SAPIENZA
UNIVERSITÀ DI ROMA

Prediction of Twitter's contents through the analysis of users' similarities

Facoltà di Ingegneria dell'informazione, informatica e statistica

Corso di Laurea Magistrale in Ingegneria informatica

Candidate

Martina Baccini

ID number 1207946

Thesis Advisor

Prof. Leonardo Querzoni

Academic Year 2013/2014

Thesis not yet defended

Prediction of Twitter's contents through the analysis of users' similarities
Master thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: martina.baccini@alice.it

Dedicato a mia madre e mia sorella
che mi hanno sempre sostenuto
e incoraggiato durante questo percorso.
A Danilo che mi è sempre stato accanto.
E a mio padre che sono sicura
sarà fiero di me.

Contents

Introduction

A social network, by definition, is a social structure made up of a set of social actors connected to each other through different types of social bonds, such as family ties or employment relationships or even a superficial knowledge.

Anthropologist J.A. Barnes describes social networks as a "set of points connected by lines. Points represent the people and also the groups and the lines indicate which people are interacting with each other".

The nodes that make up the network may consist of individuals, groups or institutions; while the relationships that link together the subjects of the social network may be unidirectional or bidirectional.

The unidirectional relationships indicate that the connection is one way, that is a person is linked to another through a bond, but this is not reciprocated. This structure is at the base of Twitter, in fact in this social network a user can follow another one (one-way link) but it is not followed in turn.

On the contrary, bidirectional relations, are relative to two people who are related to each other, for example, this type of bond is at the base of the social network Facebook, where two users are friends only if both accept the relationship of friendship.

In the last decade the online social networks have occupied an increasingly important role in people everyday life, becoming the way for connecting them on the internet. In fact, OSNs are used as a way to not only

rediscover old friends and acquaintances, but also to make new ones with whom share common interests.

In addition they allow people to get lots of information, remain up to date on what is happening around them and the people who have as friends in their social network, having the possibility of running into opportunities, such as find new jobs, in which they would never come across otherwise.

Over time sociologists expected that OSNs' structure would reflect the real-life society and relationships, allowing to study models about communities' behavior.

Computer science covers an important role in this scenario, as it provides tools and techniques for the process of data collection and their analysis.

Nowadays has become common practice for people to use social networks as a way to share with their friends messages and thoughts, often providing personal information too easily, for example their workplace or home address, or messages related to what they think about unimportant topics like the weather, or most sensitive topics such as politics, not realizing the consequences that this may cause them.

The most obvious example is to publish when someone go on vacation, thus informing that their house will remain empty and unattended.

Also information that users entrust superficially to social networks as considered unimportant and with no impact on their lives, such as their place of birth or the school attended, actually can be exploited by third parties which, through prediction techniques, can obtain private information of these same users who did not want to make public.

In this scenario take place the concept of privacy and the problem related to it that arises from the use of social networks by people in a careless and shallow way.

In fact, over time, the warnings that have spread to the users of social

networks on paying attention on what they publish online has made them more aware about their actions.

In order to remedy to this new problem, recently the OSNs have increased their privacy services, by entrusting to users the choice about what to make public or not, regarding their own information.

Facebook, for example, allows users to manage not only who can have access to certain details among their friends, but also to decide from who can be contacted, for example strangers with whom have no mutual friend, with the ability to report any abuse by anyone.

Indeed, in certain occasions a person wants that personal information about himself, such as employment status, personal photos or his list of friends, to be known only by a small circle of close friends, and not by strangers. Instead in other instances, prefers to reveal personal info to anonymous strangers, but not to who know him better.

On the contrary, other social networks, carry out privacy control in a different way.

Twitter users can communicate with others publicly or securely sending private messages, while tweets (short messages of 140-characters length) that are publicly visible by default, can be delivered in a restrictive way just to user's followers.

In this context take place the problem faced in this thesis concerns how public information of users present on the Online Social Networks can be exploited, by third parties, in order to predict users' private information.

In fact, people too often believe they are free to decide which details make public or not without knowing that, even from little information they make visible, it is possible to deduce what they do not want to reveal, as their interests.

In particular, the social network considered in this project, and to which datasets studied belonged to, is Twitter, an OSN service that enables

users to send and read messages called "tweets"; while users considered are nothing more than people who have joined it, registering their profile. The structure on which Twitter is based is different from the majority of others social networks, such as Facebook or Google+, because is not founded on the concept of friendship between users, as already mentioned. A recent study [Java et al., 2007; Krishnamurthy et al., 2008] has revealed that few users in Twitter have reciprocal relationships each other.

In fact, a Twitter user can follow on the social network anyone without needing a friendship request (usually people follow their favorite artists or celebrities). People who follow others are known as "followers" while people who are followed are known as "followee".

In 2013 Twitter was one of the ten most-visited websites, and has been described as "the SMS of the Internet". In fact, as already mentioned, the way in which users can communicate with each other on this social network is just through tweets. A tweet is a text message posted by a user, which can be attached to a video, a photo or a link with a character limit set for each uploaded to Twitter of 140 characters.

This limit has been introduced for compatibility with SMS messaging and has the advantage of "forcing" people to get right to the point of the speech that they intend to address, without getting lost in useless chatter.

Because of the very short length of messages that can be posted by people on this social network, to emphasize keywords or topics in a tweet, have been introduced special words called "hashtag". A hashtag is a word or an unspaced phrase prefixed with the hash character (or number sign), #, to form a label, e.g., #hashtag. It rapresents a form of metadata tag and allows grouping of similarly tagged messages.

This work had as its object of study the hashtag contained in the tweets published by users belonged to the datasets studied, in order to predict

their future ones.

In particular, the purpose that was wanted to achieve with the prediction of such hashtag was to show how, using public information of a user, is possible to reveal his interests that have not been disclosed.

In general, the ability to reveal users' taste can have pro and con. In fact figure out what likes to a user allow advertising companies to recommend him new pages and people to follow on Twitter that he might be interested. This makes the user satisfied for having received suggestions in line with his own interests.

However, in other situations, information predicted by third parties can be dangerous or even harmful to the users involved. For example if a user wants to keep secret his interest in a sensitive topic such as his political idea, this is no longer possible because through his tweets and pages that have been recommended to him it can be deduced its line of thought with the possibility of incurring inconvenience(for example can be considered a possible threat to the country and then be controlled by the government).

As already explained, all these repercussions for users of OSNs derive from the superficiality with which sometimes people share their personal information, and how they express their interests and their thoughts using tweets as method of communication.

This is why it is so important to study the problem related to the prediction of a user's information presenting methods more or less effective, in order to evaluate the results obtained and the impact on every day life.

For this purpose, in the first chapter of this thesis has been made an overview of the studies carried out by researchers in recent years, describing and comparing different aspects that have been treated, showing new approaches and techniques in order to recommend items to users of a social network.

In the second chapter instead has been described the framework realized, and how was possible to obtain, given a set of data as input, users' future hashtag, describing prediction tecniques used and the metrics chosen to evaluate the results obtained.

The following chapter contains the analysis of datasets considered and their structure.

Several tests were conducted on both the dataset in order to assess the accuracy of the methods used for the prediction, by varying several times predetermined parameters to verify their influence on the results obtained. Precision, recall and f-measure metrics were chosen to evaluate if hashtags predicted were correct.

Finally the two datasets were compared highlighting how, the same techniques applied to different data lead to different results.

Chapter 1

State of the art

The widespread use of online social networks in recent years, and the problem of privacy emerged from the exploitation of public information of users by third parties, has contributed to the creation of numerous articles and research related to the inference information on OSNs.

Over time, several techniques have been presented in order to demonstrate which information can be extracted from users' profile on an OSN, through details that they consciously make public, allowing others to predict their tastes and personal data.

The concept of inferring users' private attributes, such as the school they attended or the work they do, or even the city in which they live, is not so different from the problem related to predict users' tastes in music or movies.

In fact, the advertising companies, every day looking for more precise methods that can be used to recommend users objects to which they will be interested. This was possible thanks to the born and the developement of recommendation systems, whose purpose is to provide users recommendations in line with their tastes (eg Amazon), and for this reason have been used even in the field of information inference on social network.

1.1 Recommender systems

During the last ten years, the role of recommender systems has become increasingly important in the internet industry to be considered one of the most powerful business tools to recommend products to network's users.

A Recommender System (RS), by definition, is a filtering content software that guides the user through his choices, giving to him personalized results.

Nowadays, collaborative Filtering (CF) represents a widely adopted strategy to build recommendation engines. The first work on the field of CF was the Tapestry system [1], that used collaborative filtering to filter mails based on the opinion of other users, expressed with simple annotations.

In CF systems a user is recommended items based on the past ratings of all users collectively.

This method uses an approach that cares most about social implications of the recommendation proceeding. Instead of recommending to the user similar items to which he choose in the past, the system recommends items liked by users similar to the current one, assuming that people who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past.

One of the most famous examples of collaborative filtering is item-to-item CF used by Amazon.com's recommender system. This approach let Amazon to build, for each item X, a neighborhood of related items S(X); in this way when a user buy or look an item, Amazon can recommends him another one from the item's neighborhood, and with an high probability, he will like it.

Based on this, the approach adopted by collaboritive filtering appears to be the most effective way to reach the goal that is wanted to achieve with this project, that is to predict users' future hashtags. In fact, as a surrogate of items of interest to recommend to a user it can be used hashtags, which are considered as real items to predict.

The real purpose of CF resides in make predictions about unknown preferences of

users analyzing the known preferences of a group of users similar to them.
CF systems in the literature are often divided in two groups:

- memory-based;
- model-based.

Memory-based algorithms consider the all ratings available to generate predictions. The most used memory approach, in the contest of collaborative filtering, is the one based on neighborhood model.
The idea behind this methods is to obtain a similarity value between users or items, that represents the distance between them.
The two similarity metrics most used are the pearson correlation [2] and the cosine similarity [3].
Based on the neighborhood model, predictions for a user/item are made considering the ratings of the k most similar users/items (that are neighbors).
On one hand, *user-oriented methods* based the evaluation of unknown ratings using recorded ratings of similar users (neighbors) (read [4] for a complete analysis). On the other hand, *item-oriented methods* [5] make a prediction using the known ratings made by the same user on similar items.
Thanks to their simple design and implementation , memory-based methods have reported a large utilization in a lot of real-world systems even though, scalability limitations, have made impractical their use related to large amounts of data.
Model-based approaches use the known ratings to evaluate or learn a model and then use it to make predictions, in order to overcome memory-based algorithms limitations.
In particular, for this thesis work has been used a memory-based model, choosing a user-oriented method for the computation of hashtag predicted.

1.2 Papers about recommendation for OSNs

Having conquered social networks such an important role in recent years the state of art is rich of different methods proposed and procedures applied to datasets belonging to different OSNs, with the purpose of inferring attributes and study the accuracy and limitations of the data obtained.

An example lies in the study conducted by Zheleva and Getoor [8] who have treated the problem of privacy related to the prediction of private attributes of OSNs' users. In fact, their purpose was to show how an adversary can exploit an online social network with a mixture of public and private user profiles to predict private information of users.

This has been the first work that used link-based and group-based classification to study privacy implications in social networks.

Disclosing private information is an infringement people' rights to choose who can know their private information and it is very important to know how prevent this kind of attack by an adversary on a social network.

This is why different models for inferring sensitive attributes in OSNs have been proposed in this paper, and have been evaluated the effectiveness of each one.

These models can be divided into three main groups:

- privacy attacks without links and groups;
- privacy attacks using links;
- privacy attacks using groups.

Online communities considered to test these methods have been four: the photo-sharing website Flickr, the social network Facebook, Dogster, an online social network for dogs, and the social bookmarking system BibSonomy and it has been shown that, despite a person not to make public the information on their profile, affiliations to other pages or their friendships, on the contrary, are public and this leads to a surprisingly loss of information. In fact ,using group information, it was possibile to

discover the sensitive attribute values of some users with high accuracy on all the four real-world social-media datasets.

A different aspect, related to the prediction of user content in OSNs, was instead treated by Das and Datar [9] in their work. In fact their purpose was not to infer private informations about users, but study an approach to collaborative filtering in order to predict personalized recommendations for people. In particular, purpose of this work, was to recommend to users of Google News the "Top-k stories" in which they could be more interested.

Differently from the previous work in this one has been used a mix of memory based and model based algorithms to generate recommendations, using two clustering techniques as part of model-based approach, PLSI and MinHash [1] , and using item covisitation as part of memory based methods.

Each of the cited algorithms assigns a numeric score to a story in order to predict to each user those who have the highest one. The cluster approach assigns a score proportional to the fractional membership of the user, while the covisitation algorithm assigns a score to each candidate story, which is proportional to the number of times the story was covisited by the user. Finally the Top K stories to recommend to a user were chosen from the weighted list obtained combining the results of the two algorithms described.

In addition, in this work have also been considered other dataset to test the quality evaluation for the algorithms presented for recommendation and, in order to obtain the searched results, for each one of these new datasets (MovieLens, NewsSmall and NewsBig) has been created a training and a test set splitting their informations in the ratio 80% - 20% (train to test) for each user.

The metrics used to rate the results obtained for the all datasets has been precision-recall showing that,presented algorithms, although more scalable, do not incur a loss in quality, and on the contrary do better in terms of quality.

[1] MinHashing is a probabilistic clustering method that as- signs a pair of users to the same cluster with probability pro- portional to the overlap between the set of items that these users have voted for.

Observing the works described until now, it is possible to notice how the same problem has been faced by different researchers who, starting from a common point, have answered to different questions, studying different aspects applied to different contexts (in this case OSNs).

In this scenario there is the work carried out by Kywe and Hoang [10] whose purpose was to recommend hashtags in Twitter networks. Object of the study was a dataset containing information generated by a community of more than 150,000 Singapore users over a three-month period.

The method proposed selects hashtags from both similar tweets and similar users, recommending hashtags which are not only appropriate for the tweet written by a user but also match user's taste. In fact, given a user-tweet pair, their aim has been to find other similar user-tweet pairs, recommending hashtags from those user-tweet pairs.

The method proposed to select hashtags from similar users has been TF-IDF scheme where, TF(term frequency) refers to if a user uses a hashtag a lot,so the weight given to the hashtag is related to the preference's user. On the contrary IDF (inverse document frequency) assigns higher weight to a hashtag if is rarely used by other users.

Metric used to evaluate results was hit-rate and observing data obtained has been shown that, using user preferences and tweet content let to obtain better recommendation than just using tweet content alone.

Despite the purpose of this last described work [10] is similar to what has been attempted to achieve in this thesis, the project that is closest both conceptually and for metrics evaluation adopted was that of Diaz and Drumonds[7], even if the method used to predict hashtags is different.

In this work was developed the concept of "information elements" and illustrated the application of their approach to learning models factorization.

The method proposed is Stream Ranking Matrix Factorization (RMFX), an online learning algorithm based on a pairwise ranking approach for matrix factorization

that is intended for streaming data, demonstrating its usefulness on the task of recommending hashtags to Twitter users based on real world data (that is the problem faced in this thesis).

In order to obtain the list of Top-N hashtags recommendation searched for, in order to recommend to a user a few specific hashtags which are supposed to be the most attractive to him, the dataset in question was divided in two parts: training and test set. This division allows to predict future hashtags of a user,by selecting the N items with the highest score, only through the information contained in the training set, then using those contained in the test set as a term of comparison for evaluate the accuracy of the data obtained. The same approach was adopted in the thesis work described below, for which the same evaluation metrics have been adopted.

Recall metric has been used in Diaz and Drumonds work[7] to evaluate the results obtained, showing that the RMFX method achieves the best performance over all online other methods evaluated.

As already mentioned, the project carried out in this thesis was aimed to predict future hashtag published by Twitter's users, which is why it can be put more in connection with the last paper cited (Diaz[7]) than other works that have been mentioned before.

As in the project described in this article also in this work it was decided to use a k-core dataset, that means every user has used at least k different hashtags, and every hashtag has been used by at least k different users.

On the contrary, however, it was decided to work on an offline dataset and not an online one (as in the Diaz's work), this is why the tecnique choose to prediction was different. In fact, a limitation of the work described in this thesis, and a possible future development,consists in checking the effectiveness of the prediction methods used on a static dataset (constructed in a finite period of time) also on dataset obtained in real time. This will provide a real-time prediction of users involved in the study.

Chapter 2

Framework

In this chapter is presented and described the framework created fot this thesis work, analyzing in detail the major components that compose it.

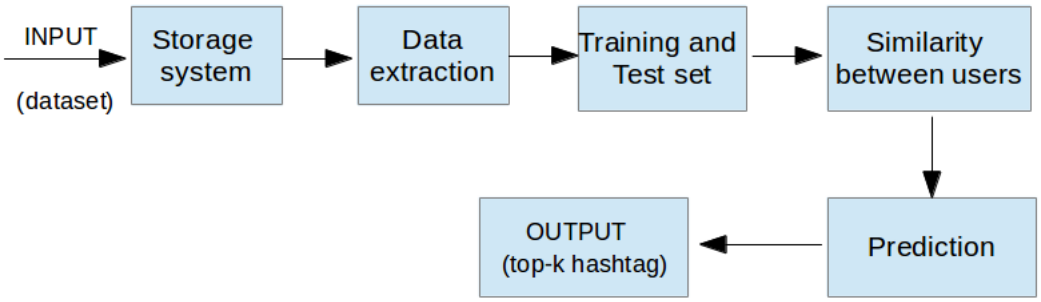


Figure 2.1. Framework's structure

As can be observed in the figure 2 how, given an input dataset relating to users information extracted from a social network, through a process of extraction and processing of data, is possible to obtain as output the top-k attributes predicted for each user, that for this project are represented by users' future hashtags.

2.1 Storage

First of all, given a dataset, is necessary to save the data that compose it in a well organized way, in order to be able to easily manage all the information that are available. The storage system to be used can be arbitrarily chosen depending on the amount of data that have to be managed and the resources that are available to process them.
For example, the dataset can be stored in a single file, or imported into a database. Specifically for this project the datasets analyzed were stored in two different ways, the first one was stored as a json file, while the second one has been imported into a document-oriented database (mongdb).

2.2 Data extraction

The following step, after data's storage, is rapresented by the extraction, from the entire dataset, of the only information really useful for the prediction. In fact, any dataset extracted from a social network does not directly contain only the data searched, but even all the information related to users and messages exchanged with their friends.
In particular, datasets studied extracted from Twitter, contained not only all the tweets written by users observed, but even relevant information about them, such as their name or nickname, and details related to tweets, like time and place when they were written.
For each user is extracted the pair of information (user ID, list of attributes), where the first value is the identifier that uniquely represents each user, while the second

one represent the list of hashtag extracted from all written tweet written by the considered user.
Subsequently, in order to apply the techniques of similarity chosen, it was necessary to perform a normalization of the data obtained relating to the list of attributes.
Specifically it was necessary to make all the same hashtag leading them to a state of lower case, and then delete all occurrences of the same word found in individual lists analyzed.

2.3 Training and Test set

After obtaining all the necessary data from the dataset studied, that is a list of hashtags for each users with no repetitions, the basic idea has been to randomized all the users' hashtags and then divide the entire dataset into two sections: the training set and the test set.
In this project it was decided to split the dataset so that 80% of the hashtag of each user belonged to the training set and the remaining 20% going to test set. The training set represents the part of the file used to determine the similarity between users, used to calculate predictions on data that belong to the test set.
In fact the prediction has been applied not to the whole dataset but just to a part of it. Subsequently, in order to verify the correctness of the hashtags predicted, these values were compared to the real hashtag contained in the test set.
In order to realize this two sets the entire dataset has been divided into 5 different folds where 4 folds merged togheter represent the training set, while 1 fold represents the test set. Because of this division in five folds, from the original dataset has been discarded all the users with less than five hashtags in their list.
In addition to obtain a prediction as impartial as possible were performed 5 different runs in which each time the test set was represented by a different fold:

- **Run 1** - Fold1|Fold2|Fold3|Fold4 as Training set
  Fold5 as Test set

- **Run 2** - Fold2|Fold3|Fold4|Fold5 as Training set
  Fold1 as Test set
- **Run 3** - Fold1|Fold3|Fold4|Fold5 as Training set
  Fold2 as Test set
- **Run 4** - Fold1|Fold2|Fold3|Fold5 as Training set
  Fold4 as Test set
- **Run 5** - Fold1|Fold2|Fold4|Fold5 as Training set
  Fold3 as Test set

2.4 Similarity

Being the purpose of this work to predict future hashtag that a user will post, according to what has previously written and what users like him have already written, the evaluation of similarity defined between users has been a crucial step. To determine this value have been implemented two different techniques in order to compare the results and choose the best one: the Jaccard similarity and the Cosine similarity.

2.4.1 Jaccard similarity

The Jaccard index, also known as the Jaccard similarity coefficient, is a statistic used for comparing the similarity and diversity of sample sets. The Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

For example, consider two users :

$$u_1 = \#A, \#B$$
$$u_2 = \#B, \#C$$

where #x represents the hashtag of the user, using the formula above the smilarity between the two users is equal to:

$$sim_{(u_1,u_2)} = \frac{1}{3}$$

Observing the Jaccard formula can be seen as the similarity between users is only linked to the number of items (in this case hashtag) they have in common. The value of this similarity may oscillate between 0 and 1, where a similarity equal to 1 represents the same two users, namely that posted the same hashtag, while 0 corresponds to two totally different users, that is with no hashtag in common.

2.4.2 Cosine similarity

On the contrary the Cosine similarity, by definition, is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them.

However, for this project, has not been considered the standard formula for calculating the cosine similarity, but on the contrary its weighted version in order to be considered for each user not only the hashtags in common, but also the number of times a particular hashtag had been posted by the same user.

In this way the weighted cosine similarity is defined as the sum of the product of the occurrences of the hashtag in common between the two users considered, divided by the product of the square root of the sum of the occurrences of all the hashtag of each user.

$$similarity = cos(\Theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

For example, consider two users :

$$u_1 = < \#A, f_1 >< \#B, f_2 >$$
$$u_2 = < \#B, f_3 >< \#C, f_4 >$$

where #x represents the hashtag of the user and f_y the number of times that the userN has written the hashtag x, using the formula above the smilarity between the two users is equal to:

$$sim_{(u_1,u_2)} = \frac{(f_1 \cdot 0)+(f_2 \cdot f_3)+(f_4 \cdot 0)}{\sqrt{(f_1)^2+(f_2)^2} \cdot \sqrt{(f_3)^2+(f_4)^2}}$$

Even for this method of similarity the value obtaibed may oscillate between 0 and 1, where 1 represents two users with same tastes, namely that posted the same hashtag, while 0 corresponds to two totally different users with no hashtag in common.

2.5 Prediction

After calculating the similarity between users, the last step is to calculate the prediction. This will allow to obtain future hashtags that the user considered will post.

The technique used for the calculation of prediction exploits the assumption of collaborative filtering that is, if two people agreed in the past will agree in the future, and they will like similar kinds of items as they liked in the past.

In order to do this it has been necessary to find the k most similar users by using the k-nearest neighbor (k-NN) approach, used in collaborative filtering systems.

This method allows to obtained for each user the list of the k-neighbors most similar to him, and is easily implementes by going to consider only the k users with higher similarity values obtained with the Jaccard and Cosine similarity method already explained.

The last step consist of consider all the hashtags belonging to these k neighbors, giving each of them the value of similarity of the user who wrote that hashtag. In this way if an hashtag was written by multiple users its final value will be the sum of the relative similarities to all users who have used it.

This value represents the probability that an hashtag has to be predicted to a user. in this way if only two hashtags has to be predicted, will be considered only the two with the highest value.

For this project, since during the normalization phase of the list of hashtags were eliminated all repetitions from each user's list, when the list containing all the hashtag

of the k most similar users is realized, before assigning the value of prediction to each hashtag will be discard all those that the user in question has already been published (taht are contained in the training test).

2.6 Parameters

In order to evaluate the values obtained by the prediction method, numerous tests were performed on the datasets presented considering various parameters.
Their values have been changed at each test to observe the impact of these changes on the results obtained. In this way it is attempted to obtained a combination of parameters' values to obtain a prediction's value as accurate as possible.
The main parameters considered to achieve this purpose are four:

- **N**: that represent the number of neighbors considered for each user;
- **k**: that represent the number of hashtag predicted for each user;
- **t**: that represent the threshold of both the minimum number of hashtag that a user has to be, and the minimum number of time that a hashtag has to be used in the dataset (that is the minimum number of users that have posted it);
- **d**: that represent the density of the matrix related to the dataset considered.

More precisely, the value of N was made mainly assume two values for both datasets: 10 neighbors and 100 neighbors. In addition, to the second dataset, were also added other parameters, namely = (5,10,50,100,200,250,300) to perform an additional test for the study of the behavior of the values obtained in the dataset to vary the density of neighbors considered.
The value of k instead assumed values $k \in$(1,10) with step 1. This value was varied to check the accuracy of the prediction made to vary the hashtag predicted, and study whether the prediction of a growing number of hashtag permettese of obtain better results or not.
Instead both t and d values are linked together as the variation of the threshold t

has allowed to obtain ever greater density always related to the matrix of the dataset considered.

For the two datasets t and d have assumed different values related to the amount of data and information contained in each dataset. More precisely for the first dataset t=(10,20) that corrisponds to a density d =(1%,12%) while for the second dataset t=(30,80,150) that corrisponds to a density d =(1%,6%,18%).

Chapter 3

Experimental evaluation

In this chapter are described in detail the datasets studied and the metrics used, discussing the tests performed and the results achieved.

3.1 Metrics

In order to evaluate the quality and the accuracy of predicted values, has been used three different performance metric:

- Precision;
- Recall;
- F-measure.

Usually, collaborative filtering algorithms are evaluated even trough another well-known method, that is the Mean Absolute Error (MAE). However, in this case, the purpose of the project has been to measure Top-N recommendation performance and not in rating prediction, this is why these methos has not been taken in consideration.

3.1.1 Precision

By definition precision is the fraction of retrieved instances that are relevant.

In this case the value of the precision is the fraction of the number of hashtags predicted for a user that matches to what he has posted for real in the test set divided by the number of hashtags retrieved.

$$precision = \frac{|relevant\,documents \cap retrieved\,documents|}{|retrieved\,documents|}$$

3.1.2 Recall

Instead, recall is the fraction of relevant instances that are retrieved.
In this case the value of recall is the fraction of the number of hashtags predicted for a user that matches to what he has posted in the test set divided by the real number of hashtags that are in the test set (that is the number of hashtag that should have been returned).

$$recall = \frac{|relevant\,documents \cap retrieved\,documents|}{|relevant\,documents|}$$

The following figure shows the relationship between these two measures:

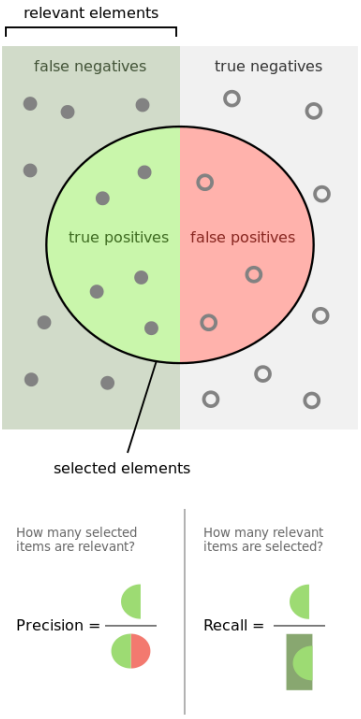


Figure 3.1. Precision and recall

The values of precision and recall can both oscillate between [0,1] and a perfect precision score of 1.0 means that every result retrieved by a search was relevant, whereas a perfect recall score of 1.0 means that all relevant documents were retrieved by the search. Usally, there is often an inverse relationship between precision and recall, in fact it is possible to increase one reducing the other.

Precision and recall are not considered just by themselves, but frequently these values are combined into a single measure. An example for measure that is a combination of precision and recall is the F-measure.

3.1.3 F-measure

By definition, the F1 score (also known as F-score or F-measure) is a measure of a test's accuracy. It can be considered as a weighted average of the precision and recall, whit a value between 0 and 1, at 1 is the best

value while at 0 is the worst one.

The general formula is:

$$F_\beta = \frac{1}{\alpha\frac{1}{precision}+(1-\alpha)\frac{1}{recall}} = (\beta^2+1)\cdot\frac{precision\cdot recall}{(\beta^2\cdot precision)+recall}$$

Instead, the formula used in this project is the traditional F-measure or balanced F-score, that is (β =2) :

$$F_1 = 2\cdot\frac{precision\cdot recall}{precision+recall}$$

3.2 Datasets

For this work has been analyzed two different datasets, both extracted from the Online Social Network(OSN) Twitter. [1]

3.2.1 First dataset

The first dataset is a collection of geotagged anonymous tweets taken from the Twitter Firehose for the 2-months period of November and December 2013, specifying as bounding box the cities of Milan and Trento, Italy and their suburbs. It consists of a total of x tweets by y users and was made available by the Telecom Big Data Challenge 2014 international competition from 14 January 2014.

The data for this dataset were initially collected in a json [2] file, where each line of this represented a tweet of a user along with numerous other information, such as his userId, the list of hashtags and symbols used in that tweet, informations about the place where the tweet was written and a list of other users related to that tweet. To be able to manage all these informations, extracting only those necessary for our purposes, has been decided to elaborate the whole dataset using the mongodb database.

[1] **Twitter**:http://twitter.com

[2] JSON (JavaScript Object Notation) by definition is a lightweight data-interchange format. It is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

Mongodb

MongoDB is a cross-platform document-oriented database. Classified as a NoSQL database, MongoDB eschews the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas (MongoDB calls the format BSON), making the integration of data in certain types of applications easier and faster.

The maximum BSON document size allow to be imported in mongodb is 16 megabytes. This limit helps ensure that a single document cannot use excessive amount of RAM or, during transmission, excessive amount of bandwidth. This is why the dataset considered has been splitted into 95 smaller json file before to be imported into mongodb.

MongoDB stores all documents in collections. A collection is a group of related documents that have a set of shared common indexes. Collections are analogous to a table in relational databases.

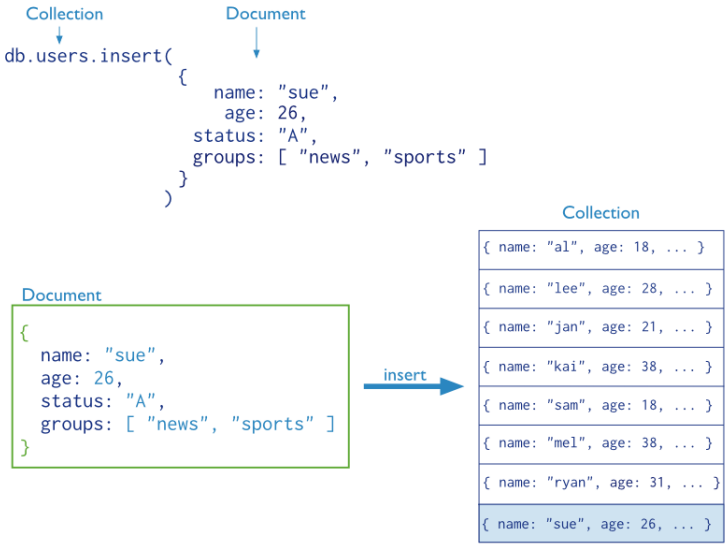


Figure 3.2. Example of mongodb's structure

The database imported in mongodb represents a collection and, each line of the json file, represents a single document. The extraction from each document in mongodb of the necessary informations related to each user (that is userId and his list of hashtag related to the analyzed tweet)

has been performed in two different steps.

- **First step** - first of all has been necessary to perform a query to the collection of documents in mongodb to extract from each one the list of hashtags of the analyzed user and his unique identifier (userId):
  - *documento.get("user").toString()*
  - *documento.get("hashtags").toString()*

  In this way have been obtained 15782 users and 57944 hashtags.
- **Second step** - afterwards the list of hashtags has been converted into an ArrayList<String>, while from the "user" informations has been extracted the userId searched.

Finally, to use the information obtained was necessary to make a further step on the list of hashtags obtained. In fact it was necessary to make all the same hashtag leading them to a state of lower case, and then delete all occurrences of the same word found in individual lists analyzed.

3.2.2 Second dataset

The second dataset, unlike the first, covers a longer period of about 5-months and it is related to the national elections .....
Even in this case the data were collected into a json file, but it was not possible to use the mongodb database as before because, being the file too large (10,7GB) the use of this db would not bring any benefit (which, however, has been possible with the first dataset), in addition the file would have to be divided into too many subsets from the original one to allow mongodb to import them.
As for the first dataset the extraction of main informations from the datasets has been divided into two steps:

- **First step** - first of all has been necessary to isolate from each line of the json file the only informations related to the userId and his list of hashtag;
- **Second step** - afterwards the information obtained was parsed in order to obtain an ArrayList<String> for the hashtags and the userId searched like in the first dataset obtaining 266887 users and 203675 hashtags.

Finally, as for the first dataset, even for these data was necessary to make a further elaboration on the list of hashtags obtained. In fact it was necessary to normalized this list turning all hashtags in a lowercase status and subsequently removing all the repetitions of the same word.

3.3 Test performed on datasets with higher density

Several tests has been run on the two datasets considered in this work, varying the parameters described in the previous chapter. In addition two different tecniques of similarity has been applied to them in order to assess the pros and cons for the prediction of hashtag users.
The dataset considered in this section is the second one described previosly.
For this one, different tests has been run, using four principal value of matrix's density and two different range of neighbors.
The value of d has been changed to verify how much the density of data present in the dataset considered could effect the reliability of hashtags predicted. For the same reason even the number of neighbors N has been changed, to show the possible connection between this parameter and the prediction's precision.
In addition, because has been used two different approach to obtained

the value of similarity between users, all the test has been run twice, to compare the advantages and disavantages to use a technique instead of another.

Tests based on Jaccard similarity

- First of all a low density [3] has been considered d=1%, studied in two different cases related to the number of neighbors considered, that is *N=10* and *N=100*:

CASE 1 - N=10 neighbors and $k \in$[0,10]with step 1

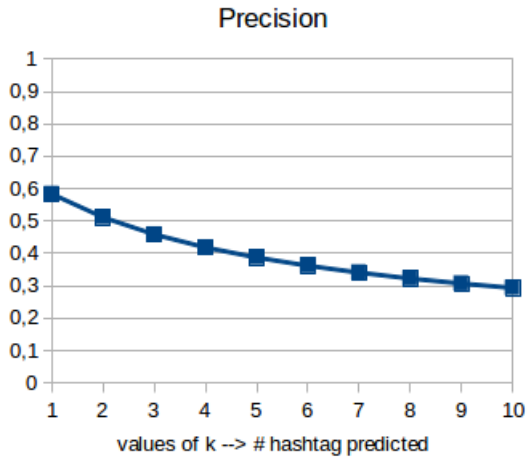
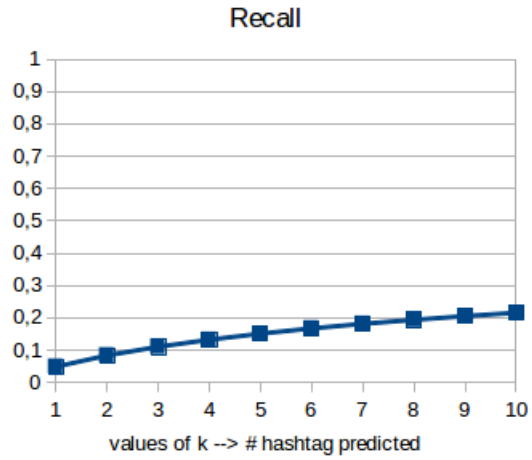


Figure 3.3. Value of Precision for N=10 neighbors and d=1%



[3] It is not necessary to consider a density value <1% because in that case the matrix will be too sparse, making prediction value too low

Figure 3.4. Value of Recall for N=10 neighbors and d=1%

CASE 2 - N=100 neighbors and $k \in$[0,10]with step 1

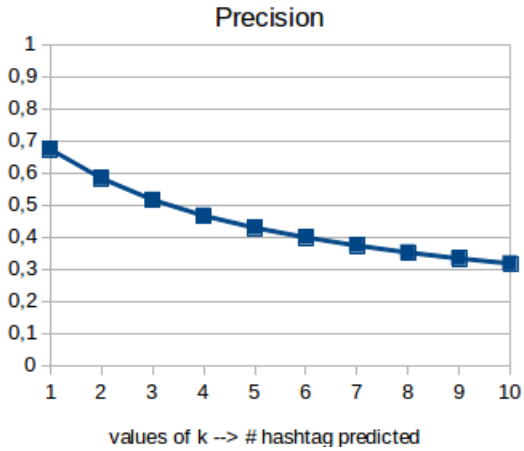


Figure 3.5. Value of Precision for N=100 neighbors and d=1%

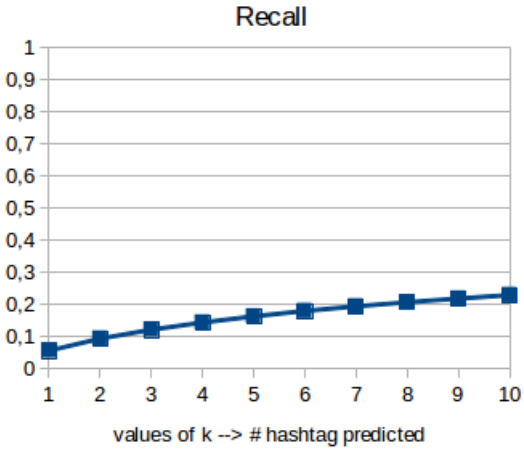


Figure 3.6. Value of Recall for N=100 neighbors and d=1%

- Following a little larger matrix's density has been considered, that is d=6%, studied in the same cases:

CASE 1 - N=10 neighbors and $k \in$[0,10]with step 1

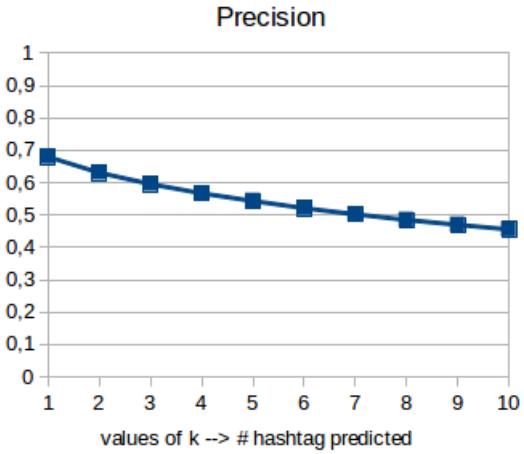


Figure 3.7. Value of Precision for N=10 neighbors and d=6%

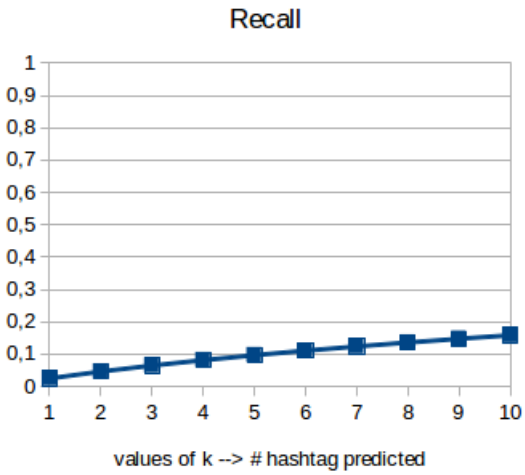


Figure 3.8. Value of Recall for N=10 neighbors and d=6%

CASE 2 - N=100 neighbors and $k \in$[0,10]with step 1

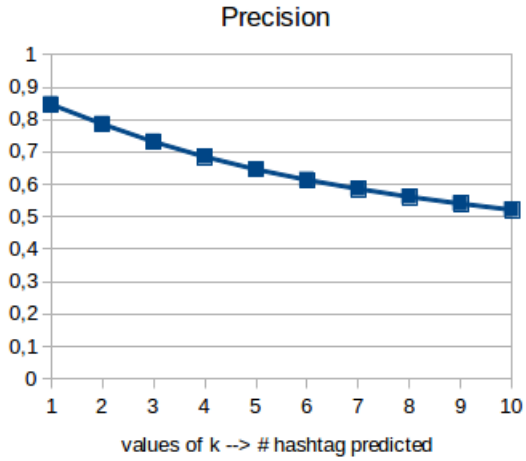


Figure 3.9. Value of Precision for N=100 neighbors and d=6%

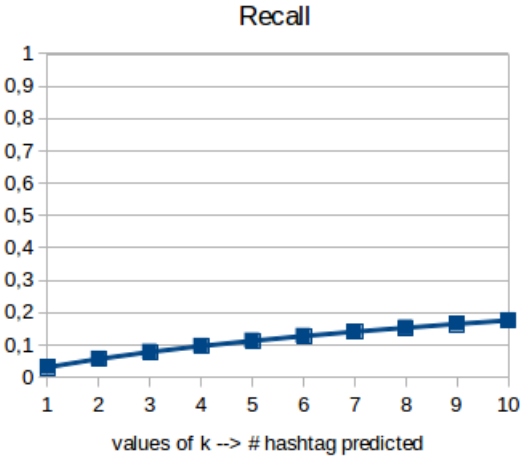


Figure 3.10. Value of Recall for N=100 neighbors and d=6%

- Finally a matrix with a higher density has been considered to verify how match this parameter effect the results (d=18%):

CASE 1 - N=10 neighbors and $k \in$[0,10]with step 1

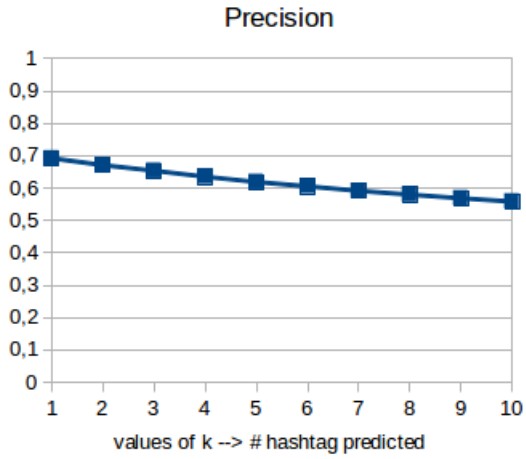


Figure 3.11. Value of Precision for N=10 neighbors and d=18%

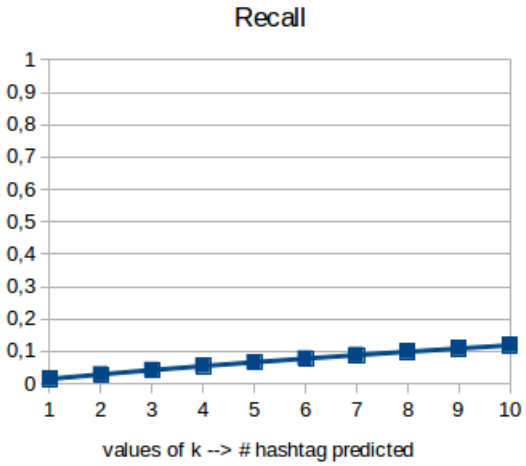


Figure 3.12. Value of Recall for N=10 neighbors and d=18%

CASE 2 - N=100 neighbors and $k \in$[0,10]with step 1

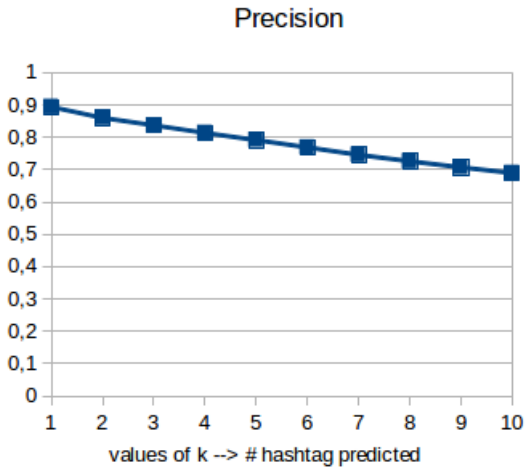


Figure 3.13. Value of Precision for N=100 neighbors and d=18%

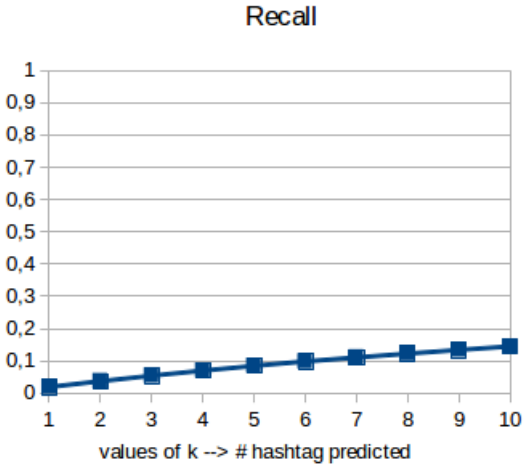


Figure 3.14. Value of Recall for N=100 neighbors and d=18%

Observing the value of Precision and Recall obtained in the different scenarios presented above, it is possible to notice that, the changing of parameter N, that represents the number of neighbors considered for the evaluation of similarity between users, has an increasing effect on the results with increasing density of matrix.

In fact, as soon as the number of neighbors considered increases, even the value of precision increases too(and parallel even the recall value). This changing happens because, if a higher number of neighbors is considered, the method of prediction has several additional informations to infer, in

a more accurate way, future hashtags for a user.

In addition it can be observed that even the value of the matrix's density affect the results, in fact higher is the density, higher is the value of precision obtained.

This can be explained by the fact that higher is the density of a matrix, higher will be the probability that users' data will have an intersection (hashtags shared among users), allowing to make a "good" prediction.

Furthermore it can be observed that, for lower density matrix's values 3.3, large variations in precision between a value k and another, correspond to smaller variations in recall between the same values of k. This difference tends to stabilize and no longer be so marked with increasing density value considered.

This is why, to point out these results, additional tests have been performed, evaluating the behaviour of values obtained related to the variation of the parameters d and N.

- First of all has been run a case in which the only value to change was the density matrix, while the others parameters considered was left fixed, in fact :
  - the number of hashtag predicted was k=10;
  - different values of density d was considered, increasing the three values studied before;
  - the number of neighbors was N = 10 and N = 100.

Variation of Precision and Recall related to density's value, k=10

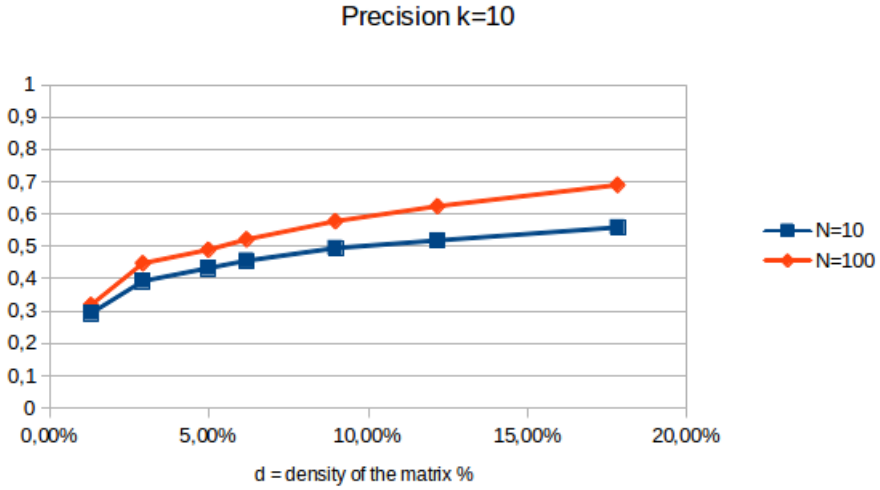


Figure 3.15. Value of Precision

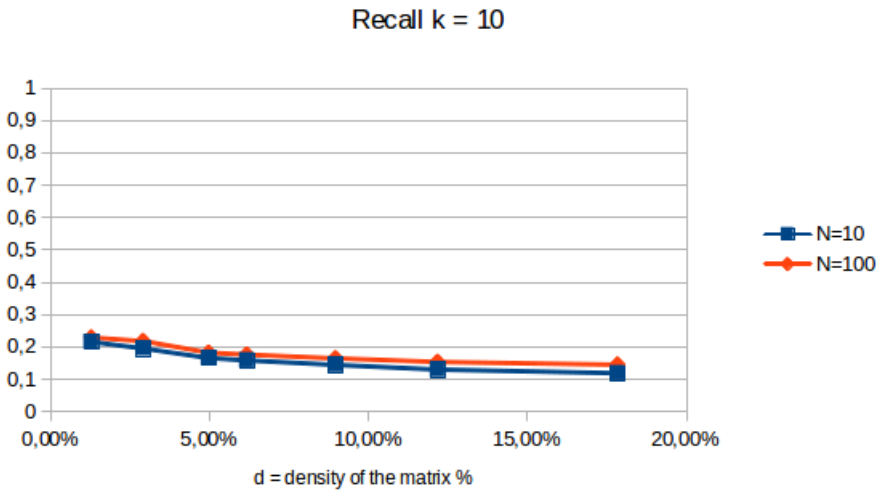


Figure 3.16. Value of Recall

From these graphics it is possibile to see how the value of density affect the results. Furthermore it can be observed how, for small values of density the increasing of precision's value is accentuated, while from bigger values on his behavior becomes asymptotic.

- Afterwards has been run a second case in which the value of density was left unaltered this time, while the value of neighbors was subjected to changes:
  - the number of hashtag predicted was k=1;
  - the value of density considered was d=18% ;
  - the number of neighbors was incresing, N=(5, 10, 50, 100, 200, 250, 300).

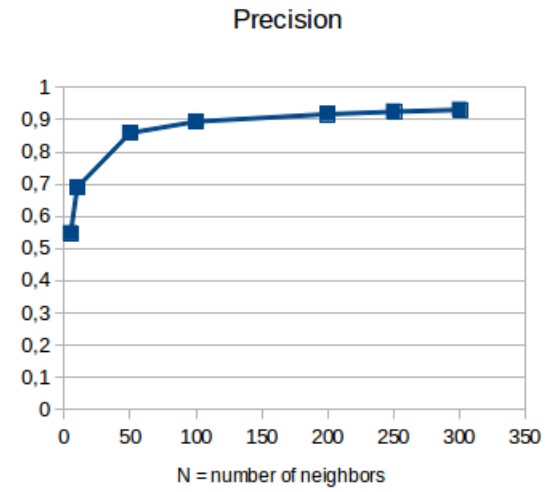


Figure 3.17. Value of Precision

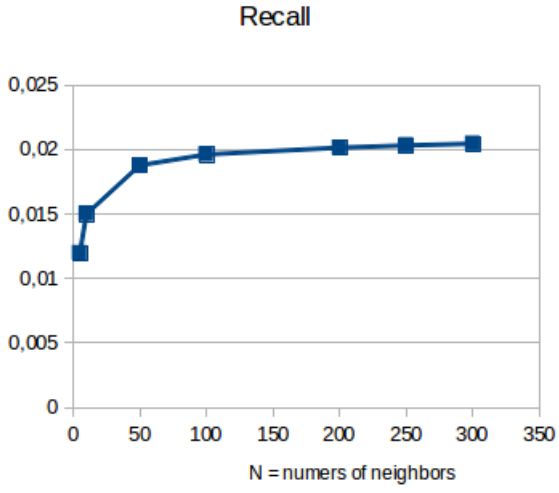


Figure 3.18. Value of Recall

From these graphics, 3.3 and 3.17, it is possibile to see how the number of neighbors affect the results. And it is possible to observe

how, for small numbers of neighbors the value of precision progressively increase, while from bigger values on his behavior becomes asymptotic as in the previous case.

Tests based on Cosine similarity

As explained in the previous section, the *Jaccard similarity* has not been the only method used to obtained the similarity between users.
In fact another tecnique have been introduced, that is the *Cosine similarity*, and to verify the accuracy of this one has been run the same tests implemented for the Jaccard similarity.

- First of all a low density has been considered d=1%, studied in two different cases considering initially N=10 neighbors with $k \in$[0,10]with step 1, and then N=100 neighbors with $k \in$[0,10]with step 1;
- Following a little larger matrix's density has been considered (d=6%) always studied in two cases with N=10 and N=100 neighbors with $k \in$[0,10]with step 1;
- Finally a matrix with a higher density has been considered to verify how much this parameter effect the results (d=18%) considering N=10 ad N = 100 neighbors before then.

To show the results obtained for Precision and Recall using the two different tecniques for the similarity have been realized specific graphics. The parameters used for these test have been:

- the value of density considered d=18%;
- the number of neighbors N = 100.

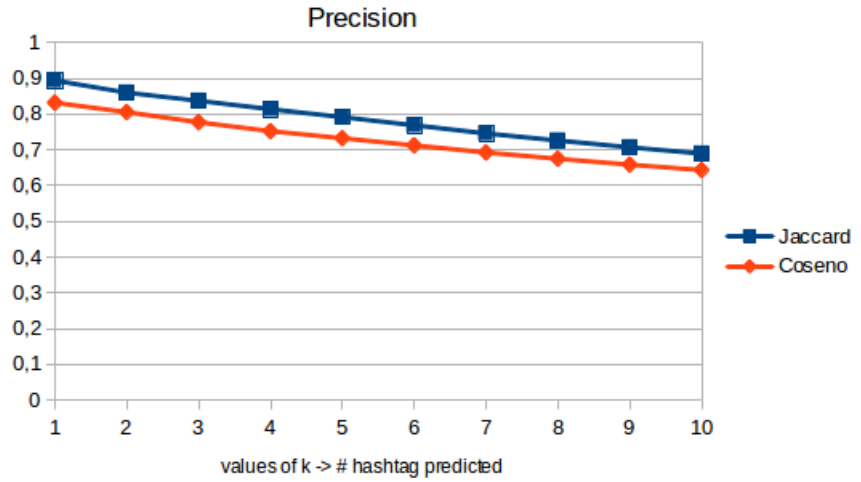


Figure 3.19. Value of Precision for N=100 neighbors and d=18%

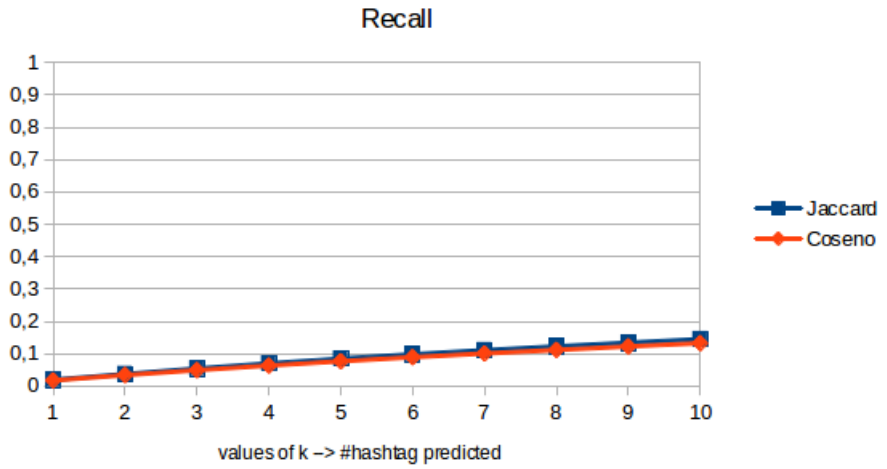


Figure 3.20. Value of Recall for N=100 neighbors and d=18%

Observing this test,3.3 and 3.19, is possibile to notice that, for the same data in the same conditions, apply the Jaccard method allows to obtained a higher value for both precision and recall.

This can be observed even for the other scenarios where different values of neighbors and density are used, this is why it were not reported the graphs.

As introduced previously, in the "metrics" section, not only the methods of precision and recall were used to assess the accuracy of the results obtained but was also used another method, the F-measure.

This was applied to all three of the density d considered (1%, 6%, 18%) evaluating it for both N = 10 and N = 100 neighbors, considering the precision and recall values obtained applying the Jaccard similarity, obtaining the following results:

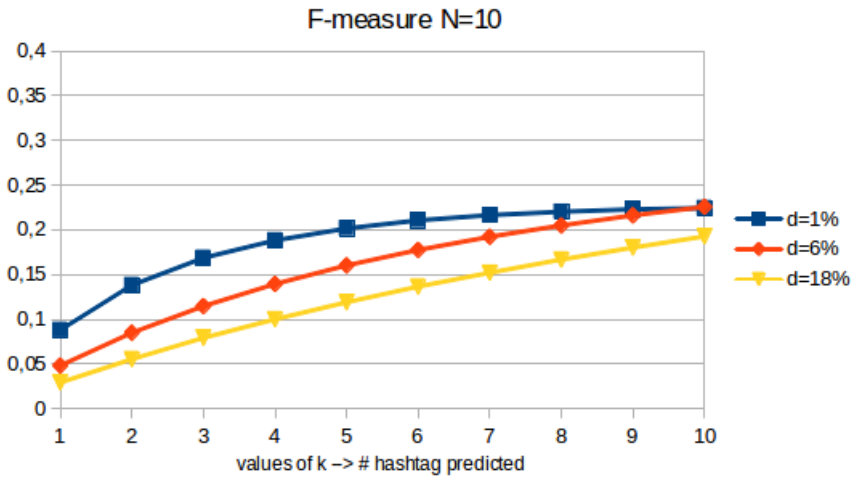


Figure 3.21. Value of F-measure for N=10

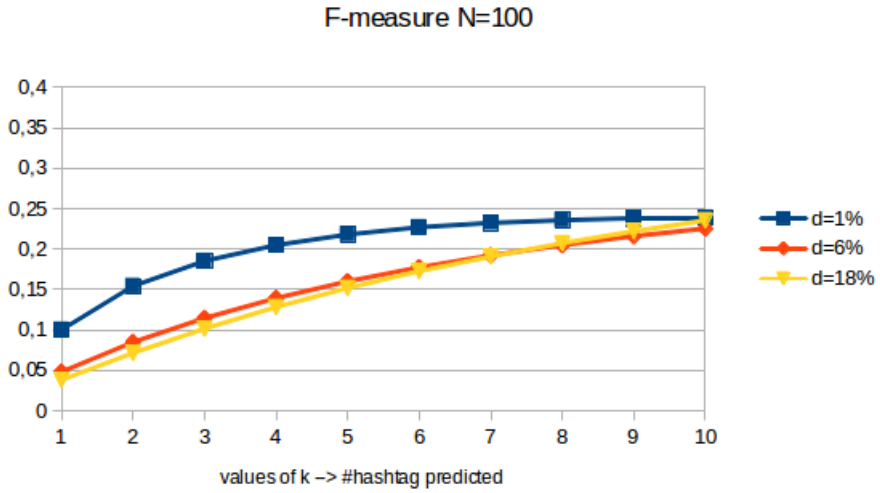


Figure 3.22. Value of F-measure for N=100

Observing these graphs, 3.3 and 3.21, it is possible to notice the value of the harmonic mean between precision and recall for the different tests run. It is also possible to notice how, to vary the density of the matrix considered, the values of the obtained Fmeasure increase, going to stabilize for values of k high, and in some cases to converge.

The same metric has been applied to the precision and recall values arising from the use of Cosine similarity, showing in this case just N=100, because the same result is obtained varying this parameter:

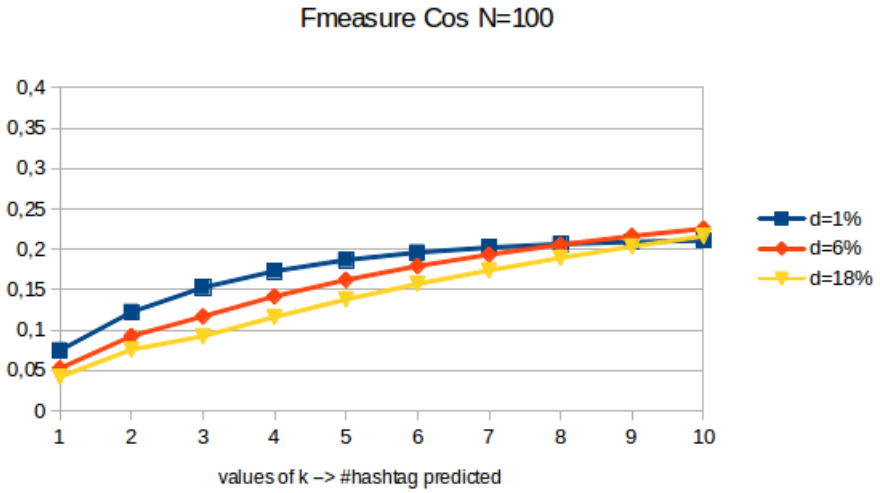


Figure 3.23. Value of F-measure for N=100

The above graph 3.3 highlight that, as it was for precision and recall metrics, the Jaccard similarity allow to obtained values better than the Cosine similarity.

3.4 Test performed on datasets with lower density

The second dataset considered is quite different from the first one, especially for the density of data it has (in fact it is minor).

As before, even in this case has been performed the same procedure in which the parameters of matrix's density number of neighbors considered and similarities tecniques have been changed to observe how these values would have affected the results obtained.

In this case has been used two different value of matrix's density and two different values for the number of neighbors considered.

Tests based on Jaccard Similarity

- First of all a low density has been considered d=1%, studied in two different cases:

CASE 1 - N=10 neighbors and $k \in$[0,10]with step 1

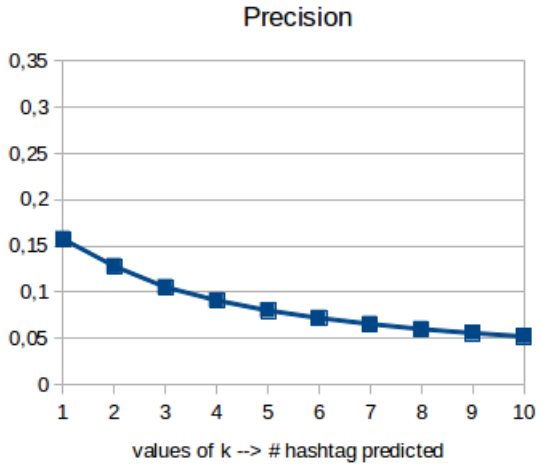


Figure 3.24. Value of Precision for N=10 neighbors and d=1%

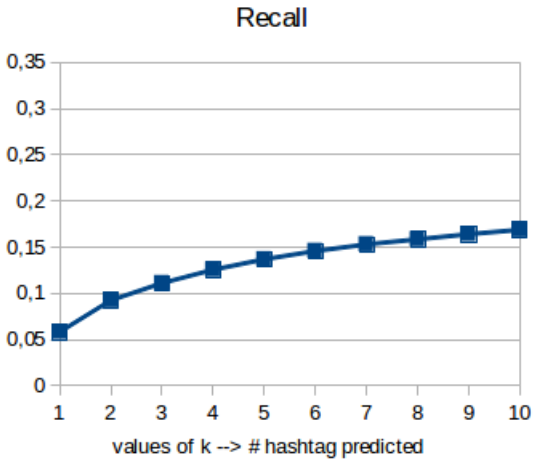


Figure 3.25. Value of Recall for N=10 neighbors and d=1%

CASE 2 - N=100 neighbors and $k \in$[0,10]with step 1

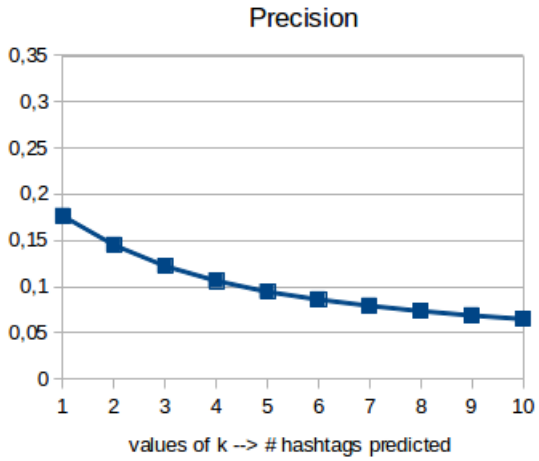


Figure 3.26. Value of Precision for N=100 neighbors and d=1%

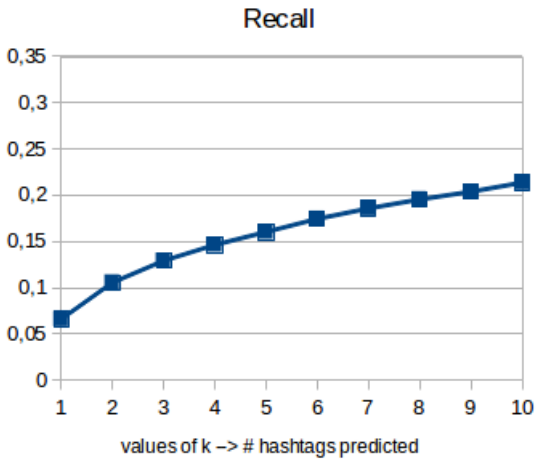


Figure 3.27. Value of Recall for N=100 neighbors and d=1%

- Then a matrix with a higher density has been considered to verify how match this parameter effect the results (d=12%):

CASE 1 - N=10 neighbors and $k \in$[0,10]with step 1

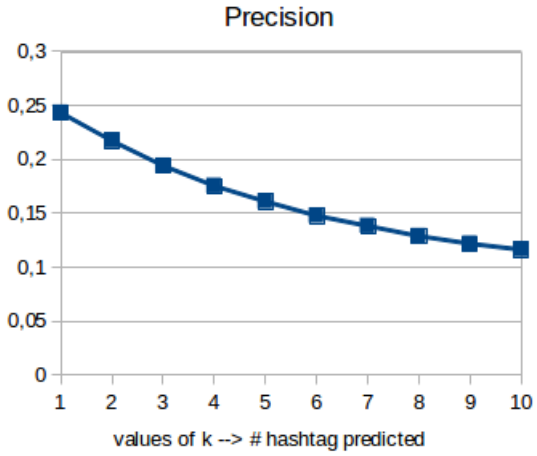


Figure 3.28. Value of Precision for N=10 neighbors and d=18%

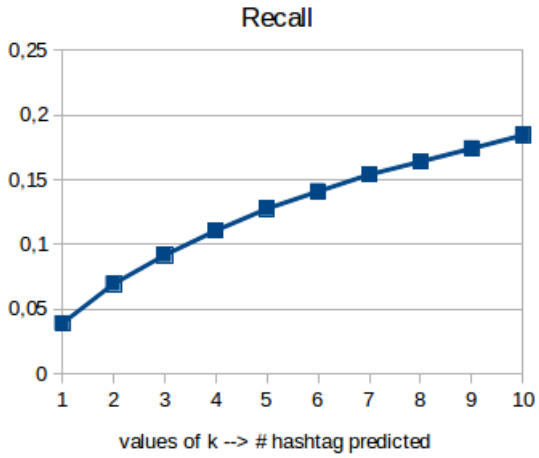


Figure 3.29. Value of Recall for N=10 neighbors and d=18%

CASE 2 - N=100 neighbors and $k \in$[0,10]with step 1

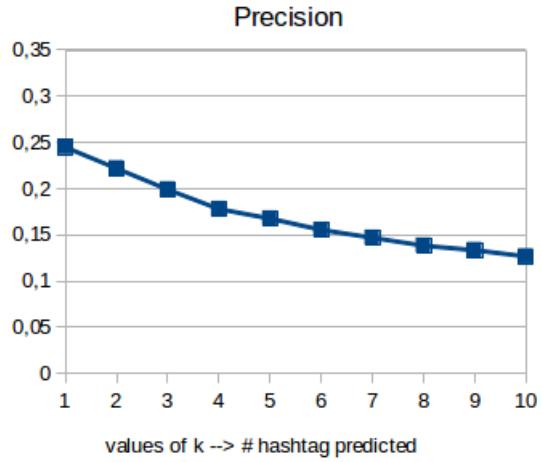


Figure 3.30. Value of Precision for N=100 neighbors and d=18%

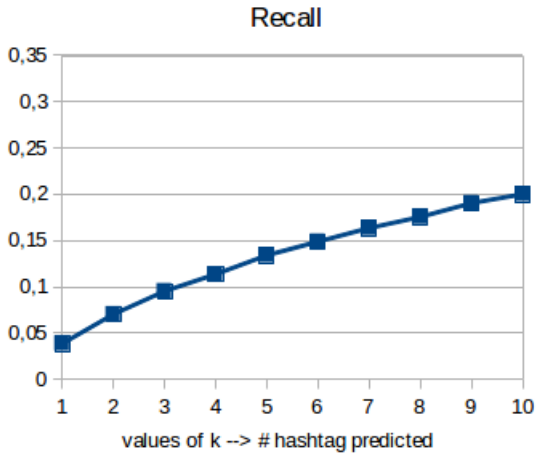


Figure 3.31. Value of Recall for N=100 neighbors and d=18%

Observing the value of Precision and Recall obtained in different scenarios for the Jaccard similarity, it is possible to observe that even for this dataset, the changing of parameter N, that represents the number of neighbors considered for the evaluation of similarity between users, has an increasing effect on the results with increasing density of matrix.

Tests based on Cosine similarity

As explained for the first datset studied, the *Jaccard similarity* has not been the only method used to obtained the value of similarity to make

predictions.

In fact another tecnique have been introduced, that is the *Cosine similarity*, and to verify the accuracy of this one has been run the same tests implemented for the Jaccard similarity.

- First of all a low density has been considered d=1%, studied in two different cases:

CASE 1 - N=10 neighbors and $k \in$[0,10]with step 1

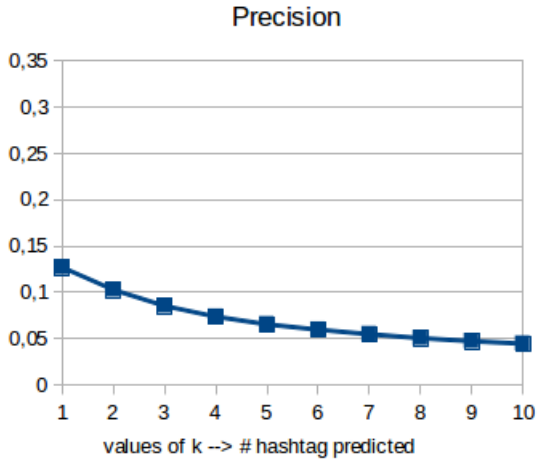


Figure 3.32. Value of Precision for N=10 neighbors and d=1%

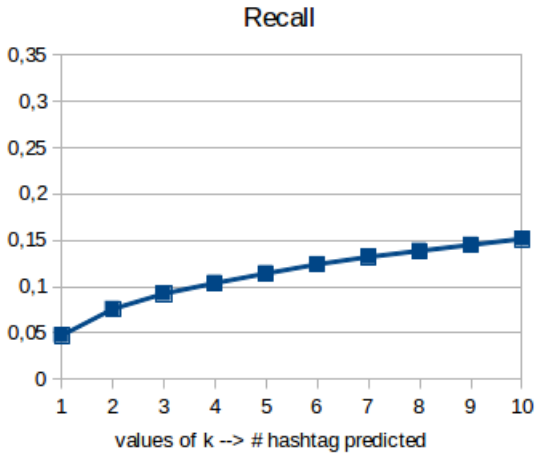


Figure 3.33. Value of Recall for N=10 neighbors and d=1%

CASE 2 - N=100 neighbors and $k \in$[0,10]with step 1

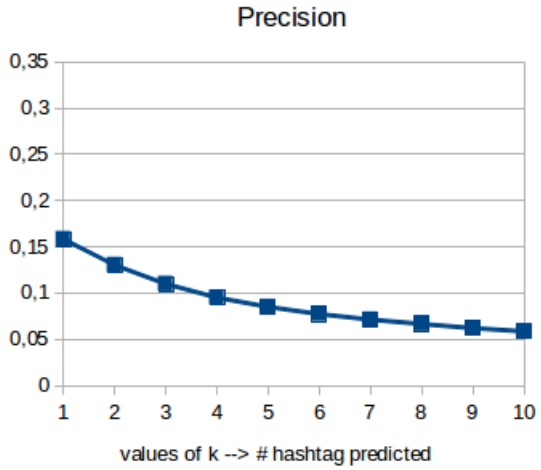


Figure 3.34. Value of Precision for N=100 neighbors and d=1%

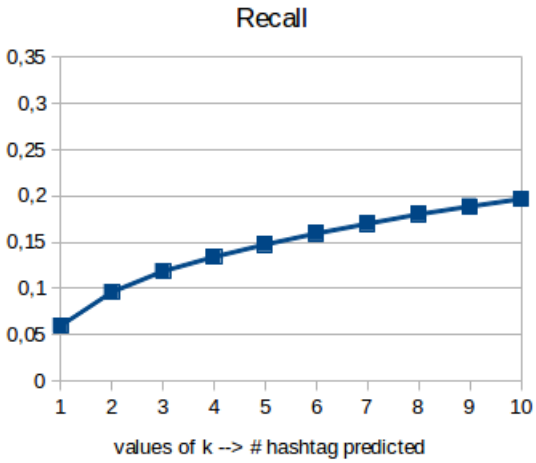


Figure 3.35. Value of Recall for N=100 neighbors and d=1%

- Then a matrix with a higher density has been considered to verify how match this parameter effect the results (d=12%):

CASE 1 - N=10 neighbors and $k \in$[0,10]with step 1

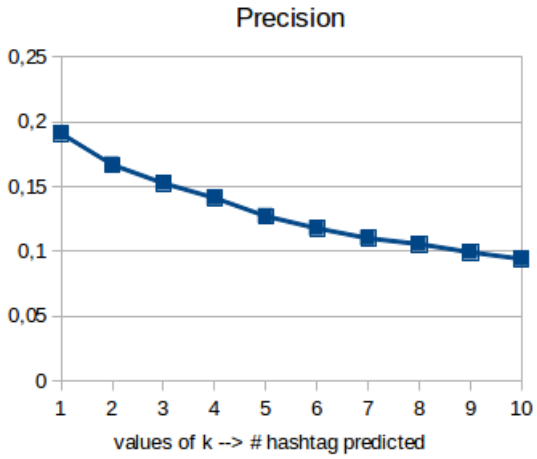


Figure 3.36. Value of Precision for N=10 neighbors and d=18%

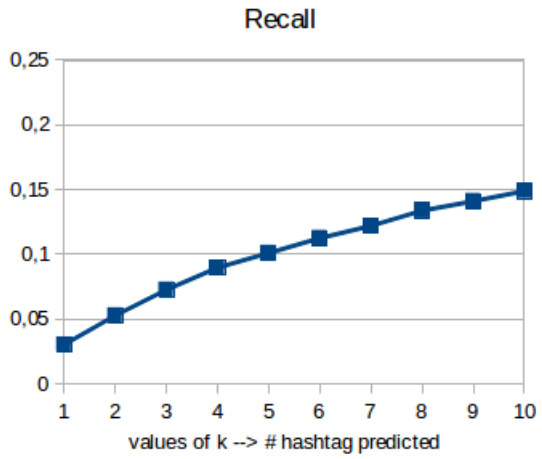


Figure 3.37. Value of Recall for N=10 neighbors and d=18%

CASE 2 - N=100 neighbors and $k \in$[0,10]with step 1

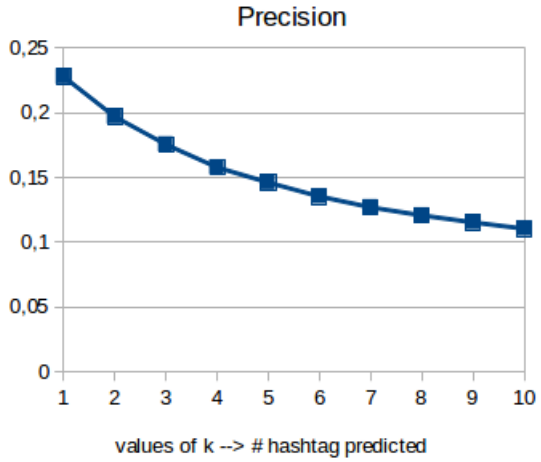


Figure 3.38. Value of Precision for N=100 neighbors and d=18%

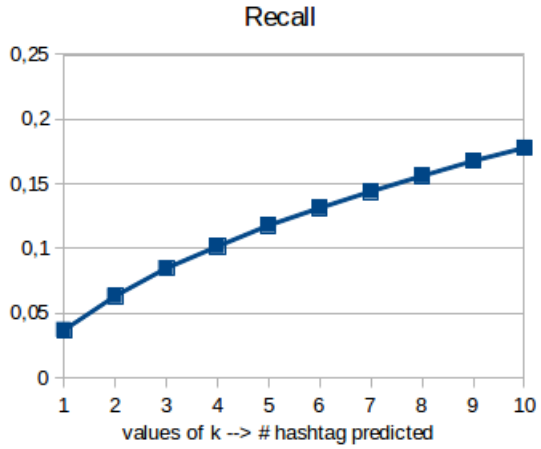


Figure 3.39. Value of Recall for N=100 neighbors and d=18%

Even in this case, using a different method of similarity, changing the same parameters d and N has brought to the same deductions, that is the values of precision and recall obtained are closely related to the density of the data considered and to the informations that can be deducted from user's neighbours.

To show the results obtained for Precision and Recall using the two different tecniques for the similarity have been realized specific graphics. The parameters used for these test have been:

- the value of density considered d=12%;
- the number of neighbors N = 100.

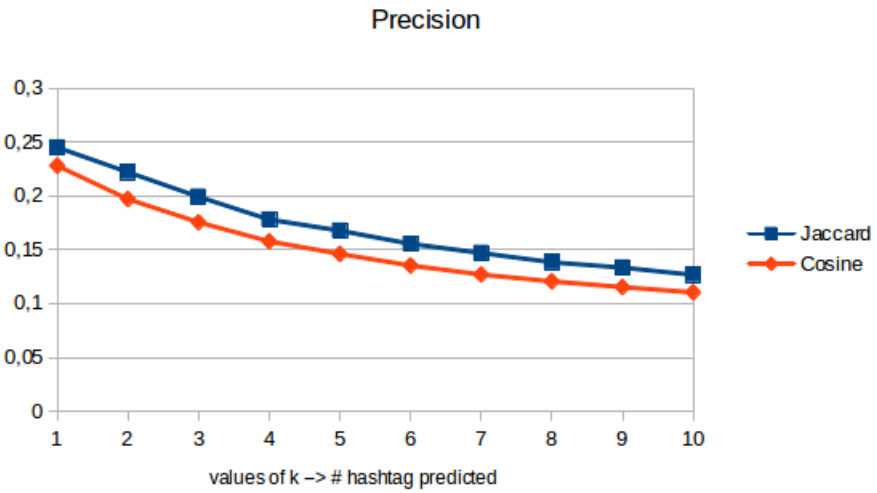


Figure 3.40. Value of Precision for N=100 neighbors and d=12%

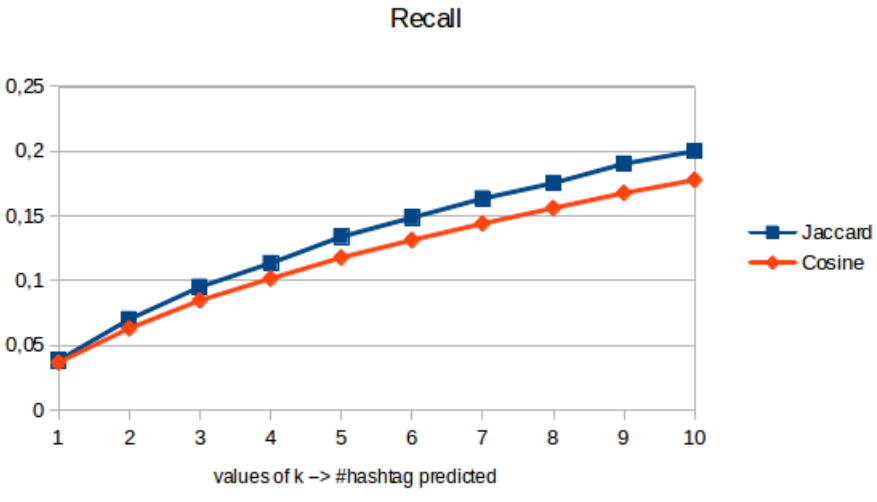


Figure 3.41. Value of Recall for N=100 neighbors and d=12%

Observing this test is possibile to notice that, for the same data in the same conditions, apply the Jaccard method allows to obtained a

higher value for both precision and recall.
This is still true even in this dataset with a density of data minor than the previous one.

As a final test has been compared the results obtained in the two datasets studied considering the same parameters for both. Below it are the graphs that compare the values of precision and recall obtained considering:

- as the number of neighbors N = 100
- as denstità array d = 1%
- as values of $k \in$[0,10]with step 1
- as methods for the evaluation of similarity both the cosine and the Jaccard similarities

Comparison using Jaccard similarity

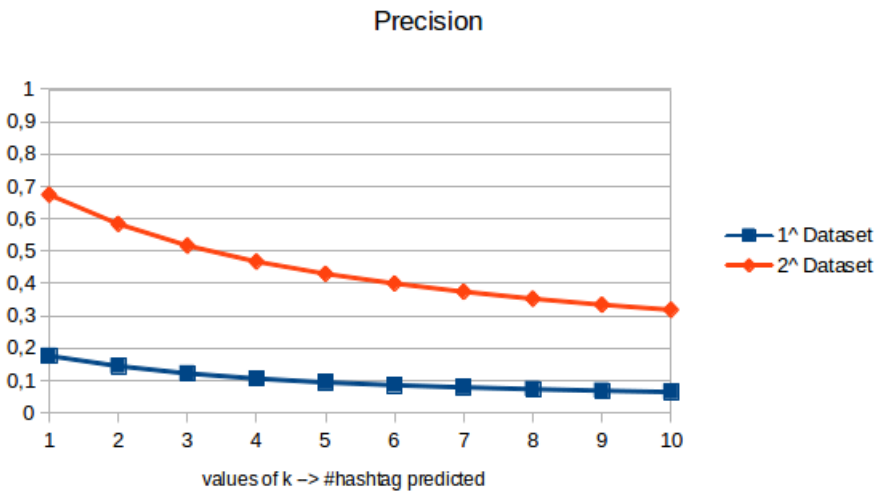


Figure 3.42. Value of Precision for N=100 neighbors and d=1% for both datasets

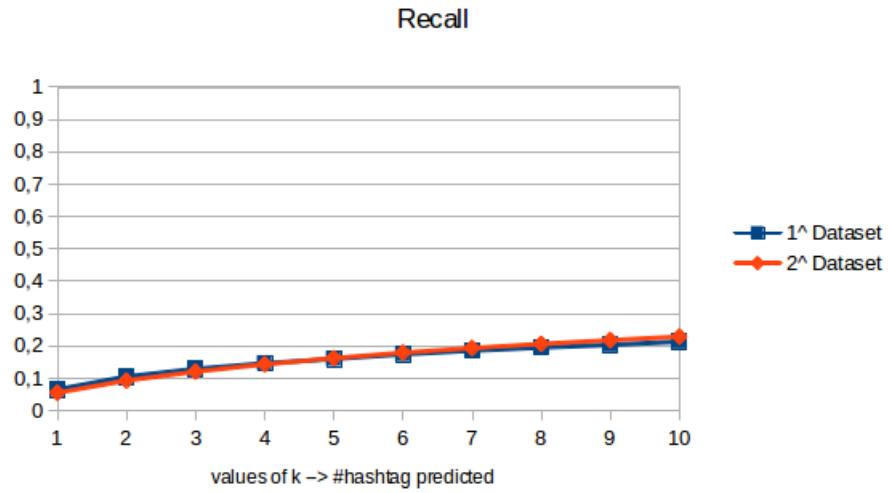


Figure 3.43. Value of Recall for N=100 neighbors and d=1% for both datasets

Observing this graph it can be seen that, the same technique of similarity applied to different datasets allows to obtain results that as k changes has the same behaviour, but quantitatively it has different values. In fact it is clear as to the second dataset have values of precision higher than in the first one. This behavior shows that, not only the technique used, but even the dataset considered influence the results obtained and because of the second dataset was richer of information (users and related hashtag) it was possible to get good results.

Comparison using Cosine similarity

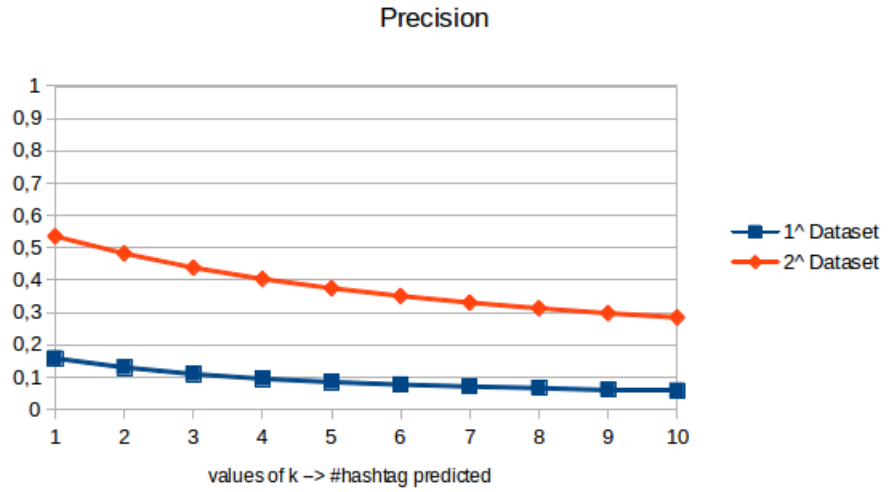


Figure 3.44. Value of Precision for N=100 neighbors and d=1% for both datasets

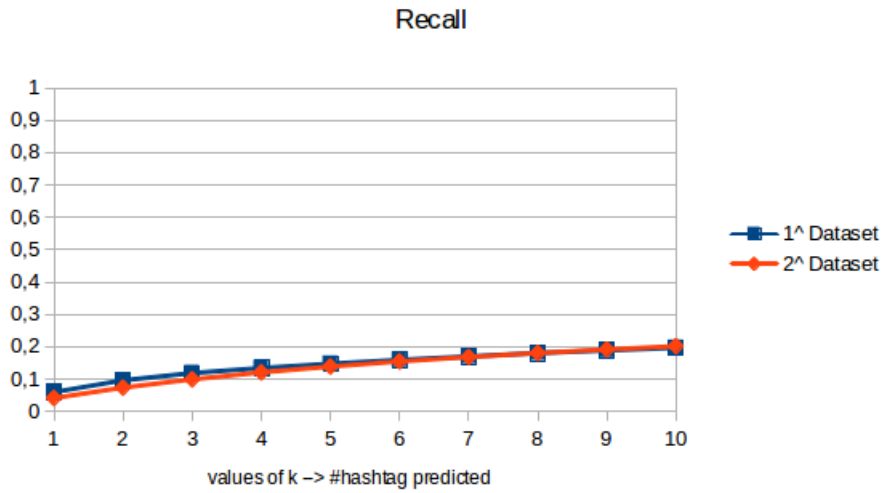


Figure 3.45. Value of Recall for N=100 neighbors and d=1% for both datasets

From this last graph, it is possible to note what already pointed out about the differences of density information between the first and the second dataset. Furthermore, as it was noticed for tests carried out previously, the method of the Jaccard similarity allows to obtain an accuracy of predicted hashtag higher than the Cosine similarity.

Conclusions

The aim of the thesis work described was to predict future hashtag of a user based not only to those who have already posted in the past, but also to those that have been written by his neighbors (users that are similiar to him with whom shares hashtags).

Different prediction techniques have been applied to the datasets studied leading to different results. In fact, by observing the tests carried out it has been possible to notice how the technique of Jaccard similarity has allowed to obtain more precise predictions compared to the technique of Cosine similarity.

Moreover, among the various parameters tested, the factor relating to the information density of the dataset d is found to be the most influential parameter, in fact increasing values of it corresponds to an increase of precision and a decrease of recall values.

In addition, values of precision and recall obtained have also shown that, for low values of k (the number of predicted hashtag) the prediction made has a higher percentage of success. This is because, the more you increase the number of hashtags to predict, the more chance of error by recommend a wrong hashtag increases, affecting the accuracy of the totality of the prediction performed.

It can be stated that the results achieved in this thesis have satisfied the question that was posed at the beginning of this project that is: "It is possible to predict the future hashtags of a user belonging to Twitter?".

In fact, the techniques used have shown how it is possible to predict with some accuracy the hashtag of a user, highlighting the existence of possible social impacts. As explained in fact this practice little lead to "reveal" hidden interests of a user by bringing to the attention of all information that he wanted to keep hidden.

Future work

As future developments of the project presented could be interesting to study the behavior of the prediction techniques applied to datasets offline in a real-time, enabling you to provide predictions to users in real time. This would allow use of this study in the field such as advertising to take advantage of users' interests considered.

Bibliography

[1] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to weave an information tapestry,” Com- munications of the ACM, vol. 35, no. 12, pp. 61–70, 1992.

[2] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “Grouplens: an open architecture for collaborative filtering of netnews,” in Proceedings of the 1994 ACM conference on Computer supported cooperative work. ACM, 1994, pp. 175–186.

[3] R. Baeza-Yates, B. Ribeiro-Neto et al., Modern information re- trieval. ACM press New York, 1999, vol. 463.

[4] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, “An algorithmic framework for performing collaborative filtering,” in Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1999, pp. 230–237.

[5] G. Linden, B. Smith, and J. York, “Amazon. com recommenda- tions: Item-to-item collaborative filtering,” Internet Computing, IEEE, vol. 7, no. 1, pp. 76–80, 2003.

[6] F. Petroni, L. Querzoni, R. Beraldi, and M. Paolucci, "LCBM: a fast and lightweight collaborative filtering algorithm for binary ratings"

[7] E. Diaz-Aviles, L. Drumond, L. Schmidt-Thieme, and W. Nejdl, "Real-Time Top-N Recommendation in Social Streams" University of Hannover, Germany, University of Hildesheim, Germany

[8] E. Zheleva, and L. Getoor, "To Join or Not to Join: The Illusion of Privacy in Social Networks with Mixed Public and Private User Profiles"

[9] A. Das, M. Datar, A. Garg,and S. Rajaram, "Google News Personalization: Scalable Online Collaborative Filtering"

[10] S. Kywe, T. Hoang, E. Lim, and F. Zhu, "On Recommending Hashtags in Twitter Networks" Singapore Management University, Singapore