UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

DESIGN AND DEVELOPMENT OF A PERSONALITY TRAITS CLASSIFIER BASED ON MACHINE LEARNING TECHNIQUES

DIEGO BENITO SÁNCHEZ

2017

TRABAJO FIN DE GRADO

Título:	Diseño y Desarrollo de un Clasificador de Rasgos de Personalidad basado en Técnicas de Aprendizaje Automático
Título (inglés):	Design and Development of a Personality Traits Classifier based on Machine Learning Techniques
Autor:	Diego Benito Sánchez
Tutor:	Carlos A. Iglesias Fernández
Departamento:	Ingeniería de Sistemas Telemáticos

MIEMBROS DEL TRIBUNAL CALIFICADOR

Presidente:

Vocal:

Secretario:

Suplente:

FECHA DE LECTURA:

CALIFICACIÓN:

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

Departamento de Ingeniería de Sistemas Telemáticos
Grupo de Sistemas Inteligentes

TRABAJO FIN DE GRADO

DESIGN AND DEVELOPMENT OF A PERSONALITY TRAITS CLASSIFIER BASED ON MACHINE LEARNING TECHINIQUES

Diego Benito Sánchez

Junio de 2017

Resumen

Recientemente, durante los últimos años el uso de dispositivos digitales, con acceso a Internet, tales como smartphones o tablets, ha ido aumentando considerablemente en el día a día de las personas. Debido a esto, el uso de Internet y por tanto de las redes sociales también ha aumentado. En las redes sociales, los usuarios comparten datos personales que además de difundir contenido entre usuarios también transmiten implícitamente información útil para estudios sociológicos de las empresas, esto hace interesante la tarea de caracterizar a los usuarios (en este caso mediante la personalidad) a través de su actividad en redes sociales.

En anteriores estudios, se ha visto que la personalidad puede influir en varios aspectos: preferencias en estilos de interacción en el mundo digital ó géneros musicales, por poner unos ejemplos. Consecuentemente el diseño de interfaces de usuario personalizadas y sistemas de contenidos musicales recomendables pueden ayudarnos a ofrecer mejores experiencias a los usuarios.

Esta memoria es el resultado de un proyecto cuyo fin principal ha sido obtener un clasificador de rasgos personalidad, cuya tarea fue planteada en un workshop en el año 2014, trabajando sobre una base de datos de YouTube, utilizando el lenguaje de programación Python y desplegando el sistema mediante un plug-in en la plataforma Senpy.

Para desarrollarlo, se han utilizado herramientas de aprendizaje automático supervisado, técnicas de procesado del lenguaje natural (PLN) y consultas a la plataforma de análisis de emociones y sentimientos, Senpy.

En referencia al workshop, se han obtenido resultados comparables lo que hacen, que predecir la personalidad sea visto como algo prometedor de ahora en adelante.

Palabras clave: YouTube, Scikit-learn, Aprendizaje automático, PLN, Rasgos de Personalidad, Python, NLTK, Senpy, Sentimientos y Emociones.

Abstract

Recently, during the last few years the use of digital devices, with Internet access, such as smartphones or tablets, has been increasing considerably in people's day after day. Because of this, Internet usage and therefore social networks usage has also increased. In social networks, users share personal data to broadcast content between users and this also implicitly convey useful information for companies studies. This makes interesting the task of characterizing users (in this case using personality) through their activity in social networks.

In previous studies, there has been seen that personality can affect users in different aspects: preferences for interaction styles in the digital world or musical genres, for example. Consequently, the design of customized user interfaces and music recommender systems can help us to provide better experiences to users.

This thesis is the result of a project whose main aim has been to obtain a personality traits classifier, whose task was set in a workshop in 2014, working on a YouTube dataset, using Python as programming language and deploying the system as a Senpy plug-in.

During the development phase, there have been used supervised machine learning tools, natural language processing techniques (NLP) and queries to the sentiments and emotions analysis platform, Senpy.

Regarding the workshop, there have been obtained similar results, what means that predicting personality can be seen as a real possibility.

Keywords: YouTube, Scikit-learn, Machine Learning, NLP, Personality Traits, Python, NLTK, Senpy, Sentiments and Emotions.

Agradecimientos

En este apartado, me gustaría agradecer a las personas que han influido de manera positiva en mí durante los cuatro años que he estado estudiando el Grado y que han permitido que pueda finalizar esta titulación de la manera más satisfactoria posible.

En primer lugar, quiero dar las gracias a mis padres, por hacer posible que yo pudiera realizar estos estudios, tanto a nivel económico como a nivel anímico. También, mencionar a mi hermano Alejandro, por compartir parte de su tiempo en atender mis dudas y conocimiento sobre el desarrollo de páginas Web.

Por supuesto, considero que merecen una mención en este documento todos mis compañeros y amigos que he tenido durante mi vida en la universidad, todos ellos han hecho que este sufrido proceso sea más ameno. En especial a Héctor Ros, que me ha acompañado durante casi todos los cursos y trabajos que hemos tenido la oportunidad de realizar.

Además, quiero dar las gracias a mis compañeros del GSI, que desde el primer día me han tratado como si fuera uno más y siempre estaban disponibles para las dudas que me surgían.

Por último, me gustaría agradecer la labor de Carlos Ángel Iglesias, mi tutor, durante el desarrollo de este proyecto, ya que gracias a él he aprendido cosas realmente interesantes, no solo relacionadas con el proyecto y me ha prestado una ayuda fundamental para acabar el trabajo.

Contents

List of Figures

CHAPTER 1

Introduction

1.1 Context

Nowadays, almost everyone has a device connected to the Net and if we focus on younger people, we find that almost everyone shares personal information in social networks [25]. This information is the result of many interactions between the users and their activity on the Net like posting, friends or their network size.

Personality can be inferred from this information. The majority of attempts have explored user's profile and text blog data only [15]. This novel approach also uses audiovisual behavioral analysis on slices of conversational vlogs extracted from YouTube. Vloggers implicitly or explicitly share information about themselves that words, either written or spoken cannot convey.

The Workshop on Computational Personality Recognition 2014 [14] aimed at performing this task. It was a challenge-based shared task. The main goals of the workshop were to define the state-of-the-art and release data/tools for a standard evaluation of computational personality recognition. Organizers warmly invited contributors working in Personality Recognition, Data Mining, Computational Psychology, Natural Language Processing, Social Network Analysis, Sentiment Analysis, Opinion Mining, Mood Detection, Deception

Detection, Information Extraction, Human-Computer Interaction, and other related areas.

In order to do so, the work was based on the Big Five personality model that states that an individual is associated with five scores that correspond to the five main personality traits and that form the acronym OCEAN (Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism). In the next section, we will describe this model in more detail.

1.2 Big Five Personality Traits

The Big Five model [7] is a personality model that suggests five broad dimensions used by some psychologists to describe the human personality and psyche. The five factors have been defined as openness to experience, conscientiousness, extraversion, agreeableness, and neuroticism, often listed under the acronym OCEAN. Under each proposed global factor, appear a number of correlated and more specific primary factors.

- **Openness to experience:** this trait appears on people who has appreciation for art, emotion, adventure, unusual ideas, curiosity, and a variety of experience. Openness reflects the degree of intellectual curiosity, creativity and a preference for novelty and variety a person has. It is also described as the extent to which a person is imaginative or independent. High openness can be perceived as unpredictability or lack of focus.

- **Conscientiousness:** people who have a high score in this dimension has a tendency to be organized and dependable, show self-discipline, act dutifully, aim for achievement, and prefer planned rather than spontaneous behavior.

- **Extraversion:** Extraversion is shown to be similar to Openness, so Energy, positive emotions, assertiveness, sociability and the tendency to seek stimulation in the company of others, and talkativeness are some of the characteristics of this trait.

- **Agreeableness:**. agreeable people tend to be compassionate and cooperative rather than suspicious and antagonistic towards others. It is also a measure of one's trusting and helpful nature, and whether a person is generally well-tempered or not.

- **Neuroticism:** The tendency to experience unpleasant emotions easily, such as anger, anxiety, depression, and vulnerability. Neuroticism also refers to the degree of emotional stability and impulse control and is sometimes referred by its low emotional stability.

1.3 Project goals

In general issues, this project aims at building a personality traits classifier using the ideas described in the workshop. With this aim, the project will use manually video transcribed texts [13].

Among the main goals of this project, we can find:

- Transform the Youtube dataset so that it can be used by the different algorithms.
- Extract semantic features from the transcriptions and include them into the dataset.
- Extract sentiments and emotions from the transcriptions and include them into the dataset.
- Use different algorithms, features and compare the results with the workshop results in order to improve the classifier and draw conclusions.
- Finally, we have been forced to design a gender classifier for reasons that we will attend later.

1.4 Structure of this document

In this section, we provide a brief explanation of the chapters included in this thesis. The structure follows this schema:
Chapter 1 explains the context in which this project is developed and the Big Five personality model. Moreover, it describes the main goals to achieve in this project.
Chapter 2 describes the main technologies used on this project. We will see in depth some Python libraries such us Scikit-Learn, Pandas or NLTK and we will also refer to Senpy.
Chapter 3 provides a general description of the classifier construction and evaluation according to the metrics later defined.
Chapter 4 explains the YouTube personality service that has been created from the the data source provided.
Chapter 5 discusses the conclusions drawn from this project, problems faced and suggestions for a future work.

CHAPTER 2

Enabling Technologies

2.1 Introduction

There are two mainly technologies used during the development of this project: Machine Learning [19] and Natural Language Processing [20]. It is usual to combine both techniques when our first aim is getting some special information from a text source. With Natural Language Processing we can extract features from a text (words, sentences, tokens, lemmatization and stemming) and given these features we can apply Machine Learning to obtain a predictive model about we tried to infer.

Machine Learning is a sub-field of computer science and a branch of artificial intelligence, whose main objective is to perform some techniques, enabling computers to *learn* and create predictive models from a data source. These techniques can be classified into supervised learning and unsupervised learning.

In unsupervised learning, no labels are given to the computer, so it is computer problem to work on its own and find patterns in the data structure. It can be used to discover patterns in data or for feature selection.

In this project, we use supervised language techniques. They consist on presenting the

computer example inputs and their desired outputs and the goal is to learn a general rule that maps inputs to outputs.

Regarding Natural Processing Language it can be defined as a field of computer science, artificial intelligence and computational linguistics that studies the interactions between computers and human languages. It can be used to obtain syntax, semantic or discourse features. Now we will continue showing how we can use these techniques into the Python programming language.

2.2 Pandas

Pandas [6] is an open source, BSD-licensed, Python data analysis library that provides fast, flexible, and expressive data structures. It is built on top of NumPy and is intended to integrate well within a scientific computing environment with many other 3rd party libraries. The two primary data structures of pandas are Series (1-dimensional) and DataFrames (2-dimensional).

- **Series** is a one dimensional labelled object, capable of holding any data type (integers, strings, floating point numbers, Python objects, etc.).. It is similar to an array, a list, a dictionary or a column in a table. Every value in a Series object has an index.
- **DataFrames** is a two dimensional labelled object with columns of potentially different types. It is similar to a database table, or a spreadsheet. It can be seen as a dictionary of Series that share the same index.

In the project, Pandas is used to read different CSV files and merge them into only one DataFrame structure so that the different algorithms can work properly. Here are some of the functions that Pandas implements.

- **DataFrame** object for data manipulation with integrated indexing.
- **Reading and writing data** in different formats: CSV and text files, Microsoft Excel, SQL databases and the fast HDF5 format.
- Intelligent **data alignment** and integrated handling of **missing data**.
- Intelligent label-based **slicing**, **fancy indexing**, and **subsetting** of large data sets.
- Columns can be inserted and deleted from data structures for **size mutability**.

- Aggregating or transforming data with a powerful **group by** engine allowing split-apply-combine operations on data sets.
- High performance **merging and joining** of data sets.

2.3 Scikit-learn

Scikit-learn [9] is an open source, BSD-licensed, Python library providing simple and efficient tools for data mining and data analysis. It is built on NumPy, SciPy, and matplotlib. Scikit-learn implements a range of machine learning, preprocessing, cross-validation and visualization algorithms.

Scikit-learn can perform classification (identifying to which category an object belongs to), regression (predicting a continuous-valued attribute associated with an object), 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.) and preprocessing (feature extraction and normalization). In this final work we focus on classification and regression.

- **Classification:** the aim is to assign each input vector to one of a finite number of discrete categories. Another way to think of classification is as a discrete (as opposed to continuous) form of supervised learning where one has a limited number of categories and for each of the n samples provided, one is to try to label them with the correct category or class.
- **Regression:** if the desired output consists of one or more continuous variables, then the task is called regression. So the main difference between classification and regression is the format of the labels, discrete or continuous respectively.

Depending if we perform classification or regression we refer to the classes as labels or values respectively. In classification and regression Scikit-learn follows the schema showed in the Fig. 2.1.

The first step is to split the original dataset into training and test sets, then using the training set we can create a model to predict new values or labels. We predict new values (or labels) using the model previously created and the test data set. Finally, we can compare the predicted labels and the expected labels to measure the classifier accuracy.

Scikit also provides some packages for extracting vectorizing features from texts. Special interest comes from the TF-IDF (Term Frequency - Inverse Document Frequency) vector-

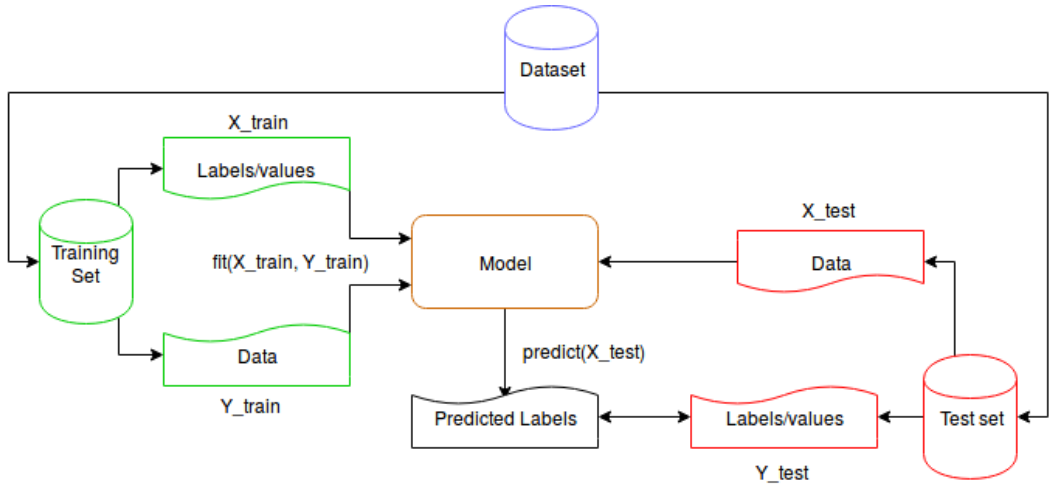

Figure 2.1: Data splitting and output prediction

izer, in order to know the words importance used in a set of documents. It works converting the textual representation of information into a Vector Space Model. This is an algebraic model that represents textual information as vectors, being their components the importance of a term. The aim is modeling documents into a vector space ignoring ordering of words but focusing on information about occurrences of words.

The first step is creating a model of the documents into a vector space creating a dictionary of terms that appear on them. The TF-IDF value increases proportionally to the number of times a word appears in the document, but this value is contrasted by the frequency of the word in the corpus, which helps us to adjust according to the fact that some words appear more frequently in general. We cannot forget that some existing words do not contribute any information, these are called stop words and are formed mainly by prepositions, pronouns and articles. These words need to be preprocessed so that our learning algorithm does not consider them.

It is also necessary to preprocess documents in another way. It consists in extracting the root or lemma of each word, this is because there is more information on a specific concept than on its variations, what could mean introducing some more noise.

The next step is calculating the TF or term frequency to represent each term into the vector space. The term frequency is the number of appearances of a specific term in the vocabulary. So, for each term of the vocabulary appearing on all the documents, each document will have a matrix of their appearances on it. This way, each document of the set is represented by a vector with zeros on the terms that did not appear and the number of appearances on the terms that did appear on it.

Once we have these vectors, it is important normalizing them because the importance of a word appearing a number of times depends on the text length. Now we are in a good position to calculate the inverse document frequency.

The inverse document is defined as: $log\frac{|D|}{1+|d:t\in d|}$ being $1 + |d : t \in d|$ the number of documents where the term t appears and $|D|$ the number of documents in the corpus. This function could variate, as it tries to get the impact of a word in the corpus and in this case it smooths it by computing it into a logarithmic scale.

When we have both calculated term frequency and inverse document frequency, we multiply both values and we obtain the TF-IDF value. So finally, a high TF-IDF is reached with a word with a high frequency in the document and with a low frequency in terms of the whole set. As a result, for each doc, we have a vector composed of the correspondence of words and its TF-IDF value. Applying these ideas we can start to extract text features from video transcriptions.

Finally, Scikit also perform some methods of measures and metrics to show the learning quality.

2.4 NLTK

NLTK [2] (Natural Language Toolkit) is a set of libraries and programs used in Natural Processing Language, NLP, written in the Python programming language.

It provides a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum. Here are some functions that NLTK implements.

- Lexical analysis: word and text Tokenizer.
- n-gram and collocations.
- Part-os-speech tagger.
- Tree model and text chunker for capturing.
- Named-entity recognition.

2.5 Senpy

Senpy [4] is a sentiment and emotion analysis server in Python. Senpy lets you analyze sentiment and emotions from a text input through a web interface, Senpy also accepts request using its simple API. It works providing the results in a JSON format. Senpy also allows to deploy new plug-ins from a developed analysis algorithm.

A relation between emotions and personality traits has been observed in past research as well [17]. The reason for including emotional features is that people with different personality traits will express themselves differently and, hence, will use different words (phrases) and emotions (anger, joy, etc) when expressing themselves. In our project, we use Senpy to extract sentiments from the video transcriptions, so can help us in both tasks classification and regression.

2.6 Related work

In this section, we show the contributors who participated in the contest and performed relevant results. All of them are groups of two to six people formed by investigators from different departments from different universities. But first, we are going to show the competition baselines suggested for the partcicipation in the workshop. The shared task was structured in two competitions:

- **Competition on multimodal personality recognition.** Contributors were required to use the Youtube personality dataset and report their predictions.
- **Competition on personality recognition from text** Contributors are required to use only the text transcriptions from the Youtube personality dataset and report their predictions.

The organizers of WCPR2014 set a reference baseline [1] to stimulate competitors overcoming their results.

Two levels of baselines were proposed for the classification task, and the metrics to overcome were: precision, recall and fi-score. These metrics are deeply explained in Sect. 3.7. These two baselines range from a basic one just better than a random one, to harder one without errors in terms of the recall score. Regarding the regression task, they used the root mean squared error metric. All this information is given in Table 2.1.

[1] These baselines are the same for both competitions.

Table 2.1: Baselines

——	Easy baselines			Hard Baselines			Regresion baselines
Trait	**Precision(Avg)**	**Recall(Avg)**	**F1(Avg)**	**P**	**R**	**F1**	**RMSE**
Extraversion	0.35	0.5	0.41	0.7	1	0.82	1.02
Neuroticism	0.29	0.5	0.37	0.59	1	0.74	0.75
Agreeableness	0.25	0.5	0.33	0.5	1	0.67	0.91
Conscientiousness	0.38	0.5	0.43	0.75	1	0.86	0.71
Openness	0.34	0.5	0.4	0.68	1	0.81	0.83
Average	0.32	0.5	0.39	0.64	1	0.78	0.84

2.6.1 A Multivariate Regression Approach to Personality Impression Recognition of Vloggers

This competitor [16] performed the first competition in the regression format and used the following features: gender, audio-video cues, LIWC (Linguistic Inquiry and Word Count), NRC, MRC, SentiStrength and SPLICE (Structured Programming for Linguistic Cue Extraction).

Due to the high correlation among the five personality dimensions given in the YouTube dataset, they decided to perform a multivariate regression, where the dependencies between the target variables are taken into account to make a combined prediction. In Table 2.2 we show the RMSE for each trait and for each algorithm employed.

Table 2.2: Multivariate Regression results

Algorithm	**Extraversion**	**Agreeableness**	**Conscientiousness**	**Neouroticism**	**Openness**
ST	0.91	0.72	0.65	0.70	0.80
MTS	0.95	0.76	0.66	0.71	0.77
MTSC	0.91	0.73	0.64	0.70	0.79
ERC	0.93	0.74	0.65	0.72	0.79
ERCC	0.91	0.72	0.65	0.70	0.80
MORF	0.98	0.84	0.64	0.75	0.83

It can be observed that all 6 algorithms are able to outperform the average baseline model for all 5 personality traits. This is interesting because previously published methods for the same dataset showed an improvement over the baseline for the majority of personality traits, but not for all simultaneously.

2.6.2 Evaluating Content-Independent Features for Personality Recognition

This group [26] participated using only text transcriptions and performed the classification task, as we do, so it can be interesting compare our results with theirs. They used content-independent features such us: token unigrams (frequency threshold), character trigrams (frequency threshold 10), LIWC features and a feature set that was shown to work well for gender prediction (character-based, word-based, sentence-based, dictionary-based and syntactic features).

They used the training set to perform tenfold cross-validation experiments with different types of features in order to tune the parameters of the machine learning algorithm and to establish which features work best for the detection of each trait. They performed support vector machine algorithm from Scikit-learn library. The results that they obtained are shown in Table 2.3.

Table 2.3: Content Independent

Class	**Precision(Avg)**	**Recall(Avg)**	**F1(Avg)**
Extraversion	0.49	0.49	0.48
Neuroticism	0.61	0.61	0.61
Agreeableness	0.69	0.68	0.68
Conscientiousness	0.45	0.46	0.45
Openness	0.50	0.50	0.49
Average	0.55	0.55	0.54

They did not overcome the baseline and they accused that the provided dataset consisting of video blog transcripts is too small for the task of personality recognition. Previous research has shown that this task is very hard even with large amounts of data. In addition tuning the parameters probably means that the systems are overfitted and does not work

good enough for new instances of data.

2.6.3 The Impact of Affective Verbal Content on Predicting Personality Impressions in YouTube Videos

This contributor [18] used the multimodal option to participate in the classification task of the competition. They used four methods to perform a 10-fold cross-validation Support Vector Machine (SVM) from WEKA toolkit using polynomial kernel.

- A. Baseline - Audiovisual + Gender.
- B. Audiovisual + Gender + Coarse-grain Emotions.
- **C**. Audiovisual + Gender + Coarse-grain Emotions + Fine-grain emotions(Valence).
- D. Audiovisual + Gender + Coarse-grain Emotions + Fine-grain emotions(Frequency/total number of affective words).

The results concerning the effects of the chosen methods on performance metrics appear in Table 2.4

Table 2.4: Precision, recall and F1 measures of the SVM personality classifier across all four methods

Class	Method A	Method B	Method C	Method D
	P R F1	P R F1	P R F1	P R F1
EXTR	56.3 75.0 64.3	64.9 73.2 68.8	82.3 76.8 79.5	82.3 76.8 79.5
AGR	76.4 76.8 76.6	78 78.6 78.3	71.2 73.2 72.2	75.8 76.8 76.3
CON	34.3 57.1 42.9	48.6 50.0 49.3	47.7 46.4 47	52.1 51.8 51.9
NEU	68.2 67.9 68	73.5 73.2 73.3	73.2 73.2 73.2	73.2 73.2 73.2
OPE	45.2 64.3 53.1	53.7 64.3 58.5	56.8 66.1 61.1	80 71.4 75.5
AVG	46.7 56.8 61.0	53.1 56.5 65.7	66.2 67.1 66.6	72.7 70.0 71.3

They concluded stating that a selection of audio and video features can be useful in predicting the Extroversion trait while selecting a multimodal set of the most significant

audio-visual features and facial expressions of emotions contributes to a successful prediction for Extroversion and Openness to Experience.

2.6.4 Predicting Personality Traits using Multimodal Information

In this case, we have a contributor [11] who participated in the multimodal competition solving the classification problem. So they experimented with audio-visual cues, features extracted from text transcriptions: lexical POS (Part Of Speech), psycholinguistic and emotional features and also using traits as features. After this combination, they applied a feature selection in order to improve the performance.

They generated a classification model for each feature set described above using Sequential Minimal Optimization(SMO), which solves the Quadratic Optimization (QP) problems, for Support Vector Machine (SVM). They used different kernels for different features sets, such us linear kernel for lexical and POS features and polynomial kernel for audio-visual, psycholinguistic, emotional and traits features. They tuned the parameters using 10-fold cross-validation on the training set. Because of the brevity, they only present the F1 score. All the results are contained in Table 2.5 and Table 2.6

They obtained comparative results among the AV, Lex, POS and LIWC feature sets but the emotional feature set (Emo) does not perform well individually. The decision combination provides better results compared to the results of any single feature set. Except for extraversion, the performance of traits features is lower compared to any other set.

Table 2.5: Results on test set using different feature sets. Baseline: Official baseline, AV: Audio-Visual, Lex: Lexical, POS: Parts-Of-Speech, LIWC: psycholinguistic, Emo: Emotion, Maj-5: Majority voting of the five models, Maj-4: Majority voting of the four best models, Maj-5-Traits: Generated traits labels using Maj-5 mode

Model	**O**	**C**	**E**	**A**	**N**	**Avg**
Baseline	40.4	42.9	41.1	33.3	37.1	39.0
AV	63.4	42.9	70.4	67.7	55.7	60
Lex	59.9	49.4	60.4	65.8	56.7	58.4
POS	57.3	54.3	57.8	69.6	61.9	60.2
LIWC	55	56	66.2	71.4	46.8	59.1
Emo	49.3	52.5	53.5	59.4	40.1	51
Maj-5	65	57.4	69.4	76.7	59.4	65.6
Maj-4	61.5	61.9	68.8	74.7	57.1	64.8
Maj-5-Traits	59.2	41.7	71	62.2	52.6	57.3
Avg	65	61.9	71	76.7	61.9	67.3

Table 2.6: Results on test set using traits as features. Ref: Reference labels of the test set. Maj-5-Traits: Generated traits labels using Maj-5 model

Model	**O**	**C**	**E**	**A**	**N**	**Avg**
Ref	77.1	41.7	77.1	58.6	58.6	62.6
Maj-5-Traits	59.2	41.7	71	62.2	52.6	57.3

2.6.5 Extrapolation to this project

Now we have already seen the different participators and the results achieved, so we can extract conclusions about which algorithms and features are candidates to be used to our

shelf performance. First of all, we must have in mind that we will only use text transcriptions in order to perform classification and regression tasks, i.e. exactly as Sect. 2.6.2, but in this case as we have told, we only add the regression performance.

Regarding the regression task, we have only one contributor to compare, this contributor addressed a very sophisticated regression which is not presented among the regressions provided by Scikit-learn so despite using the dependencies between the five traits we will perform regression for each trait independently trying to outperform baseline results and multivariate regression results.

On the classification case, we have more factors to have in account, thus we can start seeing all of them more deeply. First, content independent features combined with SVM classifier does not seem to obtain a great result, so we will try to add sentiment features and use other available algorithms presented in Scikit. Besides, we have multimodal (audiovisual) features appear to improve the results, but we cannot use them because we do not dispose of audiovisual recognizers. Finally, we have that emotional features perform good results.

CHAPTER 3

Model construction and evaluation

3.1 Introduction

During the last few years, it has become popular being a vlogger. Vlogger is a term that can be defined as a person who shares self-maked videos, in which they appear, in front of a camera usually into his room at home, narrating something interesting for a specific audience. Many people try to achieve this purpose but not all have success. If they finally are good at talking in front of the camera they usually have miles of followers and can earn a living, that is why this idea is so attractive.

Miles of these characters appear in social networks such us YouTube, so they can be viewed by their followers. Youtube is a crowdsourced shared-video social network where users can allocate their creations. There is a huge variety content since videoclips and tv shows to on-air content and what is know as youtubers (vloggers on YouTube). YouTube also allows users to share their opinion from other videos uploaded.

It is interesting to know a person through viewing his videos and comments that receive, but due to we have already stated, many people are sharing content and if we want to know about a big amount of people we have a time problem. The issue of finding automatically the different characteristics that defines a person appears at this point and is what we try

to do with the personality in our case.

In this chapter, we are going to explain the implementation of a personality traits classifier using a YouTube dataset [21]. The YouTube personality dataset consists of a collection of behavioral (audiovisual) features, speech transcriptions, and personality impression scores for a set of 404 YouTube vloggers that explicitly show themselves in front of the a webcam talking about a variety of topics including personal issues, politics, movies, books, etc. There is no restriction about the content and the language used is natural and diverse.

First of all, in the overview, we will present a general vision about the steps that have been followed. We will continue focusing on each step explaining the main tasks and goals of each one and the reasons why they are necessary.

3.1.1 Overview

In this section, we will present the global overview of the project, defining the different steps that have been followed for the creation of the project. Just a general sight of the system in order to anticipate what later comes.

The aim is to create a classification system that can predict the personality traits, in both formats continuous and binary, from a text input. In order to achieve that, we will implement a system based on Machine Learning Techniques explained in the previous section.

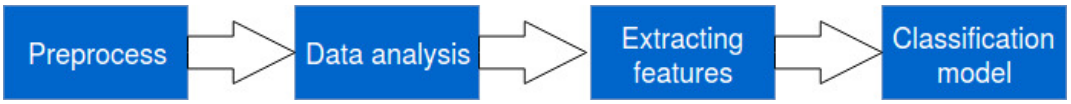

Figure 3.1: Stages followed

As we can see in the Fig. 3.1 there are four main steps: preprocess, data analysis, extracting features and classification model. Preprocess is necessary because of the dataset is composed of several CSV files and convert them into one Dataframe. Once we have only one Dataframe we can start to work in the data analysis, focusing on the traits we will see some summary stats, correlations, distributions, etc. The next step is to extract features for the classification and regression task, we will use the TF-IDF vectorizer, lexical stats, POS (Part of Speech) stats and features extracted from Senpy.

At this point, we are in disposition of performing classification and regression, but before is important to notice that TF-IDF maps the Vector Space Model into a sparse matrix and some estimators cannot work properly in these conditions so we must transform the sparse

matrix into a dense matrix. Having done all this it is time for prediction. We will use the split of the data into 348 training and 56 test instances that was suggested by the organizers of WCPR2014 and different algorithms that will be explained later.

3.2 Initial research

It all started in the year 2013 when Daniel Gatica-Perez and Joan-Isaac Biel completed a study [12] with a novel approach in personality prediction, including audiovisual behavioral analysis on vlogger's video frames. They addressed four previously unexplored issues:

- The feasibility of collecting vlogger personality impressions using crowdsourcing techniques, generating the YouTube dataset in a framework in which ordinary, diverse people (as opposed to trained annotators) make first impressions while they watch thin slices of video.
- They study how personality impresions mediate the vlog watching experience using several measures of attention from YoouTube's available metadata.
- They investigated to what extent non verbal cues (audiovisual) are useful as a lens through which personality impressions can be made.
- They addressed the task of automatic predicting vloggers personality impressions using multimodal nonverbal cues and machine learning techniques.

Fig. 3.2 summarizes the technical blocks of their approach. They started by preprocessing a set of conversational vlogs from YouTube to create "thin slices" of behavior with the extraction of the first conversational minute. On one hand, they use these vlog slices to obtain vloggers personality judgments from Mechanical Turk workers. On the other hand, they process the slices to automatically extract nonverbal cues from audio and video. Then, they divide the study into four different blocks. First, they analyze the work of MTurk annotators and study the agreement of personality impressions. Second, they investigate the relation that these impressions have with social attention, measured from YouTube metadata. Third, they measured the level of cue utilization of automatic nonverbal cues from vloggers as lenses that mediate the personality impressions of observers. Finally, they addressed the task of predicting personality from vloggers from automatically extracted nonverbal behavior.

They concluded stating that MTurk may be suitable to collect vlogger personality annotations. On the analysis of personality and social attention, they found evidence that

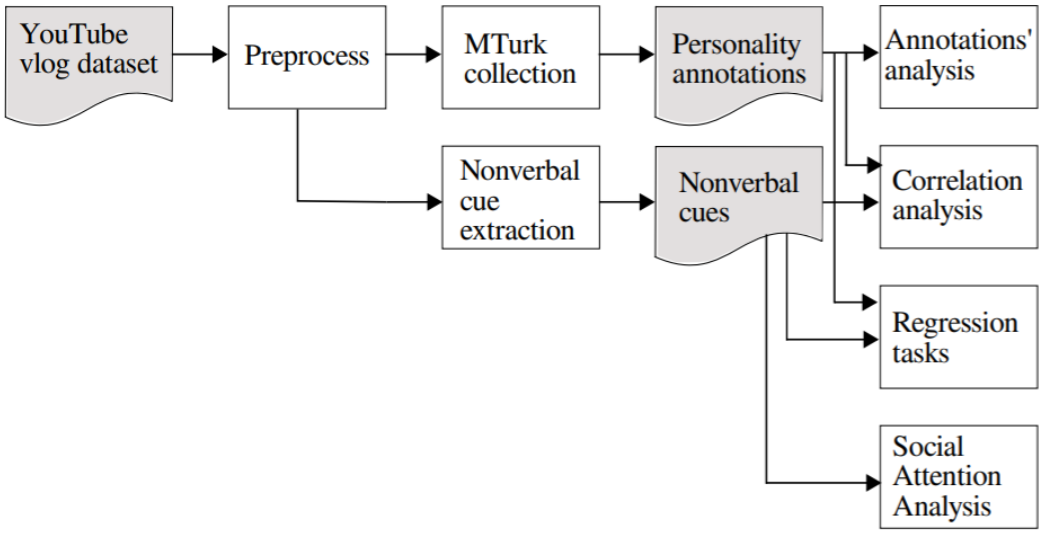

Figure 3.2: Overview for the study of personality impressions in YouTube vlogs using nonverbal cues. Figure extracted from [12]

personality impressions might mediate the YouTube vlog watching experience in a way that certain vlogger traits result on audiences watching, commenting rating and favoriting their videos more. Regarding cue utilization, they found that Extraversion is related with audio and visual cues, and Conscientiousness and Agreeableness are related to more visual information.

3.3 Preprocess

The WCPR2014 organizers provided the participants 404 data samples available in the form of YouTube video monologues (vlogs) made available by the IDIAP Research Institute. Each video is accompanied with a set of audio features that measure speech activity and prosody, and a selection of video cues related to looking activity, pose and overall motion of the vlogger (Biel and Gatica-Perez, 2013). A transcript of each vlog and the gender of the video creators were also provided. Personality perception labels were collected via crowdsourcing on Amazon's Mechanical Turk platform. The judges were self-selected from the varied population of users, or 'Turkers'. The inventory used was the TIPI [22], whose ten items (2 per dimension on a 7-point likert scaler) are designed and validated for personality in constrained environments. The judges, five per video, completed the items after viewing only the first minute, leading to first impression style assessment of personality. The crowdsourced impressions of each vlogger's personality traits were aggregated and classified

along the Big Five dimensions. It contains 348 training and 56 test instances, composing the 404 vlogs in total.

All this information is given in three CSV files (gender, impression scores in continuous format and impression scores in binary format) and the transcripts directory whose content are 404 TXT files keeping into each one the transcription associated with each instance by a vlogId. Now we find the problem of unifying all the provided information in a single structure with and adequate format in order to work easily in later tasks. Here appears Pandas library [6], which provides us facilities to read files, merge information and encode values.

The first step is to load all the data (the three CSV files as three DataFrame types) and merge them into only one DataFrame attribute. Once we have a DataFrame as a reference we can include the remaining information, ie the vlogs transcripts. But before is necessary to remove all non ASCII characters, this is because some learning plug-ins from Senpy do not accept some special UTF-8 encoded characters. As a result, we obtain something similar to the Fig. 3.3.

	vlogId	Transcription	gender	Extr	Agr	Cons	Emot	Open	Xc	Ac	Cc	Ec	Oc
0	VLOG1	You know what I see - - no, more like hear a l...	Male	4.900000	3.70	3.600000	3.200000	5.500000	n	n	n	n	y
1	VLOG3	Hey there, if you are watching this movie you ...	Female	5.000000	5.00	4.600000	5.300000	4.400000	y	y	y	y	n
2	VLOG5	Doing a standard re tweet themed, so web two p...	Male	5.900000	5.30	5.300000	5.800000	5.500000	y	y	y	y	y
3	VLOG6	Hello, and welcome to the XXXX. My name's XXXX...	Male	5.400000	4.80	4.400000	4.800000	5.700000	y	n	y	n	y
4	VLOG7	Hey you guys, um, I decided for this video I w...	Male	4.700000	5.10	4.400000	5.100000	4.700000	n	y	y	y	n
5	VLOG8	Hey guys. Well, I just got back from a Miley C...	Female	5.400000	4.80	3.800000	4.100000	4.200000	y	n	n	n	n
6	VLOG9	Hey guys, it's Monday. Have you ever recorded ...	Female	5.600000	5.00	4.000000	4.200000	4.900000	y	y	y	n	n
7	VLOG10	Hey everybody, it's Monday, July twenty sevent...	Male	5.400000	4.20	5.600000	4.600000	4.200000	y	n	y	n	n
8	VLOG11	Hello, everyone. It's XXXX. Hi, XXXX. This mor...	Male	2.900000	4.20	5.200000	5.100000	4.100000	n	n	y	y	n
9	VLOG12	So, um, ehh, I don't know -- I don't know what...	Male	4.500000	2.90	2.200000	3.600000	4.600000	n	n	n	n	n

Figure 3.3: Reference DataFrame

The next thing to do is to encode the categorical values, in our case are the gender column and the personality traits in the binary format (Xc, Ac, Cc, Ec, Oc). It is better to encode features as continuous variables, since Scikit-learn estimators expect continuous input, and they would interpret the categories as being ordered, which is not the case. We have chosen the following encoding values:

- For the gender column we have encoded “Male” as “0” and “Female” as “1”.
- For the personality traits we have encode “n” (no) as “0” and “y” (yes) as “1”.

Once we have finalized the preprocess step, we are in a good position to continue with the following tasks.

3.4 Data analysis

In this section, we are ready to perform an effective data analysis from the dataset under study. We will use some Pandas facilities tools that help us in the realization of the cited analysis, we will also be able to represent the distribution of the data in order to visualize the most important data aspects.

We can start presenting some summary statistics such us, the mean, the standard derivation, the minimum value and the maximum value. The Neuroticism trait is given by the emotional stability metric, so a high score in emotional stability means a low score in Neuroticism.

Table 3.1: Continuous scores statistics

Measure	**Extr**	**Agr**	**Cons**	**Emot**	**Open**
mean	4.624870	4.682508	4.497177	4.766231	4.664250
std	0.978368	0.880479	0.771138	0.779986	0.716353
min	2	2	1.9	2.2	2.4
max	6.6	6.5	6.2	6.5	6.3

In Table 3.1 we can observe the statistics corresponding to the continuous values, these values are into the seven-point scale, but do not fill all the range. All values have a similar mean, so along the 404 instances they have similar punctuations and no one stands out over the others.

Table 3.2: Binary scores statistics

Measure	**Extr**	**Agr**	**Cons**	**Emot**	**Open**
mean	0.403465	0.457921	0.782178	0.440594	0.349010
std	0.491202	0.498844	0.413277	0.497074	0.477248
min	0	0	0	0	0
max	1	1	1	1	1

If we do something similar with the traits in the binary format, we obtain the results presented in Table 3.2. In this case, the minimum and maximum value correspond to the both possible labels, no and yes respectively. Unlike the personality traits in the continuous format, now we have that Conscientiousness is a very representative trait (78% of vlogs have this trait) and Openness is a very poor representative trait (only 34% of vlogs have this trait). The remaining traits are more or less represented with values between forty and fifty percent.

Another interesting data could be the pairwise correlation between traits. The correlation is a measure of the relationship strength between the different columns in a dataset. In our dataset, we have the following Pearson correlation coefficients, presented in Table 3.3 and Table 3.4.

Table 3.3: Correlation in the continuous scores

——	**Extr**	**Agr**	**Cons**	**Emot**	**Open**
Extr	1	——	——	——	——
Agr	0.015282	1	——	——	——
Cons	-0.027350	0.382870	1	——	——
Emot	0.058394	0.690256	0.535069	1	——
Open	0.560913	0.287272	0.256756	0.297712	1

Table 3.4: Correlation in the binary format

——	**Extr**	**Agr**	**Cons**	**Emot**	**Open**
Extr	1	——	——	——	——
Agr	0.074522	1	——	——	——
Cons	-0.030498	0.268371	1	——	——
Emot	-0.028627	0.435211	0.323383	1	——
Open	0.392825	0.160857	0.122196	0.197445	1

We can observe that, correlation coefficients in the binary format are lower than correlation coefficients in the continuous format, that is why the granularity level to represent the data is much lower in the binary format. In the binary format we only use two labels to represent all the data, meanwhile, with the continuous format, we have infinite values into the [1, 7] interval to represent the data. Anyway, in both tables, we can see the same correlations and we make emphasis in:

- Extraversion is correlated with Opennes.
- Emotional stability is correlated with Agreeableness and Conscientiousness.

In the Fig. 3.4 is represented the distribution of each trait through a histogram plot, we also see that correlated traits have similar aspect.

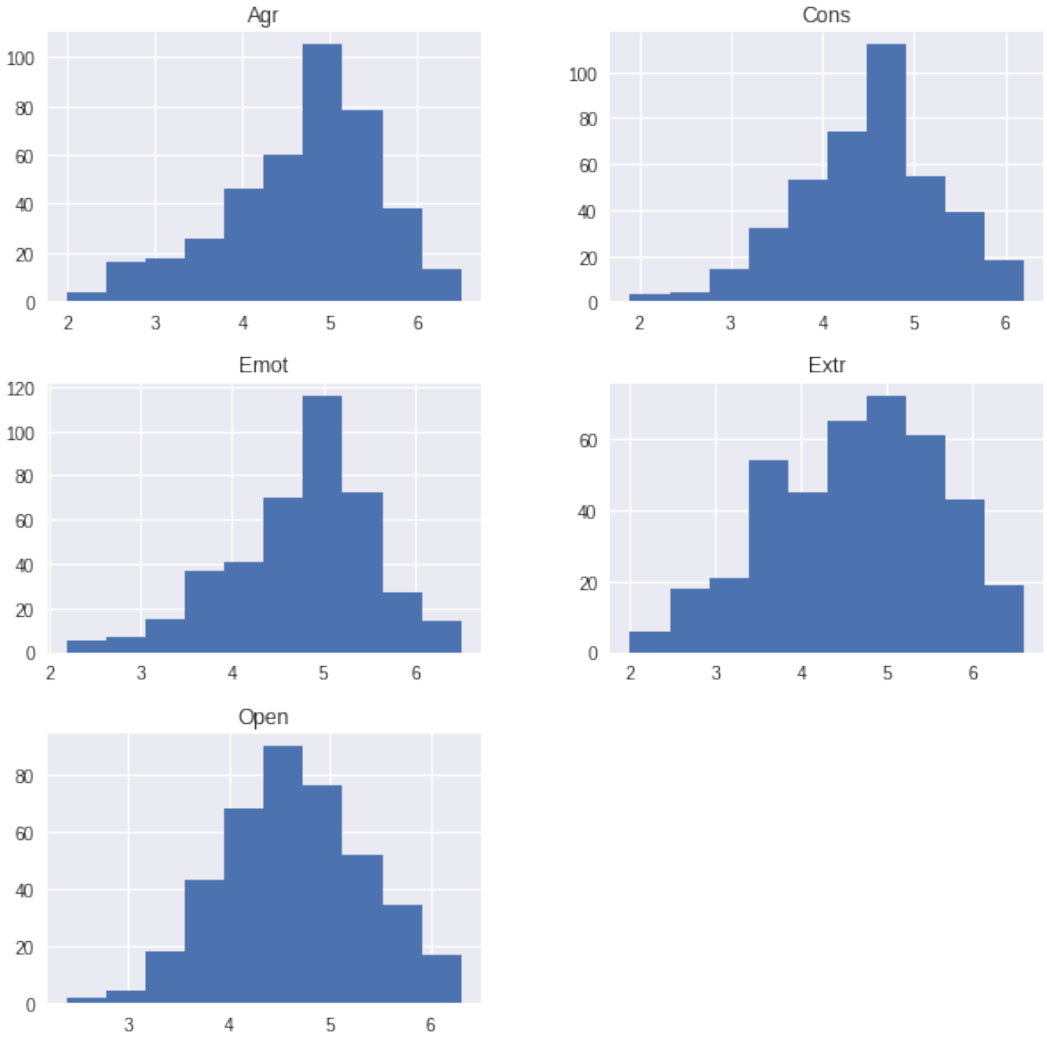

Figure 3.4: Personality traits histogram distributions

Agreeableness, Conscientiousness and Emotional stability are left skewed with a thin shape around the five value in the x-axes. Openness and Extraversion are also left skewed but have a more dense shape.

Finally, if we look from the users perspective we can observe how are the users distributed along the five traits in Table 3.5. Only the ten percent is not represented and the majority have two or three traits, so we have a well represented dataset.

Table 3.5: Users distribution along the five traits

Description	**Users**	**Percentage**
Users who do not have any trait	40	9.9%
Users who have one trait	72	17.82%
Users who have two traits	96	23.76%
Users who have three traits	97	24.01%
Users who have four traits	67	16.59%
Users who have five traits	32	7.92%

We can do something similar with continuous values if we sum the punctuation along the five traits we obtain the results shown in Table 3.6. And we arrive at the same conclusion, most users have a punctuation between 21 and 27 what it means a score per trait between four and five.

Table 3.6: Users punctuation summing the five scores

Description	**Users**	**Percentage**
Users who have a punctuation under 15	2	0.5%
Users who have a punctuation between 15 and 18	17	2.92%
Users who have a punctuation between 18 and 21	67	16.58%
Users who have a punctuation between 21 and 24	149	36.88%
Users who have a punctuation between 24 and 27	143	35.4%
Users who have a punctuation grater than 27	31	7.67%

3.5 Feature Extraction

Once we have analyzed the data, the next step is to obtain features from our unique information source input, the transcripts corpus. From a computer's point of view, the raw text gives no information at all. In order to get that information, we must perform some language processing techniques. Those consist in separating the text into words and perform some kind of selection of the words, so only the ones that provide information, or some kind of meaning to our purpose. To divide the raw text into words, we use the library NLTK, explained in section 2.3. And in order perform the selection of words that give us some kind of meaning, we must convert it into vectors that will contain all the information that will influence the decision of the personality traits prediction. Those vectors are called the features.

This section explains the extraction of all those features. That will allow in the future that the classifier can understand the information from a text input. Due to this, we are mostly going to work with textual features. Linguistic or textual features are the ones extracted exclusively from the content of the transcription. Some metrical features regarding the text of a review are number of words, words per sentence, capital letters, first and second pronouns, exclamations and questions, paragraphs, long words and punctuation marks.

3.5.1 Gender

This feature does not have to be extracted because is given in the dataset, although there are projects focused on predicting the gender in a similar way we predict personality. The gender is provided in the binary format identified as male and female. Overall, the data is balanced in terms of gender distribution and includes 210 females (52%) and 194 males (48%). As we did with the traits we can go inside the gender feature. The gender distribution along the five personality traits is presented in Table 3.7. Except for Agreeableness (females tend to be more agreeable), there is no significant difference between males and females in terms of personality traits. This may be a special case or a casualty of the dataset used because other studies [27] have demonstrated that exist significant gender differences across the ten aspects of the personality traits. In fact, women reported higher Extraversion, Agreeableness, and Neuroticism scores than men. So we include the gender to make more general predictions in order to be more adaptive with new instances non-presented in the dataset. Gender is not given in YouTube profiles, so in order to use this parameter, we have created a gender classifier which provides gender from a text source used by the personality classifier. This classifier is more deeply detailed in the appendix.

Table 3.7: Scores mean and personality traits "yes" grouped by sex

——	Scores mean		"Yes" traits	
Class	**Male**	**Female**	**Male**	**Female**
Extr	4.618385	4.630862	82	81
Agr	4.469944	4.878878	66	116
Cons	4.525749	4.470782	156	160
Emot	4.738783	0.46	84	94
Open	4.696784	4.634195	79	62

3.5.2 TF-IDF

The features obtained from this technique are, for each document, a vector with the TF-IDF of each of the terms of the vocabulary obtained from all the training documents that appear on it. As the model has to be obtained from all the training set, it is not going to be

evaluated using cross validation. Obtaining TF-IDF features has the aim of detecting words that are more relevant or used in a transcription than in other, that is to say, modelling the vocabulary used by vloggers. Notably, we have used different vectorizers for each task, regression and performance because each vectorizer performs better in one task. For the regression task the vector resultant is 67.279-dimensional and for the classification task is 46-dimensional. The difference is caused because the classification vectorizer requires a minimal appearance frequency.

3.5.3 Lexical features

From the transcription, we extracted lexical features (tokens) and then transformed them into a bag-of-words, vector space model. This is a numeric representation of text that has been introduced in text categorization and is widely used in behavioral signal processing.

- Number of characters of the transcription: large posts contain more information, so a hypothesis is that talkative (associated to Openness and Extraversion) people may speak more in their videos.
- Sentence-based features: only measure the number of sentences in a text. This is a superficial analysis of the sentences.

3.5.4 Polarity

The Senpy plugin MeaningCloud identifies positive/negative polarity in any text, including comments in surveys and social media. In order to do this, the local polarity of the different sentences in the text is identified and the relationship between them evaluated, resulting in a global polarity value for the whole text. As we have told before, the way we express in terms of sentiments may differ in one personality traits to others. In our case and with our dataset, polarity is correlated with Agreeableness (0.39) and Emotional stability(0.36), meanwhile, Extraversion is very poor correlated(0.04).

3.5.5 Part of Speech (POS)

Another way of analyzing linguistic features is focusing on lexical categories of words. The aim here is collecting the statistics about those word categories from the training dataset and see if some traits use some specific categories. The parts of speech to be analyzed are adjectives, adpositions, adverbs, conjunctions, determiners and articles, nouns, numerals,

particles, pronouns, verbs and punctuation marks. The tag set (categories) depends on the corpus annotation. Fortunately, NLTK defines a Universal targset which is presented in the Table 3.8

Table 3.8: Universal tags defined by NLTK

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
DET	determiner, article	the, a, some, most, every, no, which
NOUN	noun	year, home, costs, time, Africa
NUM	numeral	twenty-four, fourth, 1991, 14:24
PRT	particle	at, on, out, over per, that, up, with
PRON	pronoun	he, their, her, its, my, I, us
VERB	verb	is, say, told, given, playing, would
.	punctuation marks	. , ; !
X	other	ersatz, esprit, dunno, gr8, univeristy

3.6 Classification and Regression Model

The extraction of all the features gives us a new dataset containing all the information that can be read by the classifier. However, all information is useless without a classifier model, that can interpret all those features, and based on all that information, will be able to predict the vlogger's personality. I is impossible to know which classifier is going to perform the best results, so now we have to experiment selecting the best classification algorithms (optimizing its parameters) and selecting the features that give more information to the classifier. We have to take care of not falling into the over-fitting problem. This problem

appears when the classifier is too much adapted to the training set and learns by heart the training examples, but is not able to classify correctly new instances. Avoiding this issue we will create a general predictor for both, training instances and test instances.

Now we will proceed to describe all the classifiers and regressors used in the algorithm [9]. This system uses a combination of several classifiers, that are already designed by the Scikit-Learn library. All of them are going to be described in the list below. Regarding the classifiers we have:

- **Logistic Regression:** although the labels are categorical applies a regression based on the function logit and divide the labels into boundary decision intervals. Finally, with the regression result maps the interval to the label that corresponds.
- **Support Vector Machines:** it tries to separate the different labels into linear regions, based on lines, planes or hyper-planes depends on the dimension. First, tries to do it in one dimension and if it is not possible, transforms the problem in a high dimension problem by using a special function called *"kernel"*, when it can be possible. Al this is conditioned to the parameter C (Cost parameter) which is related to the error assumed.
- **Decision Tree:** the goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. These rules are called branches and are creating by splitting the labels and the features applying the best information gain at each level.
- **Random Forest:** is a meta-estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting.
- **Bernoulli Naive Bayes:** implements the Bayes theorem assuming each feature as a binary-valued variable.
- **Multinomial Naive Bayes:** the same as the previous estimator but in this case, it bases the decision on the likelihood of each feature.
- **Gaussian Naive Bayes:** the same as the previous estimator but the likelihood is assumed to be Gaussian.
- **Ada Boost:** is a meta-estimator that begins by fitting a classifier on the original dataset and then 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.

- **K-Nearest Neighbors:** it does not attempt to construct a general internal model, but simply stores instances of the training data. When a new instance is presented it computes a simple majority vote of the k nearest instances (the instances that are more similar to the presented).

Except the Naive Bayes methods and the Logistic Regression, the remaining algorithms have its own regression version. Besides Scikit-learn provides some specific regression methods. The following are a set of methods intended for regression in which the target value is expected to be a linear combination of the input variables.

- **Linear Regression:** fits a linear model with coefficients to minimize the residual sum of squares between the observed responses in the dataset, and the responses predicted by the linear approximation.
- **Ridge:** solves a regression model where the loss function is the linear least squares function and regularization is given by the l2-norm. Ridge regression addresses some of the problems of Ordinary Least Squares by imposing a penalty on the size of coefficients. The ridge coefficients minimize a penalized residual sum of squares.
- **Lasso:** is a linear model that estimates sparse coefficients. It is useful in some contexts due to its tendency to prefer solutions with fewer parameter values, effectively reducing the number of variables upon which the given solution is dependent. For this reason, the Lasso and its variants are fundamental to the field of compressed sensing. Under certain conditions, it can recover the exact set of non-zero weights.
- **LassoLars:** is a lasso model implemented using the LARS algorithm, which is piecewise linear as a function of the norm of its coefficients.
- **Bayesian Ridge:** estimates a probabilistic model of the regression problem by using bayesian regression techniques. Fits a Bayesian ridge model and optimize the regularization parameters lambda (precision of the weights) and alpha (precision of the noise).

3.7 Evaluation

The main goal of this section is to show a comparison between the results obtained during the realization of this project and the results obtained in the workshop on computational personality recognition. In order to have a real comparison, we have used the same metrics as used in the competition.

The metrics suggested for the classification task are:

- **Precision:** computes the proportion of instances predicted as positives that were correctly evaluated (it measures how right our classifier is when it says that an instance is positive). It can be calculated like this: $precision = \frac{tp}{tp+fp}$. Where tp are the number of instances that were stated as positive and actually were positive (also called true positive) and fp are the number instances that were stated as positive and actually were negative (also called false positive).
- **Recall:** counts the proportion of positive instances that were correctly evaluated (measuring how right our classifier is when faced with a positive instance). It can be calculated like this: $recall = \frac{tp}{tp+fn}$. Where fn are the number of instances that were stated as negative and actually were positive (also called false negatives).
- **F1-score:** is the harmonic mean of precision and recall ($F = 2 \cdot \frac{precision \cdot recall}{precision+recall}$), and tries to combine both into a single number.

And for the metrics proposed for the regression task:

- **RMSE (Root Mean Square Error):** the square root of the mean/average of the square of all of the error.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.1}$$

  Where y_i is the predicted value and $\hat{y}_i$ is the expected value.

3.7.1 Self Performance

In this section, we are going to test our system comparing our results with the workshop results, in order to see how good is our performance. we will show each trait result for classification and regression.

Once the test is performed, there is and important task of reading and interpreting the results. Comparing the different outcomes, we can arrive at conclusions about the performance of the classifier.

Since the results expected for the recall score are impossible to outperform and the f1 score is conditioned by the recall score we will focus on overcoming the precision score. The relevant results (those which improve the baselines) will be represented in bold.

Regarding the classification task, we got to outperform three out of five traits (Agreeableness, Conscientiousness and Neuroticism). All estimators improve the agreeableness baselines, this is because these are the less exigents and because the gender feature may help the classifier to make better decisions in this trait. The best result is achieved by the K-Nearest Neighbors classifier in the conscientiousness trait, the unbalance distribution causes a variety of rare results. The remaining results can be viewed in Table 3.9.

Table 3.9: Classification performance measuring the precision score for each trait

Classification model	**Extr**	**Agr**	**Cons**	**Neuro**	**Open**
Logistic Regression	0.62	**0.64**	0.37	0.44	0.5
Support Vector Machines	0.33	**0.55**	0.38	**0.64**	0.33
Decision Tree	0.42	**0.55**	0.43	0.43	0.59
Random Forest	0.49	**0.63**	0.37	0.48	0.54
Bernoulli Naive Bayes	0.53	**0.6**	0.43	0.34	0.64
Multinomial Naive Bayes	0.62	**0.63**	0.67	0.49	0.34
Gaussian Naive Bayes	0.57	**0.7**	0.5	0.48	0.6
Ada Boost	0.49	**0.6**	0.43	0.48	0.59
K-Nearest Neighbors	0.52	**0.6**	**0.88**	**0.63**	0.6

In the regression case, the performance is a bit better, from the point of view that the RMSE is reduced or equalized in all traits and all estimators are able to outperform the baselines in at least four traits. We can see all this information in Table 3.10.

Table 3.10: Regression performance measuring the RMSE for each trait

Regression model	**Extr**	**Agr**	**Cons**	**Neuro**	**Open**
Linear Regression	**0.94**	**0.78**	**0.71**	0.8	**0.8**
Ridge	**0.94**	**0.78**	**0.7**	0.76	**0.82**
Lasso	**1.01**	**0.83**	**0.7**	0.76	**0.82**
LassoLars	**1.02**	**0.9**	**0.71**	**0.75**	**0.82**
BayesianRidge	**1.01**	**0.78**	**0.71**	0.79	**0.82**

3.7.2 Ten-fold performance

Independent from the workshop results, we are going to measure the algorithm accuracy performing a ten-fold cross validation. This evaluation method consists in splitting the training set into ten subsets, and evaluating them in ten iterations. On each of those iterations, the test set will be one of the ten subsets that were made, and the other nine subsets will make up the training set, thus all the instances act as both training and testing. At the end, each of those subsets will have been tested with a training set of the other nine. The results will show the accuracy of the model, which means the percentage of correctly classified instances from the total number of instances. Regression and classification performance are represented in Table 3.11 and Table 3.12.

For the Extraversion trait we have results around fifty and sixty percent, a little far from the seventy percent (referred to precision) provided in the baseline which shows that we could have included more sophisticated methods for feature selection or tuning the algorithm parameters, but this was a very cost timed task having account all the traits and all the algorithms. Due to Openness and Extraversion are correlated, their results obtained are similar but Openness performs a bit better even approaching the baseline (we remind, a 0.68 of precision). In terms of Conscientiousness, this is the best performed trait maybe due to its unbalanced distribution, so it is also a good option to perform a stratified k-fold in these case, which is a variation of the original k-fold and provides stratified folds, what means that each set contains approximately the same percentage of samples of each target class as the complete set. Finally, we have Agreeableness which follows Openness and Extraversion accuracy but reaching the baseline (we remind, a precision of fifty percent) and Neuroticism which results are the worst of all traits. If we focus on the algorithms and

Table 3.11: Ten fold accuracy

Algorithm	**Extraversion**	**Agreeableness**	**Conscientiousness**	**Neouroticism**	**Openness**
Logistic Regression	0.6 +/- 0.12	0.65 +/- 0.12	0.77 +/- 0.12	0.57 +/- 0.16	0.65 +/- 0.1
Support Vector Machines	0.57 +/- 0.12	0.56 +/- 0.1	0.78 +/- 0.09	0.55 +/- 0.13	0.64 +/- 0.1
Decision Tree	0.57 +/- 0.13	0.62 +/- 0.12	0.76 +/- 0.13	0.54 +/- 0.19	0.62 +/- 0.1
Random Forest	0.57 +/- 0.13	0.59 +/- 0.13	0.76 +/- 0.1	0.6 +/- 0.13	0.62 +/- 0.08
Bernoulli Naive Bayes	0.52 +/- 0.17	0.61 +/- 0.14	0.73 +/- 0.9	0.58 +/- 0.14	0.59 +/- 0.17
Multinomial Naive Bayes	0.59 +/- 0.08	0.59 +/- 0.16	0.73 +/- 0.08	0.6 +/- 0.13	0.65 +/- 0.08
Gaussian Naive Bayes	0.5 +/- 0.13	0.61 +/- 0.14	0.64 +/- 0.17	0.58 +/- 0.13	0.56 +/- 0.65
Ada Boost	0.52 +/- 0.1	0.59 +/- 0.15	0.72 +/- 0.14	0.57 +/- 0.16	0.56 +/- 0.1
K-Nearest Neighbors	0.56 +/- 0.13	0.53 +/- 0.22	0.77 +/- 0.09	0.62 +/- 0.12	0.64 +/- 0.07

we compare Logistic Regression with the remaining estimators Logistic seems to perform properly for all traits. Besides, we could also mention K-Nearest Neighbors and Random Forest for Neuroticism and SVM for Conscientiousness.

Table 3.12: Ten fold root mean squared error

Algorithm	**Extraversion**	**Agreeableness**	**Conscientiousness**	**Neouroticism**	**Openness**
Linear Regression	0.95 +/- 0.09	0.78 +/- 0.15	0.72 +/- 0.2	0.73 +/- 0.18	0.71 +/- 0.12
Ridge	0.95 +/- 0.08	0.77 +/- 0.15	0.72 +/- 0.19	0.72 +/- 0.18	0.71 +/- 0.12
Lasso	0.97 +/- 0.08	0.87 +/- 0.15	0.75 +/- 0.19	0.77 +/- 0.24	0.71 +/- 0.13
LassoLars	0.98 +/- 0.09	0.88 +/- 0.16	0.77 +/- 0.18	0.77 +/- 0.20	0.71 +/- 0.13
Bayesian Ridge	0.97 +/- 0.09	0.77 +/- 0.16	0.72 +/- 0.2	0.72 +/- 0.19	0.71 +/- 0.13

In general issues, the results obtained are better to the results obtained with the workshop splitting proposal. Now we have an error under the unity and curiously RMSE for Openness trait is the same for all the estimators. Lasso algorithms do not follow the others algorithms tendency in Neuroticism, Agreeableness and Conscientiousness. Extraversion is the hardest trait to predict and Linear Regression and Ridge keep performing the best results for this trait.

CHAPTER 4

YouTube Personality Service

4.1 Introduction

This chapter aims to build a real functional system whose main objective is to predict or analyze the personality from a video source, provided by YouTube, using the big five personality model. The results from each trait analyzed could be presented as a binary value or as a continuous value taking advantage of the classification and regression performance already implemented in previous chapters.

In order to do so, we are going to explain in detail the real functional personality predictor system created for YouTube videos. It basically consists in a system that given a YouTube URL video extracts the video subtitles in English, it is important so it can be used for all videos which have English subtitles, then, subtitles are appended compounding the YouTube video transcription and finally the YouTube video transcription is the input received by the classifiers to generate a prediction. All this system is implemented as a Senpy plugin which is explained in the next section.

New tools have been necessary to deploy this part. We are referring to the tools which let us to download information associated to a YouTube video. We have a python script called youtube-dl, which provides us a command line interface to work, and the pysrt library

that allows us to parse YouTube subtitles.

4.2 Senpy plug-in

The purpose of this section is implementing the previous modules in one service entirely functional. Senpy allows the creation of plug-ins that can be deployed in a Senpy server. In this project, we have created one plug-in to execute our system as a service. This plug-in has two functionalities. First and principal functionality is an implementation of a personality traits classifier and a gender classifier appears as the second and last functionality. The personality traits classifier needs vlogger's gender to make a decision but this information is not given by YouTube videos, so the gender classifier will infer the vlogger's gender at the personality traits classifier request.

The main component of a Senpy analysis service is the algorithm itself. However, for the algorithm to work, it needs to get the appropriate parameters from the user, format the results according to the defined API, interact with the user when errors occur or more information is needed, etc.

Senpy proposes a modular and dynamic architecture that allows:

- Implementing different algorithms in a extensible way, yet offering a common interface.
- Offering common services that facilitate development, so developers can focus on implementing new and better algorithms.

The framework consists of two main modules: Senpy core, which is the building block of the service, and Senpy plugins, which consist of the analysis algorithm. The Fig. 4.1 depicts a simplified version of the processes involved in an analysis with the Senpy framework. The tool extracts the parameter from a NIF HTTP query and executes the plugin selected code with the inputted parameters previously validated. Then, use models to output a linked data publication in the desired format.

Senpy implements a linked data model that is usually used for sentiment and emotion analysis based on semantic vocabularies Marl [10], Onyx [24] and NIF [1]. In our case, we are modeling personality but, this system can be used as well for our purpose. Senpy can use several formats such us turtle, JSON-LD [5] and XML-RDF [8].

JSON-LD is a method of encoding Linked Data using JSON. The data is serialized in a way that is similar to traditional JSON format. JSON-LD is designed around the concept

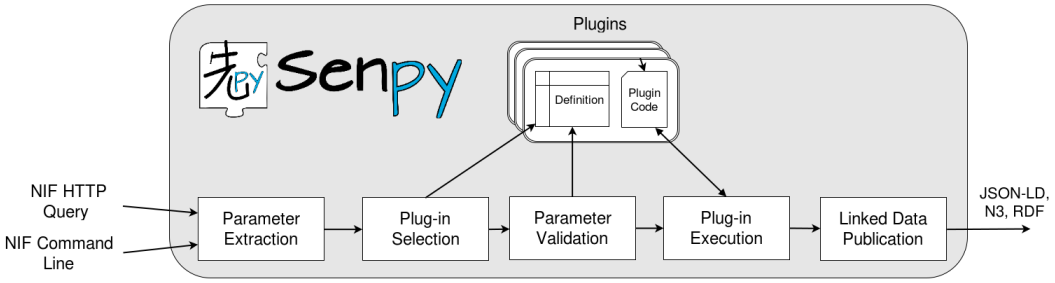

Figure 4.1: Senpy Architecture [23]

of a context to provide additional mappings from JSON to an RDF model. The context links object properties in a JSON document to concepts in an ontology, as Marl or NIF. In our project are not going to use any of the ontologies previously described. This is due to the nature of our predictor.

RDF is a family of World Wide Web Consortium (W3C) specifications originally designed as a meta data model for information interchange on the Web. RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link.

In order to implement this plug-in, a Python module was created gathering all the necessary modules and tools to execute those modules. All the modules explained in the previous sections are included in this Senpy plug-in.

Once we know how Senpy works from inside, we can explain how to use it to implement our plug-in. Fig. 4.2 shows an schematic explanation about the flows and interactions of a request realized to our system.

Trough the Senpy API the user provides an URL from a YouTube video, it can be any video but it is thought for vlogger's or youtubers. It is also necessary to provide an extra parameter indicating the prediction format choosing between regression or classification. Then, Senpy uses youtube-dl and gets the subtitles from the video identified by the URL given. YouTube-dl is a command-line program to download videos from YouTube and information associated to the video such us audio or subtitles. In order to provide subtitles YouTube has two principal options depending on the way that the subtitle is stored on the YouTube platform. When subtitles has been manually made, YouTube provides them in a srt format, is the most basic format to represent subtitles and consists in four parts and all of them are text:

- A number indicating which subtitle it is in the sequence.

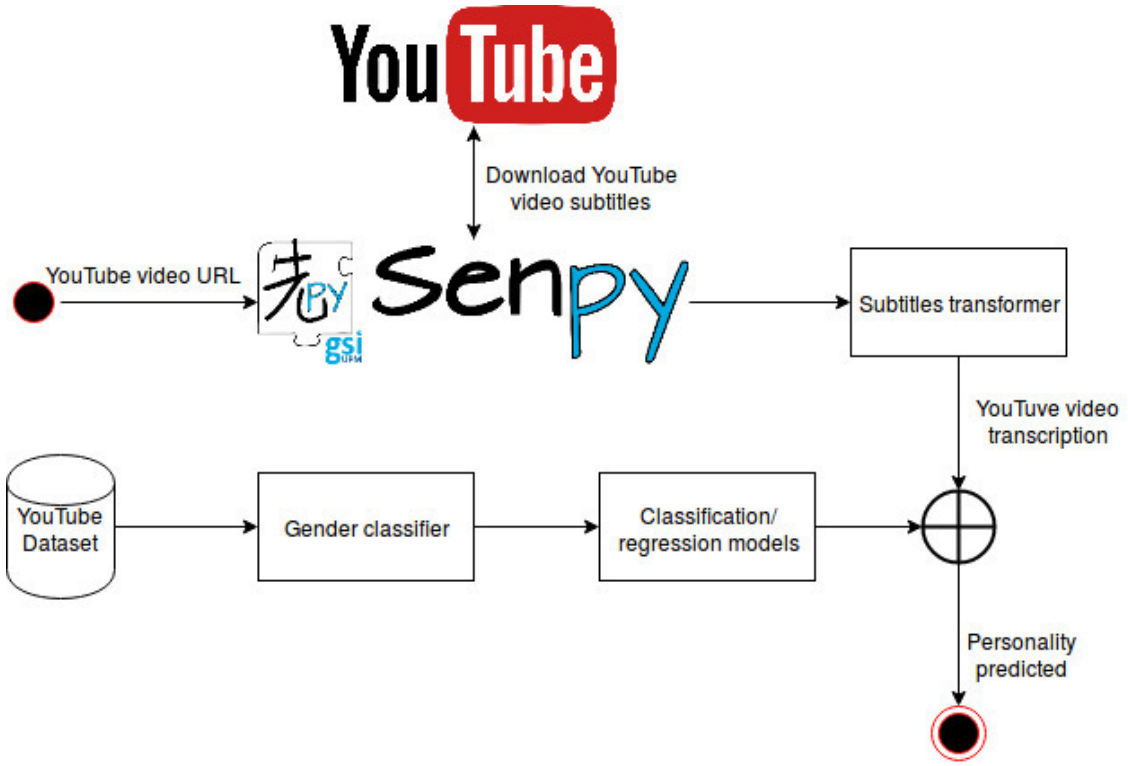

Figure 4.2: Senpy plug-in schematic architecture

- The time that the subtitle should appear on the screen, and then disappear.
- The subtitle itself.
- A blank line indicating the start of a new subtitle.

On the other hand, when subtitles are generated automatically by YouTube using its own voice recognizer system, YouTube provides subtitles in a WebVTT format. WebVTT (Web Video Text Tracks) is a W3C standard for displaying timed text in connection with HTML5. Anyway, the library pysrt implements several functionalities to extract the subtitle itself from both formats, the main difference is presented when parsing WebVTT because it is necessary to use an HTML parser. So concatenating all the subtitles from each video we can obtain its transcription, which is we are looking for. As we explained in the previous chapter the classifiers input are text, so now we can make predictions and output the results. The results are presented in JSON-LD format, the gender uses FOAF [3] ontology and each personality trait is represented using the prefix *trait*. In Fig. 4.3 and Fig. 4.4 can be visualized all the information given before.

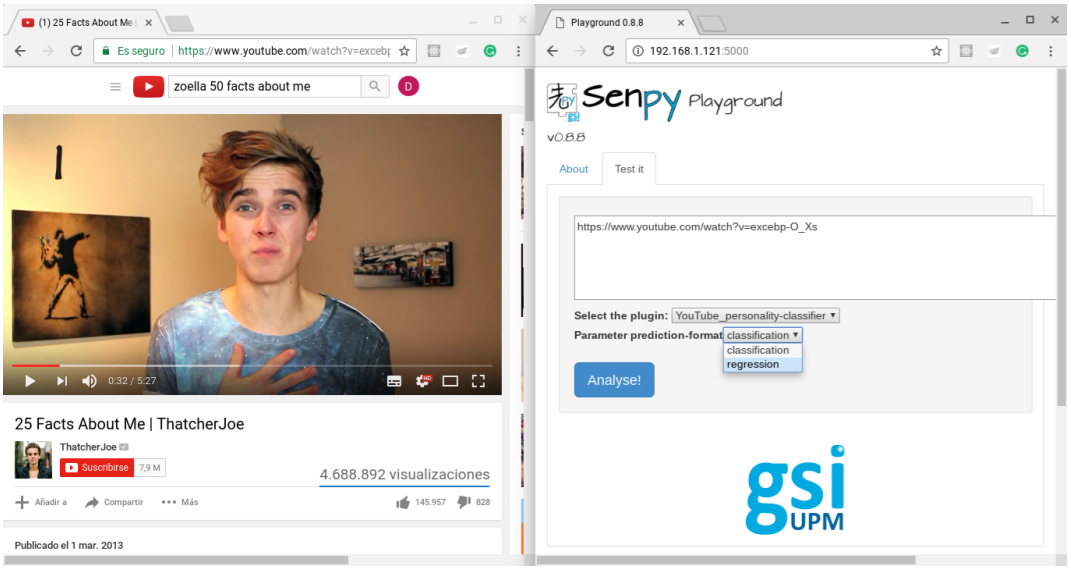

Figure 4.3: Senpy plug-in interface


```
    foaf:gender : Male
    prov:wasGeneratedBy : plugins/YouTube_personality-classifier_0.1
  nif:isString : https://www.youtube.com/watch?v=excebp-O_Xs&t=32s
▼ sentiments [0]
    (empty array)
▼ suggestions [0]
    (empty array)
▼ topics [0]
    (empty array)
▼ traits {11}
    @type : trait:Trait
    trait:Agreeableness : trait:Agreeableness
    trait:AgreeablenessValue : 4.1043530468880924
    trait:Conscientiusness : trait:Conscientiusness
    trait:ConscientiusnessValue : 4.909229847930844
    trait:EmotionalStability : trait:EmotionalStability
    trait:EmotionalStabilityValue : 4.527495789030786
    trait:Extraversion : trait:Extraversion
    trait:ExtraversionValue : 4.341292358649136
    trait:Openness : trait:Openness
    trait:OpennessValue : 4.515529548458067
```


Figure 4.4: Plug-in results

CHAPTER 5

Conclusions and future work

5.1 Introduction

In this chapter we will describe the conclusions extracted from this thesis, the achievements, problems and suggestions about future work.

5.2 Conclusions

We are going to compare the results obtained with the results obtained by other previous papers, basically the participants who were presented in the WCPR14, using the splitting test proposal. Starting with the regression case, we must have a look inside the multivariate regression, Sect. 2.6.1, the RMSE obtained were lower than in our performance. They also had the hardest difficulties in the Extraversion trait and our Openness performance are similar to their Openness performance. We can conclude stating that we are not so far from this sophisticated regression, so we could say that we have reached a state of the art performance in terms of RMSE.

In connection with the classification task, we have three participants to compare. First, Ben Verhoeven and Walter Daelemans study, Sect. 2.6.2 obtained very poor results, if we see

our performance we can see that our results are much better, but we also obtained similar results using the SVM algorithm from Scikit so they could have used other algorithms and other features. Then, Sonja Gievska and Kiril Koroveshovski paper, Sect. 2.6.3, performed very good results using audiovisual features, so we must propose to include this behavioral analysis in future work. Finally, Firoj Alam and Giuseppe Riccardi, Sect. 2.6.4, following the same lines than the previous work they did not reached the same satisfactory results but they combined features in a good and successful way. With all this information, we can infer that our system performs properly despite only using text transcriptions but it could be improved by using a multimodal analysis. Apart from all the things explained before, I would like to stand out that we are the unique who dared to perform classification and regression at the same time.

5.3 Achieved goals

In the following section we will explain the two main goals that have been achieved during the realization of this project.

Implement the system as a service. In order to show the system functionalities we have deployed a YouTube personality service which can be used by anyone interested in their favorite youtubers, for example. Also anyone who has a YouTube channel can use the system and corroborate the results.

Overcome most of the baselines. For the regression case, we improved the results for all the traits and for the classification case we improved the results for three out of five traits. The remaining traits were not so far to be improved 0.08 until 0.7 (precision) for Extraversion and 0.04 until 0.68 for Openness.

5.4 Problems faced

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

- **Senpy API Limitations:** In order to obtain the polarity, we made a request to the meaning cloud plug-in implemented in Senpy. This plug-in has a text size limit, this limit was approximately about six thousand characters. So if we wanted to use this service and the text to analyze is larger than seven thousand we had to trunk the text

and then obtain the polarity from a trunked version from the original text. Senpy also had a limit on the request rate, in order to solve this, we implemented a temporizer system to control our request rate towards Senpy.

- **Ontologies in Senpy:** Senpy is a platform designed for Sentiment and Emotion analysis, not for personality prediction. So the ontology that this project requires for their responses is not implemented. That is not a major problem, but the results given by the plug-in, will not be possible to be validated as much as if they were the results of a Sentiment or Opinion analysis system.

- **Gender in YouTube:** In order to recognize the personality traits of the user in our project we have made necessary to use the gender but this information is not given by a YouTube video, so we have had to create a gender classifier which works from a text source.

5.5 Future work

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

Include audiovisual features. This is the main point for future development, now we only work with text features and we have seen that we could improve the results using behavioral analysis. Between those, we could have in account the prosody, speaking time, the length of the speaking segments, the speaking energy, the pitch, looking activity and pose.

Analyze YouTube video comments. Another one possible next step could be, analyzing the comments that each video receives. If we think about it could improve a lot, the personality scores were given by third persons foreign to the vloggers. So it is more nearly to predict the personality in which others see in the user and not the personality who the user really has. So comments that are given by third people in the video could help to approximate the personality resolution.

Detect dialogs. In this stable version we don't have in mind the number of characters who appear in the video, so this system will work properly with videos with only one person appearance. If we are able to detect dialogs and different characters we could be able to predict personality for each character independently from each other.

Bibliography

[1] Nif ontology. http://persistence.uni-leipzig.org/nlp2rdf/.

[2] Nltk website. http://www.nltk.org/.

[3] Official foaf manual. http://xmlns.com/foaf/spec/.

[4] Official github of senpy. https://github.com/gsi-upm/senpy.

[5] Official json page for linked data. http://json-ld.org/.

[6] Pandas website. http://pandas.pydata.org/.

[7] Psychological review, vol 41, 1-32.

[8] Rdf page in w3. https://www.w3.org/RDF/.

[9] Scikit-learn online manual. http://scikit-learn.org/stable/.

[10] J. Fernando Sánchez-Rada Adam Westerski. Marl ontology. http://www.gsi.dit.upm.es/ontologies/marl/.

[11] Firoj Alam and Giuseppe Riccardi. Predicting personality traits using multimodal information. In *Proceedings of the 2014 ACM Multi Media on Workshop on Computational Personality Recognition*, pages 15–18. ACM, 2014.

[12] Joan-Isaac Biel and Daniel Gatica-Perez. The youtube lens: Crowdsourced personality impressions and audiovisual analysis of vlogs. *Multimedia, IEEE Transactions on*, 15(1):41–55, 2013.

[13] Joan-Isaac Biel, Vagia Tsiminaki, John Dines, and Daniel Gatica-Perez. Hi youtube!: personality impressions and verbal content in social video. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 119–126. ACM, 2013.

[14] Fabio Celli, Bruno Lepri, Joan-Isaac Biel, Daniel Gatica-Perez, Giuseppe Riccardi, and Fabio Pianesi. The workshop on computational personality recognition 2014. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1245–1246. ACM, 2014.

[15] Fabio Celli, Fabio Pianesi, David Stillwell, and Michal Kosinski. Workshop on computational personality recognition (shared task). In *Proceedings of the Workshop on Computational Personality Recognition*, 2013.

[16] Golnoosh Farnadi, Shanu Sushmita, Geetha Sitaraman, Nhat Ton, Martine De Cock, and Sergio Davalos. A multivariate regression approach to personality impression recognition of vloggers. In *Proceedings of the 2014 ACM Multi Media on Workshop on Computational Personality Recognition*, pages 1–6. ACM, 2014.

[17] M. Rohani M. Kosinki D. Stillwell M. Moens S. Davalos G. Farnadi, G. Sitamaran and M. De Cock. *How are you doing? Emotions and personality in Facebook.* In proc. of EMPIRE.

[18] Sonja Gievska and Kiril Koroveshovski. The impact of affective verbal content on predicting personality impressions in youtube videos. In *Proceedings of the 2014 ACM Multi Media on Workshop on Computational Personality Recognition*, pages 19–22. ACM, 2014.

[19] William L. Hosch. Artificial intelligence. `http://global.britannica.com/technology/machine-learning`.

[20] Matt Kiser. Introduction to natural language processing (nlp) 2016. `http://blog.algorithmia.com/introduction-natural-language-processing-nlp/`.

[21] The IDIAP Research. The youtube personality dataset. `https://www.idiap.ch/dataset/youtube-personality`.

[22] S. Gaddis S. D. Gosling and S.Vazire. A very brief measure of the big-five personality domains. *Journal of Research in Personality*, 37(6):504–528, 2003.

[23] 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)*, October 2016.

[24] J. Fernando Sánchez-Rada. Onyx ontology. `http://www.gsi.dit.upm.es/ontologies/onyx/`.

[25] Yasemin Koçak Usluel. Social network usage. In *Social Networking and Education*, pages 213–222. Springer, 2016.

[26] Ben Verhoeven, Walter Daelemans, et al. Evaluating content-independent features for personality recognition. In *Proceedings of the 2014 ACM Multi Media on Workshop on Computational Personality Recognition*, pages 7–10. ACM, 2014.

[27] Colin G. DeYoung Yanna J. Weisberg and Jacob B. Hirsh. Gender differences in personality across the ten aspects of the big five.

APPENDIX A

Gender classifier

In this appendix, we are going to explain the working system of a gender classifier created as an intermediate of a personality traits classifier. It works very similarly to the personality traits classifier, even using the same attributes, but we can find several clear differences between both of them.

The gender classifier has been created inside the plug-in Senpy. As we have told Senpy lets us create plug-ins that can be deployed in a Senpy server with a web interface in such a way we can execute the gender classifier as a service (a service which is used by the personality traits classifier).

The plugin receives a text input and responses with the gender inferred. It uses a semantic vocabulary called Foaf [3]. FOAF is an ontology based on linking people and information using the Web. Regardless of whether the information is, in people's heads, in physical or digital documents, or in the form of factual data, it can be linked. FOAF integrates three kinds of network: social networks of human collaboration, friendship and association. The response, i.e. the plug-in output, uses FOAF ontology in order to shape the information properly and give it in a JSON-LD format. In the format of the response, we can find that the information outputted is provided with a correspondence with the correct ontology for each case. Finally, the response obtained has the following field:

- **foaf:gender:** contains the predicted gender of the person whose text has been analyzed, ie the system output.

Finally, only remains to measure the classifier accuracy, in order to do so we are going to perform a ten-fold cross-validation using the classification algorithms explained in Sect 3.6. The result is showed in Table A.1.

Table A.1: Ten fold cross validation gender classifier accuracy

Algorithm	**Accuracy**
Logistic Regression	0.6 +/- 0.19
Support Vector Machines	0.54 +/- 0.1
Decision Tree	0.56 +/- 0.14
Random Forest	0.6 +/- 0.17
Bernoulli Naive Bayes	0.58 +/- 0.12
Multinomial Naive Bayes	0.55 +/- 0.1
Gaussian Naive Bayes	0.57 +/- 0.18
Ada Boost	0.6 +/- 0.18
K-Nearest Neighbors	0.56 +/- 0.12

The accuracy obtained is between 0.5 and 0.6, a little poor result, maybe because the features are thought to predict personality not gender. Nevertheless Logistic, Random Forest and Ada Boos perform more or less decently, comparing to the rest.