UNIVERSITÀ
DI TRENTO

Department of Information Engineering and Computer Science

Bachelor's Degree in
Computer Science

Final Dissertation

Generating Marketing Personas: an automated approach based on extracting insights from social networks

Supervisor

Alberto Montresor

Daniele Miorandi

Student

Thomas Reolon

Academic year 2019/2020

Acknowledgements

Questa tesi è dedicata alla mia mamma.

Vorrei ringraziare la mia famiglia per il supporto ricevuto durante questi anni, non sarei potuto essere qui se non fosse stato per loro.

Vorrei ringraziare tutte le persone che ho conosciuto durante questi anni, perché hanno dato colore a questa avventura.

Un ringraziamento speciale va a coloro che hanno reso possibile questa tesi: il mio relatore Alberto Montresor, i miei supervisori Daniele Miorandi e Carlo Caprini e tutto il team di U-Hopper.

Table of Content

Summary

Knowing your customers plays a central role in the development of a business. Marketing personas have been proven to be an effective tool for this purpose, in particular, they are used to identify important segments of customers. Usually, marketing personas are built upon interviews with a sample of customers to discover their needs and expectations. Unfortunately, this process is slow and expensive, so many companies do not make use of marketing personas.

The purpose for this thesis is to create a prototype capable of generating marketing personas in an automated way using social medias' data, it this way, the costs and time for creating and updating the marketing personas will be drastically cut.

Two important features that the system should posses are scalability and modularity. These characteristics have been satisfied adopting a distributed architecture; in this way, it is possible to have components running on different machines (scalability), with the components being independent between each other (modularity).

The system is divided in two parts: insights extraction and clustering. The first part analyzes users' social profiles using various models (pretrained NN, sklearn classifiers, ...) to extract valuable information about the users. The second part loads the insights from a database, clusters them and generate a dashboard that contain the marketing personas.

The analysis of the results was divided in two parts: insihgts evaluation and personas evaluation. The obtained results for the insights are:

- age: average error (real age - predicted age) of 3 years
- gender: average precision 0.93
- mbti-personality: average precision 0.72

To evaluate the goodness of a personas two approaches were taken:

- a visual approach to check the goodness of the clusters
- evaluating the generated personas against some personas created manually

These two approaches showed that the automatically generated personas represented the accounts used as their source. In conclusion, generating marketing personas using social medias' data is possible.

1 Introduction

Happy customers translate into a growing business. One of the best approaches to satisfy your clients is using a customer-centered mindset, but this is an available choice only if you know them.

As Neil Patel asserts "Consumers don't have time to waste on low-quality content they don't connect with [...] If one article doesn't fit their needs, there are dozens more they can turn to" [1]. This affirmation suggests that, in these years, with the appearance of globalization and the internet, customers had many more choices of products to buy, so, not being able to engage with clients will result in them turning to another company. Knowing your customers can also give many competitive advantages, for example customizing the user experience can lead to customers satisfaction that can increase customers loyalty and start a word of mouth chain.

Marketing personas are one of the most popular solutions adopted to understand and represent segments of customers. This approach has been proven to be highly effective, in fact, some case studies [2] showed that using marketing personas improved the competitivity of the business (increased click-through rate, increased engagement, ...). Also, Chang affirmed that using marketing personas can improve a product in its development phase [3].

Unfortunately, even if big companies use marketing personas with great results, many smaller companies do not. This is because the creation of marketing personas requires time and money that most companies can not afford. For example, an usual way to gain information to create the personas is to interview samples of customers, asking them about their needs and expectations. Many companies do not want to spend resources on this process, even if using personas could improve their competitiveness. The aim of this thesis is to address this problem proposing a system capable of automating personas generation without going through interviews and surveys.

Social medias are a goldmine of information about your customers. For example, King has proposed a way to generate personas using insights from Facebook and Google [4]. In his work, King focused on aggregated data coming from social media, but this kind of approach can reveal only general information that does not ensure any useful insights. This thesis proposes to analyze single users (that gave their consent) using machine learning techniques; in this way, we can use microtargeting marketing strategies which, as Thengs reports, are a very effective way to make successful marketing campaigns [5].

The structure of a marketing persona can widely depend on the market a company is in; for this reason, the system must be highly configurable, so that it can be changed to better address specific market needs. There are many articles that explain the best practices in building marketing personas [6, 7]. A relevant study on the effectiveness of marketing personas was conducted by Salminen [8]. This study analyzed the goodness of personas generated in an automated way (from Jensen's works [9]) and found out that too many information in a single persona could result as misleading and contrastive.

Considering these aspects we now want to define which information will compose the personas. To do so, we need to define which kind of company could use this service and what information is needed by them. Small e-commerce could definitely need it, because surveys would be too expensive and, not having a physical shop, it is hard for the owner to know his own customers. Moreover, e-commerce usually have a social login that allows user to log into the website, so this authentication system could be used to ask the user for his consent in processing his data.

To decide which insights are used to build the marketing personas a compromise must be taken: we want to use the most useful insights, but they should be easy to predict too. Hubspot and many other sources list these as useful insights that a marketing persona could contain: a person's values, needs, demographic information, interests, opinions, job/salary, communication channels, preferred brands, personality and followed trendings. Considering the approaches suggested in the literature (covered in the next chapter), demographic information, interests and personality emerged as the best insights to predict and that can bring real value to an e-commerce.

Five chapters follow this introduction: State of the Art, System Design, Implementation, System Evaluation and Conclusion. The Chapter on the *State of the Art* presents works related to automating the generation of marketing personas and researches about predicting users' insights using social media data. The *System Design Chapter* shows an high view of the system, which is divided in two main parts: the insights pipeline (which processes social medias' data) and the clustering pipeline (which uses the information extracted from the previous pipeline to generate the personas). Each pipeline is composed by individual components that interact between each other, these components are introduced in the *System Design Chapter* and covered in the *Implementation Chapter*. The Chapter on the *Evaluation* is divided in two sections, the first one evaluate the prediction results for each type of insights, while the second one assesses the goodness of the generated marketing personas. The final Chapter discusses the final results, the feasibility of this approach and looks at possible future work.

2 State of the Art

In this chapter, the state of the art in automated generation of marketing personas is presented. It is divided in three sections: insights extraction, clustering and concerns. Before these sections, an overview of similar works is presented.

Even if there is not a wide literature about *automatic personas generation*, Bernard Jansen has conducted relevant work in this field. In 'Towards automatic persona generation using social media', he and his team analyzed Facebook posts to determine users' interests and then generate personas [10]. Unfortunately this kind of approach is no longer possible due to the *General Data protection Regulament* that do not allow the processing of personal data without users' consent. To avoid this complication, more recently, he focused his research on aggregated data as in [9, 11, 12]. The main approach used in these studies is to use non-negative matrix factorization to extract information from a table with demographics and interests as dimensions. Even if this thesis do not focus on aggregated data analysis, Jansen's work is relevant because it shows the entire pipeline adopted to generate personas from data.

Another relevant work was conducted in 2017 by Koponen [13]. His work described a system capable of generating personas by clustering data about users' behaviours on a website. Koponen's work is important because it proves the feasibility of this kind of system. Unfortunately, his input data is very different from ours, so it is necessary to develop a new approach from scratch.

Aside from these papers, not many other researches have been published on this topic, so there is not a standard approach to follow. In this thesis we propose to analyze users to predict insights that will then be compared to understand if groups of clients with similar characteristics exist and generate personas from these groups. In the next section the state of the art in analyzing single users is presented.

2.1 Insights Extractions

Personas can be characterized by many attributes (eg. age, job, opinions, ...); this thesis focuses on the most common ones in the literature: gender, age, interests and personality. Once we know how to predict these insights, we can use them to compare users between each other and see if some peculiarity emarges.

Age and Gender

A fundamental part of a marketing persona is the demographic section, because it helps understanding your target and empathize with it. Several approaches have been proposed to extract relevant information; a literature review about predicting information from social data is provided by Phillips et al. [14]. Nguyen et al. have suggested a way to forecast users' age from their language [15]. Finally, Carletti et al. have presented the state of the art in extracting informations about people from photos of their faces [16].

Salary

Many researches in the topic of job and salary prediction were not GDPR compliant, so it is not possible to reuse approaches like the ones adopted in Levy et al. [17]. This problem re-emerges in many other articles that were written when there was not a strict privacy policy. An interesting approach that avoid this problem is explained in Chamberlain et al. [18], where researchers try to predict job and salary of Twitter users using their social network and posts, unfortunately the dataset they used to train the model is not available. Hu et al. have shown that personality and language used in Twitter posts correlate with their jobs; so it could be possible to use their results to infer a person's occupation

[19].

Interests

The most common approach to predict interests in the literature is to assume that, if a person posts something about a topic, that person is interested in that topic. Common approaches used in the literature to extract this insight are: using *tf-idf* to understand which words are the most significative and using *Latent Drichlet Allocation* to understand the topics of posts. A very interesting approach is used in Penas et al. [20] and in Torrero et al. [21]. These papers use an ontology to generalize the concepts discussed in the posts. In particular, the second study tried to map the concepts on Twitter's posts in 14 general categories.

Personality

There are many studies about personality prediction in social media, the most common approach is to analyze the language used in posts [22, 23].

Moon et al. suggested that users that have high regard for their social influences are more likely to behave impulsively [24]; so if you determine that a user is seeking social influence, it is more likely that he will do an impulsive purchase. Another study conducted by Chen et al. [25] relates marketing campaigns efficacy to personality, so it could be possible to infer this feature from the results obtained in the personality study.

2.2 Clustering

Once insights about single users have been extracted, it is necessary to create groups of similar users, a common adopted technique to solve this problem is clustering, as in Koponen work [13].

Some of the most common clustering algorithms are presented in George Seif's article [26], while Shehroz et al. have made a review of the most recent ones [27], finally Ahamad has explained the characteristic that a rule-mining clustering algorithm should have [28].

The most important characteristics that the clustering algorithm used in this thesis must have are: scalability, the ability to handle mixed data, the ability to handle high dimensional data.

Scalability

There are two most important points to take into consideration to assure the scalability of the system: linear time complexity and the possibility to adopt an incremental approach, with the latter one consisting in being able to update clusters without recomputing the whole dataset every time a change is made or new data is available. Time complexity is analyzed in Firdaus et al. [29], while incremental algorithms were proposed by Jhunjhunwala [30] and by Menon et al. [31].

Handling Mixed and High Dimensional Data

Most clustering algorithms use a fixed distance metric to evaluate the difference between two points; the most used one is the squared euclidean distance. Unfortunately, having to handle mixed and high dimensional data makes it difficult to find a good and reliable distance metric.

A review of the latest algorithms that can handle mixed data is proposed by Ahamad [27]. One of the algorithms discussed in the paper proposes to adjust the function that measure the similarity between two points during the clustering.

Filaire proposed an interesting collection of good practices to be adopted while solving this problem [32]. For example, he proposes how to build the similarity function and shows a code of a clustering algorithm.

Researchers have tried to find new types of algorithms to address these problems, for example Toor proposed an algorithm to cluster high dimensional data [33], Wang presented a linear complexity one [34] and Rozemberczki et al. suggested an algorithm that brought good results in the clustering of social media data [35].

2.3 Privacy Concerns

The system discussed in this thesis will handle personal data, for this reason it is fundamental to respect users' privacy. One of the most important laws on this topic is the GDPR (General Data

Protection Regulation), an european regulation that states the general guidelines that companies must follow if they process personal data of EU citizens (or if the company is located in EU). GDPR was thought to unify all EU states within a common point of view, but each state can emit new laws that respect GDPR principles. It is necessary to cite this regulation because it influenced the system design as explained below.

Personal data belongs to the user whom they refer to, so companies needs a lawfull, transparent and fair motive to process personal data. The most common way to fulfill this requirement is to ask the user for an informed consent.

For this reason we cannot train and test the models with data directly scraped from social networks, but we have to use data obtained from users that gave their consent. The easiest way to get this kind of data is by searching open public datasets. GDPR poses some limitations to the real applications of this system, in fact, profiling users require to obtain the user's consent. A way to get this form of consent in Facebook's socials, is by using the "social login" which allows the company the access to a user's personal data, like name, photo, likes, trough his explicit consent.

Another constraint posed by GDPR is to ensure the safety of the data: the company needs to plan which action must be taken to ensure the safety of the data, in proportion to the risk of a data breach. To limit this risk, the design has been thought to work without the necessity to save any personal data, so that we have to focus only on the processing phase.

To protect users' data in the processing phase, we will drop every non-needed information.

3 Solution Design

This chapter shows the choices that have been made to design the system. In the first part the system architecture is explained, while in the second part the single components are described.

3.1 High View

This section presents a high view of the system and the main factors that have influenced its design.

The proposed system can be thought as two distinct pipelines, the first one (that we will call *insights pipeline*) preprocess social media data, while the second one (that we will call *clustering pipeline*) uses the preprocessed data to build personas.

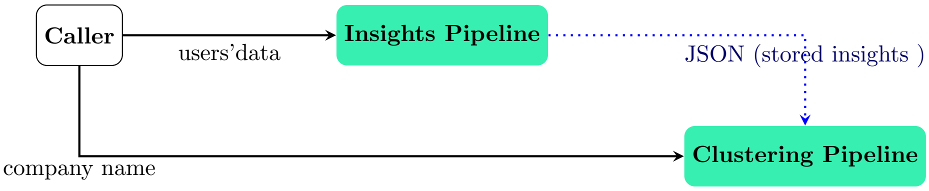


Figure 3.1: A general view of the system

Now a use case of the system is presented: The system receives a list of Twitter's users and each one of them is preprocessed by the insights pipeline, which will determine their age, gender, personality and interests. The predicted insights will then be saved persistently. When the generation of personas is requested, the clustering pipeline loads the persisted insights of the users and clusters them, then, a persona is generated for each relevant cluster.

Insights Pipeline

This pipeline receives users' data as input and saves the computed results in a storage system. There are three types of components that can compose this pipeline: *preprocessors*, *filters*, *classifiers* and *storage*.

A preprocessor takes in raw data, eg. a user social profile taken from the social network X, and transforms it in a unified formatted User class that the system is able to handle. A filter analyzes a user (eg. predicts if it belongs to a real person or not) and decides if it should be further analyzed or not. A classifier tries to predict insights that will be used in the clustering pipeline. A storage component is an interface that insert the previously extracted insights in a database.

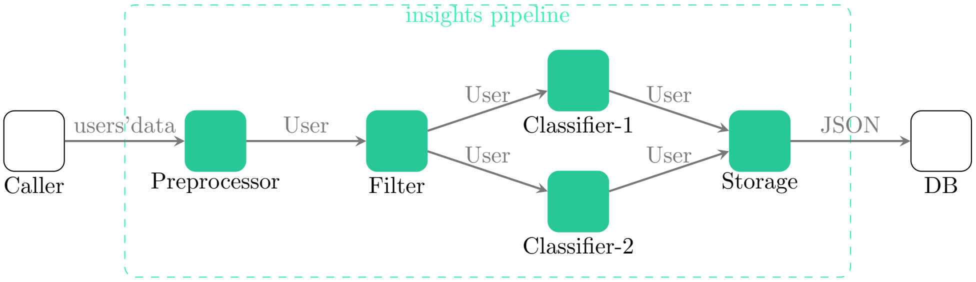


Figure 3.2: Representation of an insights pipeline

Clustering Pipeline

This pipeline receives a token to elaborate, fetch the needed data, cluster it and generate personas from the most relevant clusters. The resulting personas can then be sent to the client that requested them. This pipeline is composed by two principal parts: *clustering* and *generation*.

The clustering part fetch the insights from the database and applies a clustering algorithm; then the clusters are sent to the personas generator component. This latter component processes the most relevant clusters and generates an html dashboard that presents the personas.

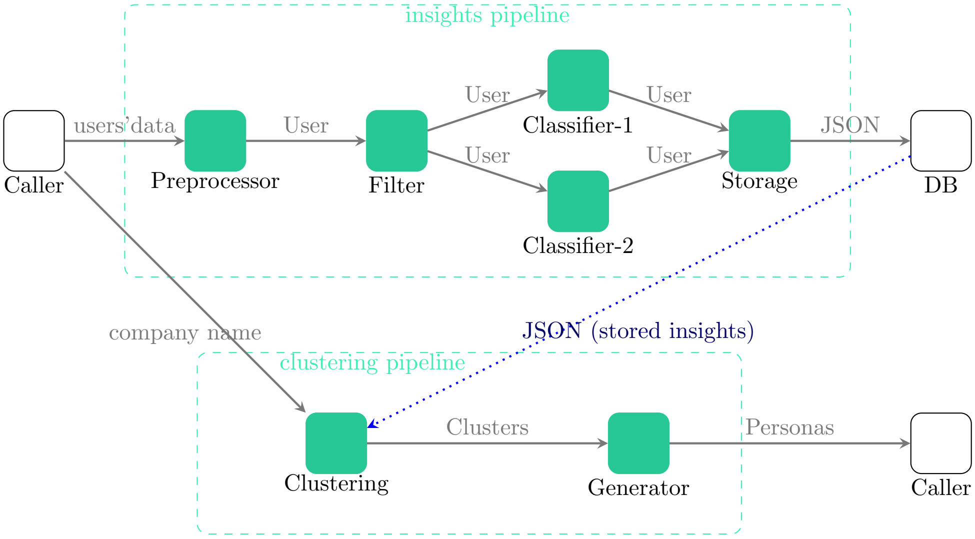


Figure 3.3: System representation example

Modularity

Personas can be characterized in many ways, for this reason it is important to have a dynamic approach while creating them. In such a way, it is possible to discover specific insights that a business could have interest in. To obtain this ability, the system must be modular, so that it is possible to add, remove and update the components that predict those characteristics and adapt them to a specific business case.

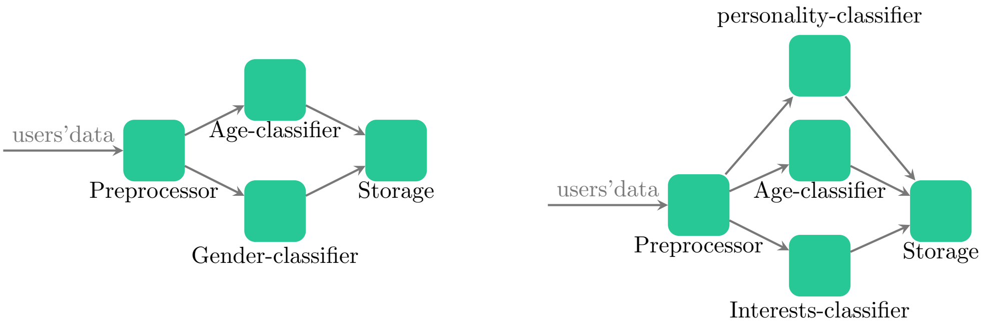


Figure 3.4: Morphing the pipeline structure

Scalability

Parallel processing can drastically improve processing time, allowing us to run each module of the system on a different machine. In this way, we can dedicate every resource of a physical machine to a single task.

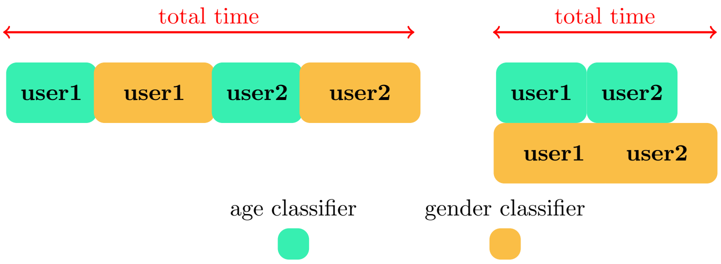


Figure 3.5: The advantage of parallelization: in the left, a single process handles all the computations, while in the right each process works on a specific task

3.2 Components Behaviour

The system is composed by independent components, each one of them handling a single task.
The output of a component will become the input for another one; in this way, chaining multiple components will create a pipeline. For example the age classifier component will only know how to predict the age of a user and, after the prediction, will send the results to the insights storage component, which is specialized in saving the insights in a permanent way.

The insights pipeline is composed by eight components:

- The twitter preprocessor which uniforms the structure of input data.
- The company filter, which uses a pretrained model to filter real people accounts from business accounts. It can also predict the gender and the age of an account.
- The general filter, which checks if a user has already been processed and saved in the database.
- The gender classifier.
- The age classifier.
- The mbti classifier, which tries to predict users' personality.
- The interests classifier, which analyzes texts and images to discover a user's interests.
- The insights storage, which stores the insights in a database.

The clustering pipeline is composed by two components:

- The clustering component, which loads the insights from the database and create clusters of users'insights. There are three variants of this component, each one using a different clustering algorithm.
- The personas generator, which uses the previously computed clusters to generate the marketing personas.

Finally it is possible to activate a *logger* component which receives the logs of the other components and saves them in a file. Components needs to exchange messages between each other and the protocol used for these exchanges must posses some characteristics: it must be asynchronous and it should allow to change the components with easiness. An easy protocol that has these characteristics is MQTT, a publish-subscribe ISO standard, that is integrated in the presented way into the design:
The age classifier subscribes to the topic '/data/formatted' and every time a message is sent to that topic, the classifier reads, processes and publishes it in the topic '/data/age'. If another component is interested in the output of the age classifier can subscribe to '/data/age' and read the data from there.

Every component needs to know an input topic, each time a message is received from that topic, the message is processed from the component. There are three kinds of outcomes of the processing

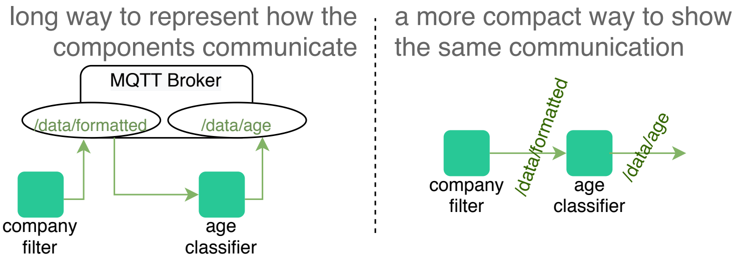
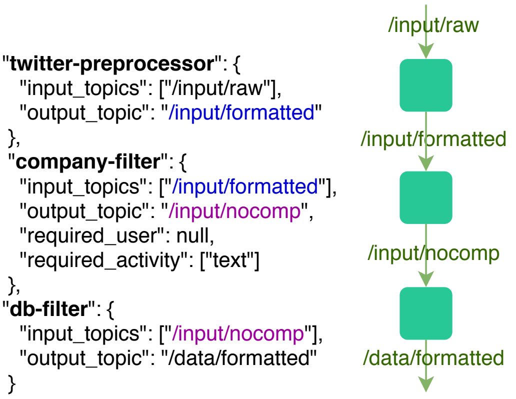


Figure 3.6: representation of how messages are sent and received

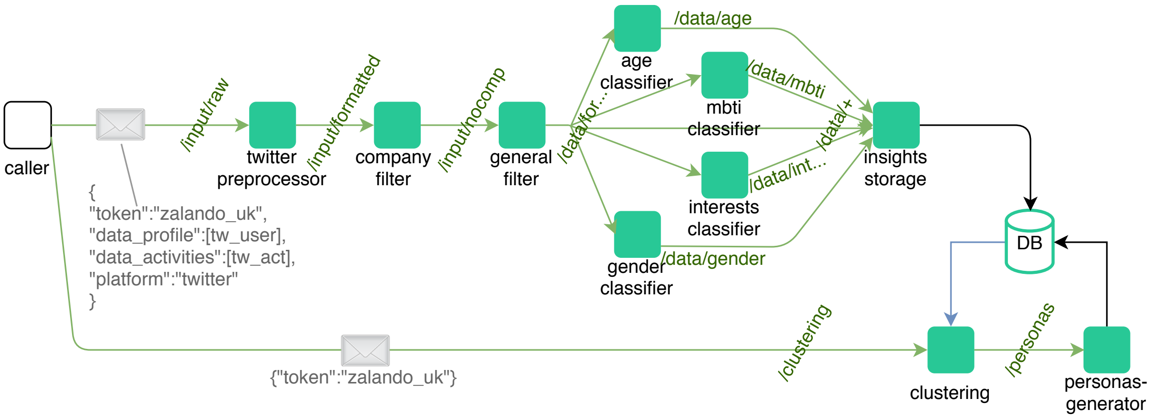


```
"twitter-preprocessor": {
   "input_topics": ["/input/raw"],
   "output_topic": "/input/formatted"
},
"company-filter": {
   "input_topics": ["/input/formatted"],
   "output_topic": "/input/nocomp",
   "required_user": null,
   "required_activity": ["text"]
},
"db-filter": {
   "input_topics": ["/input/nocomp"],
   "output_topic": "/data/formatted"
}
```

Figure 3.7: an example of settings.json file structure

phase: the component can publish the results on another topic (most common case, eg. age classifier), the component can store the data (eg. insights storage saves the insights on the database) or the component can do nothing (eg. company filter drops an invalid user).

To simplify the assignment of the topics and the behaviour of the system, a settings file (as in figure 3.7) is created: every component loads this file on startup and reads the information relevant to it (eg. its own input and output topic)



```
{
"token":"zalando_uk",
"data_profile":[tw_user],
"data_activities":[tw_act],
"platform":"twitter"
}
```

```
{"token":"zalando_uk"}
```

Figure 3.8: general structure of the system

3.3 Components Analysis: Classifiers

There are five components that try to predict users' insights using their social activities: age classifier, mbti classifier, interests classifier, gender classifier and company filter. This section presents how they work (Figure 3.8).

Every component is characterized by two functions, one that is called at startup, typically to load external resources, the other one that is called every time there is a message to process. Classifiers take a json formatted user as input message and publish a json formatted user in the output topics, containing their predictions.

The age classifier (Figure 3.9) component loads a pre-trained neural network at startup and uses it to guess users' ages. This model has been proposed by [36] and consists in a Wide Resnet [37] NN. The model's input is the picture of a human face and the output is his/her age.

The mbti classifier (Figure 3.10) component loads a pre-trained tf-idf word vectorizer and four

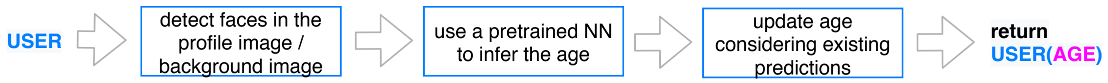

USER
detect faces in the profile image / background image
use a pretrained NN to infer the age
update age considering existing predictions
return
USER(AGE)

Figure 3.9: age-classifier chart

naive bayes classifiers, each one of them try to guess a different Myers-Briggs psychological function (sensation, intuition, feeling, and thinking).

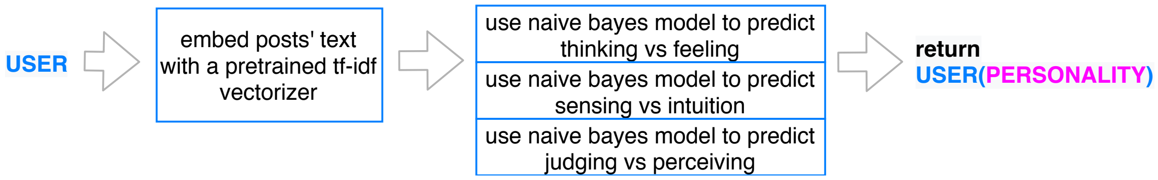

USER
embed posts' text with a pretrained tf-idf vectorizer
use naive bayes model to predict thinking vs feeling
use naive bayes model to predict sensing vs intuition
use naive bayes model to predict judging vs perceiving
return
USER(PERSONALITY)

Figure 3.10: mbti-classifier chart

The interests classifier (Figure 3.11) component works by extracting entities from users' activities in the form of wikipedia links, then compares the text of these wikipedia pages with a set of predefined documents, each one of these documents represent a different type of interest.

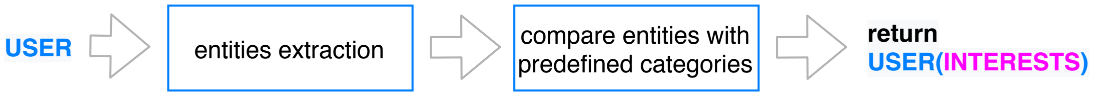


Figure 3.11: interests-classifier chart

The gender classifier (Figure 3.12) component tries to predict a user's gender using his/her name. This component extracts a user's first name from his social account, search that name in its map of names-genders and if the name is found assign the corresponding gender to the user. This approach is not very precise, so it is used when predictions made by the company filter component are insecure (explained below).

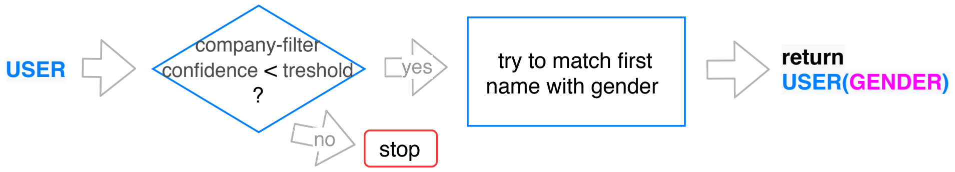

USER
company-filter confidence < treshold ?
yes
no
stop
try to match first name with gender
return
USER(GENDER)

Figure 3.12: gender-classifier chart

The company filter (Figure 3.13) main purpose is to predict if an account belongs to a real person or not. To make such a prediction a neural network is used. The NN structure was proposed in [38] and uses user-name, real name, biography and language to predict if the account is a company or not. This model is able to guess gender and age too, so these these two predictions will be used as basic insights that will be improved by the other classifiers.

3.4 Components Analysis: Clustering

There are three components that implement a clustering algorithm, each one of them: wait to receive a message containing a company's name, loads from the database users associated with that company, applies a clustering algorithm on the users and publish a message containing the clusters on a topic. Below the three clustering algorithms are presented (one for each component).

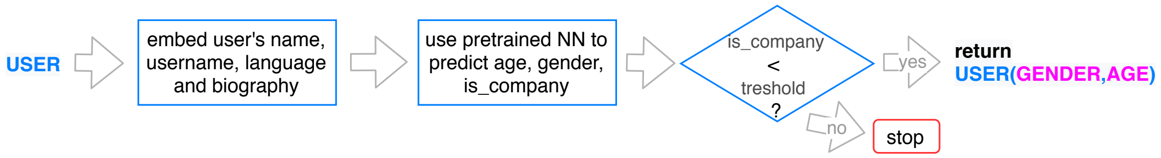

USER
embed user's name, username, language and biography
use pretrained NN to predict age, gender, is_company
is_company < treshold ?
yes
no
return USER(GENDER,AGE)
stop

Figure 3.13: company-filter chart

3.4.1 Algorithm n. 1: MIMOSA

Proposed by Marshall [39], this algorithm compares the *key* of a new point to the keys of points that the algorithm has already seen, if the similarity between the two keys is bigger than a predefined threshold, the new point is assigned to the same cluster of the other point. These keys consist in a list of strings and characterize a user in a similar way of a locality-sensitive hash. To build these keys three main components are considered: age, gender and a locality sensitive hash on the main principal components (PCA trained on a subset of the dataset).

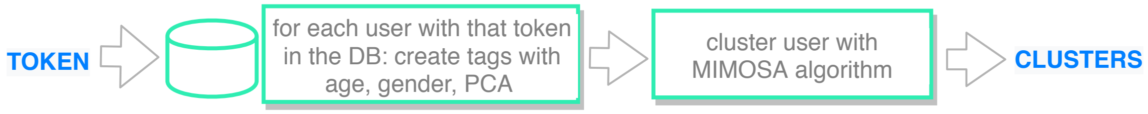

TOKEN
for each user with that token in the DB: create tags with age, gender, PCA
cluster user with MIMOSA algorithm
CLUSTERS

Figure 3.14: mimosa clustering chart

3.4.2 Algorithm n. 2: General

A very basic algorithm that compares new points with the centroids. Below, a pseudo implementation of the algorithm is presented.

Algorithm 1: General Clustering

```
general_clustering (centroids, new_point, threshold):
    max_similarity = threshold;
    centroid = new_point;
    for old_point in centroids do
        sim = sim_function(old_point, new_point);
        if sim > max_similarity then
            max_similarity = sim;
            centroid = old_point;
    end
    if centroid != new_point then
        centroids.append(centroid)
    return centroid
```

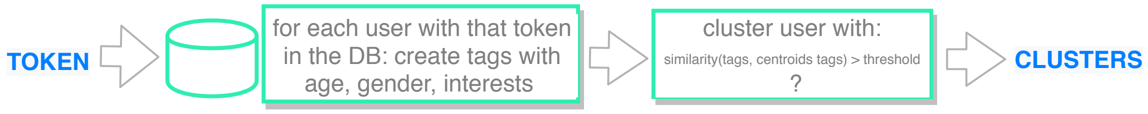

TOKEN
for each user with that token in the DB: create tags with age, gender, interests
cluster user with:
similarity(tags, centroids tags) > threshold
?
CLUSTERS

Figure 3.15: general clustering chart

3.4.3 Algorithm n. 3: Agglomerative Clustering

This is a hierarchical clustering algorithm that merges the most similar clusters into one at each step, building a tree. The distance metric used by this algorithm is the euclidean distance and before using this algorithm dimensionality reduction is used on the points.

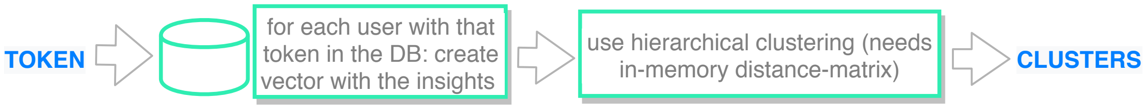
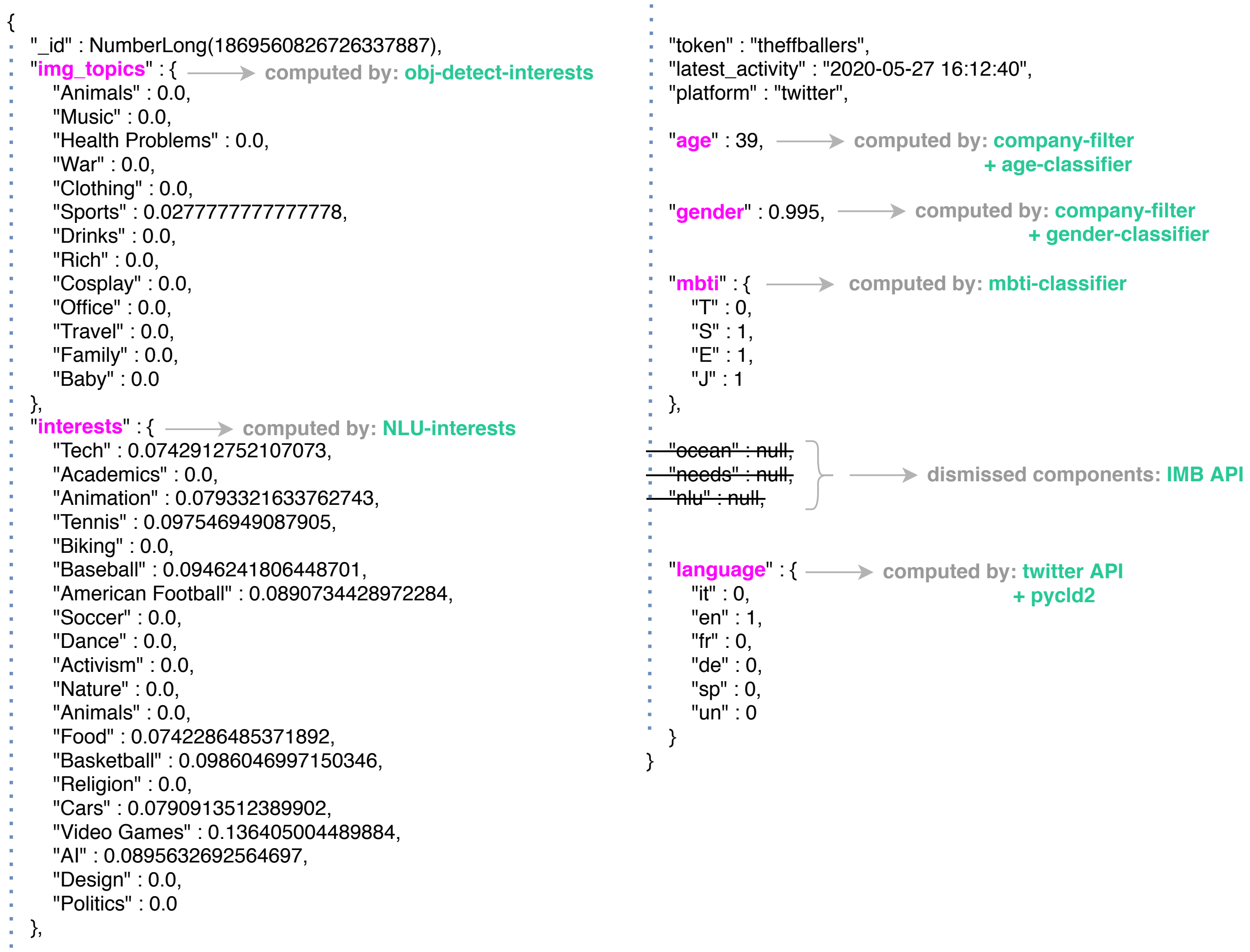

TOKEN
for each user with that token in the DB: create vector with the insights
use hierarchical clustering (needs in-memory distance-matrix)
CLUSTERS

Figure 3.16: agglomerative clustering chart

.

4 Implementation

This chapter explains how the system works and which tools have been used to build it. In the first part the general structure of the project is presented, while in the second every single component is discussed. Before diving into the details a sample of an insight is proposed (Figure 4.1). This sample represent the output of the insights pipeline that will be stored in the database and used by the clustering pipeline (along with other insights) to generate personas.



```
{
  "_id" : NumberLong(1869560826726337887),
  "img_topics" : {
    "Animals" : 0.0,
    "Music" : 0.0,
    "Health Problems" : 0.0,
    "War" : 0.0,
    "Clothing" : 0.0,
    "Sports" : 0.0277777777777778,
    "Drinks" : 0.0,
    "Rich" : 0.0,
    "Cosplay" : 0.0,
    "Office" : 0.0,
    "Travel" : 0.0,
    "Family" : 0.0,
    "Baby" : 0.0
  },
  "interests" : {
    "Tech" : 0.0742912752107073,
    "Academics" : 0.0,
    "Animation" : 0.0793321633762743,
    "Tennis" : 0.097546949087905,
    "Biking" : 0.0,
    "Baseball" : 0.0946241806448701,
    "American Football" : 0.0890734428972284,
    "Soccer" : 0.0,
    "Dance" : 0.0,
    "Activism" : 0.0,
    "Nature" : 0.0,
    "Animals" : 0.0,
    "Food" : 0.0742286485371892,
    "Basketball" : 0.0986046997150346,
    "Religion" : 0.0,
    "Cars" : 0.0790913512389902,
    "Video Games" : 0.136405004489884,
    "AI" : 0.0895632692564697,
    "Design" : 0.0,
    "Politics" : 0.0
  },
  "token" : "theffballers",
  "latest_activity" : "2020-05-27 16:12:40",
  "platform" : "twitter",

  "age" : 39,

  "gender" : 0.995,

  "mbti" : {
    "T" : 0,
    "S" : 1,
    "E" : 1,
    "J" : 1
  },

~~"ocean" : null,~~
~~"needs" : null,~~
~~"nlu" : null,~~

  "language" : {
    "it" : 0,
    "en" : 1,
    "fr" : 0,
    "de" : 0,
    "sp" : 0,
    "un" : 0
  }
}
```

computed by: obj-detect-interests
computed by: NLU-interests
computed by: company-filter
+ age-classifier
computed by: company-filter
+ gender-classifier
computed by: mbti-classifier
dismissed components: IMB API
computed by: twitter API
+ pycld2

Figure 4.1: sample of a JSON representing a user's insights

4.1 General Structure

Message Passing

There were many options to handle message passing, we choose to use the *MQTT protocol* because it is light and easy to use. Unfortunately MQTT is an IoT protocol and it is not designed for this kind of purpose, so if this system were to be taken in production, alternatives like Apache Kafka or RabbitMQ would be a better solution. The MQTT protocol has two main roles: clients and brokers. The broker implementation used was eclipse-mosquitto, an open-source project, while the client implementation used was *paho MQTT*, the most popular MQTT package for *python*.

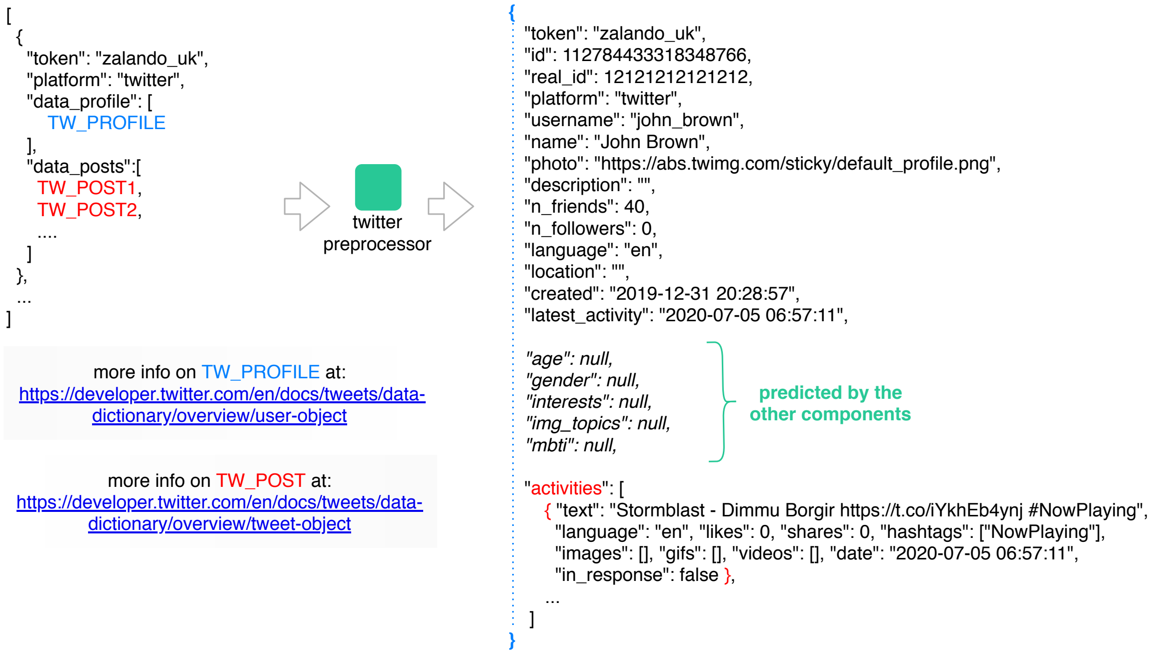
Persistence

The information extracted from the insights pipeline do not contain any personal data, so we can store and use them later in the clustering pipeline to build marketing personas. Excluding distributed storage systems like *Cassandra* (too complicated for this thesis) purpose and relational databases like *postgreSQL* and *mysql* (hard to change the database structure), a non-SQL database had to be chosen. Graph databases were a valid solution (social network are well represented with graphs), but the choice fell on *mongoDB*, a document-oriented database that stores documents in JSON format, which is the same format used in the messages between components. Another pro of *mongoDB* is the ease of configuration to run it in a *docker* container.

4.2 Single Components

Preprocessor

The first component of the system is the preprocessor, it receives JSON messages containing users' data from a social media platform and translates them in objects with a standard structure. In this way it is possible to treat data from different sources in the same way. A sample output produced by the preprocessor is shown in Figure 4.2.



```
[
 {
   "token": "zalando_uk",
   "platform": "twitter",
   "data_profile": [
      TW_PROFILE
   ],
   "data_posts":[
     TW_POST1,
     TW_POST2,
     ....
   ]
 },
 ...
]
```

more info on TW_PROFILE at:
https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/user-object

more info on TW_POST at:
https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object

```
{
  "token": "zalando_uk",
  "id": 1127844333318348766,
  "real_id": 12121212121212,
  "platform": "twitter",
  "username": "john_brown",
  "name": "John Brown",
  "photo": "https://abs.twimg.com/sticky/default_profile.png",
  "description": "",
  "n_friends": 40,
  "n_followers": 0,
  "language": "en",
  "location": "",
  "created": "2019-12-31 20:28:57",
  "latest_activity": "2020-07-05 06:57:11",

  "age": null,
  "gender": null,
  "interests": null,
  "img_topics": null,
  "mbti": null,

  "activities": [
    { "text": "Stormblast - Dimmu Borgir https://t.co/iYkhEb4ynj #NowPlaying",
      "language": "en", "likes": 0, "shares": 0, "hashtags": ["NowPlaying"],
      "images": [], "gifs": [], "videos": [], "date": "2020-07-05 06:57:11",
      "in_response": false },
    ...
  ]
}
```

Figure 4.2: a sample input (left) and a sample output (right) of the preprocessor's computation

Filters

The company filter component has been implemented using Wang et al. *python* package *m3inferece* [38]. This package is built on *pytorch* and is free for non-commercial use. When the function to process a message is called, the language of the user is predicted using the *pycld2* package, then a list of tensors is created (this list represent the embedding of the user's name, username, language and biography) sent to the *pytorch* model, composed by four LSTM NNs that are combined into a feed forward NN that produces the results. The pretrained model's weight can be downloaded from the *github* repository.

The general filter component contains a *mongoDB* client (implemented by the package *pymongo*), when a user is received the client queries the database to check if the user already exists, if it exists and does not have any new activities, it is dropped.

Demography Classifiers

The first guess on a user's age is made by the company filter component, then the prediction is improved by the gender and age classifiers.

The gender classifier uses a dictionary that maps first names to a gender. This map was built using 'Italian First Names dataset' by Alexander Cruz and 'US First Names Database' dataset by Len Fishman (both freely available at *data.world* website).

The age classifier uses the *dlib* package to predict if an image contains faces, then the portion of the image that contains the face is fed into a pretrained wide resnet that was implemented by Yusuke Uchida [36] (weights available in the *github* repository, trained on the APPA-REAL dataset).

Personality Classifiers

The MBTI personality classifier component uses the text of users' posts to predict a personality. When a user to process is received an embedding is created and then is passed to a set of pretrained naive bayes classifiers. these models have been built using *scikit learn* package and trained on two datasets: [40], [41]. To choose the machine learning model to use, four types of models were trained and tested (MLPClassifier, SVC, GaussianNB, RandomForestClassifier, from sklearn). Between these models the gaussian-naive-bayes-classifier had the highest precision and recall (the secon best model was the multi-layer-perceptron-classifier), so it was selected to be run in the component.

The OCEAN classifier used Watson IBM API to predict a user's personality. This component was dismissed because the free account was allowed to make only a thousand call per month.

Interests Classifiers

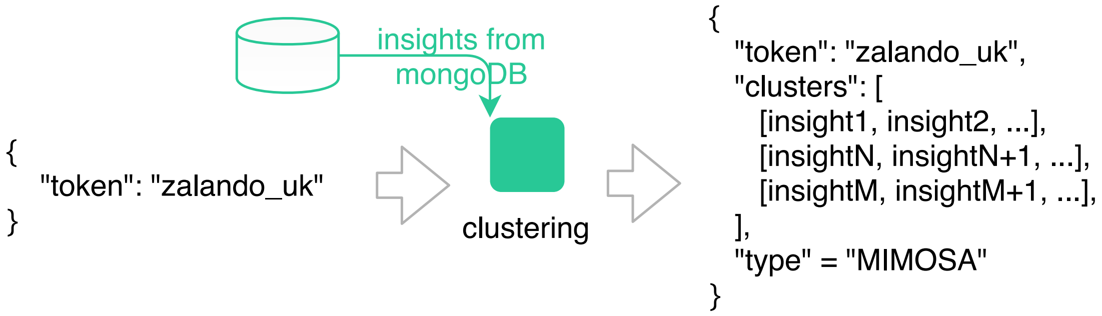
There are two components that try to discover users' interests: one uses NLU techniques to understand posts' text, while the other one uses a wide resnet model trained on the Imagenet dataset (available in the *pytorch* hub) to understand which objects are contained in posts' images.

There were two working components that calls IBM API to make these predictions, unfortunately, IBM Watson Visual Recognition does not provide a list of the classes that their default model can detect and this information was needed for computing personas in the clustering phase. IBM Watson NLU had the problem of limited calls per month, so a new component able to perform text analysis was built. This new component uses Tapoi API (a service offered by U-Hopper) to identify entities (in the form of Wikipedia's links) in texts. Extracts of these links are then downloaded using wikipedia API and compared with a set of manually curated texts representing interests. Text comparison is handled by the *whoosh* package, which uses the BM25F scoring function.

Clustering

Before clustering is applied we need to transform users' insights into computable data, for example before processing a user using agglomerative clustering, he is transformed into a vector of the form : [probIsMale, age, scoreInterest1, scoreInterest2, . . . , scoreInterestN]. There are three clustering component and the difference between them lies in the clustering algorithm. MIMOSA component uses PCA and std scaler from *scikit learn* and the MIMOSA algorithm implemented by Marshall [39]; Agglomerative clustering component uses a hierarchical clustering implemented by scikit learn and the General clustering component uses similarity between keys.

To improve the clusters' quality, the parameters used by the algorithms have been fine tuned manually. The adopted procedure to improve the parameters (called "manual" descent) is shown below.



```
{
  "token": "zalando_uk"
}
```

```
{
  "token": "zalando_uk",
  "clusters": [
    [insight1, insight2, ...],
    [insightN, insightN+1, ...],
    [insightM, insightM+1, ...],
  ],
  "type" = "MIMOSA"
}
```

Figure 4.3: a sample input (left) and a sample output (right) of a clustering component

Algorithm 2: "manual" Descent

```
improve_parameters (trainingTokens,validationToken):
    notvalidated = true;
    unsatisfiedWithClusters = true;
    while notvalidated do
        while unsatisfiedWithClusters do
            changeClusteringParameters();
            clusters = generateClusters(trainingTokens);
            unsatisfiedWithClusters = evaluate(clusters);
        end
        clusters = generateClusters(validationToken);
        notvalidated = evaluate(clusters);
    end
```

We expect that fine tuning the parameters on a sample of companies' tokens, will improve the algorithms even with new tokens that were not used in this process.

5 Evaluation

This section shows the results obtained. In the fisrt section the goodness of the insights is measured, while in the second the generated clusters are evaluated.

5.1 Insights Prediction

Age

The age of a user is predicted by 2 components: The age classifier (which uses profile's photo) and the company filter (which uses text features). The final age is calculated with the formula $age = c * ageClf + (1 - c) * compFil$ where age is the predicted age, $c \in [0, 1]$ (after some trials the best c found was 0.6), $ageClf$ is the prediction of the age classifier and $compFil$ is the prediction of the company filter.

To evaluate how good the prediction of this insights are, 100 random accounts of real people were taken and, for each of them, a ground-truth age was set. To measure how good the predictions were, the difference between the predicted age and the ground truth age was taken as error metric, showing an average error of 3 years.

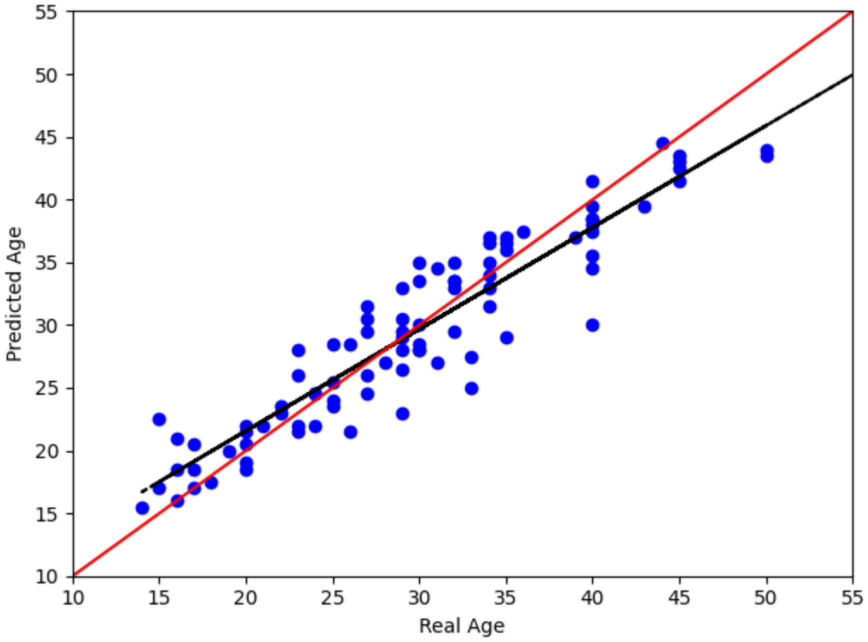


Figure 5.1: real age vs predicted age chart

Gender

Similarly to the age insight, the gender insight is predicted by two components. To obtain the final prediction, the results of the company filter and the gender classifier need to be condensed into one. The adopted formula used to merge the predictions is showed below and can be summarized with this statement: use $compFil$ result if its confidence is bigger than a threshold t, use $genClf$ otherwise (trial and error showed that 0.7 was a good threshold).

$$gender = \begin{cases} compFil & \text{if compFil} > t \\ genClf & \text{otherwise} \end{cases}$$

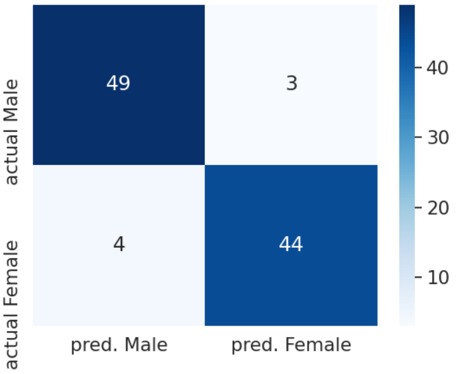
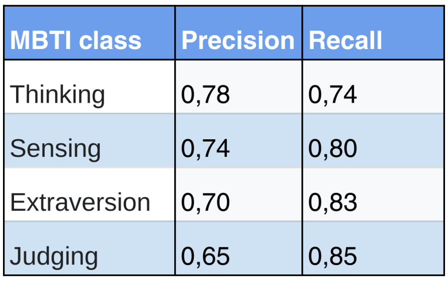


Figure 5.2: gender confusion matrix

As for the age, the predictions of this insights were evaluated against 100 random accounts (whose gender was determinated by a human), showing a precision of 0.92, recall of 0.94 in detecting males and a precision of 0.94, recall of 0.91 in detecting females. Figure 5.2 presents the resulting confusion matrix.

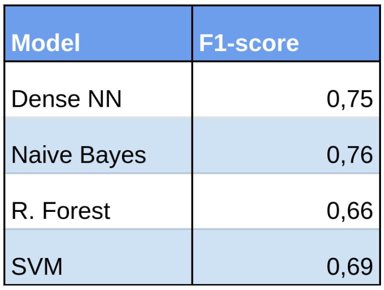
MBTI

This component was trained on a dataset containing tweets in english and the MBTI class associated to them. The dataset was split in two parts, than the models were trained on the training set and evaluated on the test set. The evaluation phase showed that the most performing model was the naive bayes classifier (Figure 5.4).



MBTI class	Precision	Recall
Thinking	0,78	0,74
Sensing	0,74	0,80
Extraversion	0,70	0,83
Judging	0,65	0,85

Figure 5.3: results on the different MBTI classes (GNB)



Model	F1-score
Dense NN	0,75
Naive Bayes	0,76
R. Forest	0,66
SVM	0,69

Figure 5.4: comparing models by F_1score on the Extraversion class

Interests

5.2 Generated Personas

If we show that clusters represents relevant segments of customers, we can expect that the marketing personas hold a strategic value.
Unfortunately, evaluating clusters is not a trivial task, especially with high dimensional data. For this reason, a visual approach was used to evaluate the clustering algorithms: the insights were transformed into vectors of floats, then dimensionality reduction was applied to transform the vectors in bidimensional arrays. Finally these arrays were plotted into a chart where each color represents a different cluster. Below (Figure 5.5, 5.6, 5.7) some charts, showing the clusters, are presented.

The visual approach showed that the general clusterig algorithm made a better job finding the clusters. We can also note that the clusters are differentiated from their position in the x-axis, in fact that axis represent the principal component, which holds most of the variance of the set of insights.

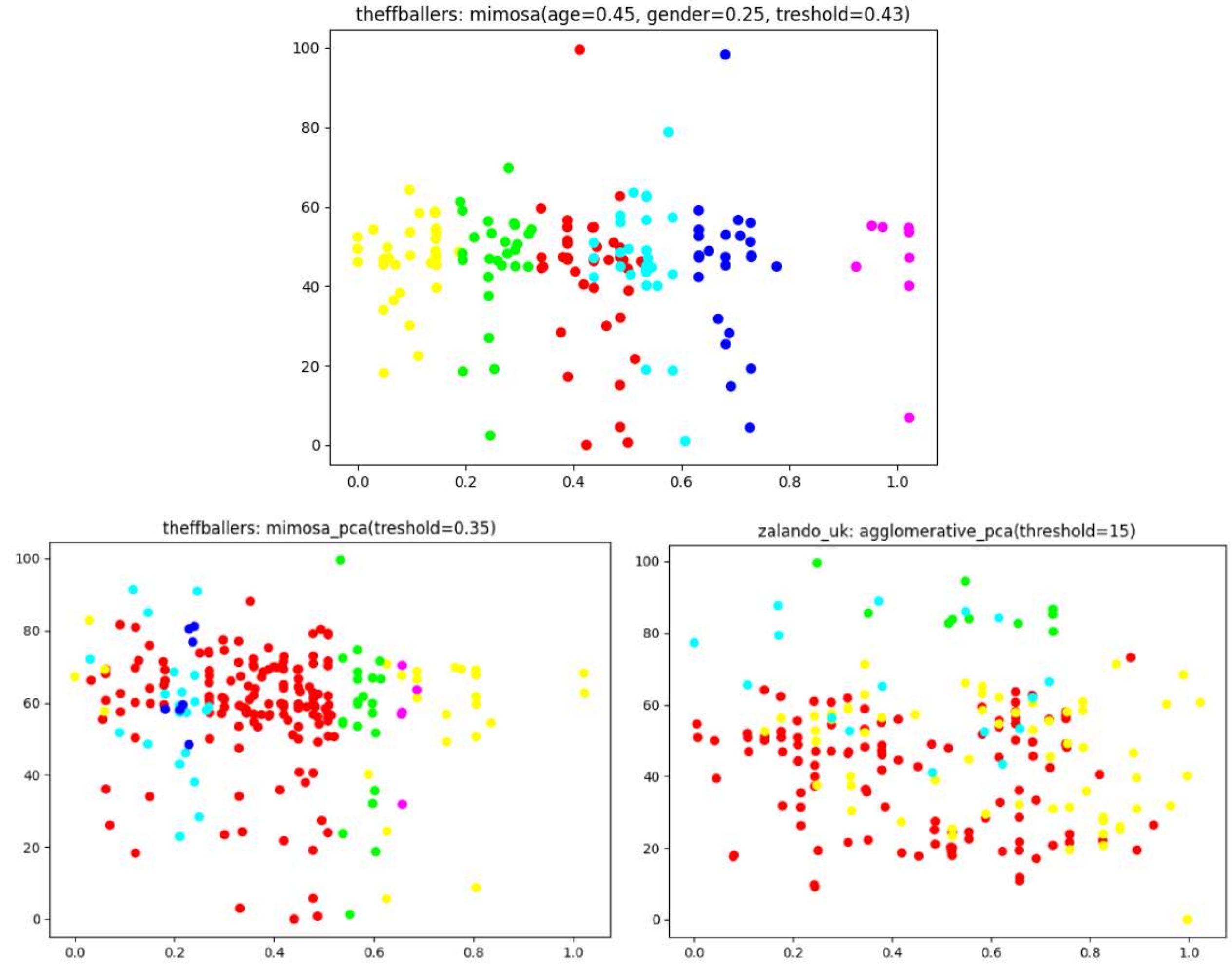
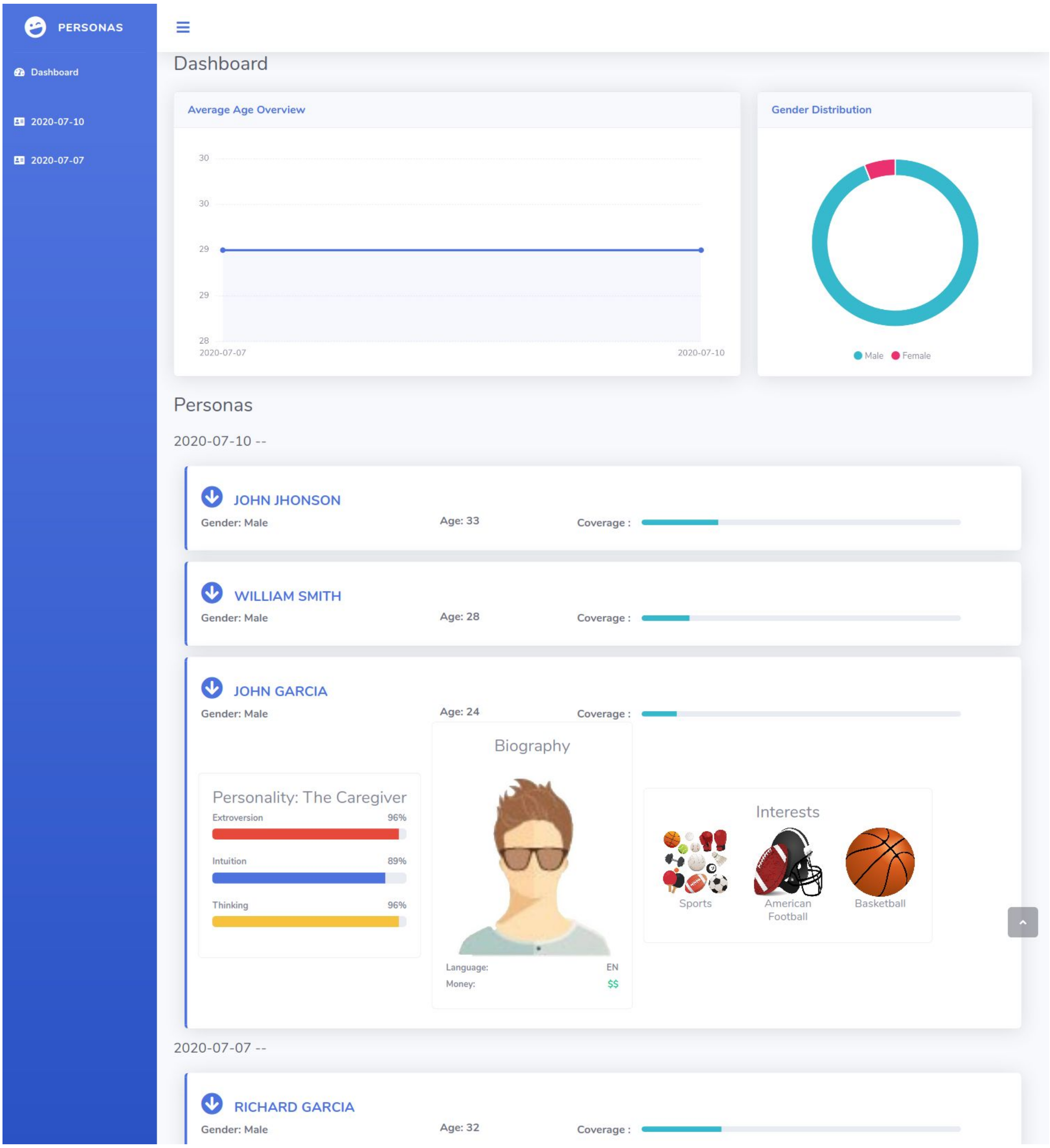


Figure 5.5: clusters found with the general (top), mimosa (bott. left), hierarchical (bott. right) algorithms

After having looked at the clusters, we must check if the personas contain useful information; but evaluating personas directly can be even harder than evaluating clusters. To accomplish this task, a non-empirical approach was used: we can try to guess what personas should emerge from an account and see if the results match with our guesses.

The accounts considered in this test were:
theffballers - an account that publishes many tweets about american football matches, so we expect to obtain personas interested in sports whose gender is male.
patmcgrathreal - an account owned by a makeup artist, so we expect to have mainly female personas.

These assumption were not based on guts, but on a manually collected ground truth, that consisted in information manually gathered looking at profiles that were following those accounts. The guesses that we made were confirmed by the dashboard created by the prototype, so we can say that the automatically generated personas hold real information in them. The personas generated from the prototype for the account *theffballers* can be seen below (Figure 5.6) and here live: https://thomasreolon.github.io/theffballersPersonas/.



2020-07-10 --

JOHN JHONSON
Gender: Male
Age: 33
Coverage :

WILLIAM SMITH
Gender: Male
Age: 28
Coverage :

JOHN GARCIA
Gender: Male
Age: 24
Coverage :

Personality: The Caregiver

Extroversion	96%
Intuition	89%
Thinking	96%

Biography

Language: EN
Money: $$

2020-07-07 --

RICHARD GARCIA
Gender: Male
Age: 32
Coverage :

Figure 5.6: example of a dashboard generated for *theffballers*

6 Conclusion

Even if the proposed system is just a prototype, it is able to analyze users, extract insights and present them in a friendly way (marketing personas). In the *Evaluation Chapter* we showed that the generated personas represent real segments of customers. For this reason, they hold a strategic value that could be used to direct advertising campaigns and marketing choices.

Being just a prototype, this system extracted only the most trivial insights (age, gender, interests, personality, language). Hence someone could argue that this kind of information do not bring real value to a company. Luckily, the system was thought to be highly customizable and upgradable, so new classifiers can be integrated to predict new insights and improve the existing ones.

Future Work

There are two principal ways to improve the actual system: change message-passing protocol and add new components. The latter one consists in creating new classifiers, filters etc. that could be added to the system to skyrocket the functionality offered by it. Among all the possible choices, the first component I would upgrade is the interests classifier that analyzes images. This component uses a wide resnet pretrained on the imagenet dataset, so the classes that this model detects are not directly linked to interests. The easiest solution would be to re-train the model on a custom dataset with more relevant classes.

Changing the message-passing protocol would prevent all the drawbacks that MQTT has: the broker is a single machine, hence it cannot handle massive data streams. A valid alternative could be Apache Kafka, because it is a distributed stream processing platform developed especially for this kind of projects.

Bibliography

[1] N. Patel, “Why understanding your customers is the only marketing strategy you need,” *Neilpatel*, 2017. [Online]. Available: https://neilpatel.com/blog/the-only-marketing-strategy-you-need/

[2] A. Wilson-Rew, “12 statistics that prove the value of creating buyer personas,” 2015. [Online]. Available: https://www.protocol80.com/blog/2015/08/14/11-buyer-persona-statistics-that-prove-personas-are-awesome/

[3] Y.-n. Chang, Y.-k. Lim, and E. Stolterman, “Personas: from theory to practices,” in *Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges*, 2008, pp. 439–442.

[4] J. King, “How to build buyer personas for free using 3rd party data,” *leahyking*, 2019. [Online]. Available: https://leahyking.com/blog/how-to-build-buyer-personas-for-free-using-3rd-party-data/

[5] J. Theng, “Artificial intelligence vs buyer personas. Who wins?” 2018. [Online]. Available: https://justintheng.com/artificial-intelligence-vs-buyer-personas-who-wins/

[6] L. Kevan, “How to create marketing personas,” *Medium*, 2015. [Online]. Available: https://medium.com/social-media-tips/how-to-create-marketing-personas-2868c8a1c6d5

[7] G. Porzionato, “Do a better (buyer) persona,” *Medium*, 2019. [Online]. Available: https://medium.com/ghostwriter-ai-content-marketing/https-medium-com-ghostwriter-ai-content-marketing-how-ai-can-improve-your-buyer-personas-62ad9a05dd

[8] J. Salminen, S. Sengün, S.-g. Jung, and B. J. Jansen, “Design issues in automatically generated persona profiles: A qualitative analysis from 38 think-aloud transcripts,” in *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, 2019, pp. 225–229.

[9] J. An, H. Kwak, S.-g. Jung, J. Salminen, and B. J. Jansen, “Customer segmentation using online platforms: isolating behavioral and demographic segments for persona creation via aggregated user data,” *Social Network Analysis and Mining*, 2018.

[10] J. An, H. Cho, H. Kwak, M. Z. Hassen, and B. J. Jansen, “Towards automatic persona generation using social media,” in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*. IEEE, 2016, pp. 206–211.

[11] J. An, H. Kwak, and B. J. Jansen, “Automatic generation of personas using youtube social media data,” in *Proceedings of the 50th Hawaii International Conference on System Sciences (HICSS)*, 2017.

[12] S.-G. Jung, J. An, H. Kwak, M. Ahmad, L. Nielsen, and B. J. Jansen, “Persona generation from aggregated social media data,” in *Proceedings of the 2017 CHI conference extended abstracts on human factors in computing systems*, 2017, pp. 1748–1755.

[13] M. Koponen, “Developing marketing personas with machine learning for educational program finder,” 2017.

[14] L. Phillips, C. Dowling, K. Shaffer, N. Hodas, and S. Volkova, “Using social media to predict the future: a systematic literature review,” *arXiv preprint arXiv:1706.06134*, 2017.

[15] D. Nguyen, R. Gravel, D. Trieschnigg, and T. Meder, “”how old do you think i am?” a study of language and age in twitter,” in *Seventh International AAAI Conference on Weblogs and Social Media*, 2013.

[16] V. Carletti, A. Greco, G. Percannella, and M. Vento, “Age from faces in the deep learning revolution,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[17] J. Levy Abitbol, E. Fleury, and M. Karsai, “Optimal proxy selection for socioeconomic status inference on twitter,” *Complexity*, 2019.

[18] N. Aletras and B. P. Chamberlain, “Predicting twitter user socioeconomic attributes with network and language information,” in *Proceedings of the 29th on Hypertext and Social Media*, 2018, pp. 20–24.

[19] T. Hu, H. Xiao, J. Luo, and T.-v. T. Nguyen, “What the language you tweet says about your occupation,” in *Tenth International AAAI Conference on Web and Social Media*, 2016.

[20] P. Peñas, R. Del Hoyo, J. Vea-Murguía, C. González, and S. Mayo, “Collective knowledge ontology user profiling for twitter–automatic user profiling,” in *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, vol. 1. IEEE, 2013, pp. 439–444.

[21] C. Torrero, C. Caprini, and D. Miorandi, “A Wikipedia-based approach to profiling activities on social media,” *arXiv preprint arXiv:1804.02245*, 2018.

[22] G. Carducci, G. Rizzo, D. Monti, E. Palumbo, and M. Morisio, “Twitpersonality: Computing personality traits from tweets using word embeddings and supervised learning,” *Information*, vol. 9, no. 5, p. 127, 2018.

[23] D. Quercia, M. Kosinski, D. Stillwell, and J. Crowcroft, “Our twitter profiles, our selves: Predicting personality with twitter,” in *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing.* IEEE, 2011, pp. 180–185.

[24] M. Moon, A. Farooq, and M. Kiran, “Social shopping motivations of impulsive and compulsive buying behaviors,” *UW Journal of Management Sciences*, vol. 1, pp. 15–27, 10 2017.

[25] J. Chen, E. Haber, R. Kang, G. Hsieh, and J. Mahmud, “Making use of derived personality: The case of social media ad targeting,” in *Ninth International AAAI Conference on Web and Social Media*, 2015.

[26] G. Seif, “The 5 clustering algorithms data scientists need to know,” *towardsdatascience*, 2018. [Online]. Available: https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68

[27] A. Ahmad and S. S. Khan, “Survey of state-of-the-art mixed data clustering algorithms,” *IEEE Access*, vol. 7, pp. 31 883–31 902, 2019.

[28] W. A. AlZoubi, “A survey of clustering algorithms in association rules mining,” 2019.

[29] S. Firdaus and M. A. Uddin, “A survey on clustering algorithms and complexity analysis,” *International Journal of Computer Science Issues (IJCSI)*, 2015.

[30] A. Jhunjhunwala, “A continuously updating k-means algorithm,” *towardsdatascience*, 2019. [Online]. Available: https://towardsdatascience.com/a-continuously-updating-k-means-algorithm-89584ca7ee63

[31] A. K. Menon, A. Rajagopalan, B. Sumengen, G. Citovsky, Q. Cao, and S. Kumar, “Online hierarchical clustering approximations,” *arXiv preprint arXiv:1909.09667*, 2019.

[32] T. Filaire, "Clustering on mixed type data," *towardsdatascience*, 2018. [Online]. Available: https://towardsdatascience.com/clustering-on-mixed-type-data-8bbd0a2569c3

[33] A. Toor, "An advanced clustering algorithm (ACA) for clustering large data set to achieve high dimensionality," *Global Journal of Computer Science and Technology*, 2014.

[34] S. Wang, B. Tu, C. Xu, and Z. Zhang, "Exact subspace clustering in linear time," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[35] B. Rozemberczki, R. Davies, R. Sarkar, and C. Sutton, "Gemsec: Graph embedding with self clustering," in *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2019.

[36] Y. Uchida, "Age estimation pytorch," 2019. [Online]. Available: https://github.com/yu4u/age-estimation-pytorch

[37] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[38] Z. Wang, S. A. Hale, D. Adelani, P. A. Grabowicz, T. Hartmann, F. Flöck, and D. Jurgens, "Demographic inference and representative population estimates from multilingual social media data," in *Proceedings of the 2019 World Wide Web Conference.* ACM, 2019.

[39] J. A. Marshall and L. C. Rafsky, "Exact clustering in linear time," *arXiv preprint arXiv:1702.05425*, 2017.

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

[41] S. Sehwag, "Twitter mbti personality classifier," 2018. [Online]. Available: https://github.com/sahilsehwag/twitter-mbti-personality-classifier