudarme durante la realización de este proyecto.

Muchas gracias a todos.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Context

The catalog of different sleep disorders is part of the main problems that must be faced by medicine today. The percentage of people suffering from some of these disorders is 31% in Western Europe, 56% in the USA and 23% in Japan [10].

A study by DOPPS (Dialysis Outcomes and Practice Patterns) concluded that there are a number of common clinical features associated with people who sleep a few hours of sleep such as increased body mass index, pain, coronary artery disease, congestive heart failure, diabetes, lung disease, psychiatric disorders, peripheral arterial disease, depression, and pruritus [4].

However, it is estimated that only some these people follow some kind of medical treatment for these disorders. This fact makes necessary to create techniques for medical professionals to predict and diagnose sleep disorders in an early way and to be able to advise healthy sleep routines and therapies.

Nowadays, social networks have become spaces where people communicate their daily activities, concerns and problems to others [5]. Recently, many studies have emerged in which social networks were used as a data source produced by users to attempt to draw conclusions on health issues [8].

In our case, we are interested in determining if it is possible to characterize potential people to suffer insomnia through the information of the messages posted on a social network.

We are also interested in determining the proportion of patients who suffer from each of the different symptoms of insomnia, allowing us to carry out a more extensive study of the phenomenon of insomnia. These symptoms have been defined in the third edition of the International Criteria for Sleep Disorders (ICSD-3) and these are: difficulty beginning sleep, short sleep duration, poor sleep quality.

The social network that we are going to use to analyse information from users is Twitter. Twitter allows its users to publish messages of a maximum of 280 characters with the possibility of accompanying them with a photo, video, and/or link.

Recently, researchers have used Twitter to draw conclusions about the personality of people with possible suicidal tendencies [1], people whose goal is to lose weight [17], influence of political candidates on voters [9], to explore user features to attract new customers to companies [16], and many others.

For all of this, we believe that it is possible to study a phenomenon such as insomnia through a platform like Twitter through a process that will be explained throughout this document.

1.2 Project goals

The main objective of this project is to make a classifier using machine learning techniques capable of classifying messages from users suffering from insomnia. These messages are taken from the social network Twitter.

The study of this project will be based on tweets and Spanish-speaking users posting about insomnia.

Another objective is to carry out a study of the insomnia phenomenon, analysing the common characteristics of users who suffer from insomnia, as well as carrying out a geographical study of the subject and analysing the importance of this in Spain.

The results obtained from these objectives will help us to carry out the deployment of a service that allows us to visualize the characteristics of the tweets related to insomnia.

1.3 Project Tasks

In order to achieve these objectives explained in the previous section, the following tasks have been carried out during the project:

- Study the state of the art in relation to Natural Language Processing and machine learning technologies.
- Collection of tweets related to insomnia in order to create a dataset.
- Geo location study of the tweets that form the dataset.
- Development of a first classifier capable of selecting tweets from a user who reports insomnia following the next steps:
  - Development of a preprocessing technique for each message to be analysed. In this preprocessing process, the characteristics that we want the classifier to learn must be extracted.
  - Dividing the processed data in two parts, with one part we will train the classifier and with the second part we will check if it performs its function satisfactorily.
  - Analysis of the results of the different machine learning algorithms.
- Development of a second classifier capable of classify tweets into a series of themes.
- Analysis of characteristic patterns of users with insomnia.
- Deployment of an insomnia monitoring service.

1.4 Structure of this document

In this section we provide a brief overview of the chapters included in this document. The structure is the following:

Chapter 1 explains the context in which this project is developed. Moreover, it describes the main goals to achieve in this project.

Chapter 2 provides a description of the main technologies on which this project relies.

Chapter 3 describes the architecture of this project, including the design phase and implementation details.

Chapter 4 describes the architecture of a visualization service and its systems detailed.

Chapter 5 discusses the conclusions drawn from this project, problems faced and suggestions for a future work.

CHAPTER 2

Enabling Technologies

This chapter aims to illustrate the technologies used throughout this project. Specifically, after an introduction point on which we will deal with the scientific fields on which this project is based, we will talk in each of the points about the libraries, tools and technologies used throughout the project.

2.1 Introduction

As the objective of this project involves the development of a tool capable of characterizing and classifying different messages from Twitter users, the technologies used throughout the project can be grouped into two major fields of artificial intelligence: machine learning and NLP (Neuro-linguistic Programming).

Machine Learning is the subfield of artificial intelligence whose objective is to develop techniques that allow computers to learn. This learning process can take different forms: supervised, unsupervised and semi-supervised [18].

As we have already mentioned, our problem is a classification problem, so our learning process is supervised, which consists of training the algorithm with a series of classified documents so that it can produce a function that establishes a correspondence between the characteristics of each document and the desired classification. The output of this function determines the classification of a new document that the system is trying to classify [23].

However, these algorithms work through a series of accounting characteristics that define each of the samples (in our case, these samples are texts) to be classified. The field in charge of extracting these characteristics from the texts is NLP.

NLP is a field of computer science that studies the interactions between computers and human language through computationally effective mechanisms for creating systems that can interpret and understand language [13].

The models applied focus on language comprehension, human cognitive aspects and memory organization. Through these models, NLP allows us to extract measurable characteristics from different human-written texts as well as identify basic rules and patterns established by the language in which the text is written.

2.2 Python libraries

We have developed the technologies in Python. One of the main reasons is because there are important libraries of our fields of interest written in this language. Some of these libraries are:

- ***Numpy***

  Numpy[1] is the main package for scientific computing with Python. It contains things such as a N-dimensional array object. This library is necessary for some other libraries such as Scikit-learn [15].

- ***Pandas***

  During the period of the data analysis we have used a library called Pandas[2]. Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language [15].

  The main advantage of Pandas is that provides extensive facilities for grouping, merging and querying Pandas data structures, and also includes facilities for time series analysis, as well as i/o and visualisation facilities.

  This library provides two different data structures:

[1] http://www.numpy.org/

[2] https://pandas.pydata.org/

  - ***Series:*** is a one dimensional object where every element has its own index.
  - ***DataFrame:*** is a two-dimensional labelled object with columns of potentially different types.

  Pandas is the library that allows us to manage all the data related to Twitter messages, allowing us to group them, filter them or even import or export them to formats like CSV or JSON when it was necessary.

- ***Matplotlib***

  Matplotlib[3] is a Python 2D plotting library which allows us to produce different data presentations from a Python script. You can generate plots, histograms, power spectra, bar charts, error charts, etc [7].

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

- ***Scikit-learn***

  Scikit-learn[4] is an Open Source library of Python with machine learning algorithms. It provides efficient tools for data mining and data analysis [18]. This library is built upon other libraries: Numpy, SciPy(Scientific Python) and Matplotlib.

  Scikit-learn includes tools such as:

  - ***Classification:*** identifying to which category an object belongs to.
  - ***Clustering:*** automatic grouping of similar objects into sets.
  - ***Dimensionality reduction:*** reducing the number of random variables to consider
  - ***Model selection:*** comparing, validating and choosing parameters and models.
  - ***Preprocessing:*** feature extraction and normalization

  This is the library we have used to implement each of the necessary steps in the classification process (for more details, Sect. 3.3).

- ***NLTK***

  NLTK[5] is a set of libraries and programs for symbolic and statistical natural language processing for the Python programming language. NLTK includes graphical demonstrations and sample data [2]. We have used some classes in this library to perform NLP area functions such as:

[3] https://matplotlib.org/

[4] http://scikit-learn.org/

[5] https://www.nltk.org/

- ***Tokenization:*** given a sequence of characters and a defined document unit, tokenization is the task of breaking it into parts, called tokens, while discarding certain characters or words in the document that contain no information about the document. Commonly the words and characters that are discarded are punctuation and stop words (such as articles, prepositions, conjunctions, pronouns, etc.) that are words that have no meaning in themselves.

- ***Stemming:*** is a linguistic process that consists of finding the corresponding stem, given a flexed form (i.e. plural, feminine, conjugated, etc.). The stem is the form that by convention is accepted as the representative of all the flexed forms of the same word. That is, the stem of a word is the word we would find as an entry in a dictionary.

- ***POS-tagging:*** this task consists of tagging each of the words of a text to its grammatical category.

- ***N-grams:*** from the point of view of language analysis, the n-grams of a document are substrings of n elements (words) which allow us to extract characteristics from sequences of words instead of isolated words.

2.3 API Twitter

For the collection of Twitter messages we have used the API[6] of this platform.

This allows you to access to read and write Twitter data, that is, through it you can create and collect tweets and read the profile of users and their followers (among other data from each profile), as it identifies the different Twitter applications and users who register using authentication and OAuth authorization [12]. The Twitter REST API responses are in JSON format.

In addition to the REST API, the public Twitter API also has a streaming API, which provides access to a high volume of tweets with low latency.

Both APIs have been used in the project, the API Rest for the collection of tweets from specific users, the streaming API for the capture of tweets related to insomnia.

[6] https://developer.twitter.com/en/docs

2.3.1 Tweepy

The number of people using APIs like Twitter has increased in recent years and this is one of the reasons why the developer community has implemented numerous wrappers to be able to use these APIs through programming languages.

In this project we have used an open source library called Tweepy[7] that allows us to use Python to communicate with the Twitter API.

Specifically, we have used it to use the Twitter streaming API. Tweepy allows the handling of authentication, connection/disconnection, errors and filtering with the Twitter API in a simple and intuitive way [20].

2.4 API Google Maps

For the study of the insomnia phenomenon, we were interested in the location of the tweets captured during the research, however, only 1% of all tweets are geolocated. To solve this problem, we decided to use the ”Location” field which is associated with the account the tweet came from.

Google Maps[8] has an associated set of APIs that allow you to use the best of this service [24]. These APIs allow you to work with:

- ***Maps:*** real static and dynamic maps, Street view and 360^o views.
- ***Routes:*** find the best routes to get from one place to another.
- ***Places:*** discover the location of places using phone numbers, addresses and real-time signals.

Specifically we have used the Geocoding API which allows us to convert the addresses that users have in the ”Location” field written in text to geographic coordinates, and so be able to work with the locations in a more effective way than with strings.

[7] http://docs.tweepy.org

[8] https://cloud.google.com/maps-platform/

2.5 Stanford Topic Modelling Toolbox (STMT)

At one point in the course of the project we were interested in classifying the tweets related to insomnia into different topics in order to group them according to the type of information provided by the tweet (this classification is explained in more detail in the chapter on Architecture).

To make this classification, we decided that one of the characteristics that was of great importance was the result of a topic model such as LDA, however this model is based on unsupervised learning, take as input a series of documents and the model groups them together in a number of topics that you want. When you want to know the topic of a new document, the model, based on the principle that a document is the result of a mixture of all the topics it has learned, gives you back the result that is most probable to be the result.

Unfortunately, this topic model was not useful for our topic classification because we wanted to tell the model not only the number of topics we wanted to classify, but also which tweets belonged to each topic. Therefore, we opted for a theme model called L-LDA which, as its name suggests, is a modification of the LDA based on supervised learning, not only you indicate the number of topics you want to group together but also the documents it takes for learning must be labelled in each of these themes [19].

To make use of this model, we have used a tool developed by Stanford University called Stanford Topic Modelling Toolbox (STMT)[9] that allows us to train in this model, thus allowing us to generate a list of words for each topic with the probability that the word belongs to that topic.

The program uses Scala language scripts, however we have made use of a Python wrapper called Topbox[10] that allows you to use the STMT tool in a more simple way without having any knowledge of the Scala language.

2.6 Senpy

Senpy[11] is a framework developed by the GSI at ETSIT-UPM for sentiment and emotion analysis services. The main advantage of Senpy is the implementation of all the tasks common to a web service that offers this type of services allowing developers to focus on the implementation of the analysis and classification algorithms [21]. In addition, all these

[9] https://nlp.stanford.edu/software/tmt/tmt-0.4/

[10] https://github.com/clips/topbox

[11] http://senpy.readthedocs.io

services that use these developed algorithms share the same API which allows you to use all the services that Senpy implements in an easy and interchangeable way.

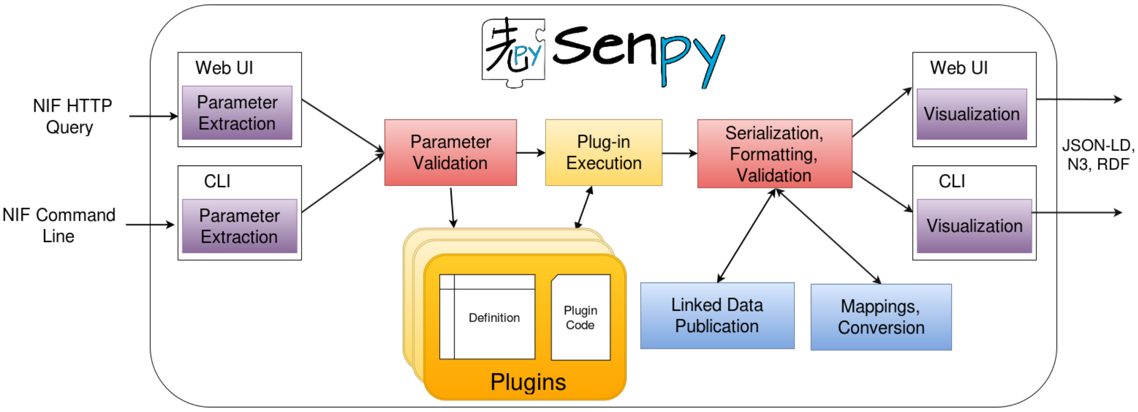


Figure 2.1: Senpy's Architecture

Senpy's architecture is based on two main components:

- ***Senpy Core:*** where the service is built and where all the tasks common to a web analysis service (data validation, user interaction, formatting, logging, etc.) are performed.
- ***Senpy Plug-in:*** where are deployed the classifiers and the analyses of a determined service.

In the case of our project, we have implemented a Senpy plug-in that allows the analysis of tweets related to insomnia.

This plug-in offers us the advantages of portability and diffusion of our service since Senpy has a user interface easy to use by any user; and an advantage for the easy analysis of new tweets related to insomnia for other developers through the API offered by Senpy.

2.7 Luigi

Luigi[12] is a Python workflow management tool. It was developed by Spotify to help create complex lines of batch job data.

The workflow for the processing and analysis of Twitter messages is always the same: capture of tweets related to the topic we are interested in, processing and classification of the message and storage of the message and analysis in some kind of storage system.

[12] https://github.com/spotify/luigi

For the management of all these tasks, Luigi is an optimal solution because it allows you to execute all of them, controlling their dependencies and protecting the execution against the possibility of any failure of them so that the others are not affected.

The two basic concepts of the Luigi tool are: **task**, that is a unit of work; and **objective** that is the output of a task which may be a file in the local file system, some data in a database, etc.

The dependencies between tasks are defined in the form of a pipeline, i.e. if a task B depends on a task A, it means that the output of task A will be the input of task B.

Another important aspect of Luigi from the point of view of task management is the possibility of being able to visualize the status of the workflow using its visualizer. Thanks to it, we get a good overview of the tasks that have been correctly executed and those that remain to be executed.

2.8 ElasticSearch

Elasticsearch[13] is a real-time, open source, distributed full-text search and analysis engine. It is developed in Java and is used by many large organizations around the world [6].

It can be accessed from the Restful web service interface and uses JSON documents without a schema to store data.

It has features such as:

- Distributed and Highly Available Search Engine.
- Various set of APIs(HTTP Restful API, Native Java API, etc).
- Reliable, Asynchronous Write Behind for long term persistence.
- Single document level operations are atomic, consistent, isolated and durable.

The parts that compose this storage system are:

- ***Node:*** refers to a single instance running Elasticsearch. A single physical and virtual server supports multiple nodes depending on the capabilities of your physical resources.
- ***Index:*** is a collection of documents.

[13] https://github.com/elastic/elasticsearch

- ***Document:*** is a collection of fields in a specific way defined in JSON format with a unique identifier called UID.
- ***Shards:*** the indexes are subdivided horizontally into shards and each shard contains the same properties as the index to which it corresponds.
- ***Replicas:*** Elasticsearch allows users to create replicas of their indexes and fragments for improved availability and performance.

2.9 Sefarad

To visualize the data stored in Elasticsearch we have used the Sefarad[14] environment developed by the GSI at ETSIT-UPM.

Sefarad consists of two main modules:

- ***Visualization:*** is the main module of Sefarad. It is structured in dashboards formed by components (Polymer Web Components) that perform the function of visualization graphics.
- ***ElasticSearch:*** represents the persistence layer of the project and where the data is obtained for visualization.

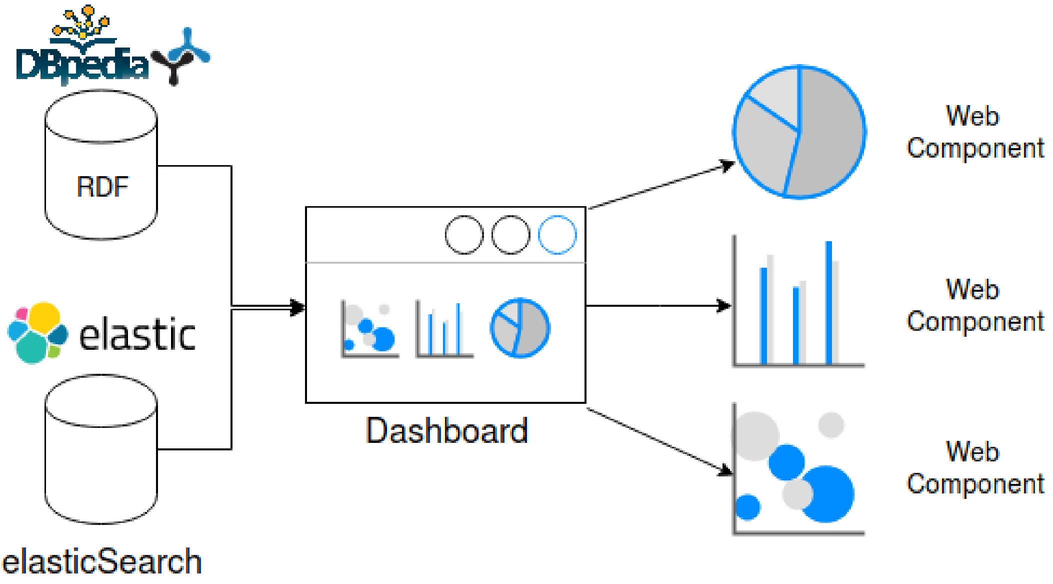


Figure 2.2: Sefarad's Architecture

[14] http://sefarad.readthedocs.io

As shown in the architecture, Sefarad is also capable to retrieve semantic data from external sources, such as Fuseki or DBPedia.

CHAPTER 3

Machine learning model building and evaluation

3.1 Introduction

In this chapter we will explain the development process of the classifiers used in this project.

We will start by describing the formation of the datasets, specifically the capture process, the analyses made of their content and the labelling process. Later we will explain the developed classifiers, explaining the feature extraction used and the different machine learning algorithms used. Then, we will discuss the results of each of them and the algorithm chosen.

Finally, we will explain the formation of a second dataset but this time only with users in Spain to make a research on Spanish users.

3.2 Dataset

The tweets were captured from December 14, 2017 to January 4, 2018. This capture was done through the Streaming Twitter API and the Tweepy library that allows us to capture tweets in real time. The tweets we captured had to meet the following characteristics:

- They must have been in Spanish.
- They had to contain the word insomnia, because it is the phenomenon we want to study.
- They could not be re-tweets (a re-post of a tweet originally posted by a different user) because we are only interested in experiences expressed by people who claim to have insomnia and therefore these tweets are not good for us.

The sample size was 54432 tweets.

3.2.1 Geography

The first study we carried out on this dataset was to know the geographical origin of the tweets captured. The process followed to determine the geolocation of the tweets was as follows:

First the tweets that were already geolocated by twitter were determined. However, the number of tweets we captured that met this characteristic was 1% of the sample.

To increase the number, we chose to geolocate the tweets from the user's location field that had written the tweet in question.

To do this we use the Google Maps Geocoding API that allows us to know from the addresses that users have (Cadiz) their geographic coordinates (-6.28, 36.5). The use of this API is in more detail in Sect. 2.4

With the combination of these two procedures we obtained 32404 geolocated tweets that is 59% of our dataset.

For the part of visualizing the geolocation of the tweets we have used the Python GeoPandas library that allows us to represent the geolocalized tweets in their corresponding country as shown in Fig. 3.1.

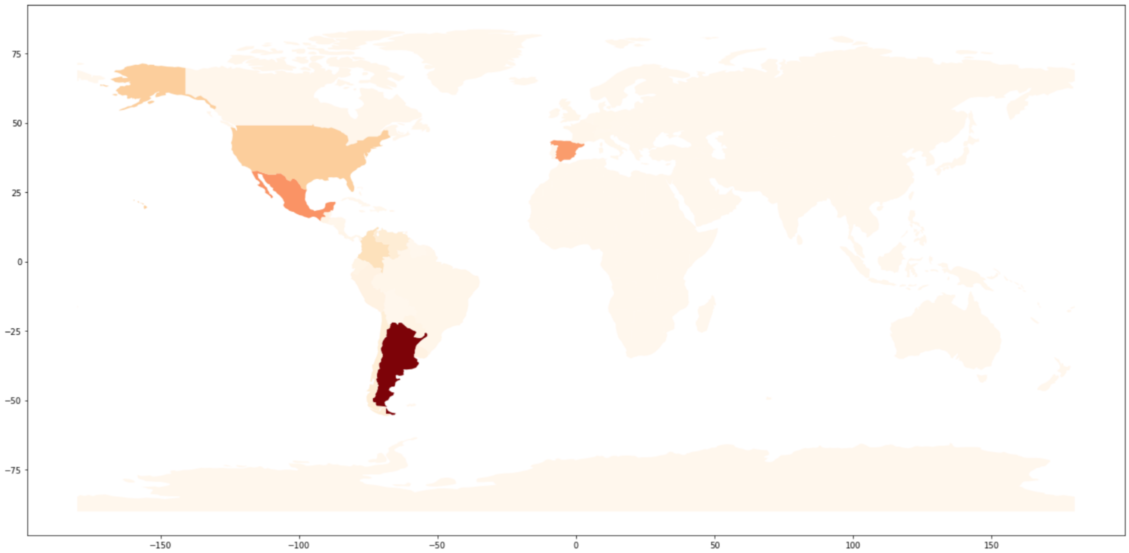


Figure 3.1: Map representing the countries according to the number of tweets with the word insomnia.

Country	**Tweets**	**Percentage**
Argentina	11132	34.53%
Mexico	5349	16.59%
Spain	5112	15.85%
United States	3052	9.46%
Colombia	1884	5.84%

Table 3.1: Top countries in tweets with insomnia

We have also highlighted the countries with more tweets with the word insomnia as shown in Table 3.1, in which it is important to highlight that Argentina, Mexico and Spain have a much greater presence than the rest of the Spanish-speaking countries.

3.2.1.1 Spain

About the previous study we have focused on tweets from Spain.

The number of tweets coming from this country is 5112 from 1982 different users. The fact that we want to determine is the importance of insomnia in the Twitter user community. To do this we have determined the percentage of users who write at least one tweet in a given period with the word insomnia.

This percentage is given by the number of different users that we have obtained from our dataset from Spain among the total number of users that have written in that country during the time of capture that has been 3 weeks.

When determining these total users who have posted during the capture time, only users who have the location field filled in should be taken into account, as our sample only includes users who have an insomnia tweet and the location field filled in. The number of users in Spain who have a written localization field is 2.9% [11].

With this number and the data that there are 7.52 million active Twitter users (they write at least one tweet) for a month in Spain[1], it can be obtained that there are 164124 active users with the location filled in during the time of capture.

From our study we have been able to extract that in the time of capture the number of users who have posted at least once the word insomnia is 1982 users which determines that approximately **1.21% of users in Spain have ever written about insomnia.**

3.2.2 Labelling

Our goal is to detect and characterize tweets from users who claim to have insomnia as a supervised classification task and two text classifiers will be trained to perform this task.

A sample of labelled tweets is required for this training process. As there is no published dataset with these characteristics, we have made our own through a labelling process:

[1] https://es.statista.com/

3.2.2.1 Insomnia

The first labelling we did consisted of differentiating between tweets that contain the word insomnia and come from a person who claims to have insomnia and those who do not. Specifically, this tag is added to tweets that match these characteristics:

- They do not come from a corporate account, that is, the user who has written the tweet is a natural person.
- In the tweet the user informs that it is the user who suffers from insomnia, so there are no valid tweets where it is transmitted that an acquaintance suffers from insomnia for example.

It is important to remember that none of the tweets that form the dataset are re-tweets, a filter that was implemented in the capture process.

Tweet	**Insomnia**
I have insomnia again (Tengo insomnio otra vez)	yes
Do you have insomnia? (¿Tienes insomnio?)	no

Table 3.2: Examples of tweets with the label insomnia

This labelling was handmade until we got 300 tweets belonging to the affirmative label ”Insomnia”. From this set of tweets we started to do the following tagging.

3.2.2.2 Themes

Another of the taks was to group the tweets by topic, according to the information transmitted by the post.

The objective of this tagging was to group tweets by type of information, allowing us to better characterize the messages and thus draw better conclusions. The topics we defined were:

- User expresses only his sleep disorder.
- User expresses his sleep disorder and gives the causes of his problem.
- User expresses his or her disorder and requests help.

- User expresses his sleep disorder and transmits what he does at night.
- User expresses his/her sleep disorder and takes some action to try to solve it.

From the labelling of our sample it can be seen that most of the tweets only give information that the user is suffering from insomnia at the moment (70%). 11% also comment on the cause of the disorder, such as a habit or emotional situation such as anxiety. On the other hand, only 4% and 12% of the posts ask for some kind of help or comment on the activity they do during the night. Finally, and more worryingly, only 2% say they should take some action to solve it.

These percentages show that there is a need to develop techniques for the detection of sleep disorders such as insomnia in order to help and diagnose those affected.

3.2.2.3 Symptoms

As mentioned above, we are interested in knowing what are the different symptoms of tweets for users with insomnia. In the third edition of International Criteria for Sleep Disorders (ICSD-3), the main symptoms of insomnia were defined as: difficulty in starting sleep; short sleep duration; difficulty in sleeping and low energy during the day.

In order to know the degree of importance of each of them, a label was made of our random sample of 300 tweets.

The first conclusion we drew was that most of the tweets did not express any of the defined symptoms. Specifically, only 14% (43) of the posts commented on any of the symptoms.

Within this small percentage the distribution of symptoms is indicated in the Table 3.3.

Symptom	**N (%)**
Difficulty in starting sleep	29 (67.44)
Short sleep duration	8 (18.6)
Difficulty in sleeping and low energy during the next day	6 (13.95)

Table 3.3: Distribution of symptoms

The conclusions of this distribution are limited due to the small size of the sample with symptoms, a fact that leads us to believe that users suffering from insomnia do not pay attention to the reasons why they suffer this disorder.

3.3 Classifiers

Two classifiers have been developed for the automation of the labelling task. The first performs the task of tagging tweets that meet the characteristics of the insomnia tag and the second classifies the tweets in the different themes defined. We have chosen not to develop a classifier for the symptom label due to the difficulty of finding posts that can form a large enough sample to be able to train the classifier.

Both classifiers perform a task of classifying texts. This task always has a similar workflow, so we have automated this workflow through the pipeline object provided by the scikit-learn library.

The different phases through which a text classification task passes are:

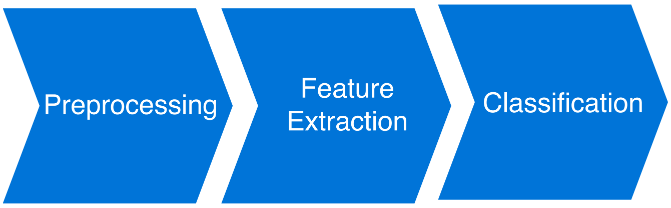


Figure 3.2: Process of text classification

3.3.1 Preprocessing

In this phase the raw text is taken and cleaned so that it can be analysed, such as the elimination of punctuation, stop words, rare characters, word stemming, emoticons or urls, as none of them are useful for classifying texts.

It is important to highlight the case of stop words, which are words that are meaningless and do not provide information about the content of the document. Within this set of words are the articles, pronouns, prepositions, etc. The list of these words, as the punctuation marks list, is provided by the NLTK library (for more details of this library, refer to Sect. 2.2).

To perform this task we have made use of the TweetTokenizer module of the NLTK library, which allows us to convert the document to a normal list of tokens, but taking into account the characteristics of a tweet such as the mention of a user or a hashtag.

Another task to be performed by the preprocessing is the stemming process, which consists of eliminating the endings in order to keep the base form of the words known as stem. To do this we have taken the PorterStemmer algorithm provided by NLTK.

3.3.2 Feature extraction

From pre-processed data is extracted some features. These feature extractions consist of converting processed and annotated text into numerical vectors, which give an automatic learning model a simpler, more focused view of the text.

Each of these extractors is known as a transformer. These transformers must be subclasses of BaseEstimator and TransformerMixin. BaseEstimator provides the necessary methods such as *get params*, to obtain the classifier feature names and the *set params*, to modify the transformer parameters. On the other hand the TransformerMixin class provides the transformer with the methods that allow us to fit and transform the data to obtain the characteristic.

To run all the transformers on a parallel text and join their results, so that they can be used in the automatic learning model, we have used the FeatureUnion module of scikit-learn library (explained in Sect. 2.2) that allows us to perform exactly this function.

The transformers used in the development of each of the two classifiers of this project are explained in the following sections.

3.3.2.1 Words

This transformer has been made using the TfidfVectorizer provided by scikit-learn. This module provides a matrix of vectors representing the relative importance of each word in a document and in the entire dataset.

This importance is given by a numerical value calculated from the normalized Term Frequency (TF) and the Inverse Document Frequency (IDF).

The term TF is calculated as the number of times the word appears in a document divided by the total number of words in the document; and the term IDF is calculated as the logarithm of the division of the total number of documents divided by the number of documents in which the word appears.

The combination of these terms will be greater in words that are repeated in a given document and low frequency in the rest of the documents.

When implementing this transformer, as we have already mentioned, we have used scikit-learn's TfidfVectorizer, passing it our preprocessor as a parameter so that it can preprocess the raw data in the way we want.

3.3.2.2 N-grams

This transformer has a very similar meaning to the previous one as it also results in a matrix of TF-IDF vectors. However, this time the importance of n-grams is measured.

N-grams is the combination of N terms together. With this transformer we help the classifier to know the relationships between most likely words. This allows us to extract characteristics from the context in which the words are found.

Although the result is the same, when implementing this transformer we have chosen a different solution to the previous one. We have developed this transformer by first passing the raw data through a CountVectorizer, whose parameters are the N range of N-grams we want to extract and our preprocessor. With this we obtain a matrix of vectors that indicate the number of times each N-gram appears in each document. This result is followed by the TfidfTransformer that converts this matrix into a matrix of TF-IDF vectors.

3.3.2.3 Latent Dirichlet Allocation (LDA)

LDA is a modeling algorithm for unsupervised learning topics. Specifically, it discovers a number of new topics in text documents. This algorithm considers each document as a mixture of several topics, and each word in the document belongs to one of the topics in the document. LDA works by calculating the probability that a document belongs to a topic, based on the probability of each of the words that make up the document [3].

LDA as a feature extractor allows the automatic learning algorithm to learn the different patterns that exist throughout the documents.

In our project we have implemented it using the LatentDirichletAllocation module that provides scikit-learn, using parameters such as the number of topics we want to discover.

3.3.2.4 Labeled LDA (L-LDA)

This transformer, as its name suggests, is a variation of the LDA theme modeling algorithm. L-LDA is an algorithm for modelling topics as well as supervised learning, i.e. not only you indicate the number of topics you want to group together but also the documents it takes for learning are already labeled in each of these themes.

It works on the same principle as LDA, considers a document as a mixture of the topics indicated and determines to which topic it belongs from the probability of each of the words belonging to each of the topics.

As we have indicated in Sect. 2.5, this transformer will be useful in the topic classifier. We have implemented it in two parts. First we used the Stanford Topic Modelling Toolbox tool which allowed us to obtain a dictionary for each tagged topic composed of all the words in the dataset and the probability that each of those words belong to that particular topic. In order to do this, we previously pre-process the texts as we have indicated in all the transformers.

Our transformer, relying on the dictionaries generated from our dataset, when analysing a document, returns a new dictionary whose keys are the 5 tagged themes and the values are the probabilities that the document belongs to each of the themes.

3.3.3 Classification

The last step in the text classification process is to train a classifier using the characteristics created in the previous step. There are many options of automatic learning models that can be used to train a final model. Basically, this classifier aims to be able to classify unknown texts through supervised learning.

Scikit-learn provides different types of classification algorithms. Each of them has different parameters that influence the level of success of this algorithm. The task of selecting each of these parameters (called hyper-parameters) cannot be chosen randomly and therefore we have used the GridSearchCV module. This module allows you to know the hyper-parameters that give the best results to a classifier when classifying a dataset with certain features. Basically it allows to optimize the algorithm to the maximum.

On the other hand, we have also measured the level of success of feature extraction and classifiers using the cross validation technique (K fold) due to the size of our dataset. K fold consists of dividing the data into k sets, and going through these sets k times, taking in each of them one of the sets as the training data and taking the other k-1 sets as test data. The estimation of the results is averaged in all tests to obtain the level of success of our model.

We can deduce this technique allows us to check the results of our classification algorithms by exploiting our dataset to the fullest and without incurring an overfitting problem.

In this project we have tested some classification algorithms provided by the scikit-learn library:

- ***Multinomial Naive Bayes:*** This classifier implements the naive Bayes algorithm for multinomial models. The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as tf-idf may also work. Scikit-learn provides a classifier of this type called MultinomialNB which is the one we have used.
- ***Support Vector Classification (SVC):*** This classifier belongs to the supervised learning Support Vector Machine (SVM) set for classification detection. Again, it is provided by scikit-learn under the name SVC.
- ***K-Nearest Neighbors(KNN):*** The classification based on neighbours is a type of learning that is calculated from a simple majority vote of the closest neighbours of each point, that is, a point is assigned the classification that has the largest number of representatives within the closest neighbours to that point.
- ***Logistic Regression:*** In this classifier the probabilities describing the possible results of a single trial are modelled using a logistics function. The implementation has been done through the LogistiRegression class of scikit-learn, which allows us to adjust to the binary logistic regression One vs Rest or Multinomial; or the type of solver used.
- ***Random Forest Classifier:*** Random Forest is a meta-estimator that adapts to a series of decision tree classifiers in several subsamples of the dataset and uses an average to improve predictive accuracy and control over-adjustment.

3.3.3.1 Evaluation Metrics

Once we have implemented the workflow of the classifiers, we must check their operation to measure if they have good enough results to deploy them or if we have to continue looking for a better combination of extraction features.

This check will be done objectively and through a series of metrics that will provide us with information on how many and what type of errors our classifiers make when classifying new tweets.

Before defining the metrics that we will use for the evaluation we must define a series of concepts:

- ***True Positives (TP):*** are correctly predicted with a positive label (Positives that were predicted as positives).
- ***True Negatives (TN):*** are correctly predicted in a negative label (Negatives that were predicted as negatives).
- ***False Positives (FP):*** are the ones predicted incorrectly with a positive label (Negatives that were predicted as positives).
- ***False Negatives (FN):*** are the ones predicted incorrectly with a negative label (Positives that were predicted as negatives).

The results of our algorithms measured in the four definitions above allow us to calculate the following metrics:

- ***Accuracy:*** is the best known and most intuitive because it measures the percentage of correct predictions in relation to the total number of predictions.

$$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FN}+\mathrm{FP}} \tag{3.1}$$

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

$$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{3.2}$$

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

$$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{3.3}$$

- ***F1 score:*** is the harmonic mean of Precision and Recall.

$$2\times\frac{\mathrm{R}\times\mathrm{P}}{\mathrm{R}+\mathrm{P}} \tag{3.4}$$

3.3.4 Insomnia Classifier

This classifier has the task of labelling the ”Insomnia” label defined in the Sect. 3.2.2.

The first thing we did in the process of developing this classifier was to form the sample that we would use in the training process.

The Insomnia labelling process resulted in 300 tweets with the 'yes' label and 160 with the 'no' label. To avoid a balancing problem we decided to use a sample of 300 tweets, with 160 with a 'no' label and 140 with a 'yes' label.

Once the sample was obtained to train the classifier, we proceeded to design the pipeline.

Preprocessing was done using the TweetTokenizer of NLTK and removing the tokens corresponding to punctuation, stop words, rare characters, emoticons, urls or user names.

Regarding the feature extraction phase, the transformers were used: Word, N-grams and LDA; all of them explained in the Sect. 3.3.2.

With these aspects, we made a pipeline and tested the commented classifiers. It should be noted that all the results obtained in the Table 3.4 were performed using K fold cross validation with K=5 and optimization of all hyper-parameters through GridSearchCV.

Classifier	**Hyper-params**	**Accuracy**	**Precision**	**Recall**
MultinomialNB	alpha=0.1	0.72 (+/- 0.15)	0.74	0.72

SVC	C=1
	kernel='linear'
	probability=True
		0.77 (+/-0.11)	0.79	0.77

Knn	n_neighbors=3
	p=1
	algorithm='ball tree'
		0.6 (+/- 0.15)	0.62	0.6

Logistic Regression	C=2
	penalty='l1'
	tol=0.1
		0.84 (+/- 0.04)	0.83	0.82

Random Forest	n_estimator=23	0.84 (+/- 0.06)	0.76	0.75

Table 3.4: Evaluation Insomnia Classifier

As we can see in the Table 3.4, the machine learning algorithm that has given the best results is the Logistic Regression algorithm, so this will be the algorithm we use when deploying this classifier.

3.3.5 Themes Classifier

The other classifier that has been developed in this project is the one that performs the task of classifying the tweets, previously labelled with the 'Insomnia' label, among the topics defined in the Sect. 3.2.2.

On this occasion it was a difficult task to form a sample with a sufficiently large size and that did not produce a balancing problem. In the section referred to in the previous paragraph, the distribution of the 300 tweets that were hand-labelled with this tag has been discussed.

Due to the disproportionate number of samples corresponding to each of the topics, a tweet capture was performed again, but this time it was filtered with typical words of each topic (for example, in the case of the topic that includes information about the action to solve it, tweets were captured that besides the word insomnia contained words such as doctor, medicine, treatment, etc.) with the objective of capturing the largest possible number of tweets of each topic in particular.

After a series of captures for each theme, a training sample of 172 tweets was formed with more or less the same proportion of tweets for each theme.

Once the sample was obtained to train the classifier, the next step was to create the pipeline. The processing we used has been the same as the one discussed in the previous sections.

Regarding the feature extraction phase, some transformers are the same as those of the Insomnia Classifier (Word and N-grams) but this time we use the L-LDA; all of them are developed in the Sect. 3.3.2.

As in the case of the previous classifier, the results of the Table 3.5 of each of the machine learning algorithms have been obtained with cross validation K fold with K=5 and optimization of all hyper-parameters with GridSearchCV. It can be seen that the results obtained are worse than those of the first classifier. This is a logical consequence because the first classifier is a binary classifier and this is a multi-class classifier. The best results have been obtained again from the Logistic Regression algorithm with an accuracy, precision and recall of 0.75, 0.74 and 0.7, which for a multi-class classifier, and with the small amount of data we had to train it is a good result.

Classifier	Hyper-params	Accuracy	Precision	Recall
MultinomialNB	alpha=0.1	0.6 (+/- 0.21)	0.67	0.6

Knn	n_neighbors=1
	p=1
	algorithm='ball tree'
		0.55 (+/- 0.13)	0.58	0.55

Logistic Regression	C=14
	penalty='l2'
	tol=0.01
		0.75 (+/- 0.14)	0.74	0.7

Random Forest	n_estimator=20	0.46 (+/- 0.18)	0.63	0.48

Table 3.5: Evaluation Themes Classifier

3.4 Dataset 2

After the development of the classifiers defined in the previous sections, these were used for the formation of a second dataset that was only formed by tweets from Spain.

Specifically, we are interested in trying to determine a pattern of activity on a social network such as Twitter of people suffering from insomnia, and compare it with the activity of a set of normal users.

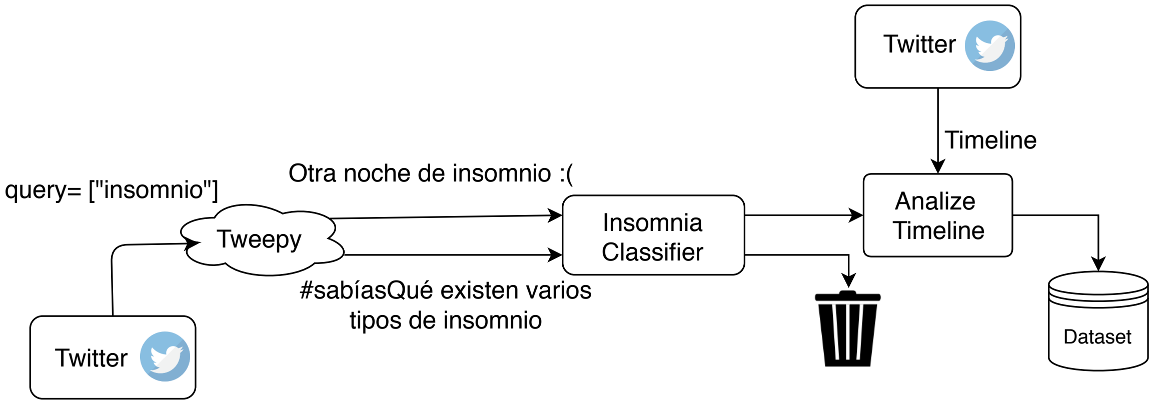


Figure 3.3: Process to elaborate the dataset of users.

The process of selecting a user for the Sleep Group (set of users with insomnia) is described in the Fig. 3.3. Basically the process consisted in making a capture similar to the one made to form the first dataset, but this time only of tweets from the region of Spain. Each one of these tweets was passed through our Insomnia Classifier, so that only the tweets of users who claim to have insomnia would follow in the process. After this first filter, we analysed the last 200 tweets of the account associated with each tweet; if in these 200 tweets there were 2 or more tweets in which the user mentioned the word "insomnia"(insomnio) or "I can't sleep"(no puedo dormir), that user became part of the Sleep Group.

As we have already mentioned we want to determine a pattern of activity of users with insomnia and normal users, so we also perform a user dataset without insomnia (Normal Group). To form this group we made a random capture of tweets from Spain, and hand filtered to have only tweets from people and thus not have in the study tweets from accounts from some kind of organisation. As we did in the case of the Sleep group, we also analysed the timeline of these users to make sure they didn't have any insomnia-related tweets.

This capture was made from April 2 to 10, 2018, and a dataset of 103 users from Sleep Group and 110 users from Normal Group was made.

The first feature that we focused on was the number of tweets related to insomnia that each of the Sleep Group users had (Fig. 3.4).

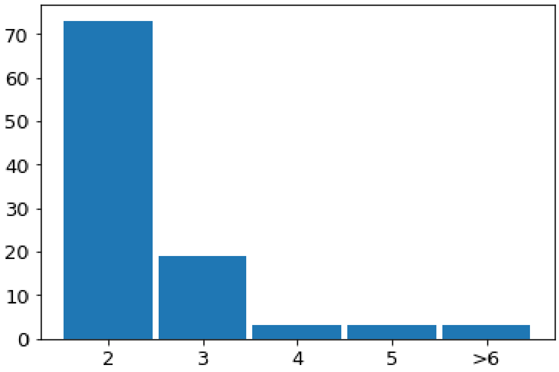


Figure 3.4: Number of tweets related to insomnia of Sleep Group

We can observe that most of the users have two relatively recent messages related to insomnia (72%).

After that, we analyse the time associated to all the tweets of the timeline of each one of the users of the two groups. In the Fig. 3.5 we have plotted the percentage of tweets per hour of average that the users of each of the groups post.

It can be seen that users who belong to the Sleep Group have a greater activity through-

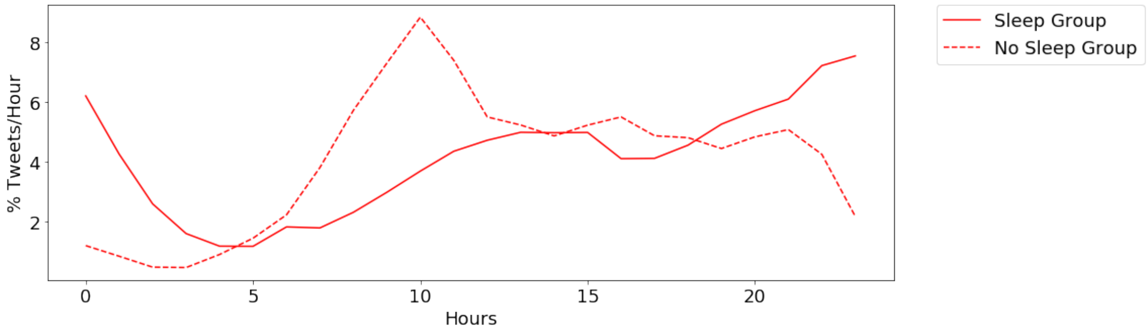


Figure 3.5: Activity on Twitter per hour in users with and without insomnia.

out the night due to the inability to fall asleep. During the morning, the Sleep Group's activity may decrease due to nighttime fatigue and in exchange the Normal Group has its highest point of activity around 10 a.m. In the afternoon, the activity is balanced, until the evening, when the Sleep Group again shows more activity.

CHAPTER 4

Insomnia Monitoring Service

4.1 Introduction

In this chapter we will explain the development of the monitoring service that we have implemented for the analysis data of messages posted on the social network Twitter related to insomnia.

Each one of the technologies used for the development of this architecture are defined in the Chapter 2.

In the first section we will talk about the architecture that forms the service, mentioning the systems that compose it and the relationships that exist between these systems.

In the following sections we will define in more detail each of these systems of the architecture and their function within it.

4.2 Architecture

In this section we will present the global architecture of the project, defining the different subsystems that participates in the project. We can identify the following subsystems:

- ***Capture system:*** This subsystem is responsible for capturing all tweets that circulate on the social network with the word insomnia.
- ***Analysis system:*** This system uses Senpy's service, defined in Sect. 2.6, to classify the content, location, feeling and emotion of each tweet.
- ***Persistence system:*** It is responsible for the storage and search of the tweets once analysed. It has been implemented with the Elasticsearch search engine (for more details, refer to Sect. 2.8).
- ***Visualization system:*** Server to visualize the analysed data. We have used Sefarad environment, defined in Sect. 2.9, which allows the visualization of data stored in Elasticsearch through a series of visualization widgets.

It should be noted that the management of this workflow is done by the orchestrator Luigi who is able to manage the inputs and outputs of each of the systems through a script that includes the necessary requirements for each of them.

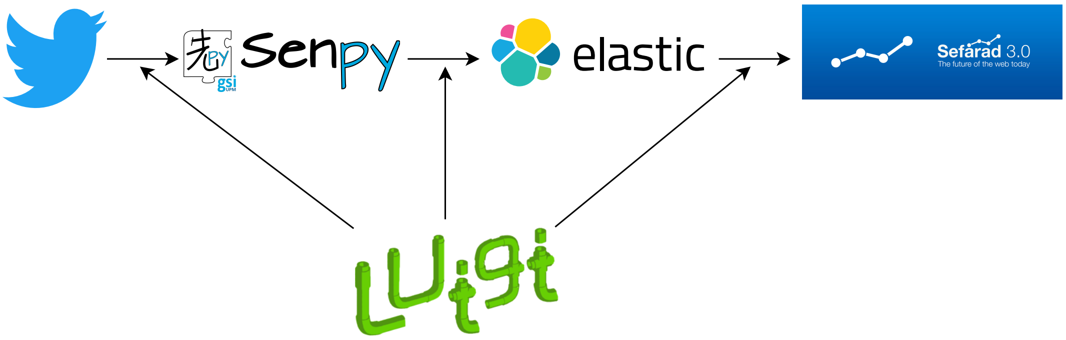


Figure 4.1: Architecture of the monitoring service

In the following sections we are going to describe deeply subsystems involved in the project.

4.3 Capture system

As we have already mentioned, this system is in charge of capturing tweets with the word insomnia.

It is developed in Python and interacts with the two Twitter APIs (Search and Streaming).

It has two modes of operation:

- ***Search:*** in this mode the system searches for all tweets with the word insomnia of the last day.
- ***Username:*** the system receives as parameter the username of a Twitter account. The system analyses your timeline and captures tweets that contain the word insomnia.

In both modes, the system groups the tweets in a JSON file that will be the input of the Analysis system.

4.4 Analysis system

As its name suggests, this system is responsible for analysing the content of captured tweets.

To do this we have used the service called Senpy which is a document analysis server developed by Intelligent Systems Group.

The internal workflow of this system consists of reading each of the tweets contained in the input and making requests to the Senpy API and the Google Maps API for the localization.

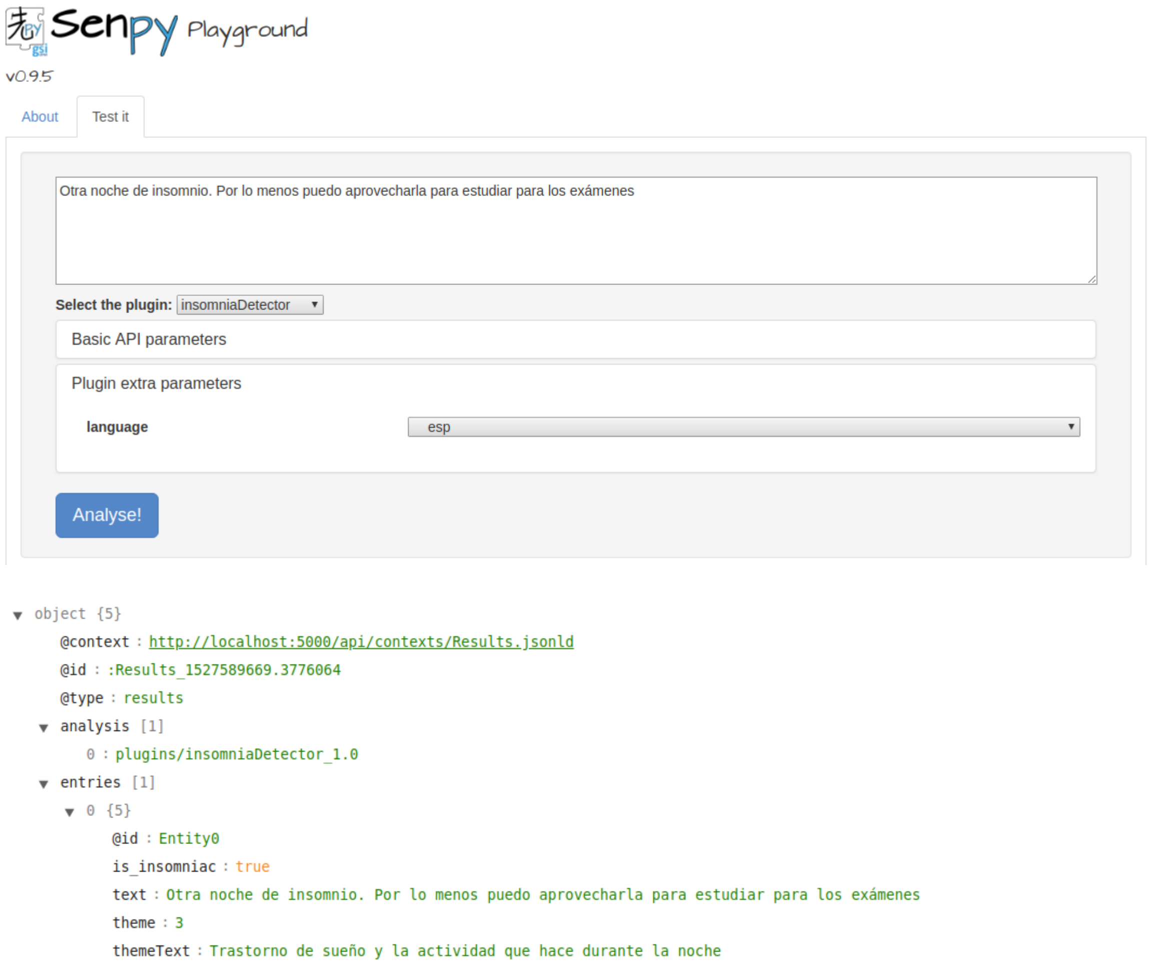
Senpy's API allows the use of each of the plug-ins developed for this service. In this project we have used three plug-ins:

- ***insomniaDetection plug-in:*** This plug-in allows us to know if the author of a tweet has insomnia and analyse it according to its content.
- ***sentiText plug-in:*** This one is used for sentiment analysis. It distinguishes between positive, neutral or negative sentiment.
- ***EmoTextANEW plug-in:*** This other one is used for emotion analysis. Emotions available are anger, disgust, fear, joy, neutral and sadness.

The insomniaDetection plug-in was developed by us. For the development of this plug-in, we first created a file with a senpy extension containing the metadata of the plug-in, with information such as name, version, libraries, parameters and author.

After that, we created a Python script containing its implementation. This implementation corresponds to the deployment of the classifiers explained in Sect. 3.3.

Both classifiers work in series, i.e. when the plug-in is going to analyse a tweet, the first classifier to analyse its content is the Insomnia Classifier, if the classification of the classifier is negative, system response with the label *isInsomniac* in negative; and if the classification is positive, the tweet is also analysed by the Themes Classifier and the response includes the labels *isInsomniac*, *theme* and *theme_code* (this last label is only a code number that identifies each of the five themes). An example of an analysis of this plug-in is found in Fig. 4.2 in which the plug-in correctly classifies a tweet in the theme *User transmits his or her sleep disorder and the activity he or she does at nigh.*



Otra noche de insomnio. Por lo menos puedo aprovecharla para estudiar para los exámenes

Select the plugin: insomniaDetector ▾

Basic API parameters

Plugin extra parameters

language esp ▾

```
▼ object {5}
    @context : http://localhost:5000/api/contexts/Results.jsonld
    @id : :Results_1527589669.3776064
    @type : results
  ▼ analysis [1]
      0 : plugins/insomniaDetector_1.0
  ▼ entries [1]
    ▼ 0 {5}
        @id : Entity0
        is_insomniac : true
        text : Otra noche de insomnio. Por lo menos puedo aprovecharla para estudiar para los exámenes
        theme : 3
        themeText : Trastorno de sueño y la actividad que hace durante la noche
```

Figure 4.2: Example of insomniaDetection plug-in input and output

It should also be mentioned that when using this plug-in in the architecture we are defining in this chapter, tweets with a negative Insomnia Classifier classification will not undergo further analysis nor will they be stored in the system s because they do not interest us in our development context.

Finally, to the global response of this system is added a *location* field formed by the longitude and latitude of the place that the tweet to be analysed comes from. These values correspond either to the provided values of the geolocation associated with the tweet, or in case of missing these fields, to the location obtained by the Google Maps Geocoding API from the "Location" field of the user who wrote the tweet.

4.5 Persistence system

This system is responsible for the storage and subsequent search of the tweets analysed by the Analysis system.

As we have already said, the technology we have chosen to perform these functions is Elasticsearch, which allows us to index and consult documents.

In Elasticsearch we have stored the tweets analysed in JSON format with the structure indicated in the Table 4.1.

Field	**Content**
created_at	This field indicates the date a time the tweet was published
id	Unique identifier of the tweet. Twitter ensures that there is no possibility that there are two tweets with the same id
user.id	Unique identifier associated with the user who wrote the tweet
text	This field is the tweet content in UTF-8 format
isInsomniac	This field contains the result of Insomnia Classifier
theme	This field contains the result of Themes Classifier
theme_code	This field contains a code associated to the field *theme*
sentiment	This field contains the result of sentiText plug-in
emotion	This field contains the result of EmoTextANEW
lat	This field contains the latitude of the origin place of the tweet
long	This field contains the longitude of the origin place of the tweet

Table 4.1: Fields of documents indexed in Elasticsearch.

The task of storing the tweets analysed by Analysis system in Elasticsearch was done using the Luigi package *luigi.contrib*, specifically we developed a class whose parent is *Copy-*

ToIndex that allows storing the tweets analysed in an index indicated as a parameter.

4.6 Visualization system

The last system is responsible for visualizing the data stored in Elasticsearch. To do this, we have used the Sefarad environment, a tool developed by the Intelligent System Groups.

Specifically, the visualization will be done through a dashboard that will be composed of a series of widgets that we will talk about in the following sections.

The first task our dashboard must perform is to be able to perform API REST requests from our Elasticsearch index so that it returns the desired documents to you. These requests are called aggregations in Elasticsearch and allow you to group and extract statistics from documents based on the value of one of the fields. Another advantage of this type of request is that it allows you to concatenate several aggregations and obtain the result of both in a single response.

The responses from Elasticsearch will be used by the widgets to represent the data they contain. These widgets have been developed through the JavaScript Polymer library.

Using Elascticsearch's ability to concatenate aggregations, widgets allow users to filter and view only tweets that meet multiple characteristics at the same time. For example, the dashboard is going to be able to display only the tweet statistics for the topic that includes users who request help, who express a feeling of sadness and an emotion of disgust. This possibility makes our visualization system a powerful tool when searching for specific data about insomnia.

The widgets that make up our visualization dashboard are:

4.6.1 Widgets

4.6.1.1 Sentiments

Widget made with the Google-chart-elasticsearch component allows to visualize the number of tweets with each of the feelings (negative, neutral and positive) through a column diagram.

The input parameters of this component are:

- ***data:*** JSON with all the documents received from Elasticsearch.

- ***field:*** field of the documents to be displayed. In our case the value of this parameter is 'sentiment'.
- ***type:*** Google Chart type. In this case the type is 'column'.
- ***filters:*** list of aggregation filters.
- ***title:*** title of the component
- ***cols:*** array with the labelling of each one of the axes.

The widget performs a filter aggregation to the Elasticsearch index when the user clicks on one of the feelings, allowing us to visualize only the tweets corresponding to the selected feeling.

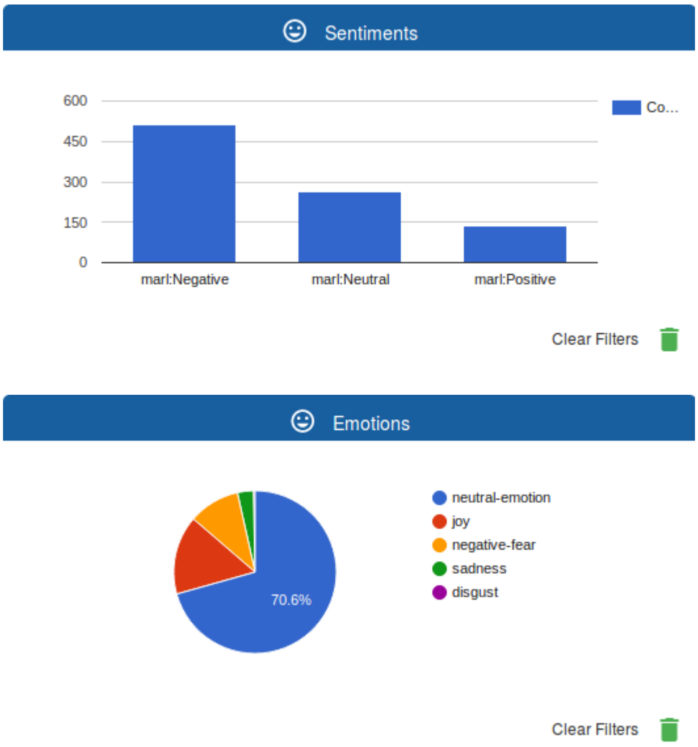


Figure 4.3: Emotion and sentiment widgets

4.6.1.2 Emotions

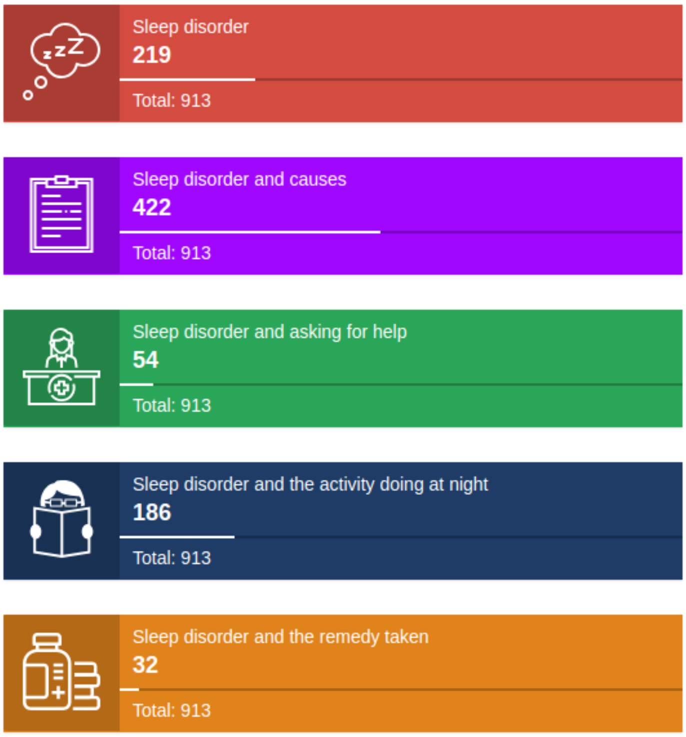
This widget has been made with the same component as the previous one but with a 'pie' **type** and displaying the percentage of tweets for each of the emotions (neutral, joy, fear, sadness, disgust).

4.6.1.3 Number chart

Five widgets have been made from this component, one for each of the themes we classify. These allow to know the number of tweets corresponding to each of the themes.

The input parameters of this component are:

- ***data:*** JSON with all the documents received from Elasticsearch.
- ***filters:*** list of with the aggregation filters.
- ***aggkey:*** documents field from which to add documents.
- ***icon:*** logo corresponding to each of the themes.
- ***title:*** title of the topic.
- ***object:*** added value of the field entered in the 'aggkey' parameter.



Sleep disorder
219
Total: 913

Sleep disorder and causes
422
Total: 913

Sleep disorder and asking for help
54
Total: 913

Sleep disorder and the activity doing at night
186
Total: 913

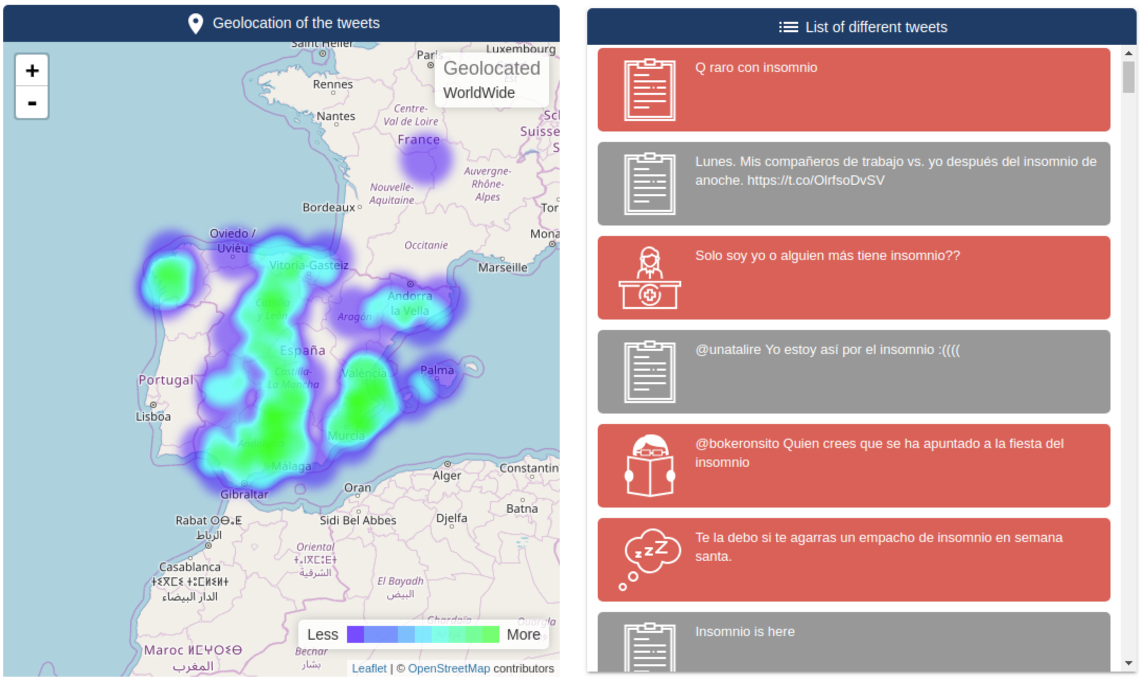
Sleep disorder and the remedy taken
32
Total: 913

Figure 4.4: Number chart widget

4.6.1.4 Geolocation

This widget has been made from the Sefarad happy-map component. It allows us to represent on a map the location of origin of the tweets analysed.

The only input parameter that it needs is **data** and corresponds to the tuple list with latitude and longitude of the tweets.



Q raro con insomnio

Lunes. Mis compañeros de trabajo vs. yo después del insomnio de anoche. https://t.co/OlrfsoDvSV

Solo soy yo o alguien más tiene insomnio??

@unatalire Yo estoy así por el insomnio :((((

@bokeronsito Quien crees que se ha apuntado a la fiesta del insomnio

Te la debo si te agarras un empacho de insomnio en semana santa.

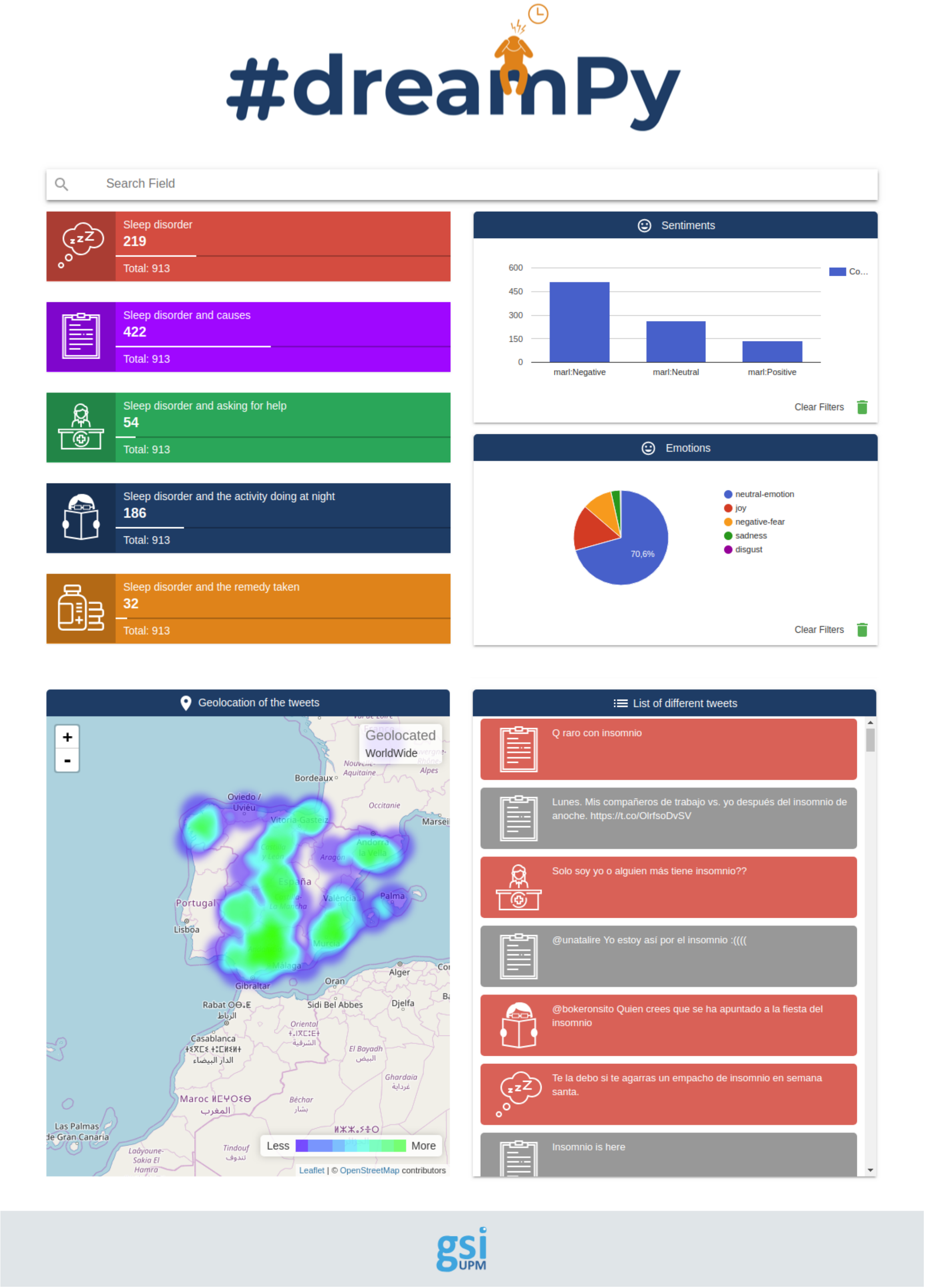
Insomnio is here

Figure 4.5: Geolocation and list widgets

4.6.1.5 List of tweets

This widget has been made from the tweet-chart component. This component allows you to view a list of tweets stored in Elasticsearch, as well as the feeling it expresses due to the background colour (red for negative, green for positive, and grey for neutral), and the theme to which it corresponds because it is represented with the logo associated with that theme.

The only input parameter is **data** to pass to the component the JSON with all the documents received from Elasticsearch.



#dreamPy

Search Field

Sleep disorder
219
Total: 913

Sleep disorder and causes
422
Total: 913

Sleep disorder and asking for help
54
Total: 913

Sleep disorder and the activity doing at night
186
Total: 913

Sleep disorder and the remedy taken
32
Total: 913

Sentiments

Clear Filters

Emotions

Clear Filters

Geolocation of the tweets

List of different tweets

Q raro con insomnio

Lunes. Mis compañeros de trabajo vs. yo después del insomnio de anoche. https://t.co/OlrfsoDvSV

Solo soy yo o alguien más tiene insomnio??

@unatalire Yo estoy así por el insomnio :((((

@bokeronsito Quien crees que se ha apuntado a la fiesta del insomnio

Te la debo si te agarras un empacho de insomnio en semana santa.

Insomnio is here

Figure 4.6: Visualization Dashboard

CHAPTER 5

Conclusions and future work

5.1 Introduction

In this chapter we will explain the conclusions that we have reached after the study. We will also talk about the goals achieved, the problems we have faced in achieving these goals and how we have solved them.

Finally, we will present the lines of future work that we have thought for the project.

5.2 Conclusions

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

The first conclusion we reached was that Argentina, Mexico and Spain are the Spanish-speaking countries with the highest proportion of tweets with insomnia.

In the case of the Spanish Twitter user community, through the penetration [1] of this social network in this country and the percentage of users that have the location field filled in [11]; together with the number of users captured during the dataset formation process,

[1] www.statista.com

we have been able to estimate that 1.21% of users in Spain write at least one tweet per month with the word insomnia, which corresponds to a total of 54450 users.

The labelling process we have carried out has allowed us to know that only 2% of the tweets handled contain information on any kind of remedy, such as medical treatment, which allows authors to try to resolve their problem with insomnia. In addition to this labelling process we have not been able to draw accurate conclusions about the levels of importance corresponding to the three most common symptoms defined in the ICSD-3 due to the small amount of data we were able to collect, however we were able to see a clear difference between the main symptom present among the users, corresponding to the *Difficulty in starting sleep*, compared to the other two, *Short sleep duration* and *Difficulty in sleeping and low energy during the day*.

The analysis of feelings that we have made in the insomnia monitoring service on Twitter explained in the Chap. 4 has allowed us to corroborate studies previously published by other researchers in which they found a relationship between twitter users who posted tweets about insomnia and the negative feelings of those users [8].

Regarding the dataset of Spanish users who suffer from insomnia, we have been able to determine that these users have a significantly higher activity posting tweets during the night hours regarding the activity of the normal user group also captured in this project. In addition, this dataset has been captured using the Insomnia Classifier developed in the project, so seeing the results we have just commented on the night activity, we can determine the correct functioning of this classifier when determining tweets about insomnia.

To conclude, we must highlight the results of the classifiers developed. These have obtained good results from datasets with not a large size, which indicates that they are able to have a good performance without the need for large amounts of data. As already mentioned, the results of the Themes Classifier are worse than those of Insomnia Classifier and this is logical due to the fact that it is a multi-class classifier and the smaller amount of data we have been able to obtain to train it. In both, the best results have been given to us by the Logistic Regression algorithm with an evaluation summarized in the Table 5.1.

Classifier		Hyper-params	Accuracy	Precision	Recall
Insomnia	Logistic Regression	C=2 penalty='l1' tol=0.1	0.84 (+/- 0.04)	0.83	0.82
Themes	Logistic Regression	C=14 penalty='l2' tol=0.01	0.75 (+/- 0.14)	0.74	0.7

Table 5.1: Summary of the evaluation of the classifiers.

5.3 Achieved goals

In the following section I will explain the achieved features and goals that are available in this project.

- ***Formation of a dataset composed of insomnia tweets from different countries.***

  The first objective we achieved was to form a dataset from messages posted on Twitter using the Python Tweepy library, which allows us to use the Twitter APIs in an easy and simple way.

- ***Geographical research of the dataset.***

  From the dataset, we distribute the insomnia tweets between the Spanish-speaking countries determining the tweets corresponding to each of them. Specifically, in the case of Spain, we were able to estimate the importance of this issue in the Spanish user community.

- ***Development of an insomnia tweet classifier.***

  As we have commented in the first objective, the dataset from which we have started has been built by us, so we have had to design and create a pipeline that allows all these raw texts, which includes NLP techniques for the extraction of characteristics and machine learning algorithms for classification.

- ***Development of a second classifier to characterize tweets by theme.***

This classifier has a great importance in the development of the project, because it allows a detailed study with large amounts of data allowing, for example, to know the number of users who ask for help during sleepless nights or the main causes they believe is the reason for it or the number of them who say to follow some kind of treatment.

- ***Formation of a dataset of Spanish users with insomnia.***

  We made a dataset of users and also this time limited to Spain, which has allowed us to conduct a study of activity on Twitter by hours or the number of tweets of each of them related to insomnia.

- ***Development of a service to monitor insomnia on Twitter.***

  This last objective has allowed us to close the process developed throughout the objectives just mentioned. It has allowed us to display the two classifiers made, to add to the study an analysis of feelings and emotions and to make an interactive visualization interface to be able to draw conclusions from the data.

  All these developed with Luigi which allows us to have all the workflow automated and allowing us to have the capture, analysis and visualization phases in a continuous integration.

5.4 Problems faced

During the development of this project we had to face some problems. These problems are listed below:

- ***Learn the technologies used in the project.***

  The first problem we had to face was the lack of knowledge and experience in most of the technologies and libraries used throughout the project. This meant that the first phase of the project was dedicated to learning what these tools consisted of and how we should use them.

- ***Formation of the dataset.***

  Unfortunately, there is no public dataset with data on insomnia on Twitter in either English or Spanish, which meant that we had to make it up. The formation of this dataset led to another problem that is the limitations imposed by the Twitter API, which specify the number of requests you can do in a given period increasing the time needed to capture.

- ***Lack of data.***

  During the process of designing the classifiers, we noticed that the number of tweets corresponding to certain themes we wanted to characterize or to the symptoms of insomnia was very small in some cases. As we have already mentioned, after a series of filtered searches we were able to form the dataset of themes but the one of symptoms it was impossible for us to obtain a size that was sufficient to be able to train a classifier.

5.5 Future work

In the following section I will explain the possible new features or improvements that could be done to the project.

- ***Increase the dataset.*** Increase the size of our dataset to improve the performance of the classifiers and even create one to classify symptoms.
- ***Development and deployment of an assistance agent.*** Development of a Twitter bot capable of interacting with users, offering them updated help based on the data obtained from the monitoring service.
- ***Detailed study of each of the users that make up the dataset.*** Carry out a detailed study of each user's timeline, including a time variable that allows for the analysis over time of both the changes and the frequency of the emotions and feelings transmitted by the user [22].

APPENDIX A

Impact of this project

A.1 Introduction

The catalogue of different sleep disorders is one of the main problems that medicine faces today. The percentage of people suffering from any of these disorders is 31% in Western Europe, 56% in the USA and 23% in Japan [10].

However, it is estimated that only some these people are following some form of medical treatment. This fact makes necessary to create techniques so that doctors can more effectively predict and diagnose sleep disorders and be able to advise on healthy sleep routines and therapies.

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

A.2 Social impact

Social networks have become the most used online platforms on the Internet. Projects such as the one we have developed allow for an analysis of the content that is shared daily to draw conclusions about humanity.

In our case, the study and the tool developed on insomnia, will allow researchers who carry out projects on this subject or with social networks to use and start from the resources created (dataset, tools, etc.) and from the conclusions made. It can be considered that we have developed the beginning of a system that allows us to diagnose people with insomnia, which would bring about a significant improvement in the treatment of this type of disorder.

Another group interested in our project are companies whose business models deal with assistance to people with insomnia, because we provide them with detailed information on what their potential customers share.

In addition, the privacy of the users from whom all information used in the project has been collected has been respected at all times, as we have only used information shared by public Twitter users. We have also taken care not to store in the dataset any user name or photo that could relate a person to the information captured.

A.3 Economic Impact

In this section we will assess the possible economic impacts that users and companies using the monitoring system developed in this project may experience.

From the point of view of users who suffer from insomnia, a system such as ours may allow them to reduce the amount of money invested in treatments for the diagnosis of this disease.

From the point of view of the companies, using a system like ours that allows you to capture, analyse and visualize information in an automated way allows you to reduce by a considerable amount the costs required to perform each of the tasks we have just mentioned.

A.4 Environmental Impact

This section aims to define the main environmental impact of the development of a system such as ours.

Computers and other information technology infrastructures consume significant amounts of electricity, adding a huge charge on our electricity networks and contributing to greenhouse gas emission. In addition to this consumption, the energy required for the cooling system associated with this equipment must also be added, which is the second main reason for the consumption of this equipment.

The main environmental impact of this project is the high consumption of the server where it is deployed.

A.5 Ethical Implications

In this section we will evaluate the ethical implications of such a project.

The first ethical problem we face, related to the economic impact (Sec. A.3) in which we comment that our system allows companies to reduce costs, is that our system can destroy jobs.

We believe that a system such as ours provokes a transformation of jobs, because although the system does work that until now has been manual, it requires people to maintain it and analyse the results it presents.

On the other hand, another ethical discussion is due to the use of Twitter data to carry out research such as the one in this project. The privacy policy used by Twitter indicates that users consent to the collection, transfer, and storage of data that is public, while each user has the ability to change their account's privacy settings. As already mentioned, this project only analysed tweets that were completely public (i.e. the user did not select any privacy settings). Public data on Twitter is considered to be from a public data source [14].

APPENDIX B

Cost of the System

B.1 Introduction

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.2 Physical Resources

The budget for the physical devices necessary for the development of this project is mainly made up of a computer whose minimum requirements allow the development and deployment of the system.

The technical characteristics of this computer can be very varied, so as an example we present the characteristics of the computer where the project has been developed:

- ***CPU:*** Intel Core i5 3.2 GHz x 4
- ***RAM:*** 8 GB
- ***Disk:*** 500 GB

It is estimated that today a computer with these or similar features has a price of approximately 800 €.

B.3 Human Resources

In this section we are going to develop the part of the budget that is oriented to the cost of the workers to develop the system and its maintenance.

The first thing is to estimate the salary of a person who can carry out a project like ours. To this end, we have based ourselves on the 360 hours of dedication stipulated in the UPM Collaboration Scholarship, through which this project has been carried out, with an amount of 1,725 €.

On the other hand, we must also consider the salary of a person who is in charge of software system maintenance. For this function you need a Telecommunication Engineer or Computer Engineer with knowledge of machine learning and NLP. The salary of a worker with this profile is approximately 24.000 € per year.

B.4 Licences

This section includes the cost corresponding to the licences of the software tools necessary for the development and deployment of the system carried out in this project.

However, all the software used in this project is open-source, so the cost of software licences is zero.

B.5 Taxes

One of the scenarios in which taxes related to a software engineering project must be taken into account is the case in which the final product is sold to an interested company.

This sale is subject to a tax of 15% of the price of the product, as defined in Statute 4/2008 of Spanish law.

Bibliography

[1] Amayas Abboute, Yasser Boudjeriou, Gilles Entringer, Jérôme Azé, Sandra Bringay, and Pascal Poncelet. Mining twitter for suicide prevention. In *International Conference on Applications of Natural Language to Data Bases/Information Systems*, pages 250–253. Springer, 2014.

[2] Steven Bird and Edward Loper. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics, 2004.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[4] Stacey J Elder, Ronald L Pisoni, Tadao Akizawa, Rachel Fissell, Vittorio E Andreucci, Shunichi Fukuhara, Kiyoshi Kurokawa, Hugh C Rayner, Anna L Furniss, Friedrich K Port, et al. Sleep quality predicts quality of life and mortality risk in haemodialysis patients: results from the dialysis outcomes and practice patterns study (dopps). *Nephrology Dialysis Transplantation*, 23(3):998–1004, 2007.

[5] Lucia Falzon, Caitlin McCurrie, and John Dunn. Representation and analysis of twitter activity: A dynamic network perspective. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 1183–1190. ACM, 2017.

[6] Clinton Gormley and Zachary Tong. *Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine.* "O'Reilly Media, Inc.", 2015.

[7] John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95, 2007.

[8] Sue Jamison-Powell, Conor Linehan, Laura Daley, Andrew Garbett, and Shaun Lawson. I can't get no sleep: discussing# insomnia on twitter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1501–1510. ACM, 2012.

[9] Sanne Kruikemeier. How political candidates use twitter and the impact on votes. *Computers in Human Behavior*, 34:131–139, 2014.

[10] D Leger, B Poursain, D Neubauer, and M Uchiyama. An international survey of sleeping problems in the general population. *Current medical research and opinion*, 24(1):307–317, 2008.

[11] Alejandro Llorente, Manuel Garcia-Herranz, Manuel Cebrian, and Esteban Moro. Social media fingerprints of unemployment. *PloS one*, 10(5):e0128692, 2015.

[12] Kevin Makice. *Twitter API: Up and running: Learn how to build applications with the Twitter API.* "O'Reilly Media, Inc.", 2009.

[13] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing.* MIT press, 1999.

[14] David J McIver, Jared B Hawkins, Rumi Chunara, Arnaub K Chatterjee, Aman Bhandari, Timothy P Fitzgerald, Sachin H Jain, and John S Brownstein. Characterizing sleep issues using twitter. *Journal of medical Internet research*, 17(6), 2015.

[15] Wes McKinney. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython.* "O'Reilly Media, Inc.", 2012.

[16] Shintaro Okazaki, Ana M Díaz-Martín, Mercedes Rozano, and Héctor David Menéndez-Benito. Using twitter to engage with customers: A data mining approach. *Internet Research*, 25(3):416–434, 2015.

[17] Sherry Pagoto, Kristin L Schneider, Martinus Evans, Molly E Waring, Brad Appelhans, Andrew M Busch, Matthew C Whited, Herpreet Thind, and Michelle Ziedonis. Tweeting it off: characteristics of adults who tweet about a weight loss attempt. *Journal of the American Medical Informatics Association*, 21(6):1032–1037, 2014.

[18] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[19] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.

[20] Joshua Roesslein. tweepy documentation. *Online] http://tweepy. readthedocs. io/en/v3*, 5, 2009.

[21] J Fernando Sánchez-Rada, Carlos A Iglesias, Ignacio Corcuera, and Oscar Araque. Senpy: A pragmatic linked sentiment analysis framework. In *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*, pages 735–742. IEEE, 2016.

[22] Elizabeth M Seabrook, Margaret L Kern, Ben D Fulcher, and Nikki S Rickard. Predicting depression from language-based emotion dynamics: Longitudinal analysis of facebook and twitter status updates. *Journal of medical Internet research*, 20(5):e168, 2018.

[23] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[24] Gabriel Svennerberg. *Beginning Google Maps API 3.* Apress, 2010.