Abstract

Artificial intelligence is increasingly playing a crucial role in helping users find products that are well suited to their interests, tastes and wishes. Nowadays users deal with a huge amount of data when they want to purchase or borrow products. The aim of recommendation system is to help the user in his decision making process, driving his focus on product that fits his needs.
Recommendation systems are models capable of suggesting personalized items to a generic user by means of filtering techniques. This tools have acquired great relevance and have been widely used in recent years as they represent a significant improvement, especially in relation to the Big Data issue. We can find their application in disparate areas, they play a fundamental role in various services: from the entertainment such as movies, videos, music, to the e-learning web sites such as newspapers. Furthermore, they are vital to improve the online e-commerce on different products. One of the main aspects of a good recommendation system is the reliability of the recommendation results, this increases the trust level of the user in the system. A plus value of recommendation systems is to surprise the user by suggesting items that have a not so evident correlation, to open his range of choice. These systems mimic the human mechanism of recommending something by suggesting similar or potentially equally interesting products.

The thesis work consists in the application of literature recommendation techniques orienting them to the world of books. Specifically, the application scenario is to recommend interesting or popular information suggested by the community of social network aNobii. The word of mouth on social networks among readers with similar tastes is crucial to extrapolate possible suggestions for the online book trade.
In addition, an analysis on library loan data was carried out thanks to

the collaboration with the libraries of the city of Turin. It was possible to analyze and merge this data-set with the aNobii social network data. The application of the recommendation techniques on this merge result data-set allowed to study the users behavior and therefore to suggest them a good book to read suitable to their preferences.
Aggregating big data is not easy challenge, due to the huge volume of data is a feat that must be overcome efficiently through essential steps. A fundamental part of the work concerns data preprocessing and cleaning. The investigative analysis on the data is of paramount importance as it permits to transform them into a more understandable, useful and efficient format for their use. This important task allows to have significantly better performance in the subsequent steps.
During the work, collaborative filtering approaches of recommendation models were applied. The results obtained showed a significant improvement between the models implemented respect the baseline model. In particular the betterment between the best performing model and the baseline model was approximately 38%.
In parallel, it was possible to perform a distinct further analysis to evaluate the similarity between different items by exploiting the content of the books metadata and using similarity metrics.
Finally, it was interesting to classify and identify the opinions of users on the books reviewed with the help of sentiment analysis techniques.
Social networks contain meaningful information, but at the same time it is difficult to extrapolate them. The sentiment analysis aims to capture the opinions relating to a single item and exploits them to improve even better the suggesting strategy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Nowadays the Internet offers a vast wealth of resources. This enormous amount of information, however, hides a crucial problem: difficulty in finding what are the potential user interests.
Recommendation tools play a fundamental role in various domains of daily-life applications like sale of products, watching videos, movies, newspapers, books. In the online e-commerce environment this tools are effective means of trading more products, think of Amazon. In the movie world such as in the case of Netflix, they take care to recommend enjoyable movies to watch. In libraries, recommendations systems support users by enabling them to go beyond catalog searches [2].
They gives a plus value to the quality of services where they are implemented offering the possibility to support the user in his decision-making.
Recommender systems are information filtering techniques that address the problem of information overload and aim to extrapolate users interests and preferences based on the observed behaviors [1].
The key feature of a recommender system is that it provides a personalized view of the data to the end user [3].
The research on recommendation systems consists of a broad variety of artificial intelligence techniques including machine learning, data mining, among others [3].

In this thesis work I analyzed two data-sets that belong to the books world, also driven by the candidacy of the city of Turin as the future capital of the book. I applied some literature techniques of recommendation system in order to provide a greater dimension to the reader. The paramount goal is allowing him to have the pleasure of choosing the next

book to read with less effort.
The data-sets in question are distinguished by type of service they offer to the user. The first data-set belongs to the world of social networks, in detail the aNobii social network. This is an online community of reading enthusiasts who love to share opinions, evaluations, reading experiences that involve them. Anobii is a web platform where the “anobians”, this is the name that identifies the members, can make reviews on the books they read, discover new ones and also interact with each other by comparing them on the books they have enjoyed or not reading. The second data-set analyzed concerns the loans of books that take place in Turin libraries. The information contained involves the loans that the city plexes provide to the registered users of the system.
The universe of books is very vast thanks also to the disparate metadata that characterize them. It becomes difficult and boring for a reader to browse through millions of titles and perhaps find on the first try what might be pleasing for himself.
Artificial intelligence and big data techniques can flank the world of reading, bringing it into a more user-oriented dimension, and data analysis plays a significant role in knowing the popularity trends among people also. Applying recommendation techniques to these data-sets enable to provide to users a faster and more refined search for the next book to read.

I structured this thesis by illustrating the state of the art recommendation techniques, in the first part. Then I presented and discussed the work done, in the subsequent chapters, in which I firstly performed an exploratory and detailed analysis of the two data-sets. This allowed me to know, highlight and discover even better characteristics and curiosities about both areas under consideration, aNobii social networks and Turin libraries. Subsequently, I moved the work focus on the application of collaborative filtering recommendation techniques, such ALS and SurPRISE, in order to observe how artificial intelligence can help and hint items in accordance with users preferences. This is done thanks to the study of users iterations and behaviors. After that, I evaluated and explored the possible affinity between various books, exploiting the content of the books metadata available and statistical similarity metrics, such Cosine similarity.
The last part of the work concerns the branch of sentiment analysis. Natural language processing and text analysis permit to catch positive and negative opinions from users. It allows to conduct a significant analysis

thanks to the presence of comments and notes posted by users on the social network during their review phase. Provide to the system with such a tool can give an extra gear, as it would enables to perceive customers' considerations about a book and identify even more their interests and wishes.

Chapter 2

Related work

Recommender systems are essential tools to support the user in the decision-making process, finding himself interacting with a huge amount of products. Recommender systems research has widely incorporated a variety of artificial intelligence techniques [3]. They can be defined as a specific type of information filtering technique that seeks to predict the "rating" or "preference level" a user would give to an item [4].
From entertainment, proposing movies and songs, to e-commerce, suggesting products for sale. From online advertisement, reporting the right contents, to e-learning, capturing the news. Recommender systems can add another dimension to the user experience.
As a proof of the paramount of recommender systems, just mention that Netflix organised a challenges a few years ago with a prize of 1 million dollars to win (the “Netflix prize” [5]). Here the goal was to produce a recommender system that performs better than its own algorithm.

2.1 Recommendation System

Recommender system is defined as a tool that helps users in search something which is related to their tastes or as a strategy of choice for users under complex information environments [1].

Recommender systems handle the problem of overload of information that users find, by providing them personalized and targeted content. For building this systems, distinct approaches have been developed. They can be collaborative filtering, content-based filtering or hybrid filtering [1].
Collaborative filtering recommends items by identifying other users with

similar tastes. On the other hand, Content-based filtering recommends elements that are similar in content to products that user appreciated in the past or matched to user attributes. The hybrid method, as said the name, combines both collaborative and content based techniques in order to reduce and overcome some limitations of this approaches [Figure2.1].

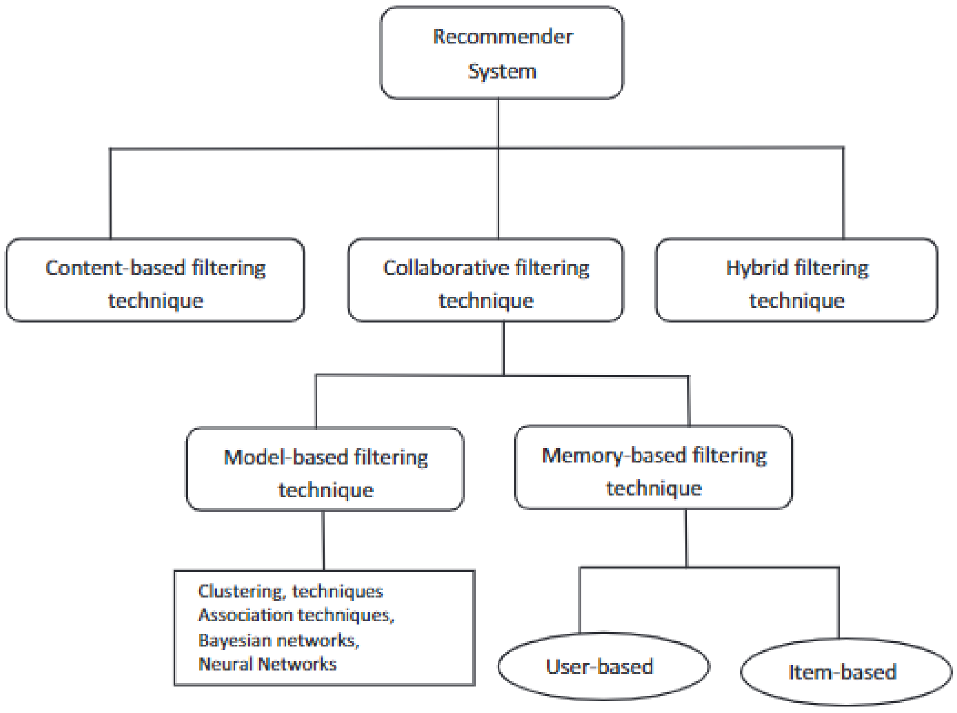

Recommender System
Content-based filtering technique
Collaborative filtering technique
Hybrid filtering technique
Model-based filtering technique
Memory-based filtering technique
Clustering, techniques
Association techniques,
Bayesian networks,
Neural Networks
User-based
Item-based

Figure 2.1: Recommendation techniques [1]

2.1.1 Collaborative Filtering

Collaborative filtering is based on the assumption that the users future preferences can be predicted by their past preferences (acquired via feed-back), in other words what they appreciated in the past will be appreciate also in the future [6]. These past user-item interactions, build through user-item matrix, are sufficient to find matches and to detect similarity between users and/or items and make predictions according on these estimated distances. User-item matrix defines the match data of m users by n items that contains the users ratings over items. So each entry (i, j) in the matrix represents the interaction between user i and item j [Figure

2.2]. Two items are considered to be similar if most of the users that have interacted with both of them did it in a similar way.

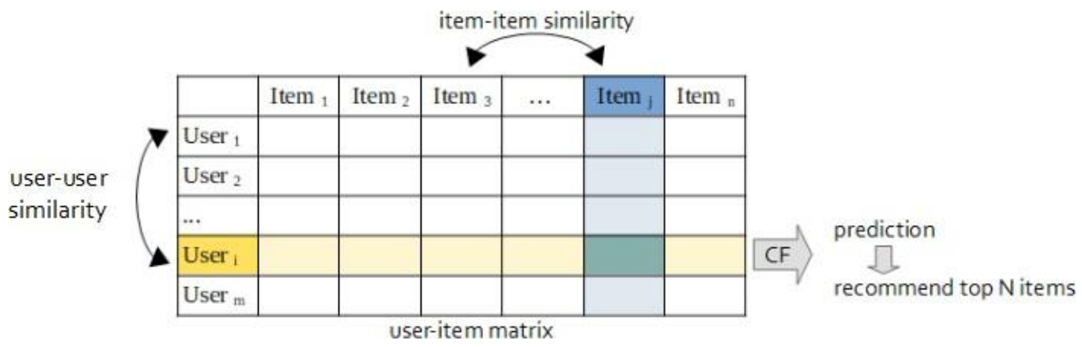


Figure 2.2: Collaborative filtering process

For measure the closeness, many algorithms are used like the K-nearest-neighbor(K-nn) approach, the Cosine similarity or the Pearson Correlation. I explained the details about the similarity metrics in the next subsection on Similarity Computation.
Collaborative filtering methods, as we can see in Figure2.1, are classified as memory-based or model-based.

Memory based approaches works directly with values of stored interactions and this techniques can be of two types: user-based or item-based. Depending on if we want to determine similarity between users or items respectively.
If we want to make a recommendation for a given user, first we represent each element with its vector in the *m* dimensional user space of interactions with different users ("its column" in the interaction matrix).
The similarity between two vectors is calculated based on the angle between them, if we are considering the nearness with the cosine similarity metric. This measure can be bounded between -1 and 1. The smaller the angle, the more equal the two vectors are. When the angle between two vectors is 0° they get maximum similarity (they are oriented in the same direction), when the angle between them is 90° they reach 0 similarity instead (they are orthogonal to one another), and when the angle is 180° they have -1 similarity (they are oriented in diametrically opposing directions) [7].
On the other hand, Pearson's correlation is the same thing as Cosine similarity if the two vectors are centered firstly by removing the mean vector.

Hence, the same practice is conducted by projecting onto the unit sphere and taking the inner product. The Pearson's correlation similarity is expressed with a value between -1 to 1 also, -1 shows negative correlation while 1 indicates positive correlation [8].
These techniques compute similarity between users or items by exploiting and comparing their similarities. The predicted rating will be obtained with approaches as nearest-neighbor or graph-based.

Model-based approaches assume the presence of a model that explains user-item interactions and seek to discover it to make new predictions. These models are developed using different machine learning algorithms to predict user ratings of unrated items. For instance Bayesian networks, Clustering models, Dimensionality Reduction techniques such Singular Value Decomposition (SVD).
In this thesis I focused my work on some approaches (based on Matrix Factorisation).

Among the problems of the collaborative approach, the most common occurs when a new user joins the system, what is called "cold start". It occurs when the system does not have enough information about the learner and will consequently produce a less accurate recommendation. Similar situation happens when a new item is added. Collaborative filtering approaches suffer from sparsity and scalability problems also; the first one because the number of items often is extremely large, the second one owing to necessity of a large amount of computation power due to enormous quantities of data [9].

One scenario of collaborative filtering application is to recommend interesting or popular information as suggested by the community. The word of mouth on social networks among readers with similar tastes is a fundamental resource for extrapolate possible suggestions for the online books. This happens in the case of aNobii social network, the reality that I analyzed and I will explain in the next chapter.

Similarity computation

The most critical step of the method is the mechanism for finding similarities using the data, so that items can be recommended based on their similarities. The calculation of similarities with Pearson's correlation and

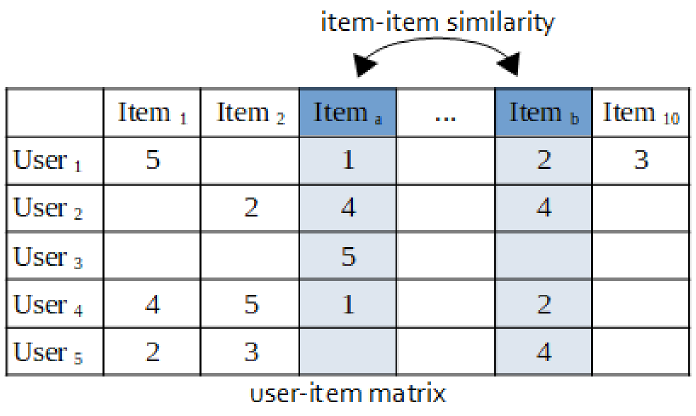
Cosine are mostly used in traditional collaborative filtering recommendation systems [10].
Pearson's correlation similarity of two users *x* and *y* is defined as:

$$w_{x,y} = \frac{cov(x, y)}{\sigma_x \sigma_y} = \frac{\sum_{j=1}^{J_{x,y}} (r_{x,j} - \bar{r}_x)(r_{y,j} - \bar{r}_y)}{\sqrt{\sum_{j=1}^{J_{x,y}} (r_{x,j} - \bar{r}_x)^2} \sqrt{\sum_{j=1}^{J_{x,y}} (r_{y,j} - \bar{r}_y)^2}}$$

Cosine similarity between two users *x* and *y* is:

$$w_{x,y} = \frac{x \cdot y}{|x||y|} = \frac{\sum_{j=1}^{J_{x,y}} r_{x,j} r_{y,j}}{\sqrt{\sum_{j=1}^{J_{x,y}} r_{x,j}^2} \sqrt{\sum_{j=1}^{J_{x,y}} r_{y,j}^2}}$$

where $J_{x,y}$ is the set of items rated by both user *x* and user *y* and $\bar{r}_x$ is the average rating of user *x* [10].
Additionally, in a context of data mining, elements in *x* and *y* can be distributed on a different scale, so mean-centering of the vectors usually leads better results. Notice that cosine similarity between the mean-centered vectors is mathematically equivalent to the Pearson correlation.
The main disadvantage of Pearson's correlation is that it does not provide an accurate result when the two users have a common rating or when a particular user has only rated one item. However, the disadvantage of the Cosine measure is that it does not consider the differences in the mean and variance of the ratings made by users *x* and *y* [11].



	Item $_1$	Item $_2$	Item $_a$	...	Item $_b$	Item $_{10}$
User $_1$	5		1		2	3
User $_2$		2	4		4	
User $_3$			5			
User $_4$	4	5	1		2	
User $_5$	2	3			4	

Figure 2.3: Collaborative filtering item-item similarity

For instance, in the [Figure2.3] the elements are ratings in a [1,5] range for each user-item pair. Item-item similarity is computed by looking onto co-rated items only. From an item-item perspective, $item_a$ and $item_b$ are similarly rated (r) by user 1, 2 and 4. If we compute the Cosine similarity the formula become:

$$sim(\vec{a},\vec{b}) = cos(\vec{a},\vec{b}) = \frac{\vec{a}\cdot\vec{b}}{|\vec{a}||\vec{b}|} = \frac{\sum_{j=1}^{J_{a,b}} r_{a,j}\, r_{b,j}}{\sqrt{\sum_{j=1}^{J_{a,b}} r_{a,j}^2}\sqrt{\sum_{j=1}^{J_{a,b}} r_{b,j}^2}}$$

where $J_{a,b}$ indicates over only common rated rows.
Still, if we calculate the similarity between the two elements with Pearson's correlation, the formula become:

$$sim(\vec{a},\vec{b}) = \frac{cov(\vec{a},\vec{b})}{\sigma_a\sigma_b} = \frac{\sum_{j=1}^{J_{a,b}} (r_{a,j}-\bar{r}_a)\,(r_{b,j}-\bar{r}_b)}{\sqrt{\sum_{j=1}^{J_{a,b}} (r_{a,j}-\bar{r}_a)^2}\sqrt{\sum_{j=1}^{J_{a,b}} (r_{b,j}-\bar{r}_b)^2}}$$

where $\bar{r}_a = \frac{1}{5} * \sum_{j=1}^{5}(r_{a,j})$ and $\bar{r}_b = \frac{1}{5} * \sum_{j=1}^{5}(r_{b,j})$, the mean values of the elements in a vectors.

Dynamic item-based recommendation

The classical Collaborative filtering approach is a static technique. In real-live, users' redefines their tastes continuously. The products perception and popularity often change over time. The time is a significant factor for learn the user's preferences trend and the item's popularity decay over time. The systems that take care of the temporal effects overcome the data sparsity issue and significantly improve the performance of collaborative filtering techniques providing better suggestion to the users in current time period [5].
The dynamic item-based recommendation algorithm utilizes the time decay to design the models and provide recommendations. These models exploit correlation-based techniques to compute similarities matrix between items and then extract top N-recommendations. One of the main motivations behind these models depends on the fact that similarity between two rated items changes according to the time interval between the two rated behaviors. This technique extends the idea of human brain memory to identify the level of a user's interests (i.e., instantaneous, short-term,

long-term) focusing on user's temporary tastes. For instance the similarity between two items that are rated by the same user on the same week differs from the similarity between two items that are rated by the same user in the same year. This is called "time decay". To quantify and utilize the effect of time decay it is necessary to consider the addition of a factor related to the time interval of the similarity calculation [12]. This factor is defined as

$$f(t) = \exp^{-\alpha(t-t_{u,i})} \text{ [13]}$$

where $t_{u,i}$ is the timestamps that record the time when user u rated the item i. In brackets, the time difference is the interval between the most recent system rating date and the evaluation date of the user u on the *i-th* item. The larger time interval is, the stronger the effect of time decay is [12]. The α factor can assume values between [0,1].
This time weight function is applied as a multiplicative factor to the review data and involves a relevant contribution when calculating the statistical similarity metric between the elements. The use of this approach allows to improve the classical collaborative filtering model identifying not only items with high similarity, items that have been rated similarly, but also items evaluated close in time [13].
An example of this approach is showed in the subsection of Dynamic item-based similarity.

2.1.2 Content-based filtering

Content-based filtering is an information filtering techniques which uses context of items. It emphasizes more on the analysis of the items' attributes in order to generate predictions [1]. The idea behind this method is to try to build a model that explain the observed user-item matches, based on the available features.

The content-based filtering technique does not need to be based on the profile of other users as they do not affect the recommendation. The main disadvantage of this approach is the need to have a rich knowledge and description of the items attributes in the profile [1].
Content based methods suffer less of the cold start problem than collaborative approaches, because new users or items can be described by their features and so can be done relevant suggestions for these new entities. Unless they have features that never appeared [14].

The system recommends new items based on how similar they are to what the user has found interesting in the past, by observing the content of the elements. From here a profile of user tastes is possible to built.
The process of recommending consists to match the attributes of the user profile with the attributes of a content of item. The result is a relevance judgment that represents the user's level of interest in that item [15]. The implementation of a content-based system requires three basic steps:

1. attribute extraction, process for extracting relevant content;
2. profile learner, to collect representative data of the user preferences and to generalize it and build the user profile;
3. generate recommendations, to compute the similarity between user and items by means of user profiles and items attributes.

One of the advantages of the content-based technique is that the system can adapts its recommendations within a very short period of time if the user profile changes. Another positive aspect is that this technique of recommendation overcome the problems of the collaborative approach. They can recommend new items, while there are no user provided ratings. However, the content-based technique suffer from some problems. One of the crucial point is the need to have a wide knowledge of the descriptions of the items features in the profile. Another essential consideration is that these techniques are strictly dependent on the items metadata, so an abundant description of them is necessary [1].

2.1.3 Evaluation of recommendation systems

Evaluating a model is a core part of building an effective recommendation system. This make part of the entire process of developing as it helps in the configuration of the parameters.
In recommendation systems where the user's interest is expressed on a scale of values, there are statistical metrics for measuring the accuracy and evaluate the model performance. Root Mean Square Error (RMSE) is one of statistical metric that is usually used to measure the error between the evaluations predicted and the actual ratings.
RMSE is one way to evaluate linear regression models by measuring the accuracy of the estimated results of a model [16]. It is calculated by squaring the error between prediction and observation. The formula of

RMSE is:

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

where y_i is the observed value for the ith observation and $\hat{y}_i$ is the predicted value. The lower the RMSE is, the better the recommendation accuracy.
Other important factors [4] must be considered evaluating recommendation system, such as:

- diversity, users tend to mostly appreciate the recommendations when there is a significant variety between the items suggested, products from different authors for give you an idea;
- trust, reliability in the recommendation system results by explaining the reason why it suggests an item and how it deals recommendations;
- recommender persistence, in some situations, it is more effective to re-show recommendations than providing new items. Due to the fact that users sometimes ignore items when they are shown for the first time;
- serendipity, measure "how surprising the recommendations are". To recommend items that are not so evident to the user, amplifying his range of choosing.

2.2 Sentiment Analysis

Sentiment Analysis is the research area that studies opinions, feelings, evaluations, attitudes and emotions of people about entities and their aspects expressed through a written text [17]. Given the constant growth of the internet and in general of models that place the user at the center of content creation, there is a rising interest from companies and organizations regarding the possibility of using this type of data to design and monitor their strategies. Sentiment analysis serves to take and identify positive or negative opinion expressed by the users, so problem of classification is identified, with the aim to use them to perceive their interests, preferences and views.
Consequently, sentiment analysis applications have spread across several domains, from financial services, to consumer products, to health care

services and social events. Opinions are central to almost all everyday activities and they influence our behaviors. Before making a decision, usually we want to know other opinions. With the explosive expansion of social media (for instance reviews, comments, forum discussions, blogs, posts), users and companies increasingly use the content present during the decision-making process [17].
The analysis of a written text is the main task of sentiment analysis and is strictly related to the Natural Language Processing (NLP). The NLP, as sayd the name, is a subfield of linguistics, computer science, and artificial intelligence concerned the process of handling information written or spoken in the natural language [18]. The target of sentiment analysis is identify sentiment and then classify them according to a polarity that it expresses the feeling level of the user about a certain topic. One of the foundamental steps in NLP is to understand the polarity of a given sentence. In sentiment classification, some words, like "interesting", "excellent", "amazing", "horrible", "boring", "disappointing", etc. , are more important because they express positive or negative judgment from the user [17]. The data preprocessing is a crucial challenge in sentiment analysis because selecting the appropriate preprocessing methods improves the correctly classified sentences [19]. Sentiment classification is essentially a two-classes document classification problem, positive and negative. The supervised machine learning approaches are the most used existing classification techniques at text-level [17].

Chapter 3

Data-set analysis

Before exploring how predict ratings, I will explain related data-sets on which I focused my workflow. The data was stored in the distributed file system of the Hadoop cluster BigData@Polito.
The considerable amount of data present made it necessary to use Big Data techniques for the exploration, knowledge and analysis of them.

In this thesis, the data processing and analysis was carried out with the aid of PySpark programming language on the JupyterLab interface. Furthermore, the use of the PostgreSQL DBMS, an open source object relational database system that uses and extends the SQL language, allowed to have an overview of the data and their typology [20].

3.1 Data of aNobii overview

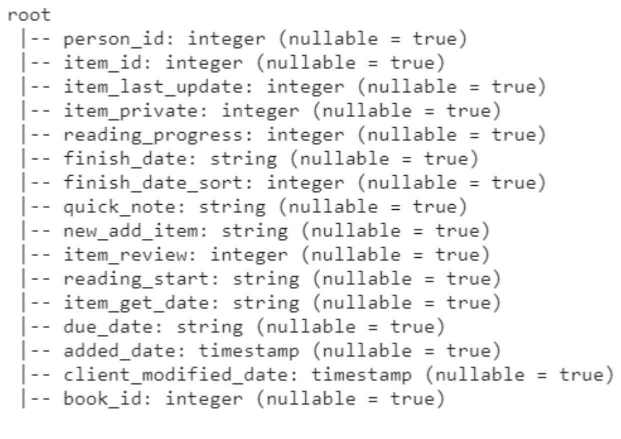
The first dataset under analysis belongs to the aNobii network service. Anobii is a social network where users can share their passion for literature. The aNobii online platform changed its name to Anobii in the following years, but since the data that are provided to me are related until the year 2016 I will refer in the thesis to the social network with its previous name aNobii.
This online community was been created for book lovers in 2005 in Hong Kong with the Slogan: "*Anobii: Together We Find Better Books*" [21].
The term "aNobii" rises from the name of *Anobium punctatum,* the "paper worm", the one who feeds on cellulose. Meaning like someone who spends a lot of time on books metaphorically [22].

«The idea of Anobii came to me after reading a book on web 2.0. I lived in Hong Kong and felt the need to know what other people had read it in my city. I wanted to talk to them, I wanted to exchange my impressions with them and to point out other books on the subject. That's how the idea was born, quite simply.»
(Greg Sung, public meeting in Milan, 4 February 2009 [22])

Anobii has readers in over 20 countries, but is more popular in Italy [23]. Registered users can find useful information on millions of books, exchange opinions with other readers, see what "anobians" think, read reviews and make ratings as well [21].

3.1.1 Preliminary data exploration

A fundamental part of the work concerns data preprocessing and cleaning. The data analysis allows to transform the big volume of data into a more understandable, useful and efficient format for their use, so it is an essential task useful to have significantly better performance in the subsequent steps.
The data-set provided to me contains a set of tables of the social network aNobii data. I mostly focused my attention on the "*link_person_item*" table, the one containing the features that characterize the reviews, among the tables that have been supplied to me. The [Figure3.1] shows the list of the data-frame columns of the table mentioned and the corresponding type of data contained in each one.



```
root
 |-- person_id: integer (nullable = true)
 |-- item_id: integer (nullable = true)
 |-- item_last_update: integer (nullable = true)
 |-- item_private: integer (nullable = true)
 |-- reading_progress: integer (nullable = true)
 |-- finish_date: string (nullable = true)
 |-- finish_date_sort: integer (nullable = true)
 |-- quick_note: string (nullable = true)
 |-- new_add_item: string (nullable = true)
 |-- item_review: integer (nullable = true)
 |-- reading_start: string (nullable = true)
 |-- item_get_date: string (nullable = true)
 |-- due_date: string (nullable = true)
 |-- added_date: timestamp (nullable = true)
 |-- client_modified_date: timestamp (nullable = true)
 |-- book_id: integer (nullable = true)
```

Figure 3.1: Features of aNobii 'link_person_item' table

In details, the column:

1. "item_private" is the item private status (0: public; 1: private);
2. "reading_progress" is an integer which specifies the reading progress (from 1 to 6);
3. "quick_note" is a private note for the book;
4. "item_review" is the item rating (up to 5 stars);
5. "reading_start" is the start date of the reading.

As first step, I did a descriptive statistic that is a study of data analysis to describe, show and summarize data in a meaningful way. The dimensions of the data-frame was (47 482 387, 16), in terms of samples and features (rows and columns). The data-frame has 8 numeric columns, 6 categorical columns, while the remaining 2 are of timestamp type, as demonstrated by "*count_column_types()*" function. Moreover, the number of distinct users and items was respectively 507 719 and 3 373 428.
To have a more detailed overview, I analyzed some basic statistical details like mean, standard deviation, min and max value of each column of which the data-set is composed. These analysis let to make some deductions. In

summary	item_review
count	47 482 355
mean	1.67
stddev	2.03
min	0
max	55

Table 3.1: Statistics on columns of interest

particular, observing the *item_review* feature of interest, it was possible to note from the table above [Table3.1] that there is an massive presence of 0 values, considering the mean value of approximately 1.67 on 47 482 355 samples. Observing the max value it is significant to notice the presence of some anomalous value also. This values are called outliers in statistics. An outlier is a value clearly distant from the other observations available. From here we can infer the need to carry out a data cleaning process a priori.
Data Cleaning phase process detects and removes the errors and inconsistencies present in the data removing noisy, invalid data and outliers.

During this first general survey in addition, it is possible to notice a particular aspect of the data-frame that should not be underestimated: the presence of a significant amount of null or missing values that must be handled. To overcome this aspect the rows that contained inconsistent data were removed.

3.1.2 Exploratory Data Analysis

One of the steps in the exploration of the data-set is to have a rough idea of how the features are distributed.
The exploratory phase was divided into two stages: exploration of numerical and categorical features respectively.

Exploring Numerical Data

For numerical features is done an analysis of distribution combining the histogram plots with the kernel density estimate graphs. The latter are used for visualizing the Probability Density of a continuous variable. The empirical PDF (Probability Density Function) and the CDF (Cumulative Distribution Function) have been considered to measure and visualize the distribution of the features. The first is the probability of a given continuous outcome, it is often expressed in terms of an integral; the second is the probability that the variable takes a value less than or equal to a given outcome.
With this analysis the more massive presence of 0 values in the *item_review* feature clearly emerged.
In general, observing the distributions it is clear that the features have a rather different scale of values and therefore it is not possible to make a direct comparison with box-plots. So, to get further confirmation of these observations, I computed the box-plot analysis considering each feature separately.
The box plot is a further statistic technique for summarize the distribution of values in the data-set highlighting the minimum and maximum range values, the upper and lower quartiles, and the median [24]. The box is bounded by the first and third quartiles and divided inside by the median, that is the second quartile. The segments, which come out of the box, are delimited by the minimum and maximum value of the feature. The Box-Plot provides to show also the presence of outliers in the data

distribution [24].

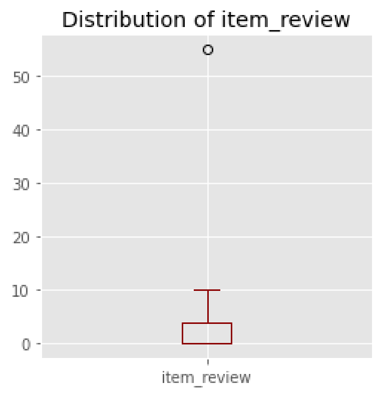


Figure 3.2: Box-plot distribution

The [Figure3.2] shows the box-plot relative to the feature *item_review*. Note the presence of outliers.

Exploring Categorical Data

Categorical data are discrete data. Most of the categorical features represent dates in string format, so it is not meaningful to plot their distribution. The extra *quick_note* feature is a very variable field as it expresses a note, a comment posted by a user.

This preliminary analysis of the data allowed to have a first overview of the distribution of the data and in particular of the variables that make up the data-set.

3.1.3 Further analysis on the data-set

The data-set contains 507 719 distinct values of total users and 3 373 428 unique values of total items.
First of all, I analyzed the distribution of the values assumed by the *item_reviews* feature, not considering the value 0 and after removing the outliers values [Figure3.3]. From the graph it can be seen that the amount of the evaluations make by the users of type 4 and 5 is significantly higher than the 1 or 2 types.

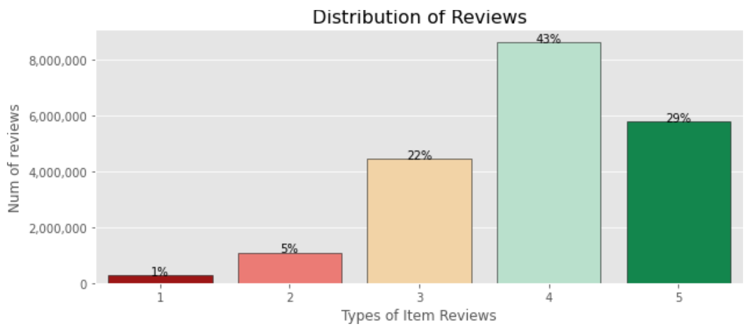
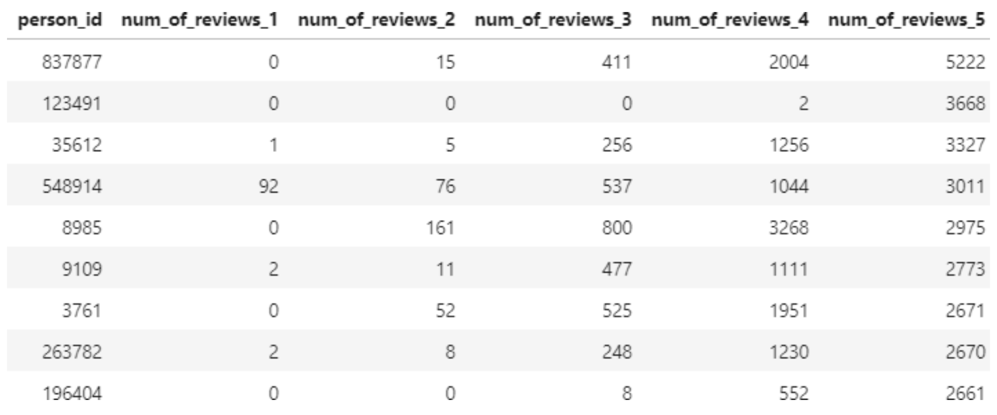


Figure 3.3: Distribution of reviews carried out by the users

After having created a dictionary with as keys the identifiers of the users and as values the number of reviews made, distinguished by type, it was possible to perform a further analysis on the distribution of the reviews assigned by type. In the [Figure3.4] is shown a portion of output obtained with this analysis. I sorted the values according to number of reviews 5 in descendent order for to see first the users that have made more positive ratings.

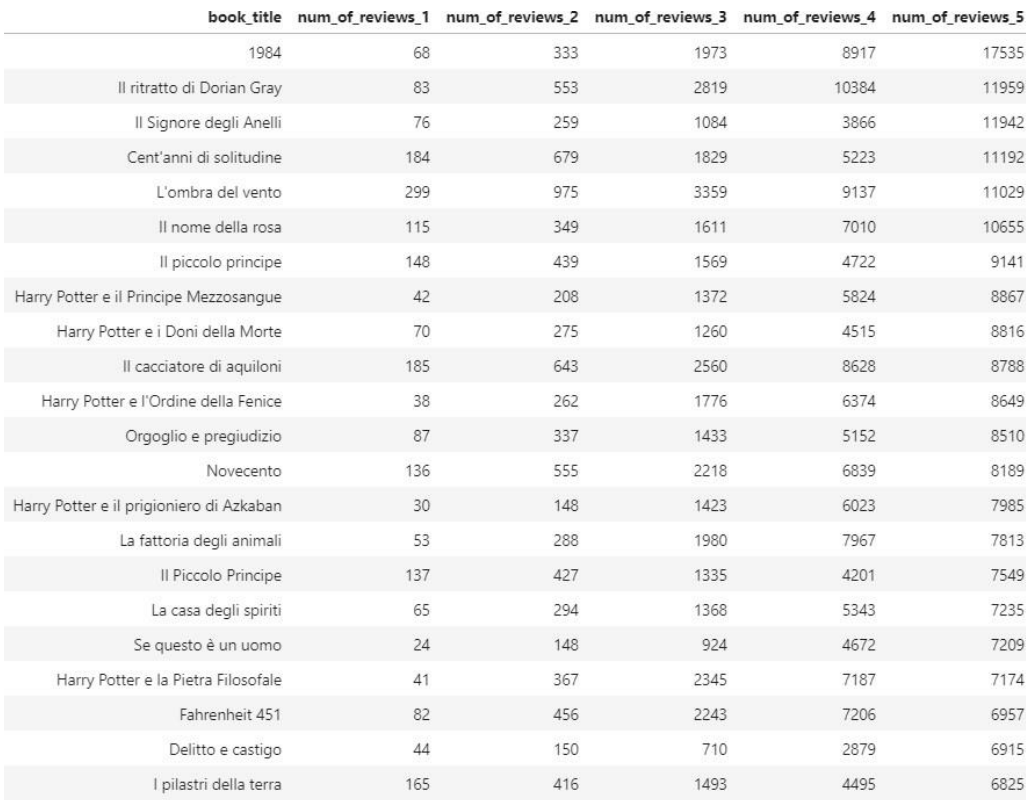


person_id	num_of_reviews_1	num_of_reviews_2	num_of_reviews_3	num_of_reviews_4	num_of_reviews_5
837877	0	15	411	2004	5222
123491	0	0	0	2	3668
35612	1	5	256	1256	3327
548914	92	76	537	1044	3011
8985	0	161	800	3268	2975
9109	2	11	477	1111	2773
3761	0	52	525	1951	2671
263782	2	8	248	1230	2670
196404	0	0	8	552	2661

Figure 3.4: Type of reviews done by users distinguished by type

Later, the analysis was concentrated on the three columns of interest: *person_id*, *item_id* and *item_review*. Wanting to have a more interpretable analysis format, I joined these columns with those in the items details table through an inner join. In this way, I was able to subsequently create a new dictionary containing the book titles as keys and the number of reviews obtained as values, in this case again distinguished by type. From this dictionary I created a new data-frame related to this analysis.

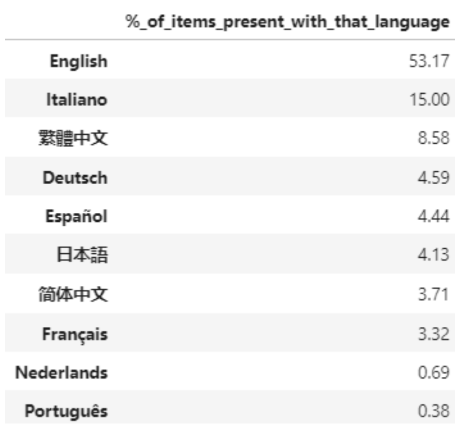
I sorted the values according to number of reviews 5 in descendent order for to see first the titles most positively rated. In the [Figure3.5] is shown an extract of the output obtained.



book_title	num_of_reviews_1	num_of_reviews_2	num_of_reviews_3	num_of_reviews_4	num_of_reviews_5
1984	68	333	1973	8917	17535
Il ritratto di Dorian Gray	83	553	2819	10384	11959
Il Signore degli Anelli	76	259	1084	3866	11942
Cent'anni di solitudine	184	679	1829	5223	11192
L'ombra del vento	299	975	3359	9137	11029
Il nome della rosa	115	349	1611	7010	10655
Il piccolo principe	148	439	1569	4722	9141
Harry Potter e il Principe Mezzosangue	42	208	1372	5824	8867
Harry Potter e i Doni della Morte	70	275	1260	4515	8816
Il cacciatore di aquiloni	185	643	2560	8628	8788
Harry Potter e l'Ordine della Fenice	38	262	1776	6374	8649
Orgoglio e pregiudizio	87	337	1433	5152	8510
Novecento	136	555	2218	6839	8189
Harry Potter e il prigioniero di Azkaban	30	148	1423	6023	7985
La fattoria degli animali	53	288	1980	7967	7813
Il Piccolo Principe	137	427	1335	4201	7549
La casa degli spiriti	65	294	1368	5343	7235
Se questo è un uomo	24	148	924	4672	7209
Harry Potter e la Pietra Filosofale	41	367	2345	7187	7174
Fahrenheit 451	82	456	2243	7206	6957
Delitto e castigo	44	150	710	2879	6915
I pilastri della terra	165	416	1493	4495	6825

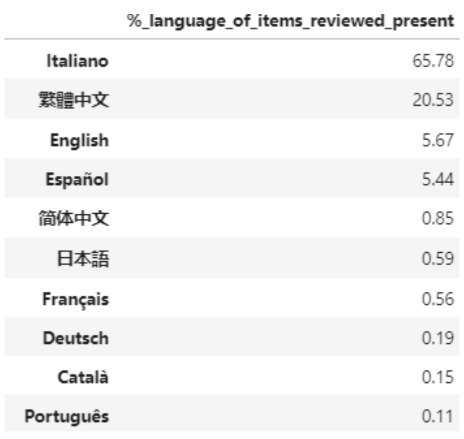
Figure 3.5: Type of reviews received from the books

An additional interesting analysis was carried out on the language of the books present in the data-set to get an idea of the quantitative percentage of books present, distinct by languages [Figure3.6]. This was possible thanks to the merge with the available language table.
Through this survey I was able to ascertain that, out of a total of 78 different languages, 53.17 % of the books present are English, followed by a considerable percentage of Italian books [Figure3.6a]. The latter are also those most valued by users. From the [Figure3.6b] it is possible to deduce that 65.78% of the books evaluated are Italian.
Finally, other curiosities analyzed have been developed about the person table. In particular, attention was paid to the gender of users of the social network [Figure3.7] and the distribution of their ages [Figure3.8]. The age factor was calculated considering the date of birth of the users.



	%_of_items_present_with_that_language
English	53.17
Italiano	15.00
繁體中文	8.58
Deutsch	4.59
Español	4.44
日本語	4.13
简体中文	3.71
Français	3.32
Nederlands	0.69
Português	0.38

(a)



	%_language_of_items_reviewed_present
Italiano	65.78
繁體中文	20.53
English	5.67
Español	5.44
简体中文	0.85
日本語	0.59
Français	0.56
Deutsch	0.19
Català	0.15
Português	0.11

(b)

Figure 3.6: Top languages most present

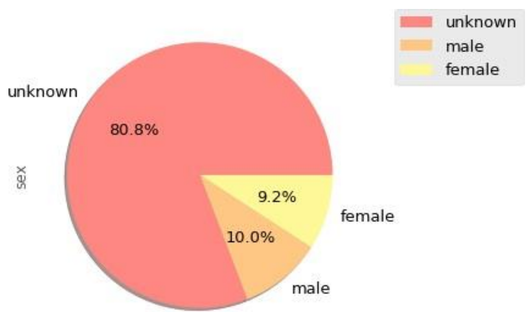


Figure 3.7: Gender of users present

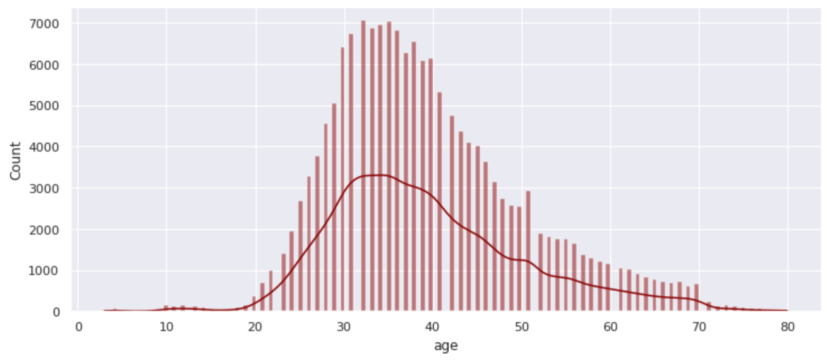


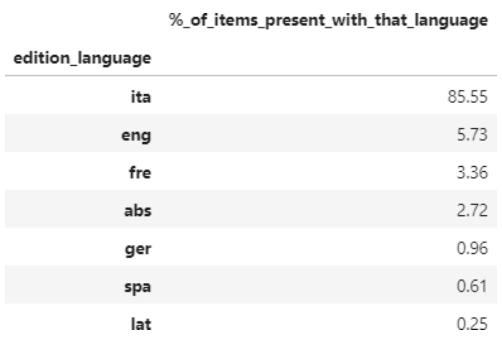
Figure 3.8: Age groups of users present

3.2 Data of loans of Turin libraries overview

Complementary to the aNobii data-set, the data-set of loans from the libraries of the city of Turin was also analyzed during the thesis work. This data-set consists of the following tables:

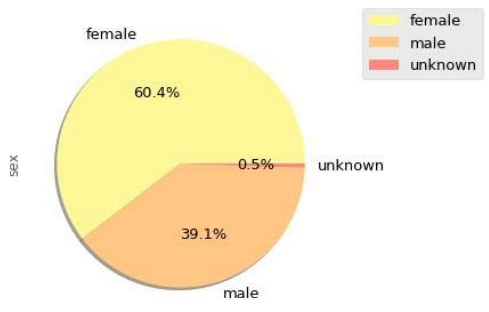
- items: containing the details of the copies of the books;
- manifestations: containing the books details;
- patrons: having the details of the users registered in the system;
- libraries: containing the names of the libraries in the city;
- loans: having information about the details of the loans.

Initially, for this data an exploratory phase was performed also. This was useful for understanding their nature and evaluating their size and distribution. Through this preliminary analysis, it was interesting observe the percentage distribution of book languages in libraries, visible in [Figure 3.9a], and the users gender distribution shown in [Figure3.9b].



edition_language	%_of_items_present_with_that_language
ita	85.55
eng	5.73
fre	3.36
abs	2.72
ger	0.96
spa	0.61
lat	0.25

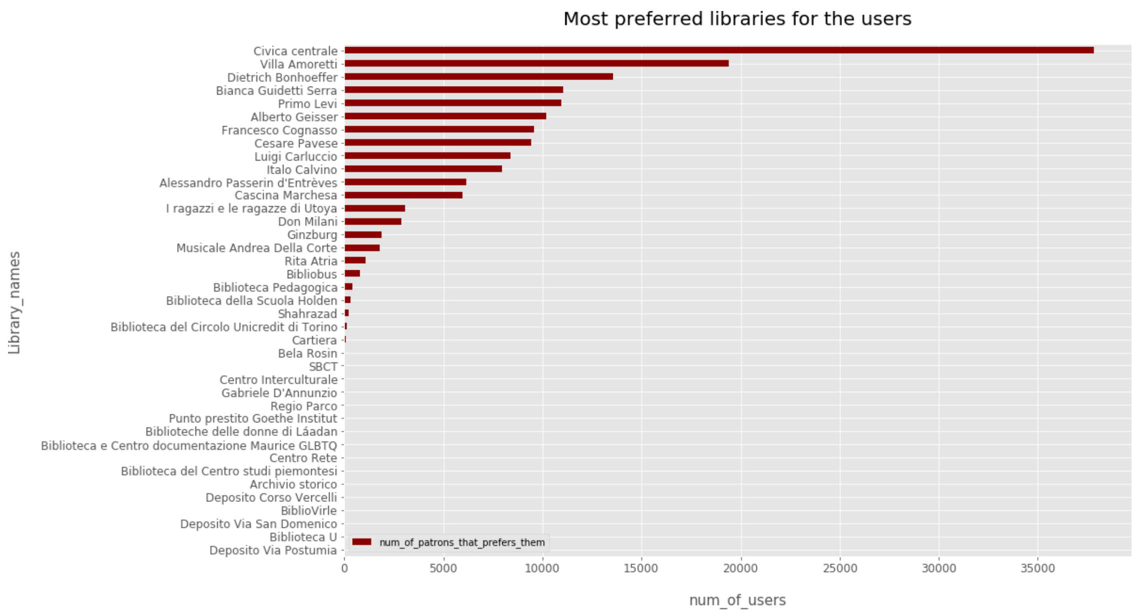
(a)



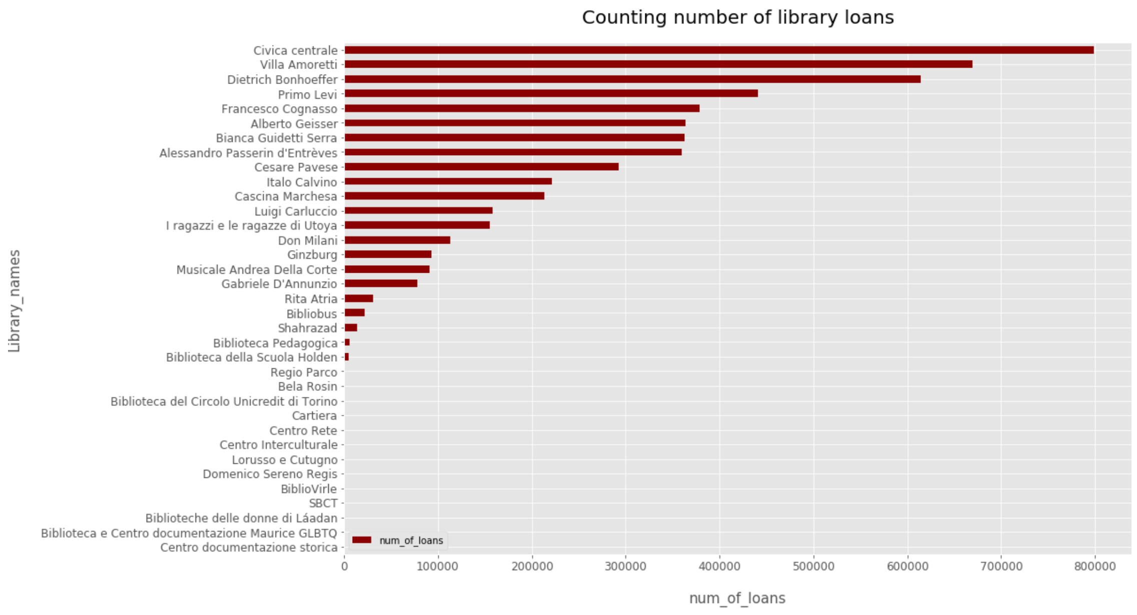
(b)

Figure 3.9: Some distributions of Data-set on loans from Turin libraries

In addition, thanks to the features contained in the database, it was curious to make further analysis on the most classified libraries by users as preferred, on the amount of books that each library has, on the frequency of loans in libraries and on their average duration (for instance, it turned out that the average duration of a loan was approximately one month). In [Figure3.10] the graphs on these considerations are displayed.



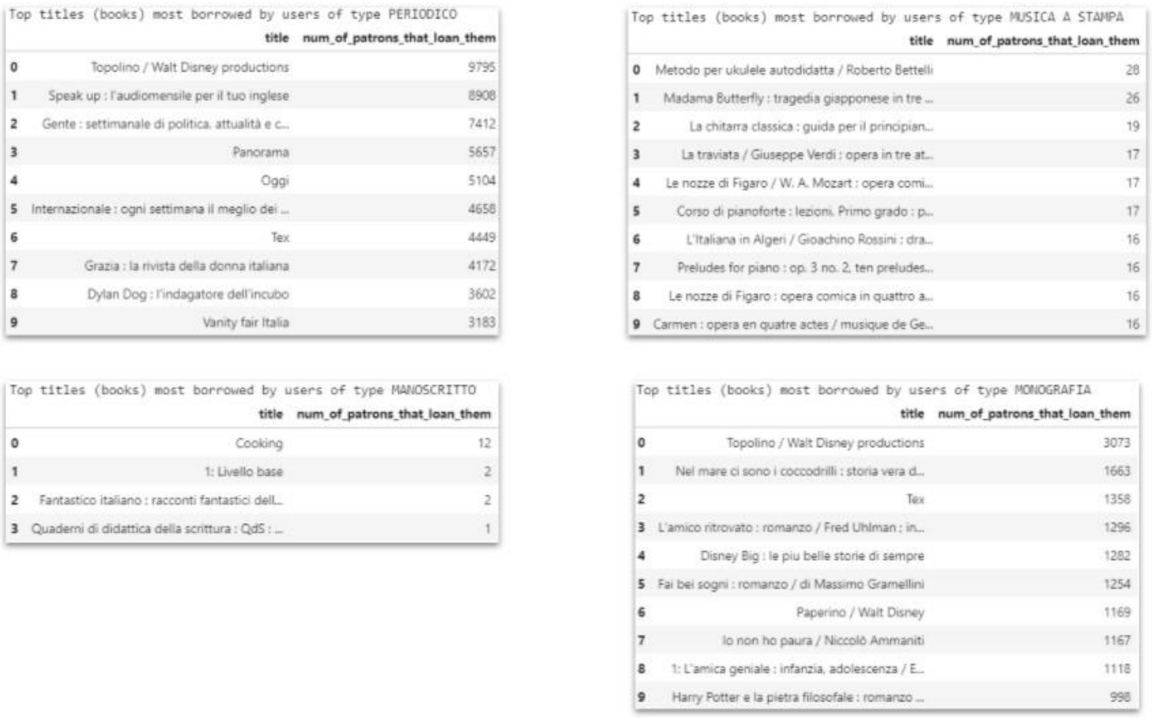
(a)



(b)

Figure 3.10: Further analysis on the data

Subsequently, it was curious to discover which books are most borrowed by library users, making a distinction by type of book [Figure3.11] and as well as considering the books in general [Figure3.12].



Top titles (books) most borrowed by users of type PERIODICO

	title	num_of_patrons_that_loan_them
0	Topolino / Walt Disney productions	9795
1	Speak up : l'audiomensile per il tuo inglese	8908
2	Gente : settimanale di politica, attualità e c...	7412
3	Panorama	5657
4	Oggi	5104
5	Internazionale : ogni settimana il meglio dei ...	4658
6	Tex	4449
7	Grazia : la rivista della donna italiana	4172
8	Dylan Dog : l'indagatore dell'incubo	3602
9	Vanity fair Italia	3183

Top titles (books) most borrowed by users of type MUSICA A STAMPA

	title	num_of_patrons_that_loan_them
0	Metodo per ukulele autodidatta / Roberto Bettelli	28
1	Madama Butterfly : tragedia giapponese in tre ...	26
2	La chitarra classica : guida per il principian...	19
3	La traviata / Giuseppe Verdi : opera in tre at...	17
4	Le nozze di Figaro / W. A. Mozart : opera comi...	17
5	Corso di pianoforte : lezioni. Primo grado : p...	17
6	L'Italiana in Algeri / Gioachino Rossini : dra...	16
7	Preludes for piano : op. 3 no. 2, ten preludes...	16
8	Le nozze di Figaro : opera comica in quattro a...	16
9	Carmen : opera en quatre actes / musique de Ge...	16

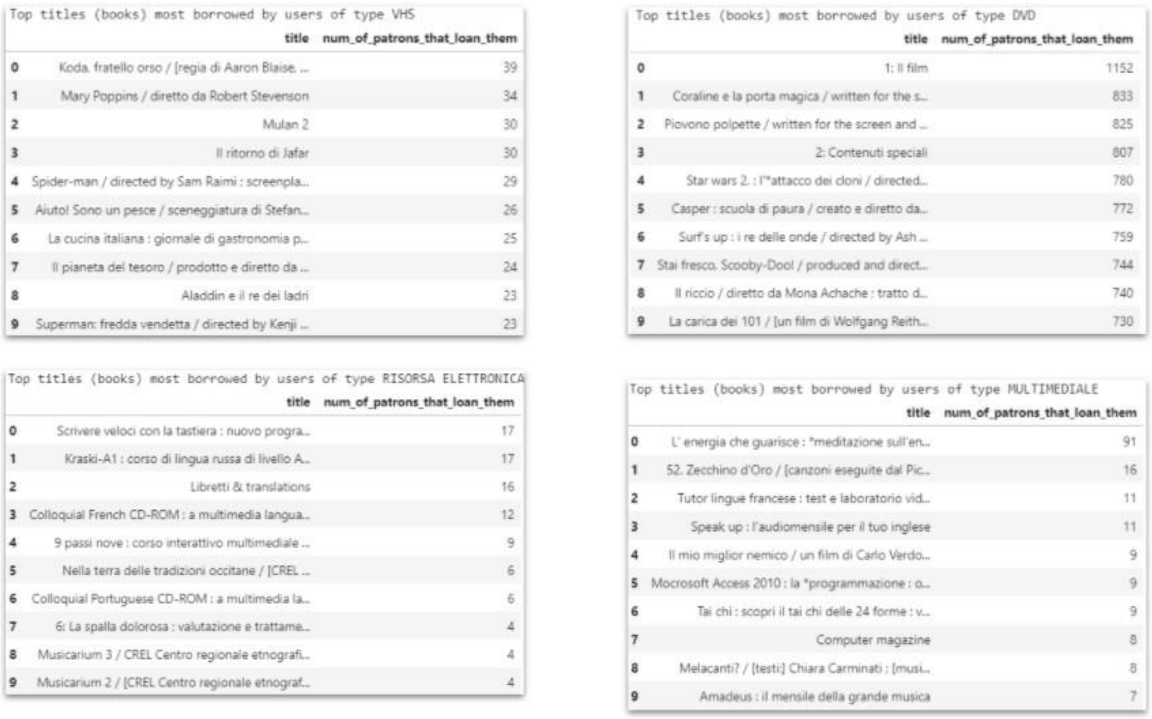
Top titles (books) most borrowed by users of type MANOSCRITTO

	title	num_of_patrons_that_loan_them
0	Cooking	12
1	1: Livello base	2
2	Fantastico italiano : racconti fantastici dell...	2
3	Quaderni di didattica della scrittura : QdS : ...	1

Top titles (books) most borrowed by users of type MONOGRAFIA

	title	num_of_patrons_that_loan_them
0	Topolino / Walt Disney productions	3073
1	Nel mare ci sono i coccodrilli : storia vera d...	1663
2	Tex	1358
3	L'amico ritrovato : romanzo / Fred Uhlman ; in...	1296
4	Disney Big : le piu belle storie di sempre	1282
5	Fai bei sogni : romanzo / di Massimo Gramellini	1254
6	Paperino / Walt Disney	1169
7	Io non ho paura / Niccolò Ammaniti	1167
8	1: L'amica geniale : infanzia, adolescenza / E...	1118
9	Harry Potter e la pietra filosofale : romanzo ...	998

(a)



Top titles (books) most borrowed by users of type VHS

	title	num_of_patrons_that_loan_them
0	Koda, fratello orso / [regia di Aaron Blaise, ...	39
1	Mary Poppins / diretto da Robert Stevenson	34
2	Mulan 2	30
3	Il ritorno di Jafar	30
4	Spider-man / directed by Sam Raimi : screenpla...	29
5	Aiuto! Sono un pesce / sceneggiatura di Stefan...	26
6	La cucina italiana : giornale di gastronomia p...	25
7	Il pianeta del tesoro / prodotto e diretto da ...	24
8	Aladdin e il re dei ladri	23
9	Superman: fredda vendetta / directed by Kenji ...	23

Top titles (books) most borrowed by users of type DVD

	title	num_of_patrons_that_loan_them
0	1: Il film	1152
1	Coraline e la porta magica / written for the s...	833
2	Piovono polpette / written for the screen and ...	825
3	2: Contenuti speciali	807
4	Star wars 2. : l'"attacco dei cloni / directed...	780
5	Casper : scuola di paura / creato e diretto da...	772
6	Surf's up : i re delle onde / directed by Ash ...	759
7	Stai fresco, Scooby-Doo! / produced and direct...	744
8	Il riccio / diretto da Mona Achache : tratto d...	740
9	La carica dei 101 / [un film di Wolfgang Reith...	730

Top titles (books) most borrowed by users of type RISORSA ELETTRONICA

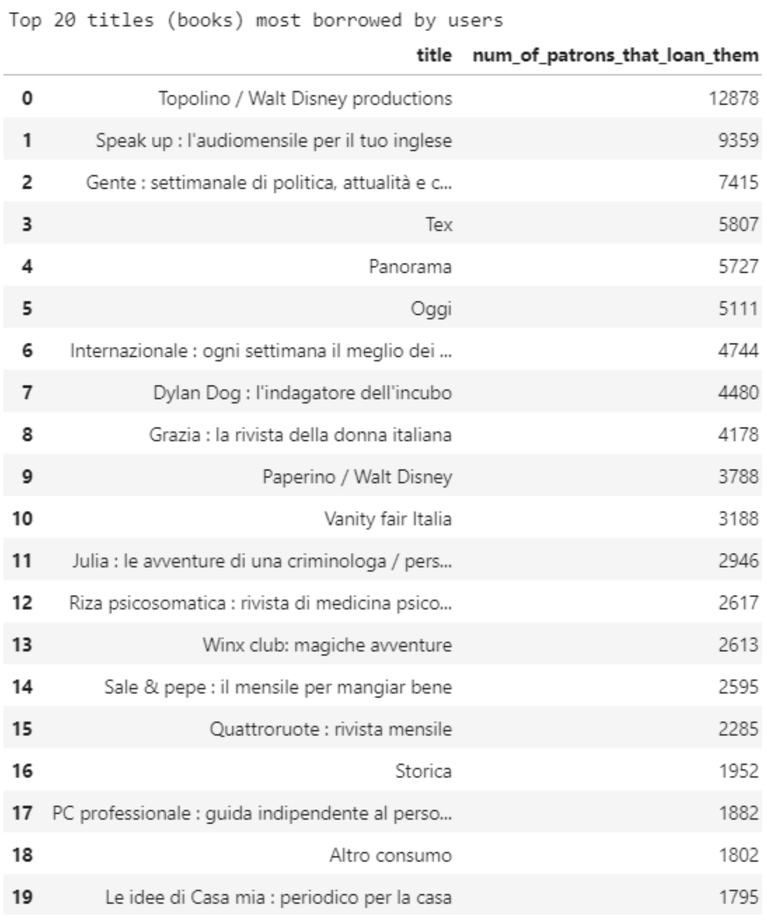
	title	num_of_patrons_that_loan_them
0	Scrivere veloci con la tastiera : nuovo progra...	17
1	Kraski-A1 : corso di lingua russa di livello A...	17
2	Libretti & translations	16
3	Colloquial French CD-ROM : a multimedia langua...	12
4	9 passi nove : corso interattivo multimediale ...	9
5	Nella terra delle tradizioni occitane / [CREL ...	6
6	Colloquial Portuguese CD-ROM : a multimedia la...	6
7	6: La spalla dolorosa : valutazione e trattame...	4
8	Musicarium 3 / CREL Centro regionale etnografi...	4
9	Musicarium 2 / [CREL Centro regionale etnograf...	4

Top titles (books) most borrowed by users of type MULTIMEDIALE

	title	num_of_patrons_that_loan_them
0	L' energia che guarisce : "meditazione sull'en...	91
1	52. Zecchino d'Oro / [canzoni eseguite dal Pic...	16
2	Tutor lingue francese : test e laboratorio vid...	11
3	Speak up : l'audiomensile per il tuo inglese	11
4	Il mio miglior nemico / un film di Carlo Verdo...	9
5	Mocrosoft Access 2010 : la "programmazione : o...	9
6	Tai chi : scopri il tai chi delle 24 forme : v...	9
7	Computer magazine	8
8	Melacanti? / [testi:] Chiara Carminati ; [musi...	8
9	Amadeus : il mensile della grande musica	7

(b)

Figure 3.11: Top books most borrowed distinct by some item type



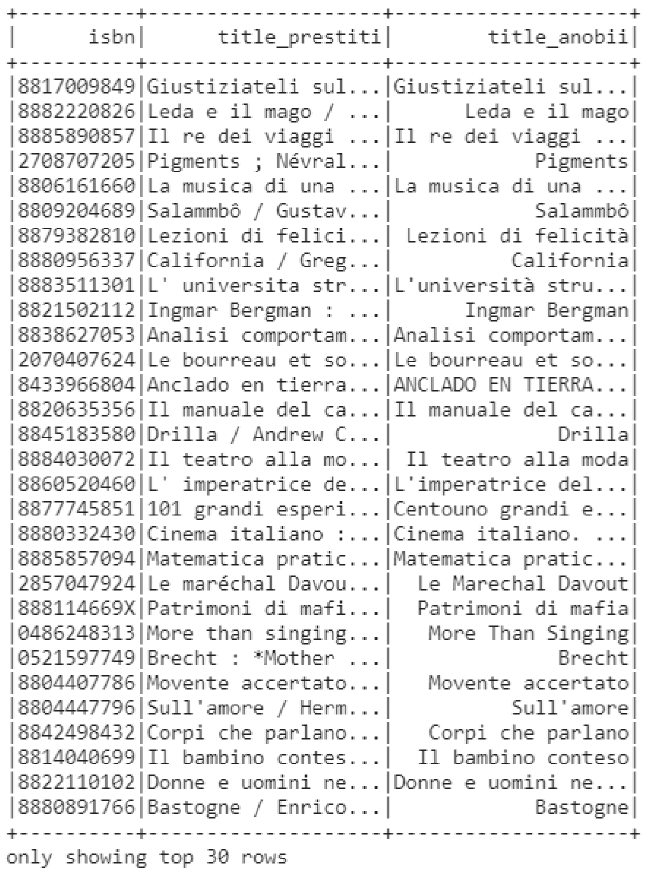
```
Top 20 titles (books) most borrowed by users
```

	title	num_of_patrons_that_loan_them
0	Topolino / Walt Disney productions	12878
1	Speak up : l'audiomensile per il tuo inglese	9359
2	Gente : settimanale di politica, attualità e c...	7415
3	Tex	5807
4	Panorama	5727
5	Oggi	5111
6	Internazionale : ogni settimana il meglio dei ...	4744
7	Dylan Dog : l'indagatore dell'incubo	4480
8	Grazia : la rivista della donna italiana	4178
9	Paperino / Walt Disney	3788
10	Vanity fair Italia	3188
11	Julia : le avventure di una criminologa / pers...	2946
12	Riza psicosomatica : rivista di medicina psico...	2617
13	Winx club: magiche avventure	2613
14	Sale & pepe : il mensile per mangiar bene	2595
15	Quattroruote : rivista mensile	2285
16	Storica	1952
17	PC professionale : guida indipendente al perso...	1882
18	Altro consumo	1802
19	Le idee di Casa mia : periodico per la casa	1795

Figure 3.12: Top 20 books most borrowed by the users

3.3 Merge between aNobii and loans of Turin libraries

After this exploratory phase of data-sets analysis, a merge was performed between the last data-set of Turin libraries loans with the previously analyzed data-set relating to aNobii. The merge was done considering the ISBN code of the books and the recommendation models were developed on the resulting data-set. The data-frame after the merge operation is composed of 94 928 samples of different titles. In the [Figure3.13] there is an extract of this resulted data-frame.



```
+----------+--------------------+--------------------+
|      isbn|      title_prestiti|         title_anobii|
+----------+--------------------+--------------------+
|8817009849|Giustiziateli sul...|Giustiziateli sul...|
|8882220826|Leda e il mago / ...|      Leda e il mago|
|8885890857|Il re dei viaggi ...|Il re dei viaggi ...|
|2708707205|Pigments ; Névral...|            Pigments|
|8806161660|La musica di una ...|La musica di una ...|
|8809204689|Salammbô / Gustav...|            Salammbô|
|8879382810|Lezioni di felici...| Lezioni di felicità|
|8880956337|California / Greg...|          California|
|8883511301|L' universita str...|L'università stru...|
|8821502112|Ingmar Bergman : ...|      Ingmar Bergman|
|8838627053|Analisi comportam...|Analisi comportam...|
|2070407624|Le bourreau et so...|Le bourreau et so...|
|8433966804|Anclado en tierra...|ANCLADO EN TIERRA...|
|8820635356|Il manuale del ca...|Il manuale del ca...|
|8845183580|Drilla / Andrew C...|              Drilla|
|8884030072|Il teatro alla mo...| Il teatro alla moda|
|8860520460|L' imperatrice de...|L'imperatrice del...|
|8877745851|101 grandi esperi...|Centouno grandi e...|
|8880332430|Cinema italiano :...|Cinema italiano. ...|
|8885857094|Matematica pratic...|Matematica pratic...|
|2857047924|Le maréchal Davou...|  Le Marechal Davout|
|888114669X|Patrimoni di mafi...|  Patrimoni di mafia|
|0486248313|More than singing...|   More Than Singing|
|0521597749|Brecht : *Mother ...|              Brecht|
|8804407786|Movente accertato...|   Movente accertato|
|8804447796|Sull'amore / Herm...|          Sull'amore|
|8842498432|Corpi che parlano...|   Corpi che parlano|
|8814040699|Il bambino contes...|  Il bambino conteso|
|8822110102|Donne e uomini ne...|Donne e uomini ne...|
|8880891766|Bastogne / Enrico...|            Bastogne|
+----------+--------------------+--------------------+
only showing top 30 rows
```

Figure 3.13: Extract of the data-frame resulting from the merge

Chapter 4

Application of literature recommendation models

Recommender system is the most effective method for information overload problems. Even if the "Cold-Start" is still an open question and has become a significant issue present in the analysis of social networks [25].

Generally, once we like a book, we subsequently go in search of similar one so that we will surely like it, perhaps asking for advice from those who also have tastes or interests similar to ours, thus influencing our possible choices.
The recommender tools mimics the human mechanism of recommending something by suggesting similar or potentially equally interesting products. The artificial intelligence offers the possibility to help the user in their choices, recommending items based on data from the past.

In this section, I will explain how I used the books data to build a recommendation engine in PySpark.
To build the recommendation system I implemented two models. The first one realized is based on Alternating Least Square(ALS) algorithm from the Spark library which is the implementation of the paper published by the Netflix competition winners [5]. The second model was built using the Python SurPRISE library.

4.1 Matrix Factorization technique

The concept of the recommendation system is based on the construction of the Matrix Factorization. In collaborative filtering approaches, Matrix Factorization is the state-of-the-art solution to the sparse data problem. This algorithms work by decomposing the user-item interaction matrix into the product of two lower dimensional rectangular matrices.
Matrix factorization model maps both users and items to a joint latent factor space, such that user-item interactions are modeled as inner products in that space [5]. The Matrix factorization is a sparse matrix where each user is a row, each item is a column and the values are the known ratings.
Each item i is associated with a vector $I_i \in \mathrm{R}^{n \times k}$, and each user u is associated with a vector $U_u \in \mathrm{R}^{m \times k}$. The resulting dot product $r_{u,i} = U_u^T \cdot I_i$ captures the interaction between user u and item i [Figure4.1].

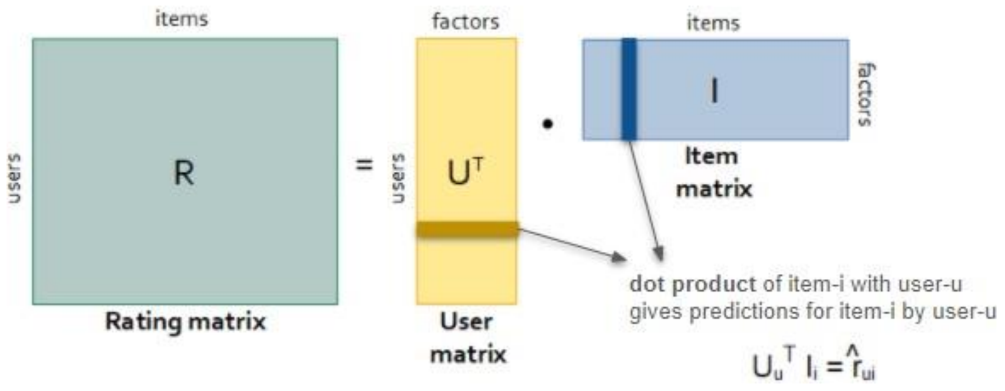

items
R
users
Rating matrix
=
factors
U^T
users
User matrix
items
I
factors
Item matrix
dot product of item-i with user-u gives predictions for item-i by user-u
$U_u^T I_i = \hat{r}_{ui}$

Figure 4.1: Matrix Factorization in recommender system

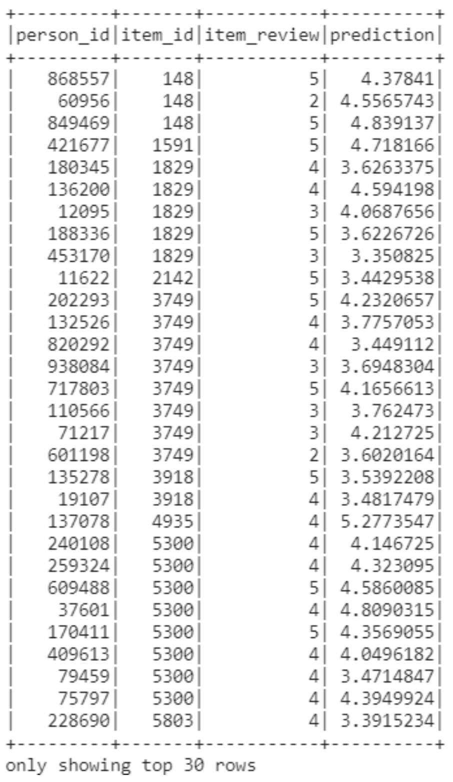
The idea behind matrix factorization is to use latent factors to represent user preferences or items in a much lower dimension space. Matrix factorization is one of very effective dimension reduction techniques in machine learning.
By multiplying the two matrices, users matrix and elements matrix, the rating matrix is reconstructed and factored in such a way as to minimize the loss between the multiplication of the two and the real rating matrix. Each user and item is projected onto a latent space. The more similar the two latent vectors are, the more correlated is the preference of the corresponding users [26].

4.1.1 Recommendation models on aNobii

For the data-set in exam, firstly I calculated the data sparsity and I obtained that the 99.9956 % is empty.
To effectively evaluate the performance of a machine learning model, the way is to use new data. Through the model evaluation phase it is determined how well the model has generalized.
In order to provide the system with new data I splitted the "*link_person _item*" table of the data-set into train, validate and test sets before moving into recommendations. The train-set is used during the training phase to allow the model to learn from data; the validate-set is employed to evaluate the performance of the model, calculating the error between the predicted and actual results; and finally the test-set is used in the testing phase to evaluate the performance of the model considered best in the previous phase. The split percentage of the data-set was 60% of train, 20% of validate and 20% of test. After that, I applied the ALS algorithm.
ALS is an user-item based association technique that solves the problems that comes out from the management of sparse matrices, very common in the field of recommendation systems where the data-sets often have several missing values [27].
Collaborative filtering is commonly employed for recommender systems. These techniques aim to fill the missing entries of a user-item association matrix.
In PySpark users and products are described by a small set of latent factors that can be used to predict missing ratings. The alternating least squares algorithm is used to learn these latent factors [28].
The implementation of the ALS model has these parameters:

1. iterations: the number of iterations to run;
2. implicitPrefs: specifies whether to use the explicit feedback ALS variant or one adapted for implicit feedback data;
3. coldStartStrategy="drop": is used when we have no data for a user that could lead to a null prediction (to make sure you do not get NaN ratings).

After defining the model, I fitted it with train-set. The input validate data are transformed in order to generate predictions [Figure4.2] applying the transform function.



```
+---------+-------+-----------+----------+
|person_id|item_id|item_review|prediction|
+---------+-------+-----------+----------+
|   868557|    148|          5|   4.37841|
|    60956|    148|          2| 4.5565743|
|   849469|    148|          5|  4.839137|
|   421677|   1591|          5|  4.718166|
|   180345|   1829|          4| 3.6263375|
|   136200|   1829|          4|  4.594198|
|    12095|   1829|          3| 4.0687656|
|   188336|   1829|          5| 3.6226726|
|   453170|   1829|          3|  3.350825|
|    11622|   2142|          5| 3.4429538|
|   202293|   3749|          5| 4.2320657|
|   132526|   3749|          4| 3.7757053|
|   820292|   3749|          4|  3.449112|
|   938084|   3749|          3| 3.6948304|
|   717803|   3749|          5| 4.1656613|
|   110566|   3749|          3|  3.762473|
|    71217|   3749|          3|  4.212725|
|   601198|   3749|          2| 3.6020164|
|   135278|   3918|          5| 3.5392208|
|    19107|   3918|          4| 3.4817479|
|   137078|   4935|          4| 5.2773547|
|   240108|   5300|          4|  4.146725|
|   259324|   5300|          4|  4.323095|
|   609488|   5300|          5| 4.5860085|
|    37601|   5300|          4| 4.8090315|
|   170411|   5300|          5| 4.3569055|
|   409613|   5300|          4| 4.0496182|
|    79459|   5300|          4| 3.4714847|
|    75797|   5300|          4| 4.3949924|
|   228690|   5803|          4| 3.3915234|
+---------+-------+-----------+----------+
only showing top 30 rows
```

Figure 4.2: ALS model prediction

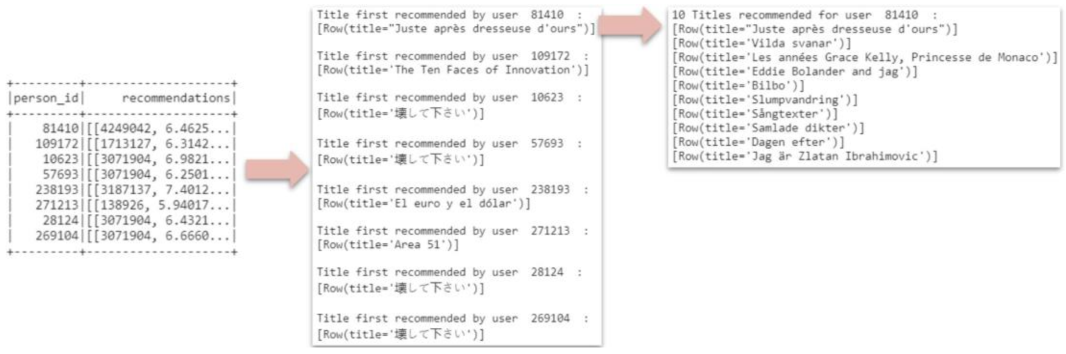
Evaluating a model is a core part of building an effective machine learning model. The statistic metric used to evaluate the model performance was the Root Mean Squared Error (RMSE). It is calculated by squaring the error between prediction and observation:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \quad (2.1.3)$$

where y_i is the observed value for the ith observation and $\hat{y}_i$ is the predicted value. This score tells how off estimated ratings are on average from the actual ratings. In the case under consideration the Root-mean-square error value obtained is 0.8395.

Like any other machine learning algorithm, ALS also has its own set of hyper-parameters that can be optimized using the cross-validation technique.
The goal is to minimize the RMSE. With the cross validation technique the model is trained for each of the k parts for avoiding overfitting issue.

The principal problem is choosing the reasonable *k* value. If I choose a value too small for *k* , I will run into the selection bias issue. On the other hand, too big value for k means overfitting, so high variance and low bias. Usually, the recommended value of *k* is between 5 and 10.
In the SparkML Engine, there are a series of parameters used to run the ALS procedure. The objective of hyper-parameter is to produce a good output [29]. To test several values for those hyper-parameters and choose the best configuration, it's common practice to define a grid of parameter associations and to run a grid search over the combinations to evaluate the resulting models and comparing their performance. It works by searching exhaustively through a specified subset of hyper-parameters [30].
For this analysis I choosed a *k* value of 5 and I fit the model. After application of cross validation, the parameters of the best model obtained were: rank 1, max iteration 20 and regularization parameter 0.1.
The root mean squared error obtained is 0.8196. This value is calculated on the difference between predicted and actual evaluations after configuring the model with the best evaluated parameters and evaluating it on the test set.
After evaluating the ALS model, the final step was computing the recommendations. The model has a "*recommendForUserSubset()*" function that allows you to generate the top N recommendations for users passed in input, the recommendation are made by observing the past behavior of this users. I can also provide recommendations for every user of the data-set with the "*recommendforAllUsers()*", although this take longer time. In the [Figure4.3] is possible to see some recommendations given after passing a subset of users as input.



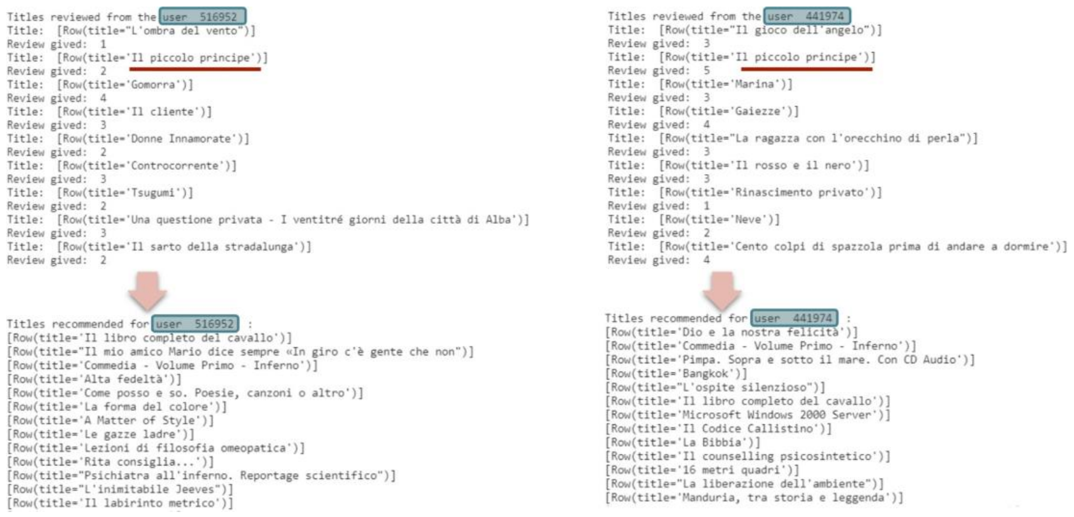
```
+---------+--------------------+
|person_id|     recommendations|
+---------+--------------------+
|    81410|[[4249042, 6.4625...|
|   109172|[[1713127, 6.3142...|
|    10623|[[3071904, 6.9821...|
|    57693|[[3071904, 6.2501...|
|   238193|[[3187137, 7.4012...|
|   271213|[[138926, 5.94017...|
|    28124|[[3071904, 6.4321...|
|   269104|[[3071904, 6.6660...|
+---------+--------------------+
```

```
Title first recommended by user  81410  :
[Row(title="Juste après dresseuse d'ours")]

Title first recommended by user  109172  :
[Row(title='The Ten Faces of Innovation')]

Title first recommended by user  10623  :
[Row(title='壊して下さい')]

Title first recommended by user  57693  :
[Row(title='壊して下さい')]

Title first recommended by user  238193  :
[Row(title='El euro y el dólar')]

Title first recommended by user  271213  :
[Row(title='Area 51')]

Title first recommended by user  28124  :
[Row(title='壊して下さい')]

Title first recommended by user  269104  :
[Row(title='壊して下さい')]
```

```
10 Titles recommended for user  81410  :
[Row(title="Juste après dresseuse d'ours")]
[Row(title='Vilda svanar')]
[Row(title='Les années Grace Kelly, Princesse de Monaco')]
[Row(title='Eddie Bolander and jag')]
[Row(title='Bilbo')]
[Row(title='Slumpvandring')]
[Row(title='Sångtexter')]
[Row(title='Samlade dikter')]
[Row(title='Dagen efter')]
[Row(title='Jag är Zlatan Ibrahimovic')]
```

Figure 4.3: ALS recommendations

In the [Figure4.4], on the other hand, it is possible to observe, in the upper part, some titles of books that the corresponding user has reviewed (with below the vote attributed to the item), while in the lower part there are some titles suggested by the system for them. In this example, both users have read and reviewed the book "*Il piccolo principe*", but they have given it two different ratings. The system for both provides almost different books, despite the book rated in common.



```
Titles reviewed from the user  516952
Title:  [Row(title="L'ombra del vento")]
Review gived:  1
Title:  [Row(title='Il piccolo principe')]
Review gived:  2
Title:  [Row(title='Gomorra')]
Review gived:  4
Title:  [Row(title='Il cliente')]
Review gived:  3
Title:  [Row(title='Donne Innamorate')]
Review gived:  2
Title:  [Row(title='Controcorrente')]
Review gived:  3
Title:  [Row(title='Tsugumi')]
Review gived:  2
Title:  [Row(title='Una questione privata - I ventitré giorni della città di Alba')]
Review gived:  3
Title:  [Row(title='Il sarto della stradalunga')]
Review gived:  2

Titles recommended for user  516952  :
[Row(title='Il libro completo del cavallo')]
[Row(title="Il mio amico Mario dice sempre «In giro c'è gente che non")]
[Row(title='Commedia - Volume Primo - Inferno')]
[Row(title='Alta fedeltà')]
[Row(title='Come posso e so. Poesie, canzoni o altro')]
[Row(title='La forma del colore')]
[Row(title='A Matter of Style')]
[Row(title='Le gazze ladre')]
[Row(title='Lezioni di filosofia omeopatica')]
[Row(title='Rita consiglia...')]
[Row(title="Psichiatra all'inferno. Reportage scientifico")]
[Row(title="L'inimitabile Jeeves")]
[Row(title='Il labirinto metrico')]
```

```
Titles reviewed from the user  441974
Title:  [Row(title="Il gioco dell'angelo")]
Review gived:  3
Title:  [Row(title='Il piccolo principe')]
Review gived:  5
Title:  [Row(title='Marina')]
Review gived:  3
Title:  [Row(title='Gaiezze')]
Review gived:  4
Title:  [Row(title="La ragazza con l'orecchino di perla")]
Review gived:  3
Title:  [Row(title='Il rosso e il nero')]
Review gived:  3
Title:  [Row(title='Rinascimento privato')]
Review gived:  1
Title:  [Row(title='Neve')]
Review gived:  2
Title:  [Row(title='Cento colpi di spazzola prima di andare a dormire')]
Review gived:  4

Titles recommended for user  441974  :
[Row(title='Dio e la nostra felicità')]
[Row(title='Commedia - Volume Primo - Inferno')]
[Row(title='Pimpa. Sopra e sotto il mare. Con CD Audio')]
[Row(title='Bangkok')]
[Row(title="L'ospite silenzioso")]
[Row(title='Il libro completo del cavallo')]
[Row(title='Microsoft Windows 2000 Server')]
[Row(title='Il Codice Callistino')]
[Row(title='La Bibbia')]
[Row(title='Il counselling psicosintetico')]
[Row(title='16 metri quadri')]
[Row(title="La liberazione dell'ambiente")]
[Row(title='Manduria, tra storia e leggenda')]
```

Figure 4.4: ALS recommendations with book in common

The second approach implemented was achieved using the SurPRISE library. Surprise [31] is a Python scikit for building and analyzing recommender systems, most of which deal with explicit rating data. The name SurPRISE stands for Simple Python RecommendatIon System Engine. It provides a collection of estimators (or prediction algorithms) for rating prediction. With SurPRISE classical algorithms are implemented such as the main similarity-based algorithms (Aggarwal & others, 2016), algorithms based on matrix factorization like SVD (Koren, Bell, & Volinsky, 2009) or NMF (Lee & Seung, 2001) and also provides different tools to evaluate, analyse and compare the algorithms performance [31].

Owing to development of Web 2.0 nowadays, the size of users and products available is growing fast, causing extreme sparsity of user rank datasets. Traditional similarity measurement methods are less successful in this situation, poor user ratings are the main reason for poor quality. To overcome this problem, techniques on collaborative filtering algorithms based on singular value decomposition are used. This approach helps to predict the rating of items for which the user has not yet been rated, and then uses the singular value decomposition method as well as Pearson similarity measurements to find the user's target neighbor, finally generating recommendation [29].

Singular Value Decomposition is a matrix decomposition by breaking down a matrix into two unitary matrices U and I, and a diagonal matrix S containing a scale factor called a singular value. The translation of the concept algorithm can be seen in equation:

$$R = U * S * I^T$$

U is the matrix containing the representations of the users in the space of latent features, I is the one containing the representations of the items in the space of latent features instead. Assuming that we have m users and n items, the R ratings matrix is $m \times n$. U $(n \times n)$ is orthogonal matrix containing eigenvectors of RR^T . $S(n \times m)$ contains ordered singular values in the diagonal. $I(m \times m)$ is orthonormal matrix containing eigenvectors of $R^T R$ [32].

By focusing on the aNobii data-set under analysis, I obtained an RMSE value equal to 0.7948 applying the Surprise SVD model.

Again I applied the cross-validation technique and I valuated the paramenters of the best model. I setted the value of kFold to 5 and exploited the GridSearchCV() function. This method calculates a score for each combination of hyperparameters on a k-fold cross validated data-set and returns the set of parameters that minimize the mean score across folds. I obtained that the best model was achieved with 20 epochs, setting a learning rate of 0.005 and a regularization of 0.4 as parameters. With the best model I obtained an RMSE equal to 0.8002.

By virtue of the large number of data it was not possible to build a *pivot_table* which is indispensable for the creation of the correlation matrix to generate and visualize possible recommendations through this approach.

Comparing the recommendation models implemented, the collaborative filtering approach with the SurPRISE library gave better performances in terms of RMSE. The evaluation on test data performs on it showed that this model estimates rating values that are more similar and less deviating from the review level that the user has actually assigned.

For a further comparison I calculated also the relative error between the best model, the approach obtained with SurPRISE, and the baseline model. This allowed me to perceive the improvement achieved by the implemented model and the naive one. The result of the comparison between the baseline model of recommendation system and the approach obtained with the best performances showed a 38% improvement of the evaluation parameter.

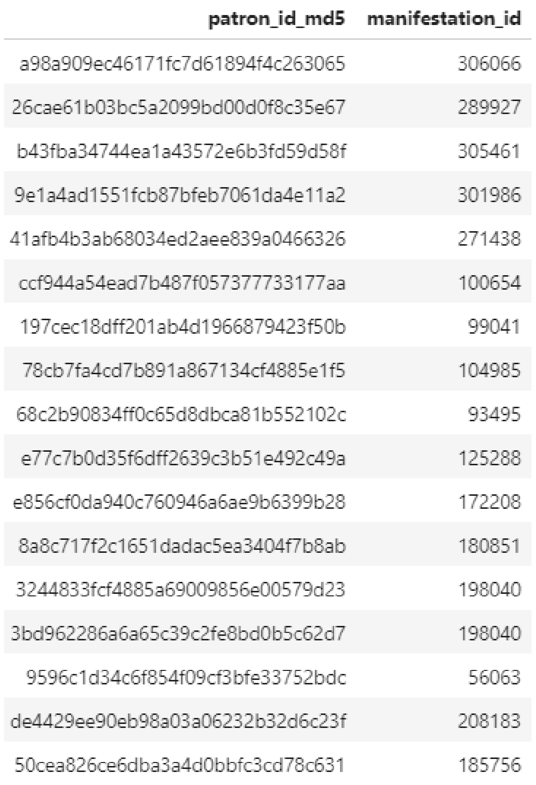
There are also other popular recommendation libraries with similar functionalities like OpenRec (Yang, Bagdasaryan, Gruenstein, Hsieh, & Estrin, 2018) and Spotlight (Kula, 2017) that support neural-network inspired algorithms; Implicit is specialized in implicit feedback recommendation, and LightFM (Kula, 2015) implements a hybrid algorithm based on matrix factorization [31].

4.1.2 Recommendation models on merge data-sets

I applied the ALS and SurPRISE models also for the data-set resulting from the merge between aNobii and the loans of the Turin libraries.
The resulting data-frame after the merge operation is composed of 94 928 samples of titles.
From the table relating to the loans made by the Turin libraries I extracted the rows relating to the loans occurred on all those common books with the aNobii data-set, books with ISBN code value in the merge. The resulting data-frame is composed of 1 351 680 samples and 2 columns: the identifier of patrons (*patron_id_md5*) and the identifier of books (*manifestation_id*) [Figure4.5]. I used these data to design and test the recommendation models.

Given the lack of the reviews factor for loan data, the recommendation system developed aims to consider this data implicitly, which means interactions not necessarily indicate user's interest it's just interaction. The only interaction between user and item, which took place through a loan, is considered as interest by that user for that particular item.

The data-set was divided into train, validate and test sets with respectively 60%, 20% and 60% as percentages. Then, I applied the ALS algorithm and I evaluated the model through the RMSE metric. Initially



patron_id_md5	manifestation_id
a98a909ec46171fc7d61894f4c263065	306066
26cae61b03bc5a2099bd00d0f8c35e67	289927
b43fba34744ea1a43572e6b3fd59d58f	305461
9e1a4ad1551fcb87bfeb7061da4e11a2	301986
41afb4b3ab68034ed2aee839a0466326	271438
ccf944a54ead7b487f057377733177aa	100654
197cec18dff201ab4d1966879423f50b	99041
78cb7fa4cd7b891a867134cf4885e1f5	104985
68c2b90834ff0c65d8dbca81b552102c	93495
e77c7b0d35f6dff2639c3b51e492c49a	125288
e856cf0da940c760946a6ae9b6399b28	172208
8a8c717f2c1651dadac5ea3404f7b8ab	180851
3244833fcf4885a69009856e00579d23	198040
3bd962286a6a65c39c2fe8bd0b5c62d7	198040
9596c1d34c6f854f09cf3bfe33752bdc	56063
de4429ee90eb98a03a06232b32d6c23f	208183
50cea826ce6dba3a4d0bbfc3cd78c631	185756

Figure 4.5: Data-frame of book loans in Turin libraries with ISBN in common with aNobii

obtaining a value equal to 0.1043 on the validation set. After the application of the cross validation technique with k set to 5 (on a best model having rank 5, regular parameter equal to 0.1 and max iterations 20 as parameters) the value of RMSE was equal to 0.1032.

After training the ALS model, it was possible to view the values of titles that the system recommends for a set of users or for a single user provided as input. An instance of some book titles that the user, identified with 593, has borrowed in the past is showed.

Some books borrowed by the user 593:

→ I Malavoglia / Giovanni Verga ; testo critico e commento di Ferruccio Cecco

→ Parker Pyne indaga / Agatha Christie ; traduzione di Grazia M. Griffini ; prefazione e postfazione di Alex R. Falzon

→ Mastro-don Gesualdo 1889 / Giovanni Verga ; in appendice l'edizione 1888 ; a cura di Giancarlo Mazzacurati

→ Masaniello : trionfo e caduta del celebre capopopolo nello sfondo della tumultuosa Napoli del Seicento / Giuseppe Campolieti
→ Il sergente nella neve : ricordi della ritirata di Russia ; e Ritorno sul Don / Mario Rigoni Stern
→ Il grande libro di Carosello : e adesso tutti a nanna... / Marco Giusti
→ Sogni di Bunker Hill / John Fante ; prefazione di Pier Vittorio Tondelli ; traduzione di Francesco Durante
→ Odore di cipria / Enzo Biagi
→ Uno che passa di qui / Julio Cortazar ; traduzione di Flaviarosa Nicoletti Rossini
→ Le vite del conte di Cagliostro / Constantin Photiadès ; traduzione di Anna Zanetello
→ Conversazione con Primo Levi / Ferdinando Camon
→ Cuore di pietra / Sebastiano Vassalli

Below it is illustrated an example of the output obtained, for the same user 593, from the recommendation system implemented with the ALS model.

Titles recommended for the user 593:
→ Istituzioni di diritto pubblico / Paolo Caretti, Ugo De Siervo
→ 7! : *Coriolano ; Alcibiade / Plutarco ; introduzione e note a Coriolano! di Maria Cesa ; introduzione e note a Alcibiade! di Luisa Prandi ; traduzione e note di entrambe le opere! di Lucia Maria Raffaelli ; con il saggio Plutarco come lo leggeva Shakespeare, di John Denton e contributi di Barbara Scardigli e Mario Manfredini
→ 2: I luoghi e le culture
→ Grecia / Mario Torelli, Theodoros Mavrojannis
→ Il concerto per pianoforte e orchestra : da Haydn a Gershwin / Piero Rattalino
→ La cittadinanza societaria / Pierpaolo Donati
→ Storia culturale dello sport / Richard D. Mandell
→ I speak Italian / [texts Carlo Mella!
→ La rivoluzione dei quanti : una nuova era nella storia della fisica / Victor Weisskopf
→ FileMaker pro 7 : corso avanzato / Roberto Celano

The second model evaluated on the merge data-set was SurPRISE. In this case, the first RMSE value obtained later the training phase was 0.0549 and next it dropped to 0.0125 with the application of the cross validation technique, again with k set to 5.
Also in this case, it was then possible to observe N top recommendations that the system provides to a group or a single user passed as input. Below

are showed some suggestions that the recommendation system has given to the user 593.

Titles recommended for the user 593:

→ La botta in testa / Tiberio Mitri ; nota introduttiva di Massimo Raffaeli

→ Tartarino di Tarascona, seguito da Tartarino sulle Alpi e Tarascona a mare / Alphonse Daudet ; traduzione di Aldo Palazzeschi ; con un saggio di Antonio Faeti

→ Perché diciamo le bugie / Gianna Schelotto

→ Calci al vento / Ezio Vendrame

→ Faccetta nera : storia della conquista dell'Impero / Arrigo Petacco

→ L' immagine al potere : vita di Giovan Lorenzo Bernini / Maurizio Fagiolo dell'Arco

→ Giuliano / Gore Vidal ; postfazione di Domenico De Masi ; traduzione di Chiara Vatteroni

→ Quattro a tre : Italia-Germania storia di una generazione che andø all'attacco e vinse (quella volta) / Nando dalla Chiesa

→ Di casa in casa, la vita : 30 racconti / Piero Chiara

→ La distrazione : romanzo / Luciano De Crescenzo

4.2 Suggestions by item-cosine similarity

In this section, I describe and show the method used to design a system that recommends items that are considered similar to particular one passed as input.
To accomplish this, I computed the pairwise cosine similarity scores for the subset of items based on their available metadata descriptions and I recommend so items based on that similarity score.

The metadata that I have taken into consideration for each item are: language, author, binding and publisher of the objects for compute the similarity on items of aNobii data-set; and language, item_media, author, publisher and mean_review of the books for compute similarity on items of loans of Turin libraries data-set.
The main problem in this case is the Natural Language Processing problem. I extracted some features from the data before calculating the possible similarity between them. To do this, it is necessary to calculate the word vectors of each tuple, row of the data-frame. Word vectors are vectorized representations of words in a document that have semantic meaning with them. I calculated the Term Frequency-Inverse Document Frequency

(TF-IDF) vectors for each tuple. The output of this operation is a matrix in which each column represents a word in the general vocabulary (all words appearing in at least one row), and each row represents a tuple [33]. The TF-IDF score is the frequency with which a word occurs in a tuple reduced by the number of tuples in which it occurs. This is done to reduce the weight of frequently occurring words in the metadata and, therefore, their meaning in the calculation of the final similarity score. Python scikit-learn [34] has a built-in TfIdfVectorizer class that produces the TF-IDF matrix [33].
Having done this, I defined a function which generates a list of the 10 most similar books taking the title of a book as input.

Below you can see an instance of aNobii items that the system suggests as similar to the book "*La pista di sabbia*".

Considering the book: “La pista di sabbia”

Top most similar books (Cosine Similarity):

→ La vampa d’agosto

→ La linea della palma

→ Sola fra le donne

→ Siracusa: città e fortificazioni

→ I peccati di Tommaseo e altri studi sulla confessione letteraria

→ Pianificazione e sviluppo nelle comunità montane del Mezzogiorno. Schema-guida di ausilio metodologico e pratico alla pianificazione finalizzata allo sviluppo economico delle comunità montane del Mezzogiorno interno - vol. 3

→ Codice delle leggi sul lavoro

Title	Language	Author	Binding	Publisher
La pista di sabbia	Italiano	Camilleri	Mass Market Paperback	Sellerio
La vampa d’agosto	Italiano	Camilleri	Paperback	Sellerio
La linea della palma	Italiano	Camilleri	Others	Bureau
Sola fra le donne	Italiano	Warner	Mass Market Paperback	Sellerio
Siracusa : città e fortifica□ioni	Italiano	Dufour	Mass Market Paperback	Sellerio
I peccati di Tommaseo e altri studi sulla conf...	Italiano	Maria	Others	Sellerio

Table 4.1: Example of cosine similarity between item of aNobii

Here, it is easy to note that the first two titles proposed are books written by the same author and furthermore the first also has the same publisher. But also the remaining suggestions appear to be linked to the book in input as we can see in the [Table4.1].

I carried out the same analysis considering only the loan books of Turin libraries and the [Table4.2] gives an idea of a possible output obtained.

Considering the book: “Mastro-don Gesualdo 1889 ; in appendice l’edizione 1888 ; a cura di Giancarlo Mazzacurati”

Top most similar books (Cosine Similarity):

→ I Malavoglia ; testo critico e commento di Ferruccio Cecco

→ Storia di una capinera ; introduzione e note di Giulio Carnazzi

→ Eros ; introduzione di Gilberto Finzi

→ Tutte le novelle ; introduzione, testo e note a cura di Carla Riccardi

→ I Malavoglia ; introduzione di Carla Riccardi

→ Due sceneggiature inedite ; a cura di Carla Riccardi

→ I Malavoglia ; a cura di Vincenzo Guarracino

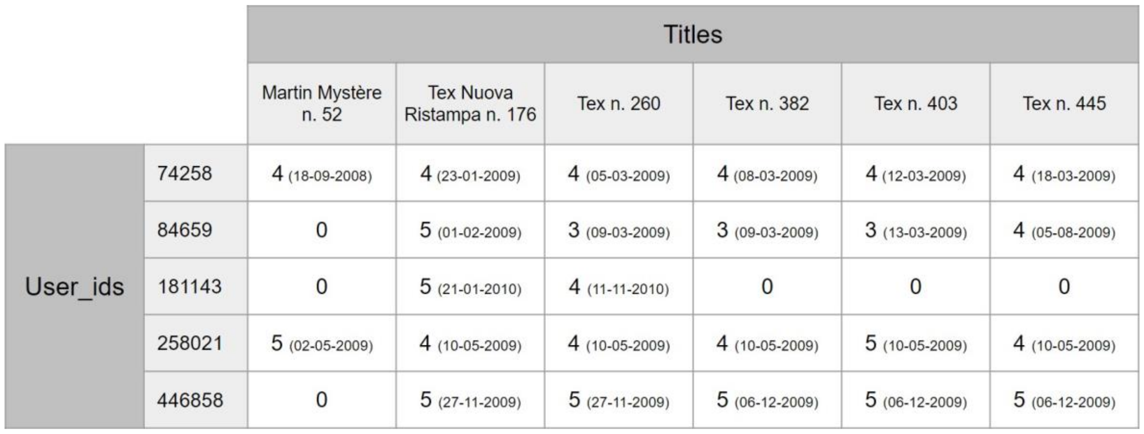
Title	Language	Item_media	Author	Publisher	mean_review
Mastro − don Gesualdo 1889...	ita	Monografia	Verga, Giovanni	Einaudi	3.0
I Malavoglia ; testo critico e...	ita	Monografia	Verga, Giovanni	Einaudi	3.0
Storia di una capinera ; introdu⍰ione e...	ita	Monografia	Verga, Giovanni	B.U.R.	3.0
Eros ; introdu⍰ione di...	ita	Monografia	Verga, Giovanni	A. Mondadori	3.5
Tutte le novelle ; introdu⍰ione...	ita	Monografia	Verga, Giovanni	A. Mondadori	3.5
I Malavoglia ; introdu⍰ione di...	ita	Monografia	Verga, Giovanni	A. Mondadori	3.5

Table 4.2: Example of cosine similarity between item of loans of Turin libraries

4.3 Dynamic item-based similarity

The discussed models are static models. In real-life, users continuously change their preferences. The dynamic item-based recommendation technique is a collaborative filtering approach that considers and takes into account the moment in which a user rate an item. This technique can be used to improve the recommendations provided by the collaborative filtering approach. To evaluate the similarity between items, these techniques identifies similar products also in terms of users that have rated them in the same time span [13]. The system takes care of the temporal effects that reflects into the dynamic, time-drifting nature of user-item interactions [5]. User tastes often change over time and products can decline in popularity. This type of approach allows to know the similarities between items and provide recommendations for the user more in agreement over time. I applied this dynamic item-based similarity approach to the reviews data from the aNobii data-set.

Below, in [Figure4.6] is displayed the *pivot_table* of an extract of the reviews. This user-item interaction matrix also contains the timestamp on which each evaluation was made, near to each rating value.



		Titles					
		Martin Mystère n. 52	Tex Nuova Ristampa n. 176	Tex n. 260	Tex n. 382	Tex n. 403	Tex n. 445
User_ids	74258	4 (18-09-2008)	4 (23-01-2009)	4 (05-03-2009)	4 (08-03-2009)	4 (12-03-2009)	4 (18-03-2009)
	84659	0	5 (01-02-2009)	3 (09-03-2009)	3 (09-03-2009)	3 (13-03-2009)	4 (05-08-2009)
	181143	0	5 (21-01-2010)	4 (11-11-2010)	0	0	0
	258021	5 (02-05-2009)	4 (10-05-2009)	4 (10-05-2009)	4 (10-05-2009)	5 (10-05-2009)	4 (10-05-2009)
	446858	0	5 (27-11-2009)	5 (27-11-2009)	5 (06-12-2009)	5 (06-12-2009)	5 (06-12-2009)

Figure 4.6: Extract of user-item interaction matrix of the system with the time span of each rating

From this matrix, it was possible to compute the similarity correlations between items. Considering the timestamp t in terms of months, I firstly defined the $f(t) = \exp^{-\alpha(t-t_{u,i})}$ [13] function as explained and mentioned in the subsection of dynamic item-based approach, I set:

$$\alpha = \frac{1}{t_{max} - t_{min}}$$

where t_{max} and t_{min} are the maximum and the minimum timestamp values of the system respectively. In this way the α value is very small and the contribution of the time period assumes an important weight in the computation. Having the minimum and maximum timestamps values equal

to *2008-09* and *2010-11* respectively in the example, the α value obtained was 0.03882.
To measure similarity between items by incorporating temporal factors, the times weight function $f(t)$ were multiplied by each rating value. Then the cosine similarity between items is calculated.
For instance, considering the title "*Tex n. 403* " and the corresponding reviews column, in array [4, 3, 0, 5, 5], the first step was to incorporate the time factors by means of $f(t)$ through the product with the review values. In this way, a new column is generated from the product between the rating column in the dataframe and each corresponding time weight function. The values of this new column are calculated as $r_{u,i}\ f_{u,i}(t)$. The $r_{u,i}$ factor means the corresponding values in the reviews column mentioned above, the review posted by user u on the i *th* item.
Subsequently, a new *pivot_table* was defined using the new column obtained after the incorporation of the time factors and thus a new user-item interaction matrix is obtained. Finally, the cosine similarity with the other items was calculated exploiting this matrix. The similarity formula thus becomes:

$$sim(i,j) = \frac{\sum_{u\in(U_i\cap U_j)}(r_{u,i}\cdot f_{u,i}(t))\cdot(r_{u,j}\cdot f_{u,j}(t))}{\sqrt{\sum_{u\in(U_i\cap U_j)}(r_{u,i}\cdot f_{u,i}(t))^2}\cdot\sqrt{\sum_{u\in(U_i\cap U_j)}(r_{u,j}\cdot f_{u,j}(t))^2}}$$

The correlation results are showed in the tables below. To provide a comparison, on the [Table4.3] the similarity values without considering the time factor in the computation are displayed; instead on the [Table 4.4] are showed the values with the incorporation of the time factor using the formula $f(t)$ mentioned above.

Title	Correlation
Tex n. 382	0.994937
Tex n. 445	0.986577
Tex n. 260	0.892607
Tex Nuova Ristampa n. 176	0.848381
Martin Mystère n. 52	0.739369

Table 4.3: Cosine similarity without the influence of the time decay factor

Title	Correlation
Tex n. 382	0.995512
Tex n. 445	0.978512
Tex Nuova Ristampa n. 176	0.795310
Tex n. 260	0.741080
Martin Mystère n. 52	0.653974

Table 4.4: Cosine similarity incorporating temporal factor

The book "*Tex n.382* " is resulted very similar to the one taken into consideration, the "*Tex n.403* ", even considering the time moments of the reviews. This is not the case of the book "*Tex n.260* ", which undergoes a decrease in similarity if we consider the time instant.
Further targeted studies would be needed to understand how much this dynamism takes place in the books' world. If you have enough data, this could be probed in an algorithmic way, evaluating the improvement of RMSE parameter for instance. Another possible way to evaluate it could be by exploiting psychological studies or through volunteers, to understand if exists a shift of interest and how often it occurs, this would also be useful for estimating the alpha value.

Chapter 5

Sentiment Analysis on aNobii

The social media comments are one of the most significant sources of text analysis in data science and machine learning. This due to the fact that the notes posted by the users are the representation of the human behavior. Sentiment analysis is thus related to Natural Language Processing (NLP) and text analysis.
The social networks contains text posted by the users that it is a vital source for analyzing their opinions and sentiments. Comments and notes hold meaningful information, but at the same time it becomes difficult to extrapolate them. Having a tool that automatically reveals opinions can allows to perceive the users' considerations on an item and identify users' interests and wishes. The study of sentiment analysis aims to capture opinions, perceive emotions and desires from text [17].
Sentiment analysis is considered as a classification problem because it identifies positive or negative opinion expressed by the users, i.e. two-classes.
Opinions are central to almost all real-life activities specially with the explosive growth of social media on the Web (for instance reviews, comments, forum discussions, blogs, posts). Users and companies make widely use of the content of these media during the process of decision making [17].
Before being able to analyze a written text, it is necessary to carry out some text cleaning and preprocessing steps in order to reduce the amount of terms present in the data. In [Figure5.1] is shown the flow of steps followed during the whole process of sentiment analysis.

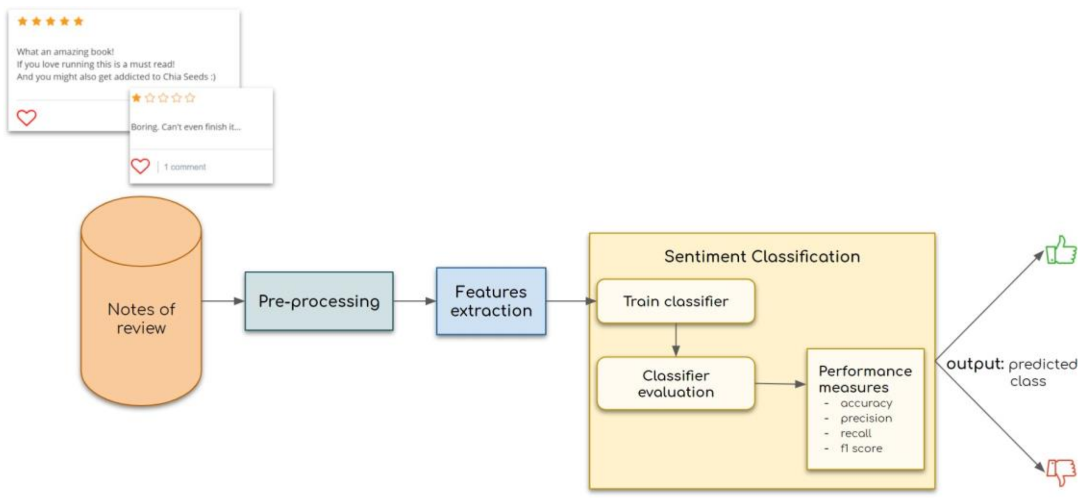


Figure 5.1: Steps of sentiment analysis

Before of the preprocessing phase I prepared the data useful for this analysis. I extract the whole list of review notes. On the available 256 551 notes, the average length of words contained in one note was four words. I explored NLP starting with NLTK (Natural Language Toolkit) package [35] in Python to perform the preprocessing phase. This is a collection of Python libraries and programs for processing the written language and to measure the frequency of words [36].
The first step is to cancel all those terms that do not add any fruitful information needed for analysis of each document (sample of a note). The preprocessing phase consists of:

- data labeling, to find the polarity of a user's note. Polarity is something that expresses the emotion of a particular sentence by using the words that compose it. This can be done using a python module, called TextBlob, which provides a function to find the level of polarity;
- convert each word into its lower case;
- strip, to remove all the trailing spaces from the notes;
- remove stopwords, the words that are repeated often in a particular language (for give you an idea, articles, pronouns, conjunctions) and which are not useful for the purposes of sentiment analysis.

This enable to have a set of notes, called corpus, easily interpretable by the algorithm.

The TextBlob [37] module is a Python library for processing textual data. This method allows to define a polarity score associated with each input note. The polarity score is a float within the range [-1.0, 1.0].
Some examples of the described preprocessing steps carried out on the notes under analysis are shown below:

Example of a note posted by a user about the book "*My Life as a Fake*" :
"An excellent book. The audiobook version was very good."
Data labeling phase → Polarity: 0.955
Lowercase and Strip phase → Note becomes: "an excellent book. the audiobook version was very good."
Remove stopwords phase → Note becomes: "excellent book audiobook version"

Example of a note posted by a user about the book "*Crash*" :
"noioso, monotono e ripetitivo. manca una storia alla base"
Data labeling phase → Polarity: -0.8
Lowercase and Strip phase → Note becomes: "noioso, monotono e ripetitivo. manca una storia alla base"
Remove stopwords phase → Note becomes: "noioso monotono ripetitivo manca storia base"

Then, it was necessary to convert the polarity score value obtained into an integer because the "*fit()*" classifier function does not accept labels with continuous values. The [Figure5.2] illustrates the amount of notes presents distinguished by polarity assigned.

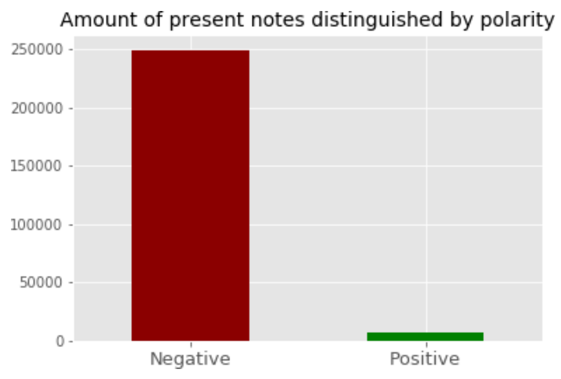


Figure 5.2: Distribution of polarity of user's notes

After the preprocessing phase, I splitted the data-set into train data and

test data using scikit learn's *train_test_split* module [34]. In order to test the accuracy of the trained model I need a test dataset. The split percentage set was 80% of train data and 20% of test data.
The next phase consists of extract functionality from text train data. I used scikit learn's CountVectorizer module, which create a vocabulary from the text data and store the occurrences of each word that appears in the text. CountVectorizer performs three basic steps [38]:

- tokenization, that involves splitting sentences in words. This is useful for the creation of a matrix having the rows corresponding to the documents (the notes of the reviews) and the columns to the words contained in them;
- build a vocabulary, containing all the words present in the document;
- encode, encodes the entire document creating a vector with the same length of the vocabulary.

The length of the encoded vector was 288 222, it means that total words in vocabulary are 288 222.
Once the text cleaning and preprocessing phase has been completed, there is the classification phase. Classification is a supervised learning approach in which the program learns from the data received in input and uses the learning to classify new data. The classification consists of two phases, a learning phase and an evaluation phase. During the first, the classifier trains his model on a given set of data; in the second, it checks the performance of the classifier. Sentiment classification is usually formulated as a two-class classification problem, positive and negative, liked and disliked. The two machine learning algorithms applied consist of Logistic Regression and Random Forest. Both models have been implemented successfully in disparate domains for classification and regression purposes [39].

Logistic Regression is a supervised machine learning algorithm in which there is a data-set labeled with the target variable. Logistic regression is a predictive analysis. It is used in binary classification. Binary because logistic regression models the probabilities of classification problems with two possible outcomes. A famous example of using logistic regression is the classification of spam. The same can be applied for other cases such as in sentiment analysis, where there are two classes to be classified, positive (1) and negative (-1) [38].
After training the model, the evaluation phase was performed. The accuracy score on the train set was 98.6%, which implies the model is predicting

98.6% accurate results, which is pretty good. Next, I also evaluated the accuracy score on the test set and the result was 97.9%, which means that the model predicts 97.9% of correct results on the test data-set.

Several studies on the use of stand-alone classifiers for comments sentiment analysis are available in the literature and the ensembles lead to more accurate classifiers [40]. The other classifier model applied is the Random Forest, an overall learning ensemble method for classification and regression which consists in the construction of a multitude of decision trees and which gives the class as output. When a sample passes through the random forest, each decision tree makes a prediction on which class the sample belongs to (in this case, negative or positive review). The final prediction is made according to a combination rule on the predictions of each individual tree. The combination rule can be majority vote or average of class predictions, for instance. It is indicated by the symbol$^{\mathbf{q}}$ in the [Figure5.3] that illustrates this concept [40].

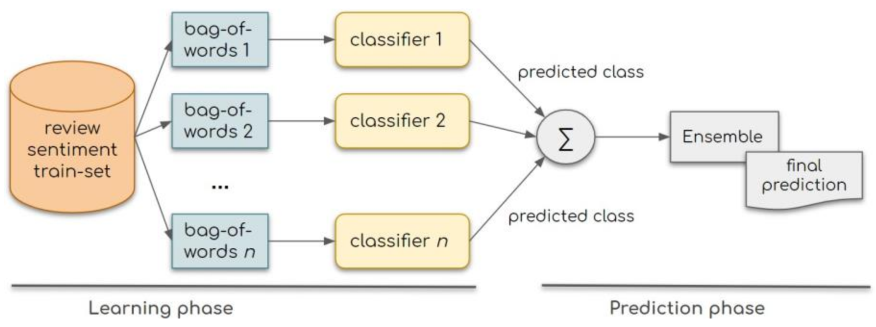


Figure 5.3: Classifier Random Forest model schema

Individual decision trees are usually prone to overfitting. A good model will fit the training data well enough to pick up on good trends, but not so well that it picks up noise. It is also essential to avoid underfitting, in which are missing out on relevant trends. In reality, it is usually difficult to perfectly balance over and underfitting. It is useful to select the best hyperparameters and select finally the model with the highest score [41]. I used the GridSearchCV of sklearn.model_selection [34] that implements a "*fit()*" and a "*score()*" methods. Cross-validation splits the training set into multiple train/test folds, 5 in my case, and train and evaluate each model. The one with the highest average score in the CV splits is then selected. In this way, I got a score of 0.9970 with the *gini* criterion and num_estimators equal to 100 as best parameters. This implies the model is predicting 99.7% accurate results. Next, I also evaluated the accuracy score on the test dataset that was 98%.

To have a practical example on the ability of the classifier to predict the positivity or negativity of a note, I used the "*predict_proba()*" scikit [34] method applied to the two classifiers. It provides as output: [probability of negative, probability of positive].
I extracted two sentences from the available notes, one positive and one negative, to give you an idea: an extracted positive sentence was "*I liked the way we could enter in people thoughts and see their different opinions about a same event.*", while a negative one "*deludente, inferiore alle aspettative, anche un po' noioso a tratti*".
I applied the vectorizer and then the '*predict_proba*' function mentioned above to observe the prediction that the Logistic Regression classifier previously trained on the reviews notes gives as output results. It gave as result [0.17544923 0.82455077] for the positive sentence and [0.9947795 0.0052205] for the negative one. This means that the selector understood both reviews correctly, giving to negative sentence a 99% chance of being negative and to positive sentence an 82% chance of being positive.

The same computation with the Random Forest classifier gave as results [0.3 0.7] for the positive sentence and [0.98 0.02] for the negative one. For the positive sentence the classifier indicates that 98% it is positive and for the negative sentence that at 70% it is negative.

After running the data-set through each model, to learn more about the performance of the models, the confusion matrix is needed. Confusion Matrix [42] is a two by two matrix, that shows the probability of predicting the correct class for both classes. In this matrix are shown:

- true positive (TP): measures the proportion of actual positives that are correctly identified;
- true negative (TN): measures the proportion of actual negatives that are correctly identified;
- false positive (FP): measures the proportion of actual positives that are not correctly identified;
- false negative (FN): measures the proportion of actual negatives that are not correctly identified.

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

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

Performance is also evaluated based on per-class evaluation metrics like precision, recall and f1 score. Precision expresses how often the prediction is correct, while recall is the classifier's ability to find all positive samples. F1 score is equal to the harmonic mean of precision and recall.
In mathematical terms:

$$precision(positive) = \frac{TP}{TP + FP}$$

$$recall(positive) = \frac{TP}{TP + FN}$$

$$f1\ score(positive) = 2 * \frac{precision * recall}{precision + recall}$$

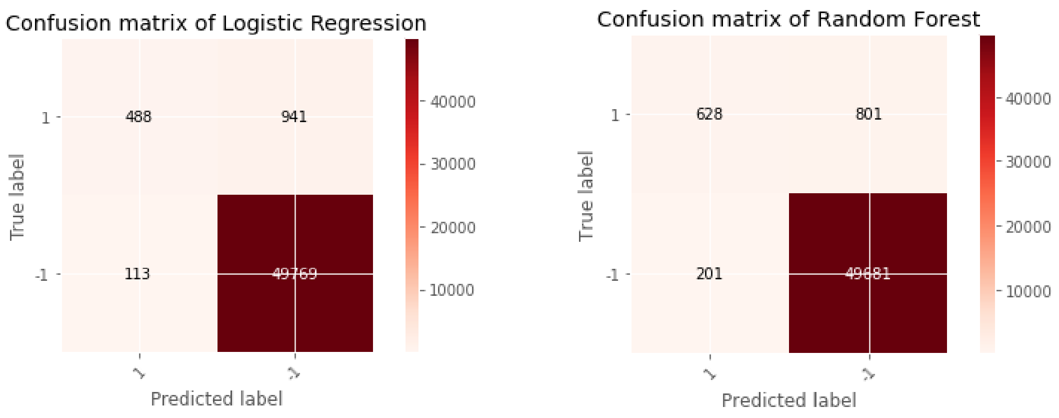
With the Logistic Regression model I obtained the confusion matrix in [Figure5.4a] and values of prediction, recall and f1 score showed in [Table 5.1]. While, with the Random Forest model I obtained the confusion matrix in [Figure5.4b] and the evaluation parameters illustrated in [Table 5.2].

	presision	recall	f1 score
-1	0.98	1	0.99
1	0.81	0.34	0.48
accuracy			0.9794

Table 5.1: Classification report of Logistic Regression

	presision	recall	f1 score
-1	0.98	1	0.99
1	0.75	0.43	0.55
accuracy			0.9805

Table 5.2: Classification report of Random Forest



(a) (b)

Figure 5.4: Confusion matrix of Logistic Regression and Random Forest models

The results were graphically represented using the Receiver Operating Characteristic curve (ROC curve) as shown in [Figure5.5].

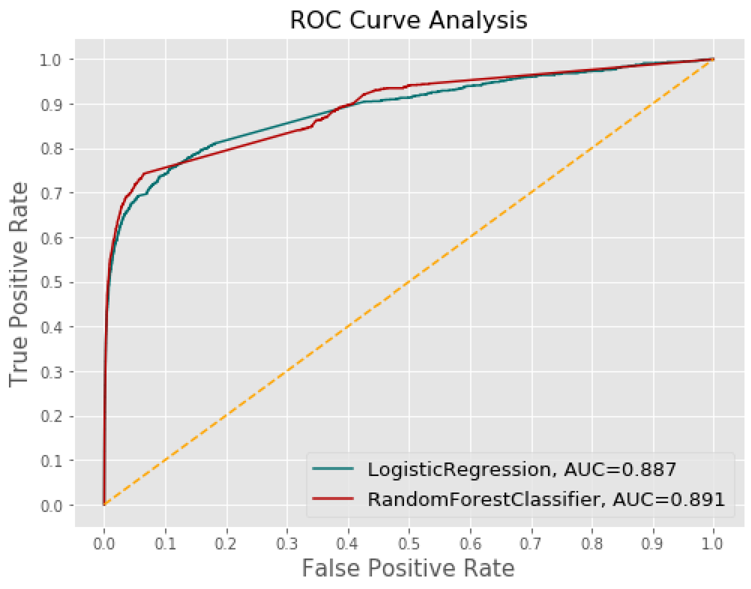


Figure 5.5: ROC curve of analyzed classification models

The ROC curve is a graph having the x and y axes that assume values from 0 to 1. The x axis represents the false positive rate and the y axis the true positive rate. The AUC (Area Under Curve) represents the ability of a model to discriminate between positive and negative classes. The orange line represents the ROC curve of a purely random classifier. A good classifier stays as far away from that line as possible (towards the upper left corner). The AUC metric is more descriptive than accuracy because it expresses a balance between the accuracy and the false positive rate [39]. By comparing the two classification models implemented it is possible to notice that the Logistic Regression classifier rises lightly than the Random Forest when False Positive Rate value assumes certain values, such as in the range between about 0.75 and 0.87; while the Random Forest reaches higher values in other range, such between about 0.37 and 0.74. However, looking at the general trend of the two curves, the Random Forest ensemble classifier is resulted slightly more accurate and it gives best performance than the Logistic Regression one.

Chapter 6

Conclusions

Currently, recommender systems cover a fundamental part in disparate realities. In a world where there is more and more growth of social media and where the user is flooded with thousands of data, the use of these recommendation techniques give a paramount plus on the system where they are implemented. Recommendation tools play a significant role on different platforms. Their inclusion into a system enables to create more user-oriented applications. The importance of recommendation systems is also that they permit a personalized and closer view to the end user. Recommendation systems have been used to mitigate and alleviate the problem of information overload by suggesting related and relevant elements to users [2]. These filtering techniques learns from the data coming from the users' behaviors and take advantage of this information to recommend them something close to what they like and near to their preferences, tastes and interests. Thanks to these supports, it becomes less tedious for the user to choose the one that matches his wishes from millions of products.

The goal of this thesis was to apply artificial intelligence techniques, in particular recommendation techniques, to the world of reading in order to flank the book and library system and to improve the quality of the advice shown to users.

The opportunity to use the vast dataset of the aNobii social network in parallel with data from Turin libraries, each containing millions of books and information, proved to be a great challenge.

This work allowed me to apply and use the knowledge on machine learning, artificial intelligence, big-data and data analysis learned during the

course of studies and in addition it allowed me to learn new ones.
This thesis gave me the opportunity to further enrich my knowledge on the field of artificial intelligence and to know this new branch of recommendation systems and sentiment analysis, techniques and machine learning tools that I had not had the opportunity to meet before.
In addition, providing practical support to both realities, aNobii and libraries, in order to better satisfy readers and thus be able to recommend the next book to be read in a personalized way and near to the interests and preferences of the individual user is an extra satisfaction.
Recommendation systems open up new opportunities to retrieve personalized information for the user. They also helps to overcome the problem of information overload which is a very common phenomenon in information retrieval systems [1].
The results of the developed analysis showed that, between the two types of collaborative filtering models, ALS and SurPRISE, the latter records best performance in terms of RMSE.
The results obtained showed a significant improvement by comparing the models implemented with the baseline model.
The global analysis carried out, especially regarding the similarity between items in the system, still reveals good outputs. I was able to see notable similarities between the items that the system suggested.
Finally, the idea of combining sentiment analysis with the recommendation system grew by studying these two tools, which I consider essential in a world dominated by a huge amount of data and information contained in them, especially if you think about the centrality of online reviews in the choices of next possible interesting products of almost all users.

Artificial intelligence techniques can therefore support the library system. Furthermore, analyze further metadata on the content of the books, for example on the plot or on the summary of it, could improve the performance of this tool even better. This can improve the quality of the suggestions shown to users and thus recommending items even more suitable to their wishes. The next step would be to really experience this recommendation engine in libraries so that we can support them by providing an additional tool that can help the readers to choose their next reading.
This could be achieved by developing a simple and appealing graphic interface integrated into the library system and environment thanks to the

aid of devices located in appropriate areas dedicated to the readers. This would allow library users to view the list of their top N books that the system would recommend them.
In addition, another tool implemented allows to generate a list of books related to a specific book requested by the user.

Bibliography

[1]F. Isinkaye, Y. Folajimi, and B. Ojokoh, “Recommendation systems: Principles, methods and evaluation,” *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261–273, 2015. [Online]. Available:https://www.sciencedirect.com/science/article/pii/S1110866515000341

[2]S. Khusro, Z. Ali, and I. Ullah, *Recommender Systems: Issues, Challenges, and Research Opportunities*, 02 2016, pp. 1179–1189.

[3]R. Burke, A. Felfernig, and M. Göker, “Recommender systems: An overview,” *Ai Magazine*, vol. 32, pp. 13–18, 09 2011.

[4]Wikipedia contributors, “Recommender system — Wikipedia, the free encyclopedia,” 2021, [Online; accessed 7-February-2021]. [Online]. Available:https://en .wikipedia.org/w/index.php?title=Recommender_system&oldid=1001354799

[5]Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, vol. 42, no. 8, pp. 30–49, Aug 2009.

[6]S. Luo, “Introduction to recommender systems.” 2018. [Online]. Available:https://towardsdatascience .com/intro-to-recommender-system-collaborative-filtering-64a238194a26

[7]M. Nandi, “Recommender systems through collaborative filtering,” *Domino*, 2017. [Online]. Available:https://blog .dominodatalab.com/recommender-systems-collaborative-filtering/

[8]B. Vintch, “What similarity metric should you use for your recommendation system?” May 11, 2020. [Online]. Available:https://medium .com/bag-of-words/what-similarity-metric-should-you-use-for-your-recommendation-system-b45eb7e6ebd0

[9]S. Lee, J. Yang, and S.-Y. Park, “Discovery of hidden similarity on collaborative filtering to overcome sparsity problem,” in *International Conference on Discovery Science*. Springer, 2004, pp. 396–402.

[10]J. S. Breese, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” *arXiv preprint arXiv:1301.7363*, 2013.

[11]L. Hassanieh, C. Abou Jaoude, J. Bou abdo, and J. Demerjian, “Similarity measures for collaborative filtering recommender systems,” 04 2018, pp. 1–5.

[12]C. Xia, X. Jiang, S. Liu, Z. Luo, and Z. Yu, “Dynamic item-based recommendation algorithm with time decay,” in *2010 Sixth International Conference on Natural Computation*, vol. 1. IEEE, 2010, pp. 242–247.

[13]D. Grönberg and O. Denesfay, “Comparison and improvement of time aware collaborative filtering techniques: Recommender systems,” 2019.

[14]B. Rocca and J. Rocca, “Introduction to recommender systems.” 2019. [Online]. Available:https://towardsdatascience .com/introduction-to-recommender-systems-6c66cf15ada

[15]P. Lops, M. de Gemmis, and G. Semeraro, *Content-based Recommender Systems: State of the Art and Trends*, 01 2011, pp. 73–105.

[16]T. Chai and R. R. Draxler, “Root mean square error (rmse) or mean absolute error (mae)? -arguments against avoiding rmse in the literature,” 2014.

[17]B. Liu, “Sentiment analysis and opinion mining,” *Synthesis lectures on human language technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[18]Wikipedia contributors, “Natural language processing — Wikipedia, the free encyclopedia,”https://en .wikipedia.org/w/index.php?title= Natural_language_processing&oldid=1007284878, 2021, [Online; accessed 23-February-2021].

[19]A. Krouska, C. Troussas, and M. Virvou, “The effect of preprocessing techniques on twitter sentiment analysis,” in *2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA)*. IEEE, 2016, pp. 1–5.

[20]PostgreSQL, “Postgresql: The world’s most advanced open source relational database.” [Online]. Available:https://www .postgresql.org/

[21]“Anobii.” [Online]. Available:https://www .anobii.com/about

[22]Arduini, Barella, and Simonelli, “Librovisioni. quando la lettura passa attraversa lo schermo,” *Torino*, vol. p. 46-48, 2009. [Online]. Available: https://editrice.effata.it/libro/9788874024810/librovisioni/

[23]Wikipedia contributors, “Anobii — Wikipedia, the free encyclopedia,” 2021, [Online; accessed 8-February-2021]. [Online]. Available:https://en .wikipedia.org/w/index.php?title=ANobii&oldid=1002276839

[24]K. Potter, H. Hagen, A. Kerren, and P. Dannenmann, “Methods for presenting statistical information: The box plot,” *Visualization of large and unstructured data sets*, vol. 4, pp. 97–106, 2006.

[25]L. Yanxiang, G. Deke, C. Fei, and C. Honghui, “User-based clustering with top-n recommendation on cold-start problem,” in *2013 third international conference on intelligent system design and engineering applications*. IEEE, 2013, pp. 1585–1589.

[26]H. S. Kung-Hsiang, “Paper review: Neural collaborative filtering explanation implementation,” 2018. [Online]. Available:https://towardsdatascience .com/paper-review-neural-collaborative-filtering-explanation-implementation-ea3e031b7f96

[27]A. Provino, “Alternating least square | als recommender system.” Marzo 26, 2020. [Online]. Available:https://andreaprovino .it/alternating-least-square/

[28]“Apache spark, collaborative filtering.” [Online]. Available:https://spark.apache.org/docs/2.2.0/ml-collaborative-filtering.html

[29]S. F. Ramadhan, “Lodging recommendations using the sparkml engine als and surprise svd,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 4, 2020. [Online]. Available:https://ejurnal.stmik-budidarma.ac.id/index.php/mib/article/view/2257

[30]J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *J. Mach. Learn. Res.*, vol. 13, no. null, p. 281–305, Feb. 2012.

[31]N. Hug, “Surprise: A python library for recommender systems,” *Journal of Open Source Software*, vol. 5, no. 52, p. 2174, 2020.

[32]K. Baker, “Singular value decomposition tutorial,” *The Ohio State University*, vol. 24, 2005.

[33]A. Sharma, “Beginner tutorial: Recommender systems in python.” datacamp, 2020. [Online]. Available:https://www .datacamp.com/community/tutorials/recommender-systems-python

[34]F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and

E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[35] “Natural language toolkit,” 2020. [Online]. Available:https://www.nltk.org/

[36] “Processing text using nlp | basics.” 2019. [Online]. Available:https://www.geeksforgeeks.org/processing-text-using-nlp-basics/?ref=rp

[37] S. Loria, “textblob documentation,” *Release 0.15*, vol. 2, 2018.

[38] R. Agrawal, “Sentiment analysis of youtube comments.” 2020. [Online]. Available:https://www .analyticssteps.com/blogs/sentiment-analysis-youtube-comments

[39] K. Kirasich, T. Smith, and B. Sadler, “Random forest vs logistic regression: binary classification for heterogeneous datasets,” *SMU Data Science Review*, vol. 1, no. 3, p. 9, 2018.

[40] N. F. Da Silva, E. R. Hruschka, and E. R. Hruschka Jr, “Tweet sentiment analysis with classifier ensembles,” *Decision Support Systems*, vol. 66, pp. 170–179, 2014.

[41] “Sentiment analysis tutorial in python: classifying reviews on movies and products.” 2018. [Online]. Available:https://www .tensorscience.com/nlp/sentiment-analysis-tutorial-in-python-classifying-reviews-on-movies-and-products

[42] Wikipedia contributors, “Confusion matrix — Wikipedia, the free encyclopedia,”https://en .wikipedia.org/w/index.php?title=Confusion_matrix&oldid=999451492, 2021, [Online; accessed 7-February-2021].