POLITECNICO DI TORINO

Corso di Laurea in ICT for Smart Societies

Master Degree Thesis

Machine learning and data mining techniques for big retailer customer sales analysis and predictions

Supervisors
Prof.ssa Baralis Elena Maria
Dott. Scinicariello Vincenzo

Candidate
Rossi Mariagrazia

Aprile 2018

mediaMENTE
consulting

POLITECNICO
DI TORINO

Contents

List of Figures

56figure.4.28 56figure.4.28 56figure.4.28 56figure.4.28

List of Tables

Introduction

With the progress and the birth of new technologies, the business scenario has undergone a drastic change. The internet has given birth to new communication channels, new technology like the RFID (Radio Frequency IDentification) gives the possibility to monitor things and humans. Customers become more conscious and their pretenses become higher. Companies have had to approach these changes: the mass market and mass production policy leave place at a new customer-centric policy.

The birth of the e-commerce brought advantages both for companies and for customers: the customer is able to choose acquiring a 360 degree feedback of the product, the companies are able to monitor the consumers and collect data about them. The new business policy requires the continuous customer monitoring, the companies strategy is to use the fidelity campaigns, so that through the fidelity card the company get in contact with the customers desires and is able to know his lifecycle. The main consequence of this change is the birth of a new strategy oriented to the customer: the Customer Relationship Management (CRM). The second is the rise of new techniques that support the extraction of information in this huge amount of data: the Data Mining and Machine Learning techniques.

One of the main pretense of the company is to extract the potential profit of a customer in the near future. Going deeper there is a continuous research by the companies giving the customer its needs and that it can be profitable for the company.

Forecast techniques are used in order to know the sales trends and recommendation techniques, such as association rules [4], are utilized. The recommendation means to apply a Cross-selling strategy that consists of proposing a service or a product to the customer looking his shopping history. The recommendation is targeted to the customers satisfaction, but at times it carries minimal or no profit for the companies. This thesis focuses on the necessity to assign an economic value with a targeted recommendation. The use case is a big retailer which holds historical data about their clients. In this use case the recommendation is trivially the "suggestion of a new product" to the customer extracted through the market basket analysis. The thesis was develop in collaboration of Mediamente Consulting s.r.l.

The thesis can be divided in three phases that will be explained better in the next chapters:

- **Association Rules extraction**. Scanning the transactional database, the main associations between the products in the market baskets were extracted as association rules.
- **Clients clustering and sales market forecast**. From the registry and the shopping behavior of the customers, algorithms of clustering have been implemented splitting clients in few clusters. Afterwards, for each cluster, daily sales have been forecasted, investigating statistic and data mining methods.
- **Rule value estimation**. An economic value to the association rules has been assigned. The estimation of the economic value was performed using a Decision tree algorithm on the clients dataset and taking into account the shopping behaviour of each cluster.

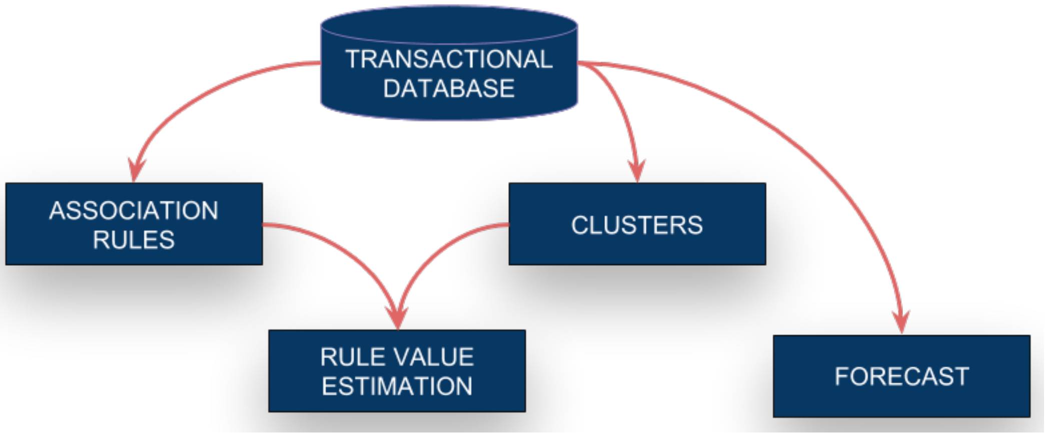


Figure 1. Schema of the architecture

The thesis is organized in the following way. *Chapter 1* is an overview of the Advanced Analytics and in particular of the use of the data mining techniques in the Customer Relationship management contest. *Chapter 2* presents the common Data mining methodology in general and a focus on the application of the method in the thesis use case. *Chapter 3* is an overview of all the algorithms exploited. *Chapter 4* presents the results of the experience.

Chapter 1

Advanced Analytics for Business

Advanced Analytics is a macro-category of technologies used to improve the performance of the traditional analytical tools such as the Business Intelligence (BI).

The companies nowadays focus on the potential of data, they own huge databases containing large amounts of data that most of the time are not analyzed due to problems in scalability, high dimensionality and heterogeneity issues [1]. The BI gives the chance to collect, scale and prepare the data, showing aspects that are not evident at first glance in order to create effective business strategies. However BI is limitated to examination only of the data.

Advanced analytics improves the analysis of the BI bringing an enrichment of the data and providing predictive analysis tools [5][6][7]. The Advanced analytics tools aim at solving many business problem, such as:

- Customer profiling and segmentation. Once that a class of similar client profiles are picked up it can be adopted for targeted marketing campaigns.
- Forecast. Predictive analysis of the sales makes it possible to adapt the future business strategy.
- Market Basket analysis. It allows and suggests products to clients or how to organize products on their shelves [7].

Advanced analytics is a mixture of tools and techniques combined with one another to analyze and gain information and predict solution of the problem. Advanced analytics stands behind data integration and data mining.

Data integration consists of the integration of the correlated data coming from various sources to gain a unified view of the information. The problem is that companies are not able to deal the huge amount of information coming from their systems. Most of the time each department of the firm use his own tools to deal or access to the information in order to reach his own aim, so the systems are overloaded of redundant and heterogeneous data. Data integration represents a solution strategy to this problem and the main approach to guarantee integration is the ETL (Extract Transform and Load) strategy. ETL is a series of tool that allows to manage the extraction, the transformation and the load of the data. The data sources are in local and here are managed, consolidated and then are cached in a database or datawarehouse. The figure 1.1 shows the ETL strategy. Firstly the data are extracted from three types of external sources: the flat file, the Enterprise resource planning (ERP), that collect all the data about the business processes of the company, and the CRM (Customer Relationship Management) database, that is a cache of all the customer information. The ETL process is in the middle between the database sources and the Datawarehouse (DWH) that consists of a different representation of the raw data in order to support the decision making. The final step is the data analysis

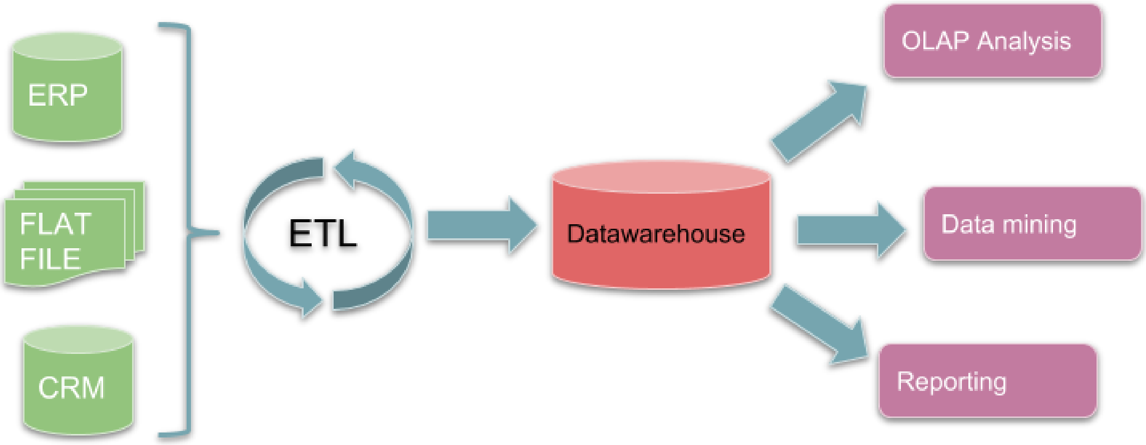


Figure 1.1. ETL strategy

that can be performed into three way:

- OLAP Analysis stands for On-line Analytical Processing (OLAP), it allows a multidimensional view of the data; from [8]: "OLAP enables end-users to perform ad hoc analysis of data in multiple dimensions, thereby providing the insight and understanding they need for better decision making"
- Reporting consists of querying the datawarehouse for providing information in a human readable way
- Data mining consists of the extraction of "patterns, associations, changes, anomalies and significant structures from data"

This thesis focus on the last part of the flow shown in figure 1.1 and in particular on the Data mining process. As before said, Data mining is an integral part of advanced analytics; it consists in discovering useful information from large repositories, it is a confluence of many disciplines. "It draws upon ideas, such as (1) sampling, estimation, and hypothesis testing from statistics and (2) search algorithms, modeling techniques, and learning theories from artificial intelligence, pattern recognition, and machine learning" [1]. In figure 1.2 is depicted a visualization of the combination of the disciplines that give rise to data mining: the database is the container of the data source, the statistical analysis allows to discover similar pattern and trend in the data and new technology like AI, pattern recognition and neural networks allow to make strategy in advance through prediction of outcomes.

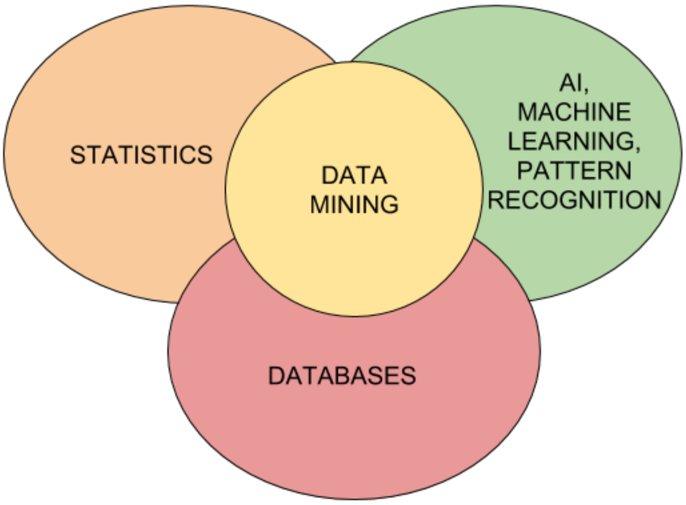


Figure 1.2. Data mining as a confluence of many disciplines [1]

Data mining born with the diffusion of the data storage on computers. The first stage data collection allows to do simplest operations to answer needs like knowing what was the revenue of the company of last month or the average total avenue of the last year. All the answer are related to the necessity on extracting information from the past. The information systems used to answer this need were the RDBMSs (Relational Database), the companies could query the system about the sales during a period and doing reporting operations.Then the diffusion of the OLAP systems gives the possibility to navigate the data and do comparison between different situations, for example to compare between daily sales of two different retailers. With the diffusion of new technology the companies are involved in a continuous competition in order to acquiring and retaining customers. This implying the necessity of predicting future trend and discovering behavioural common patterns in order to make marketing decision or policies that provides better services to customers. At this scope data mining algorithms diffuse side by side with the storage systems. Data mining algorithms for business can be differ into three macro area [9]:

- Discovery searching common behavioural patterns that are not evident, it comprehend the association and clustering algorithms.
- Prediction to forecast future trend looking at the historical data
- Anomaly detection to detect anomaly in the dataset and discover unusual trend

In this thesis techniques of data mining related to the first two area have been adopted, in particular the algorithms used are : the association and clustering algorithms, to describe behaviour, and prediction algorithms to describe possible trend.

1.1 Data mining for Customer Relationship Management

The CRM, as mentioned before in figure 1.1, has been considered only as a simple database, but behind that stands a more large and complex system made by strategy and process as well as hardware systems like the databases.

Customer Relationship Management (CRM) is the process that comprehends the strategies and decisions related to acquiring, retaining, and partnering with selective customers to create superior values for the company and the customer [10]. Today, due to the complexity and extent of the activities of each organization and with the increase of the number of organizations and variety of their services, using CRM is a necessity. Moreover the advance of the technology is the principal actor in the change of the companies strategy: Internet give rise to free information diffusion as consequence the consumer becomes well informed about the product and the services and so more conscious and their pretenses become higher. A well studied strategy can improve service quality and deepen relationships with customers, making the best out of a competitive environment while satisfying customers.

The CRM foundations are essentially all that agents that relieve companies to improve the front-end relationship with customers and the today technology offers multiple opportunity of one-to-one interactions, let's think of a web platform that allows to obtain an immediate feedback of the service or the product by the client. The main challenge is to get connect all this agents in order to integrate all the information in an unique platform able to output new strategy that increase customer loyalty and benefit for the company. In figure 1.3 there is an example of this kind of integrated platform in which all the information collected from different sectors of the company aim to unified objectives.

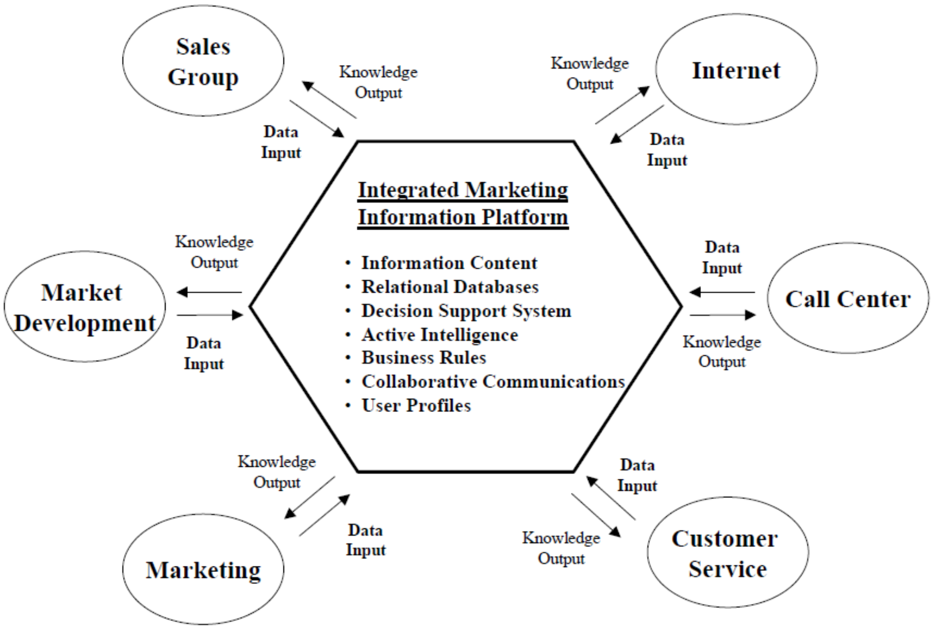


Figure 1.3. Information platform for CRM [2]

The technologies used are all services and tools, hardware and software, through which is possible to implement a CRM strategy. CRM can be divided in three main areas:

- Operational. It includes all methods for managing the direct communication with the customer during his entire lifetime. It is based on the automation of the communication channel between the front-office and the customers.
- Analytical It uses the information extracted during the operational CRM. It is based on the customer information analysis.
- Directional. It consists in the management of the planning procedures with the customers [11].

In figure 1.4 there is a global representation of the main area of the CRM previously descripted, instead in figure 1.5 there can be seen how the Operational and Analytical CRM interact to each other: the analytical CRM is the back-end part of the operational CRM. In the thesis it has been dealt only the analitycal stage of the CRM and in particular data mining techniques have been analyzed in order to propose new strategy for increasing the company profit.

Data mining tools fit in the analytical CRM in order to analyze the customers and making the firms conscious about them. The central idea of data mining for CRM is that historical data contains information that will be useful in the future. It works because customer behaviors captured are not random, but reflect the differing needs, preferences, propensities, and treatments of customers. The goal of data mining is

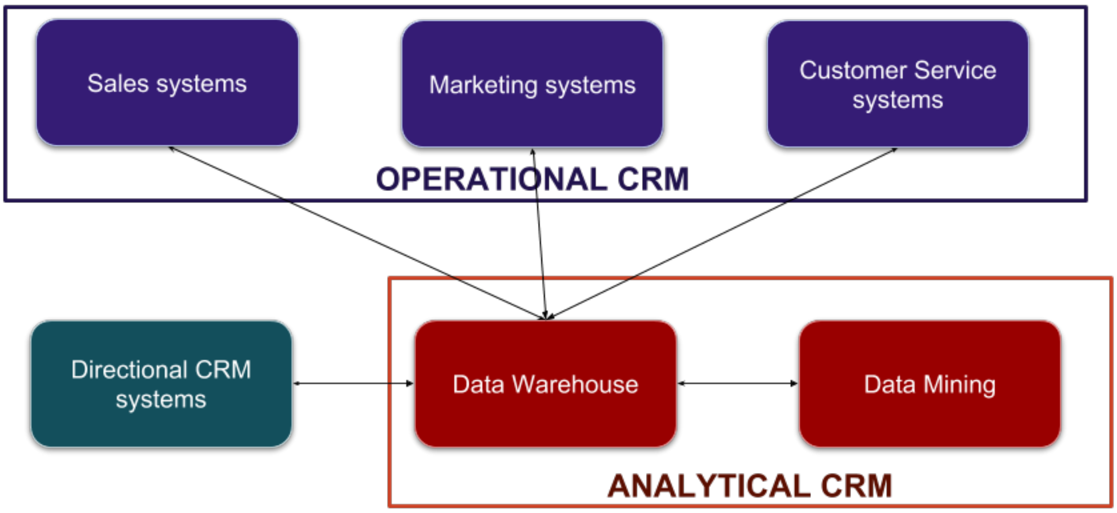


Figure 1.4. Visualization of the CRM areas

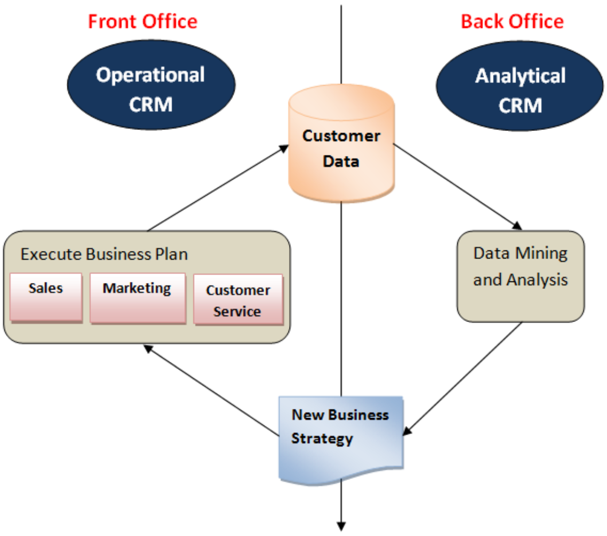


Figure 1.5. Relationship between the Operational and analytical CRM

to find patterns in historical data and build generalized model from them [12].
At the basis of the data mining in CRM there is the concept of "customer lifecycle" that refers to the description of the various stage of the relationship with the customer, so the company can understand if a new strategy is needed for increasing customer value, as example for minimizing risk of churn. There are a lot of way

to increase the customer lifecycle, Cross-selling strategy is one of those: it consists of proposing a new service or product to the customer looking at this historical behaviour. In the thesis we focus in particular on the recommendation of a new product if it correlated with the shopping habit of the customer.
The analytical CRM deals with many functionalities tools, presented in [12], devoted to analyze different problems. In this thesis only some functionalities are investigated:

- Customer segmentation. It consists in trying to identify segments of customer using clustering algorithm, starting from customer list and shopping behaviour.
- Cross-selling. It consists in selling to the customer a new product or service that brings economic profit to the company. It is referred as up-selling if the suggestion comes from past sales.
- Market Basket analysis. It concerned with analyzing the basket of the customers in order to discover correlations between products. It uses the association rules algorithms and helps the market strategy as the promotional campaigns.

Chapter 2

Data Mining Methodology

As said before the Data Mining is the result of a combination of different disciplines and from the beginning it has been applied to various frameworks. It climbed the necessity to standardize the data mining process because scientists were aware that a common line was needed. In 1996, the Cross Industry Standard Process for Data Mining (CRISP-DM) was established. The figure 2.1 shows the CRISP-DM flow: it is composed of six stages (that are not necessarily sequential). Below there is a description for each step. [13] [14][15].

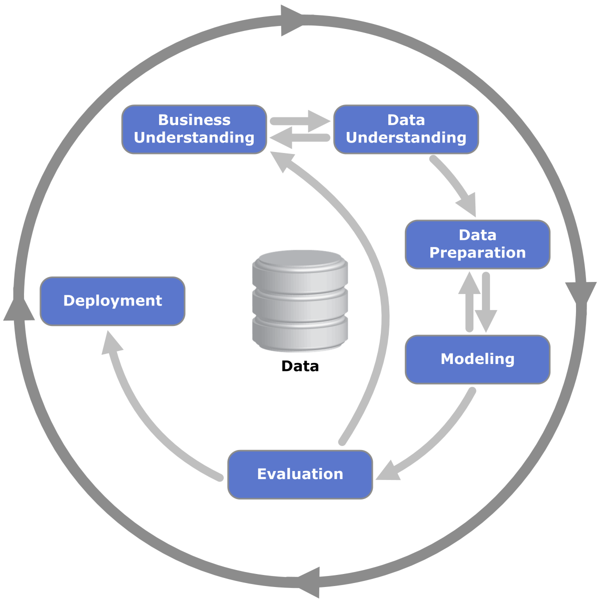


Figure 2.1. CRISP-DM process [3]

Business Understanding The initial step is the problem of understanding. Here the data scientists have to make clear the problem and try to transpose it in a data mining problem, possibly splitting the initial task in subtasks, using a divide et impera strategy. A huge number of algorithms were investigated in recent years, but only few of those address the main business problems [15].

Data Understanding The second step, strictly connected with the first, is to understand and get familiar with the data. The raw data are very noisy and some times they seem pointless, but most of the time to extract information is necessary to cross combine different data: think of a customer database and a transactional database of a company, taken singularly they provide limited information, but taken together they can be used to extrapolate their market behaviour. In these steps the key is to discover the set of data that would be interesting to investigate. In practice after this analysis usually there is a jump back to the step of formulation of businesses problem and the procedure is iterated a certain number of times [16].

Data Preparation Once that the problem is clear and the useful data are identified, the raw data must be cleaned and prepared. This is the step that requires a lot of time and the quality of the final results strictly depend on how well the data are prepared.

Some of the main aspects to deal with are: elimination of the noise, change scale and the conversion of the data in a exploitable format. Usually data must be converted from a format to another; or data must be joined, for example when dealing with sequential data it can be necessary to aggregate in weeks or months the measurements.

Moreover the raw data present outliers and missing values. A method to detect outliers is the clustering: the outliers are those values that differ significantly from the others, in general when an outlier is detected it is eliminated. The missing values are those values not recorded they can be handled or ignored, it remains a scientists choice, and much of the time one of these choices brings more benefits in the evaluation than the other[17].

Modeling In this step the data mining technique will be applied. Because each technique and data mining tools requires a specific format of the data it is necessary to return to the data preparation step. The dataset has to be divided into a training set used to build the model and a test set used to evaluate the model. It is necessary to test the model to avoid the occurrence of the *overfitting* problem. The aim of the data mining in general is to create a model that allows predicting future or unknown data; *overfitting* is the tendency of the algorithm to tailor models to the training data at the expense of generalization, this problem occurs when increasing the model complexity [15].

Evaluation and Deployment In the previous steps a certain number of techniques have been applied, here the appropriated technique is chosen, the evaluation of one method with respect to an other is based on the performance evaluation. Moreover the obtained results must be useful for the initial business problem. In the deployment stage the model is used in a real business process.

2.1 Application of the method

In the case of the big retailer analyzed in this thesis the main business problems analyzed are:

- **Discovering possible future sales** It is a typical Market Basket Analysis problem that consists in investigating the historical purchases of the customer and extracting sales information as the products that are usually purchased together and thus that are strong correlated to each other. Market Basket Analysis helps to plan the future marketing actions as the promotions; a popular algorithm designed for solving this kind of problem is the Apriori Association Rules algorithm [1] [4]: this algorithm extracts the more correlated product as association rules from the transactional database. As first to apply the input data must be cleaned: we need to delete the unregistered customer and the products that does not carry profit to the company. The rules extracted from the Apriori algorithm are of the form *Product_A* → *Product_B* meaning that

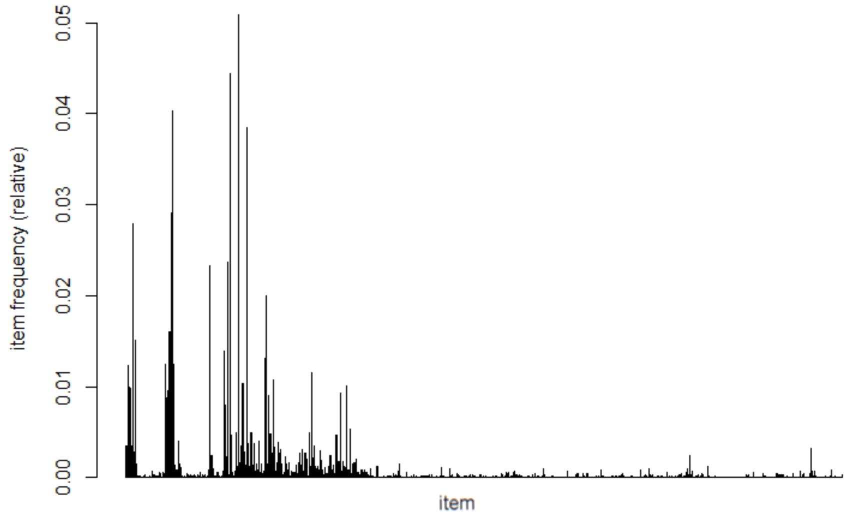


Figure 2.2. Frequency of each product in the transactional database

the two products co-occur frequently in the datasets and that the $Product\_A$ trails the $Product\_B$. The $Product\_B$ is the product to propose to the clients that have bought the $Product\_A$.
In a market basket there are products of the same commodity, for example *pizza* and *bread* belong to the *bakery products* commodity. So first of all, it must be established how to deal the product in the basket taking in to account the product category. In the transactional database the products are identified by an ID that is unique, in figure 2.2 there is a plot of the frequency of the product in the transactional database. It can be seen that few products are more frequently purchased than the others, moreover the frequency is of the order of 10^{-2}, so the apriori algorithm input parameter must be very slow unless only few rule will be extracted. Scaling the product hierarchy and grouping the products into the upper category, the frequency of the single item maybe will be increased, but the rule will be of the form $Category\_A \rightarrow Category\_B$ loosing the reference with the single product. However the final choice depends on the company needs and remain a choice of the user. For this use case the products is identified in the transaction with is ID. The Apriori algorithm requires a as input a transactional format of the type $< TransactionalID, listOfProductID >$, the algorithm doesn't require any other data pre-processing.

- **Searching for the profitable sales for the company** It consists in selecting the obtained association rules that bring economic advantage to the company. Considering the rule $Product\_A \rightarrow Product\_B$ as before, most of the time to propose the $Product\_B$ carries minimal profit for the company and it rise the necessity to rank the rules. It can be assigned an economic value to each rule and then to choose the more productive ones, but how? A possible method is to search the rule value as function of certain weighted variables, so using a sort of heuristic method. But questions rise: how to assign the weights? What are the most relevant variables? There is not a precise way to assign the weights, they are assign by a coarse consideration of the user.
  The different method used in this thesis is a probabilistic method built thought a Decision tree algorithm. As a matter of fact each customer gives a more or less large contribution to the value of the rule: some customer actives the rule frequently, others never. The probability p_i that a customer i actives the rule represents this contribution, the total economic value of the rule can be seen as $\sum_i p_i$ multiplied for the price of the rule. So the proposed method is to use the Decision tree in order to solve a binary decision problem consisting in the activation or not of the rule by the customers, also the tree returns the score probability p_i.
  For each rule it is constructed a decision tree for the customers, where the attributes are extracted from the clients list and the transactional database

and they include customers personal information, as gender, age, etc., and commercial information, as number of item purchased, average amount of the outlay, etc.. Because the dimension of information to deal is huge, *client clustering* techniques is applied to improve the performances, so other clients attributes are added related to the cluster membership.
The decision tree requires numerical and categorical format so it is not required any particular conversation of the data, but it is needed to deal with the "missing value" and the outlier. In the case of a missing of a categorical value, as *gender, education level, etc.*, the record is discarded, instead if the missing value is numerical it is tried to estimate it comparing with the cluster values. other attributes are obtained aggregating the existing ones, the aggregation allow to generate a kind of attributes called *trend attribute.* The *trend attribute* are variables able to represent better the analysis of the target.

- **Predicting the future sales trend** It is a simple estimation of future trend. At this purpose a large number of forecast techniques were born in order to solve various problems in various environments. In this thesis has been analyzed two techniques: a first classical method as Linear Predictor and a novel method as Recurrent Neural Network.
  The forecast sequence is shaped grouping the daily sales, it present a periodic behaviour that it is an advantage for the final estimation. Also here firstly customers are clustered for reducing dimensionality, then we perform prediction of the future sales for each cluster. The forecast techniques require operation of data pre-processing as the previous point, also the sequence must be scaled.

Chapter 3

Algorithms overviews

3.1 Association Rules

Association Rules mining is a technique to find associative pattern and it was introduced in [4]. It is intended to identify strong hidden rules discovered in transactional databases using some measures of interest. An example of transactional database is in Table 3.1 where a transaction is a set of items, not necessarily in order, purchased by a customer in a day identified by a TID.

TID	ITEMS
1	Bread,Milk
2	Bread, Milk, Diapers
3	Diapers, Beer, Milk
4	Beer, Diapers, Bread
5	Milk, Beer

Table 3.1. An example of transactional database

For example, from the transactions in table can be extracted an association rule of the form $Diapers \rightarrow Beer$. This rule suggests a strong relationship between *Beer* and *Diapers* items because they are jointly present in a large percentage of transactions [1].

3.1.1 Problem Definition

Let $I = i_1, i_2, ..., i_n$ be the set of n binary attributes called *items*. Let $D = d_1, d_2, ..., d_n$ be the set of all transactions. Each transaction in D has a unique transaction ID and contains a subset of the items in I.

Every rule is composed by two different sets of items, also known as itemsets, X and Y, where X is called antecedent or left-hand-side (LHS) and Y consequent or right-hand-side (RHS). Both X and Y can identify a subset of I and not only an unique item i_i, for this reason we refer to X and Y as itemsets. The graphical convetion for indicating a rule is

$$X \Rightarrow Y \tag{3.1}$$

where $X, Y \subseteq I$ and $X \cap Y = \emptyset$.
The strength of a rule is defined in term of three main components:

- *Support* it defines how popular an itemset is, it refers to the number of transaction that contain both the LHS and the RHS. Mathematically the support of the rule $X \Rightarrow Y$ is defined as:

  $$support(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \tag{3.2}$$

  where N is the number of all transactions in the database and $\sigma(X \cup Y)$ is the number of transactions in the database in which itemset X or Y are present. It is defined as:

  $$\sigma(X) = |\{d_i | X \subseteq d_i, d_i \in D\}| \tag{3.3}$$

  A low support rule is also likely to be uninteresting from a business perspective because it means that rarely the two items are bought together.
- *Confidence* measure the reliability of the inference made by a rule, or the strength of a rule, in terms of correlation between the LHS and the RHS. It is defined as:

  $$confidence(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \tag{3.4}$$

- *Lift* is defined as ratio of the observed support to that expected (if $A\&B$ were independent)

  $$lift(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X) \cdot \sigma(Y)} \tag{3.5}$$

  – $lift(X \Rightarrow Y) > 1$: So that X and Y are positively correlated, i.e. the occurrence of one implies the occurrence of the other.
  – $lift(X \Rightarrow Y) < 1$: So that the occurrence of X is negatively correlated (or discourages) with the occurrence of Y.
  – $lift(X \Rightarrow Y) = 1$: So that X and Y are independent and there is no correlation between them.

The Association Rules mining problem can be formally stated as: given a transactional database, extract the rules having

$$support \geq min\_support \text{ and } confidence \geq min\_confidence$$

A first brute-force approach is to generate all possible rules computing each support and confidence and then prune the rules that do not satisfy the minima constraints.

It is computationally expensive because the number of rules that can be extracted are exponential in the number of items. Therefore, a common strategy is to divide the problem in two subproblems [1]:

1. **Frequent Itemset Generation** find all the itemsets that satisfy the *minimum support* threshold
2. **Rule Generation** from the itemsets found in the previous step determine the rules that satisfy the *minimum confidence* threshold

In general the frequent itemset generation is more expensive in term of computation. Due to this a lot of technique have been created in order to reduce the number of candidates to be explored during this step. The figure 3.1 show an example of 4-item lattice where $2^4 = 16$ possible itemsets are generated. With n items there are 2^n possible itemsets to explore.

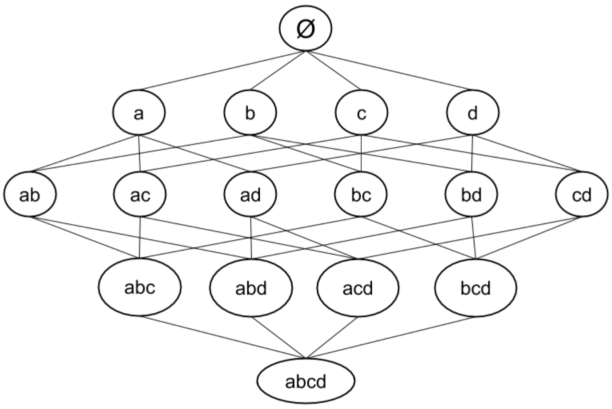


Figure 3.1. Example of itemset lattice with 4 item

3.1.2 Apriori algorithm

The Apriori algorithm goal is to find the sets of items that are purchased together frequently. It was introduced by [4] in order to create association rules for market basket analysis. It is based on the principle that:

if an itemset is frequent, then all of its subsets must also be frequent

or

if an itemset is infrequent then all of its supersets must also be infrequent

Looking at an itemset in figure 3.1 a superset is a child and a subset is a parent of it. The principle is inferred from the antimonotone property of the support:

Property 1 *if* $A \subseteq B \Rightarrow supp(A) \geq supp(B)$

In practice the infrequent itemsets are those who have support$\leq$ *minimum support*. Looking the figure 3.2, the itemset $\{c, d\}$ is infrequent. Thus, the itemsets that contain it must be discarded because for sure they will have a support less or equal than $\{c, d\}$; at the same $\{a, b, c\}$ is frequent and so its subsets are frequent too.

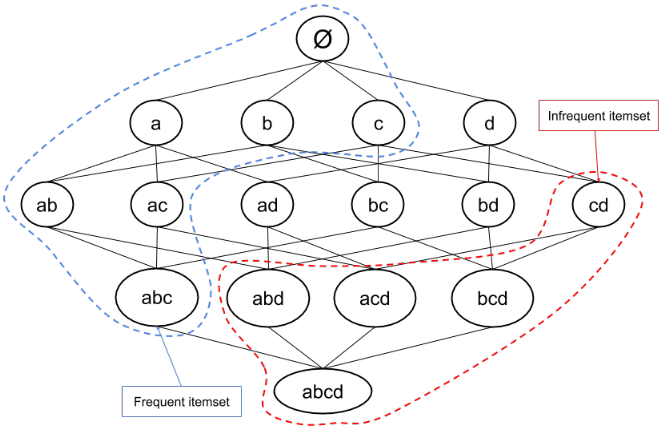


Figure 3.2. Example of itemset lattice with 4 item. The infrequent itemset are inside the red shape, the frequent itemset are inside the blue shape

The pseudocode of the Apriori algorithm is shown in Algorithm 1. It is defined as level-based approach because at each iteration itemsets of a given length are extracts. The first step of the algorithm is trivial and consists in determining the L_1 set of frequent 1-itemsets. Then the algorithm calculates iteratively the candidate (k+1)-itemsets from the previous frequent k-itemsets using the function *apriori_gen()* and then increment the support count σ of the candidate. Finally the new set of frequent k-itemsets is extracted considering the itemsets with support greater than *minsup*. The candidate generation, performed by the *apriori_gen()* function is a **join step** to join each set in frequent itemsets L_k and obtaining all the candidates for the $(k + 1)$ step. Then, after the increment of the support count, there is a **pruned step** that consists in the use of the apriori principle eliminating the candidate with a support lower than *minimum support*.

Some Definitions

- **Maximal itemset** if none of its supersets is frequent. In other words if none of the combination containing the above-mentioned itemset has a support greater than the *minsupp* it is maximal.
- **Closed itemset** if none of its supersets has the same support.

If an itemset is maximal than certainly it is closed, but not the contrary.

```
Data: $F_k$={$candidate \quad itemsets \quad of \quad size \quad k$}
Data: $L_k$={$frequent \quad itemsets \quad of \quad size \quad k$}
$L_1 = \{i|i \in I \wedge supp(\{i\}) \geq minsup\}$ [find frequent 1-itemsets]
for ($k = 1; L_k! = \emptyset; k++$) do
    $F_{k+1} = apriori\_gen(L_k)$; [generate candidate itemsets]
    for (each transaction $t \in T$) do
        $F_t = subset(F_{k+1}, t)$; [identify the candidates that belong to t]
        for (each candidates itemset $f \in F_t$) do
            $\sigma(f) = \sigma(f) + 1$; [increment support count]
        end
    end
    $L_{k+1} = \{f|f \in I \wedge \sigma(\{f\}) \geq N \times minsup\}$ [find the frequent k-itemsets]
end
Result: $\cup_k F_k$
```

Algorithm 1: Apriori algorithm

3.2 Clustering

Clustering can be defined as the process of partitioning a set of elements into a generic number of (possibly with nonzero intersection) subsets. It is an unsupervised method, since there is no knowledge of

- what the clustering mechanism should be, i.e. it is possible to use euclidean distance, entropy or many other metrics
- what is the true region of appartenence of the single element

There is not a single state of the art for the method for clustering: first there is no one fits all algorithm capable of performing a good clustering and second if also the computational cost is taken into account several choices are possible.

With practice we learn that the different clustering algorithms must be adapted to the specific use case in sinergy with a reasonable preprocessing. In order to split fairly the clients for the use case presented in this thesis, three main different clustering techniques were exploited:

- K-means. It is the more used method in literature due to its simplicity and it is based on the minimum distance criterion.
- Mini-Batch K-means. A variant of K-means that reckons on the usage of mini-batches of the origin dataset in order to improve the time of convergence.
- K-medoids It is a variant of K-means too, in which the centroids are represented by one of the dataset point.
- Agglomerative Clustering. It is the most popular hierarchical clustering technique.

3.2.1 Statement of the clustering problem

Clustering is an unsupervised method, this means that it doesn't require a training stage and there isn't a target to reach. The aim of the algorithm is to map the items into regions such that the items in a region will be similar to one another and different from the items in the other regions.

Formally the clustering problem can be stated as:

Definition 1 *Given the dataset T with $y : y \in T$, if y belongs to the region R_k then $f(y) = k$, with $k = 1,2,...,K$ that are the region tag, and*

$$R_k \cap R_j = \emptyset, \bigcup_{k=1}^{K} R_k = T \tag{3.6}$$

A decision criteria must be defined in order to map the items; different decision criteria give rise to different decision regions. The algorithms considered in this thesis use decision criteria based on distance each with his variant, so the results will be different.

3.2.2 K-means

For a better explanation it is considered the simple case of a 2-dimension problem: the items in the dataset can be represented by point like in figure 3.3, where there are 3 clusters with the centroids in black. The cluster centroids are the means of the point in the regions

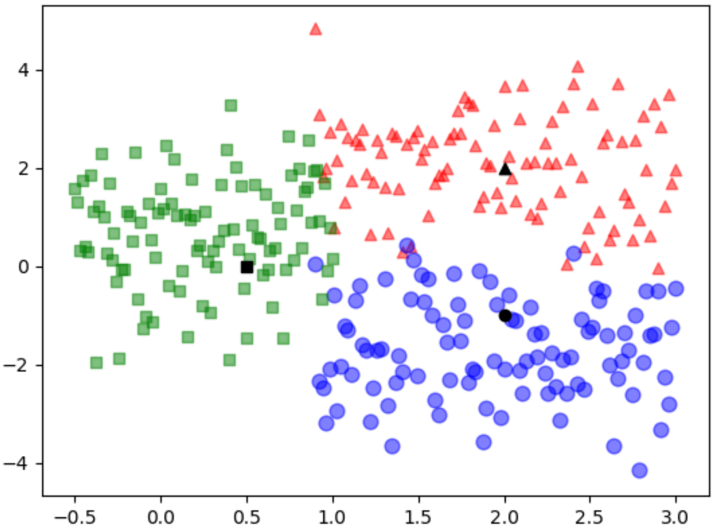


Figure 3.3. Example of clusters with centroids in black

The algorithm works as follow:

1. start from an initial centroids $x_k(0)$ with $k = 1, .., K$, K is the number of the cluster
2. **assignment step**. the point y is associated to the k region if y is closer to the $x_k(i-1)$ than the other centroids
3. **update step**. update the x_k centroids as:

$$x_k(i) = \frac{1}{N} \sum_{k \in R_k} y_k$$

4. go back to step 2 until **stop criterion**

The **stop criterion** is chosen by the user, some possible criteria would be:

- the centroids doesn't change too much: $\|x_k(i-1) - x_k(i)\| < \epsilon$
- after p iterations, etc.

Some problem of the K-means algorithm is due to the choice of the initial guess $x(0)$, it can be set randomly or fixed to known values, surely a different initialization affects

the final result. Another problem of the K-means is the sentitivity to outliers that greatly reduce the performance of the clustering method.
The version of K-means used for this thesis is provided by the scikit learn library of Python [18]. The K-means algorithm aims to choose centroids that minimize the inertia, or within-cluster sum of squared criterion:

$$\sum_{j=0}^{N} \min_{x_k \in K}(\|y_i - x_k\|^2) \tag{3.7}$$

The minimize the inertia can be identify of how internally coherent clusters are.

3.2.3 Mini-Batch K-means

Mini-Bacth K-means is a variant of K-means where at each iteration step only a random subset of the totality of the datapoints. Hereafter the algorithm description:

1. start from an initial centroids $x_k(0)$ with $k = 1, .., K$, K is the number of the cluster
2. **random subset selection step**. The set of considered datapoints for this iteration is a randomly sampled subset of y of dimension b
3. **assignment step**. the point y is associated to the k region if y is closer to the $x_k(i-1)$ than the other centroids
4. **update step**. update the x_k centroids as:

$$x_k(i) = \frac{1}{N} \sum_{k \in R_k} y_k$$

5. go back to step 2 until **stop criterion**

3.2.4 K-medoids

K-medoids is a variant of K-means where the centroids (called in this case medoids) are forced to be one of the observed datapoints. This clustering techinique have mainly two advantages with respect to k-means:

- It is more robust to outliers: since we force the centroids to be in one of the datapoint the presence of a (few) outlier possibly do not distort at all the clustering results
- It has better convergence properties with respect to classical k-means, especially for more exotic distance measures

However one drawback is that k-medoids is computationally heavier than k-means.
The brief description of the algorithm is the following:

1. start from an initial medoids set $m_k(0)$ with $k = 1, .., K$, K is the number of the cluster chosen from the available datapoints

2. **assignment step**. the point y is associated to the k region if y is closer to the $m_k(i-1)$ than the other medoids
3. **update step**. update the m_k medoids as:

$$m_k(i) = argmin_{h \in \{y_1, y_2, \ldots\}} \sum_{k \in R_k} d(y_k, h)$$

4. go back to step 2 until **stop criterion**

3.2.5 Agglomerative Clustering

Agglomerative clustering is the only hierarchical clustering technique considered. It produced a binary tree called dendrogram which display the cluster and subcluster relationship [1]. An advantage of this method is that it must not be defined a number of desired cluster K, but it is sufficient to cut the tree at a certain depth. The cluster are agglomerate according to the rule:

"agglomerate the 2 objects at minimum distance"

where an object can be a single datapoint or an agglomerate of datapoints. The algorithm works as follow:

1. start with all N datapoint
2. evaluate the distance matrix between datapoint
3. **agglomerate** the objects with the minimum distance
4. **update** the distance matrix
5. go to 3 until it remains one object

The key of the method is how to define the distance not only between the datapoints but also between the clusters called linkage.

For the distance between the datapoints x and y, the common measures used are:

- Euclidean: $d = (x-y)^2$
- Manhattan: $d = |x-y|$
- Minkowski: $d = [(x-y)^p]^{1/p}$

For the linkage between clusters C_i and C_j and defining $x \in C_i$, $y \in C_j$, the measures are:

- Maximum linkage: $D = max\{d(x,y)\}$
- Minimum linkage: $D = min\{d(x,y)\}$
- Mean linkage: $D = \frac{1}{|A\dot{B}|} \sum d(x,y)$

3.3 Forecast

After the creation of the clients clusters there will be an attempt to estimate the cluster future expenditure. Forecast is a common strategy to predict future data looking at the historical past data, the future trend is predict as function of the past data. Unlike clustering, forecast is a supervised method: there is a training stage during which a model is created using a training dataset and then a test stage during which the model is tested and validated. A disadvantage of the unsupervised methods is the $overfitting$ problem that consists in the development of a model that tailor the training set perfectly as results it goes wrong in the testing phase. In literature there are a variety of techniques use for forecast from the classical statistical analysis to the machine learning technique, they all are under a category of statistic named predictive analytics.

In this thesis two techniques are investigated:

- Recurrent Neural Network (RNN) [19], an effective type of neural network designed to handle dependent sequential inputs. In particular the focus on the Long-Short Term memory network that is a type of RNN able to handle long-term dependencies, as dependencies between point that are far apart; in fact common RNN suffer of the vanished gradient problem.
- Linear Predictor, is a classical statistical method based on the estimation of the future trend as a linear function of the previous data.

3.3.1 Recurrent Neural Network

Recurrent Neural Network (RNN) is a class of the Artificial Neural Network where the inputs (and the outputs) are dependent from each other as in the case of temporal dependent input.

In classic Neural Networks the main consideration is that all the datapoints are not correlated to each other: given an input X it tries to predict the desired output Y minimizing a certain loss function previously defined. A neural network can have one or multiple layer. In figure 3.4 there are two representations of a 1-layer network: in each layer the inputs are multiplied by weights parameter w_i and a bias b_i is added, then the results are passed to a non linear function $z()$, as a $tanh$ or $sigmoid$, called activation function [20]. The equation that dominates the network is:

$$Y = f(WX + B) = h(X) \tag{3.8}$$

In RNN the information of the previous input persist, this allows it to exhibit dynamic temporal behaviour. An example of RNN can be seen in figure 3.5, the cycle in the hidden layer represents the persistence of the previous information. To have a better understanding, we can have a text mining example: given a sentence

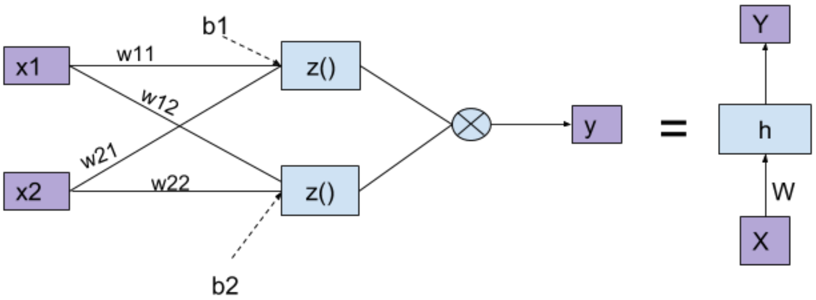


Figure 3.4. A 1-layer Neural Network example

in input, we want to predict the next word. The single input $x(t)$ is a word of the sentence, to estimate the next word $y(t)$ it must be considered the current and previous read words. This gives the possibility of considering the context of the sentence. The black box in figure can be thought as a memory of the previous steps that at each time is aggregated with the next step input [21] [20]. The hidden node h is here called state node or state unit and is a function of the previous and current state. A definition of the state h at the t time is:

$$h_t = f(Wh_{t-1} + x(t)) \tag{3.9}$$

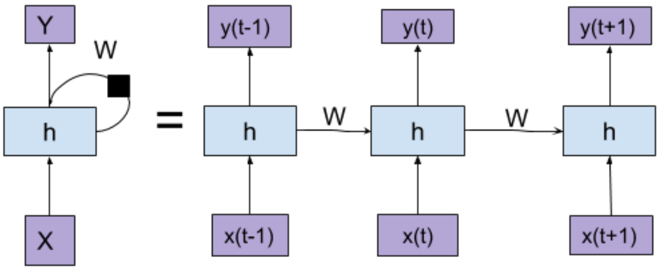


Figure 3.5. A folded recurrent neural network and its unfolded version

The RNNs are used for dealing with sequential information. Examples of RNN usage as generation of images are in [22] [23], text generation like the example done before are applied in [24], in [25] the RNN was used to support GPS and detect the next position in the map. In this thesis the RNN are used to forecast the daily profit of a company, focusing on particular category of customer.
We have to consider a limitation of the RNN explained in [26]: the longer is the depth the harder is the training. A version of RNN capable to overcome this issues is the Long short term memory network proposed in [27].

Long Short-Term Memory Network

The Long Short-Term Memory Network (LSTM) is a variant of the RNN, in which the ”memory” previous described is sufficiently large so that a long sequence of the previous information can be carried.
The base of the LSTM network is the LSTM cell figured out in figure 3.6. In addition to the external recurrence loop, the LSTM cell have a self-loop with a weight governed by a gate called *forget gate*.
In the figure 3.6 the $h(t)$ signal is called state unit, it contains the information that is carried in all the network cells. The strategy is to take or not the information carried by $h(t)$ using a *sigmoid* function which outputs are {0,1} where 0 means ”carry nothing” and 1 ”carry everything”. There are three main gates inside the LSTM cell [21] [20]:

- **forget gate** it is used to has make the decision about carrying or not the information: looking at the input sequence $x(t)$ and at the previous output $y(t-1)$ it returns a 0 or a 1 for each number in the sequence $h(t-1)$. As said the function that performs the decision is the *sigmoid*:

$$f_t = \sigma(W_f[y(t-1); x(t)] + b_f) \tag{3.10}$$

  where W_f, b_f are the forget gate parameters
- **input gate** it updates the $h(t-1)$ state; it is composed by a *sigmoid* and a *tanh* function:

$$i_t = (\sigma(W_s[y(t-1); x(t)] + b_s)) * (\tanh(W_C[y(t-1); x(t)] + b_C)) \tag{3.11}$$

  where also here the W_i, b_i are the gate parameters. So the state unit is updated as:

$$h(t) = f_t * h(t-1) + i_t \tag{3.12}$$

- **output gate** computes the output of the cell as:

$$y(t) = (\sigma(W_o[y(t-1); x(t)] + b_o)) * \tanh(h(t)) \tag{3.13}$$

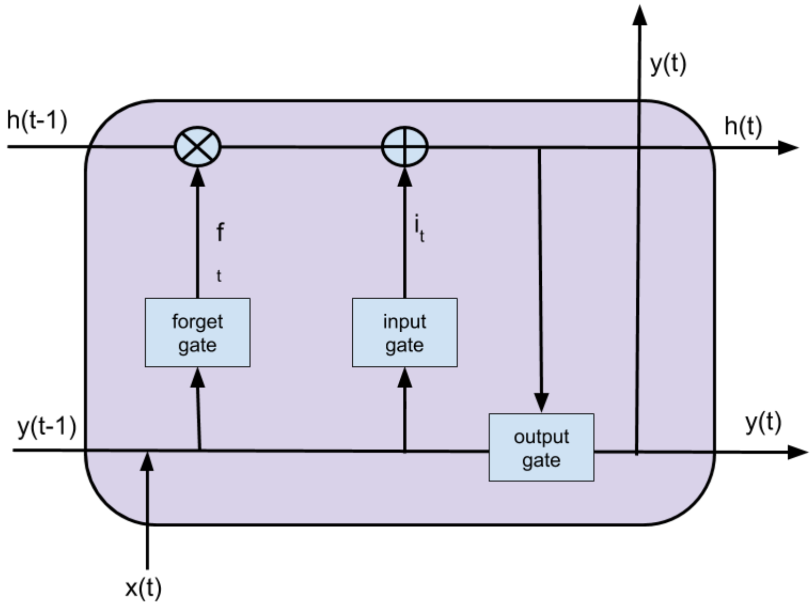


Figure 3.6. LSTM cell

3.3.2 Linear Predictor

Given a discrete time random process $X[n]$ with $n = 1, .., N-1$ the linear predictor of order P allows to predict the future sample $X[n+1]$ using a linear combination of the previous P samples [28]:

$$\hat{x}[n+1] = \sum_{p=0}^{P-1} h_p x[n-p] \tag{3.14}$$

The above equation can be rewritten in matrix form as

$$\hat{x}[n+1] = \mathbf{h}^T \mathbf{x}_p \tag{3.15}$$

where

$$\mathbf{x}_p = [x[n-P+1], x[n-P+2], ..., x[n-1], x[n]]^T \tag{3.16}$$

The linear predictor coefficients **h** are obtained using as a cost function the Mean Square Error, or, in equations

$$\mathbf{h} = \arg\min C = E(||\hat{x}[n+1] - x[n+1]||^2) = E(||\mathbf{h}^T \mathbf{x}_p - x[n+1]||^2) \tag{3.17}$$

The cost function can be expanded as

$$C = E(||x[n+1]||^2) + E(||\mathbf{h}^T \mathbf{x}_p||^2) - 2E(\mathbf{h}^T \mathbf{x}_p x[n+1]) \tag{3.18}$$

The first term does not impact the minimization process and the other two terms can be expanded and used in a new, equivalent cost function as

$$C_1 = \mathbf{h}^T E(\mathbf{x}_p \mathbf{x}_p^T)\mathbf{h} - 2\mathbf{h}E(\mathbf{x}_p x[n+1]) \tag{3.19}$$

We notice that the content of $E(\mathbf{x}_p \mathbf{x}_p^T)$ is the autocorrelation matrix,

$$\mathbf{R_x} = E(\mathbf{x}_p \mathbf{x}_p^T) = \begin{bmatrix} R_x[0] & ,R_x[1], & ..., & R_x[P-1] \\ R_x[1], & R_x[0], & R_x[1], & ... \\ ... & & & \\ ... & & & \end{bmatrix} \tag{3.20}$$

where $R_x[n]$ is the autocorrelation function,and that the vector $E(\mathbf{x}_p x[n+1])$ is equal to

$$\mathbf{r_d} = E(\mathbf{x}_p x[n+1]) = [R_x[1], R_x[2], ..., R_x[P]]^T \tag{3.21}$$

Without going into the derivation details the optimal vector $\mathbf{h}$ is determined as

$$\mathbf{h} = \mathbf{R_x}^{-1}\mathbf{r_d} \tag{3.22}$$

The pseudocode for the algorithm is the following

- *Training phase* Estimate the autocorrelation $R_x[n]$ for $n = \{0, ..., P\}$. Use the values estimated for computing $\mathbf{R_x}$ and $\mathbf{r_d}$. Compute $\mathbf{h}$.
- *Prediction phase* At time instant n, the considered window is equal to $\mathbf{x}_p = [x[n-P+1], x[n-P+2], ..., x[n-1], x[n]]^T$. Predict the next value as $\hat{x}[n+1] = \mathbf{h}^T\mathbf{x}_p$.
- *Autocorrelation update (optional)* Refine the estimate of the autocorrelation to keep track of eventual changing in the statistical properties of the process

The linear predictor is a powerful method in that combines a computationally light training algorithm with good statistical prediction performances.

3.4 Decision tree

A decision tree is a widely used classification method. A decision tree consists in building a tree in which a non-leaf node represent a question to a subsets of attributes, an edge corresponds to an outcome of a test performed on the subset, and a leaf node corresponds to a class prediction [29] [1].

More specifically, the total set of attributes T is partitioned into non-overlapping subsets S_i, i.e. $T = \bigcup_i S_i$. Each of these subset in a decision tree represents a node. The aim of the algorithm is to

1. find a meaningful partitioning of the attribute set T
2. find meaningful functions f_i to be applied to the various subsets for deciding how to explore the tree, i.e. the decision d is equal to $d = f_i(S_i)$

Hopefully the following pseudocode will make the comprehension of how a decision tree works clearer:

1. start from the root node (corresponding to S_0)
2. **Decision step**. the current decision d, that can take values "descend right" or "descend-left" is calculated on the current node as $d = f_i(S_i)$. From the current node descend along the edge to the node corresponding to the decision.
3. **Termination control**. If the current node is a leaf a final decision has been taken, return to the user the final decision (with according confidence probability) and exit. Otherwise go to step 2.

We will see in greater detail in the following subsection how the subsets and the decision function are determined according to an information theoretic criterion.

Unlike clustering, a decision tree technique is a supervised method. There is first a training phase in which the model is trained, starting from know values, and a testing phase that consist in the application of that model to a new dataset.

In this thesis the decision tree is used to solve a binary decision problem related to the activation or not of a rule by the client. The set of attributes are:

- client personal information, such as the age, gender, marital status, number of children,...
- client commercial information, such as the list of acquired items, the date of the last acquisition, the registration date to the commercial service...
- the cluster of appartenence (this point will be explained in gretaer detail in results section)

and the target is a set of binary variables: *yes* if the client actives the rule, *no* if the rule is not activated (for each rule a different tree is built).

The target scores probability s_r is defined as the probability the customer will activate the rule. It can be used for two main scopes:

- suggest a new product only to the clients with whose probability is higher than a given threshold
- estimate the potential gain of the rule when proposed to the particular client

as $E(gain) = s_r c_r$ where c_r is the price of the sold item.
As a side note, in this thesis, the algorithm only determines the decision rules f_i while the subsets S_i are pre-determined by the user.

3.4.1 C 4.5

In this thesis, the decision tree algorithm investigated is the $C4.5$, an algorithm created by Ross Quinlan [30]. It is based on the entropy concept of a random variable X defined as:

$$H(X) = -\sum_i P(X = i) \log_2 P(X = i) \tag{3.23}$$

where $p_i = P(X = i)$ is the probability that the variable X takes the value i. An important related concept is the mutual information (called information gain by the author of the algorithm), defined as:

$$I(X;Y) = H(X) - H(X|Y) \tag{3.24}$$

where $H(X|Y)$ is the conditional entropy of the random variable X given the random variable Y:

$$H(X|Y) = -\sum_{i,j} P(X = i, Y = j) \log_2 P(X = i|Y = j) \tag{3.25}$$

Qualitatively, the mutual information can be understood as: how much information do we gather about a random variable Y when we observe X?. In our case the random variable Y is the binary variable that determines if the client satisfies the rule, while the variable X is one of the subsets S_i (in the following we call S_i as a variable).
Without going into technical details, we can present the pseudocode of the algorithm:

1. **Initialization step**. All the variables S_i are marked as unused
2. From the set of unused variables, select the one that maximize the information gain:
$$\tilde{S}_i = \arg\max_i I(Y; S_i) \tag{3.26}$$
The selected variable is assigned to the current node
3. For the selected variable, if it is non-binary, select a threshold τ such that, if a new random variable from S_i is created as
$$L_i = \begin{cases} 0 & if \quad S_i \leq \tau \\ 1 & otherwise \end{cases} \tag{3.27}$$

then, τ is the value that maximize

$$I(Y; L_i(\tau)) \tag{3.28}$$

The variable S_i is marked as used and the tree is augmented with two new nodes. The decision function is:
if S_i is binary: "is $S_i = 0$?,then go to left, otherwise go to right"
if S_i is non binary "is $S_i < \tau$?", then go to left, otherwise go to right"

4. Explore the first unvisited children. If no unvisited children are present the algorithm has terminated.
5. Is the current conditional probability on the node above a certain threshold? If yes, the current node is a decision node and go back to 4. If not go to 2 repeat the procedure on the subset of the new node

The author of the algorithm suggests that instead of maximixing the mutual information it is possible to maximize the normalized mutual information, defined as

$$I_{norm}(X; Y) = \frac{I(X; Y)}{H(X)} \tag{3.29}$$

Notice that the above pseudocode is a slight simplification of the actual algorithm: the tree can be non binary and categorical variables with cardinality higher than 2 can be used. However from a conceptual point of view nothing is different.

Chapter 4

Results

In this chapter the results are presented. The objective of the work was to predict the customer behaviour at two level of granularity. The first, the single client behaviour analysis, was predicted by using a combination of results of association rules algorithm and C4.5 classification tree: namely given information about the client what is the probability that the considered customer satisfies a given association rule? Once this probability is quantified the expected monetary value of the rule for the single customer can be derived by multiplying the probability by the average rule price. The second level of granularity considered is the set of customers as a whole, clients are divided into clusters and two information can be extracted: first the daily sales on a cluster base forecast and second the predicted value of a given rule on a cluster base.

In section 4.1 the association rule evaluation is exposed, outlining the data preprocessing procedure. From the evaluation 18 rules was extracted and than for each of that a decision tree algorithm was performed as in section 4.3. In parallel, in order to reduce the dimensionality of the problem, it is adopted a clustering technique to group the client with the same attitude. In section 4.2 the clustering techniques, before mentioned in chapter 4, were evaluated in performance in order to extract the best one. The section 4.3 exposes the evaluation of the monetary value of the rule according to the proposed probabilistic approach based on the Decision Tree. In the section 4.4 the forecast results of the LSTM network and the LP were presented and also some performance consideration was done, finally here it is attempted to predict the expected profit of a rule forecasting the daily cluster transactions and joining with the price of the rule.

4.1 Association Rules results

The algorithm used for the rules generation is the Apriori algorithm performed in the R software environment [31]. As previously explained the algorithm requires as

input a minimum confidence and support, so the rules with parameter lower than the minima required were pruned by the algorithm.
The input database is a collection of the last year transactions of some retailers situated in a predefined zone of the city. The choice derives from the assumption that usually customer shops in various supermarkets situated in the same area.
The provided dataset contains the transactions of both signed up and not signed up clients, so at the beginning it is needed some pre-processing in order to discard the not signed up clients denoted with an ID equal to 1. At first glance, wrongly, it was assumed that unregistered clients could not influence the evaluation of the association rules. However, since a portion of the products that are purchased the most are acquired by unregistered customers (such as coffee at the bar), if the unregistred customers are not removed the Apriori algorithm extract mainly the rules that are applied by unregistered customers. From a practical point of view, the elimination of the unregistered client was thus needed. Another cleaning operation in the dataset is the elimination of all the service products like shoppers and products used by the staff of the retailer. Moreover seasonal products have been discarded, as the Christmas or the Easter products, because they are in shelf only for a short period.
To select reasonable values as input, firstly some considerations were done. In table 4.1 statistics about the frequency of the single item in the dataset are presented. It stands out that the items in general have a low frequency: the maximum value is about 0.004, it can be deduced from the antimonotone property of the support, presented in Section 3.1.2, that the *itemsets* will have at most a support with the max value in table.

Minimum	$3.030e^{-06}$
Maximum	$4.549e^{-02}$
1st. Quantile	$1.213e^{-05}$
Median	$3.334e^{-05}$
Mean	2.382^{-04}
3th. Quantile	$1e^{-04}$

Table 4.1. Items frequency statistics in the transactional database

In figure 4.1 are plotted the frequency values for each products. Most item have a frequency lower than 0.001, however with the value of the minimum support and confidence set to 0.001 the results are 18 rules that is a plausible number, considering that we want create a decision tree for each rule. In figure 4.2 the obtained rules are represented: each product, identified by is ID, is connected to another if a rule exists. The bigger the circle, the higher the support and the more intense the color of the circle, the higher the lift.

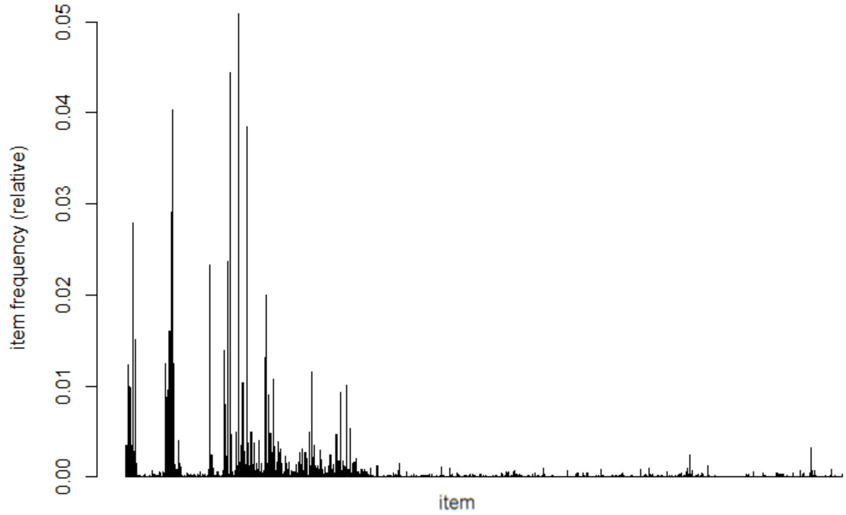


Figure 4.1. Frequency of the item in the dataset

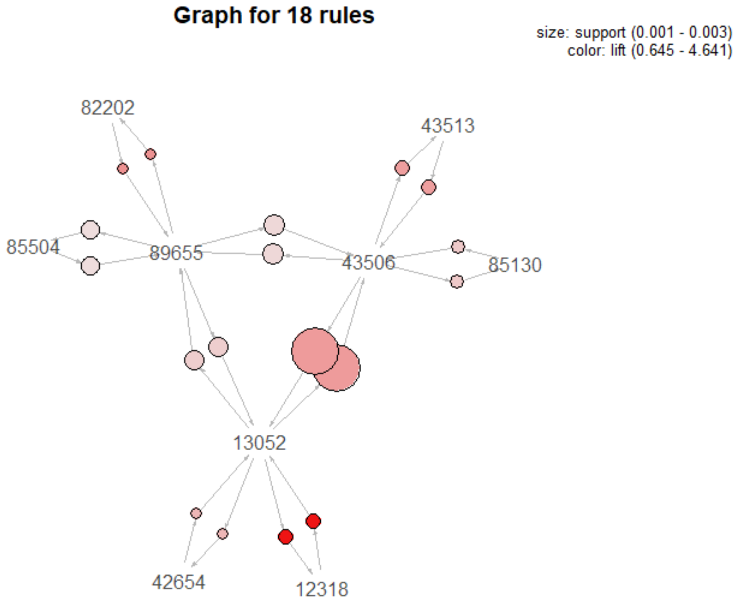


Figure 4.2. Items connected by the association rules

4.2 Comparing the different clustering techniques

The clustering is performed in order to split clients into 4 sub-categories. To perform clustering the *scipy* [32] and *sklearn* [18] python libraries were used. The input table contains customer features both from the customer lists and from its shopping in the last year. The main features are:

- ID_CLIENT: is the client identification number
- OUTLAY_DEPARTMENT_i: is the year's consumer spending for the department category i
- N_PROD_DEPARTMENT_i: is the year's number of products bought for department category i
- N_PROD_PROMO: is the year's number of items bought in promotion
- OUTLAY_PROMO: is the year's consumer spending for item in promotion
- N_TRANSXCLIENT: is the number of transactions done by the client
- JOB: is the job of the customer. There are 5 possible types of job indicated with an integer from 1 to 5
- EDUCATION_LEVEL: is the grade of scholar education of the customer, here there are 4 possibilities expressed with an integer from 1 to 4
- CARD_POINTS: is the points number collected in the customer card
- AGE: the age of the customer
- FAMILY_COMPONENTS: is the number of family components
- LAST_PURCHASE: is expressed in month and correspond to the last shopping time
- ENTRY_DATE: is the month number from the sign up of the client in the retailer program
- OUTLAY_AVG: is the average of the consumer spending

The well known problem of the Clustering is that this technique strictly depends on the dataset. The pre-processing stage has an important role. The input data must be numerical, some of the previous features, as JOB and EDUCATION_LEVEL, must have been converted from categorical to numerical. Moreover some features are inconsistent, for example some costumers during the compilation of the entry module did not specify the FAMILY_COMPONENTS or the AGE, so these information was been missed in the customer lists. The decision is to trying to estimate the missing value considering the similarity with the other customer or through user consideration. It is a common practice to normalize the data before clustering, so in this case the dataset was scaled between [0,1]. Regarding the dimensionality reduction, it is possible to reduce the number of features, at this purpose a Principal Component Analysis (PCA) is performed. The PCA allow to discard the contribution of the features that have a small variance, it consists of two phase: the first is the analysis where we apply the Karhunen-Loeve decomposition that allow to obtain $Z = U^T X$ where X is the input dataset and U is the matrix of eigenvectors of $R_x = E\left\{XX^T\right\}$,

in the second phase we reduce the number of features keeping only the L features with the largest variance. Note that the variance of the z_k feature corresponds to the λ_k eigenvalue of R_x. What we do is to change point of view of the problem, the new matrix Z is a projection of the original one.
The effectiveness of the clusters has been left to the user. However there can be defined some parameters to evaluate the performances based on the intra-cluster similarity and inter-cluster dissimilarity.

SSE

It is the sum of the squared error (SSE), if SSE is small than the clusters are compacted and well separated. It is measured as:

$$SSE = \sum_{k=1}^{K} \sum_{y_i \in C_k} \|y_i - x_k\|^2 \tag{4.1}$$

Where K is the number of clusters, C_k identifies the k cluster and x_k is centroid of the k region as previously defined.
In figure 4.3 can be seen the normalized SSE values, for the different techniques and for different numbers of cluster. It shows that the values converge increasing the number of clusters, in particular the K-medoids performs the worst with the number of clusters lower than 4.

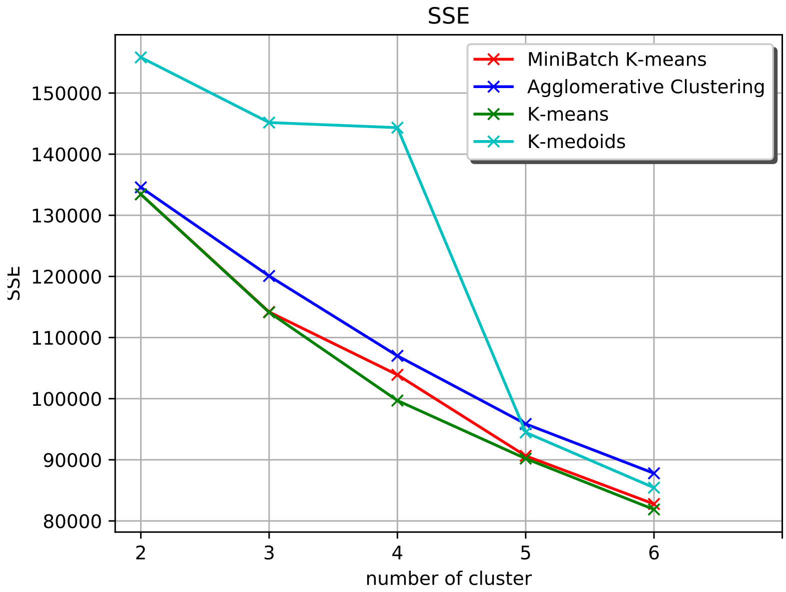


Figure 4.3. Normalized SSE for different number of clusters

Cohesion

The cohesion measures how strong is the similarity intra-cluster, it is defined in [1]. The cohesion evaluation differs considering center-based or hierarchical clustering, because for each technique is different the meanings of cohesion of the items in a cluster.

In center-based clustering the cohesion is defined as:

$$cohesion(C_i) = \sum_{x,y \in C_i} dist(x,y) \tag{4.2}$$

In hierarchical clustering the cohesion is defined as:

$$cohesion(C_i) = \sum_{x \in C_i} dist(x,c_i) \tag{4.3}$$

where c_i is the centroid of the cluster. Lower is the cohesion shorter is the distance between points of the same cluster. In figure 4.4 the value of the mean of the normalized cohesion for different number of cluster can be seen, increasing the number of clusters the performance deteriorate, meaning that intra cluster distance increases and the items are not so "similar".

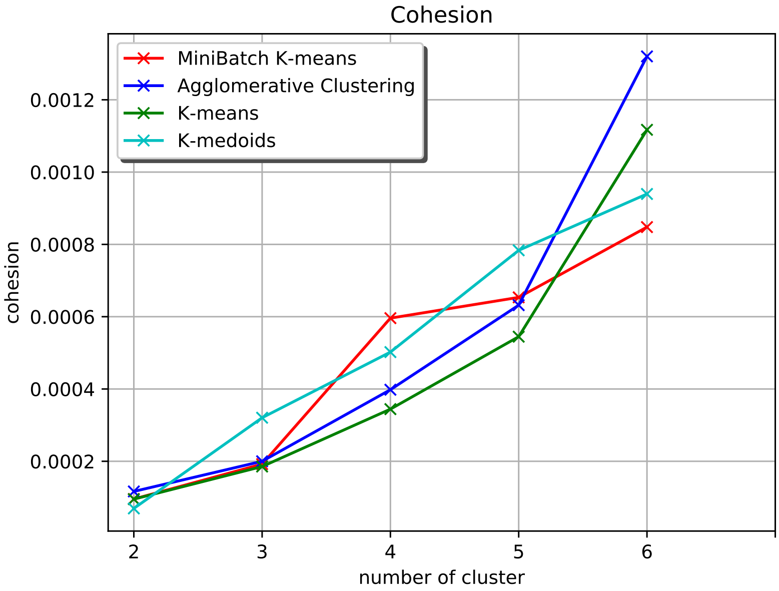


Figure 4.4. Cohesion intra cluster for different number of clusters

Separation

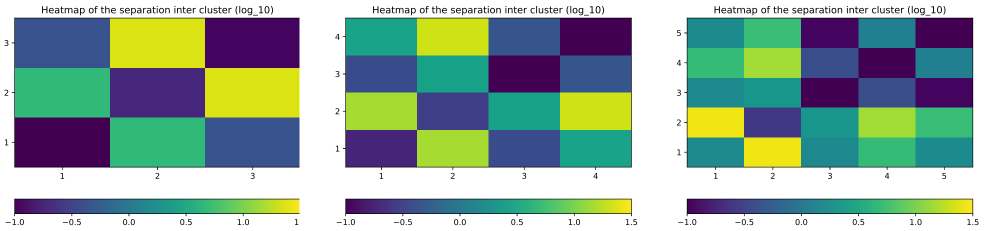
It measures the difference in term of distance between each pair of clusters. As the cohesion it is represented as distance measure and it differs considering the clustering technique used. In center-based clustering it is defined as:

$$separation(C_i, C_j) = dist(c_i, c_j) \tag{4.4}$$

In hierarchical clustering it is defined as:

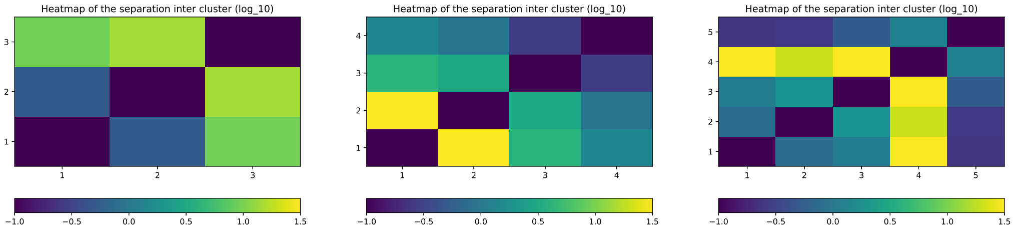
$$separation(C_i, C_j) = \sum_{x \in C_i, y \in C_j} dist(x, y) \tag{4.5}$$

Below in figures 4.5, 4.6, 4.7 and 4.8 the heatmaps of the separation values obtained are shown. In the diagonal of each heatmaps are the cohesion values. In general the algorithm that performs the worst is the Agglomerative because it has higher values.



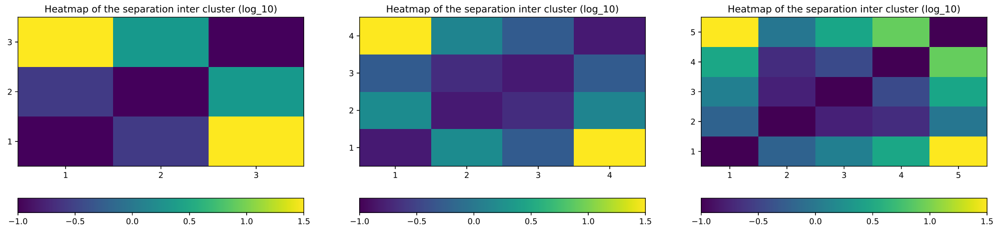
(a) 3 cluster (b) 4 cluster (c) 5 cluster

Figure 4.5. Separation inter cluster for Agglomerative



(a) 3 cluster (b) 4 cluster (c) 5 cluster

Figure 4.6. Separation inter cluster for K-means

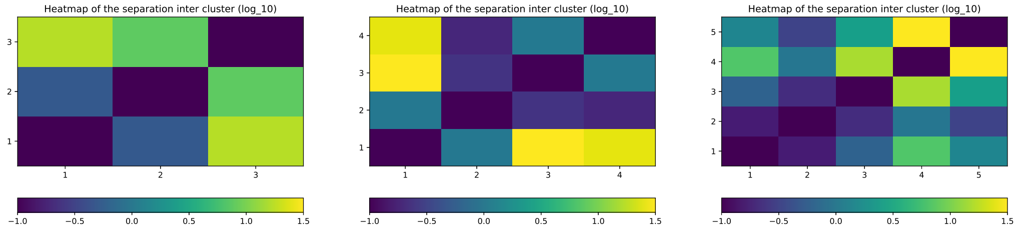


(a) 3 cluster

(b) 4 cluster

(c) 5 cluster

Figure 4.7. Separation inter cluster for K-medoids



(a) 3 cluster

(b) 4 cluster

(c) 5 cluster

Figure 4.8. Separation inter cluster for Mini Batch K-means

Correlation coefficient matrix

It is a square matrix containing the Pearson correlation coefficients that are defined as:

$$r_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{4.6}$$

where $cov(X,Y)$ is the covariance evaluated for the X and Y value and σ_X, σ_Y are the standard deviations.
The correlation matrix R is a symmetric matrix because $r_{X,Y} = r_{Y,X}$. In figure 4.10,4.9,4.12 and 4.11, it is possible to observe the correlation coefficient matrix between the datapoints sorted by cluster labels for each cluster technique tested. The higher is the value, the more correlated are the two points. Looking at the graph it is easy to deduce the dimension of each cluster by ”inspecting the yellow shapes”. It can be seen that in Mini Batch and K-medoids algorithms the number of points in each cluster is unfair, moreover the correlation coefficients intra cluster is not so high.

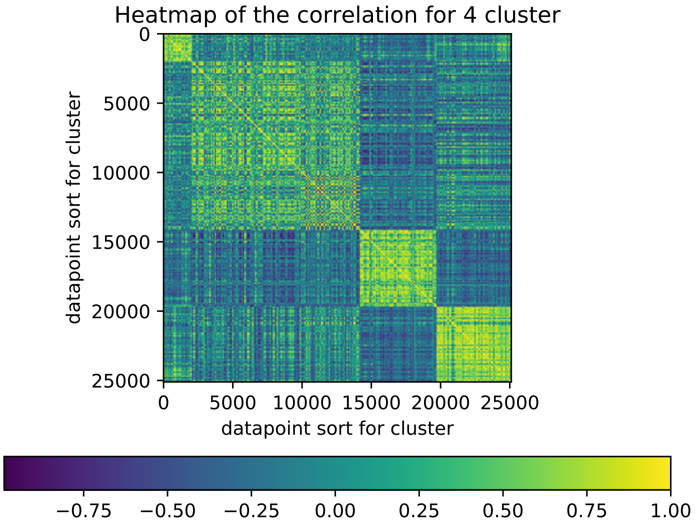


Figure 4.9. Datapoints correlation sorted for MiniBatch cluster labels

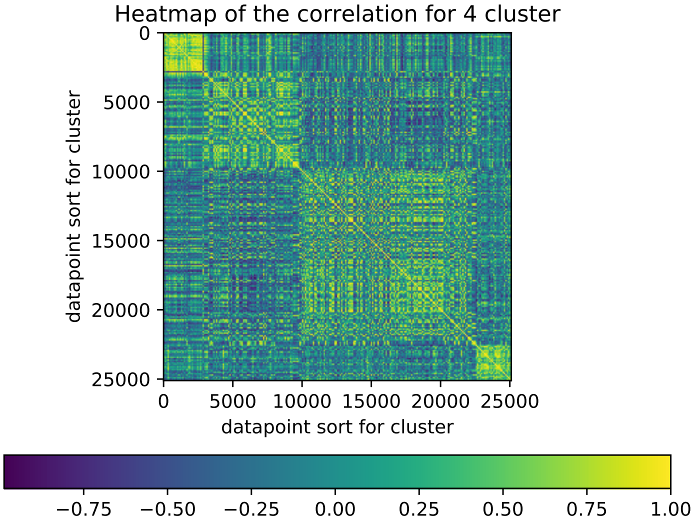


Figure 4.10. Datapoints correlation sorted for K-medoids cluster labels

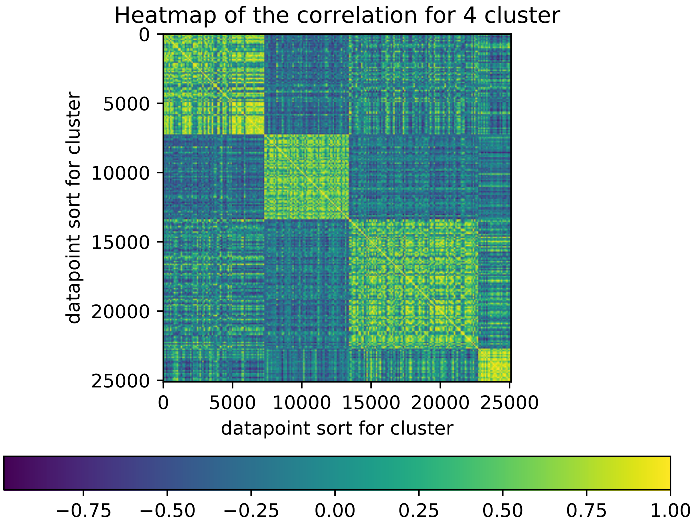


Figure 4.11. Datapoints correlation sorted for Agglomerative cluster labels

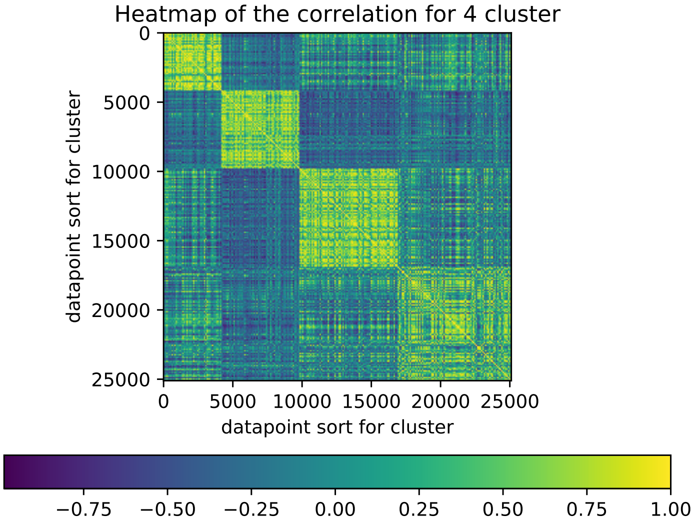


Figure 4.12. Datapoints correlation sorted for K-means cluster labels

After all these considerations there is no absolute winner between the 4 algorithms investigated. Some performances metrics are better for one than the others. However focusing on the case of 4 cluster, there is a method that always performs better and this is the K-means.

4.3 Decision tree results

As before explained the decision tree is performed using the C 4.5 algorithm in the IBM SPSS Modeler tool. It is used here in order to classify clients that active or not a rule between the set of rule discovered in section 4.1. Then it is assigned an economic value to that rule based on the probability the client activates it.
The main attributes as input to the C 4.5 algorithm are the same used for the clustering defined in section 4.2, in addition there are:

- CLUSTER_NUM:is the label of the belonging cluster
- FREQ:is the value of how frequently the customer make a purchase
- AVG_PRODXTRANS: is the average number of product in customer transaction
- AVG_PRICEXTRANS: is the average price of the customer transaction
- N_TRANSXRULEXCLU: is the total number of transaction of the cluster in which the rule is activated
- SUPPORT_RULEXCLUSTER: is the support of the rule in the cluster
- RULE_VERIFIED: is the target, it is 'Y' if the customer active the rule, otherwise is 'N'

The algorithm returns as result the expected value of the target with a certain probability P:

- if 'Y' is returned, P represents the probability that the customer actives the rule
- if 'N' is returned, $1 - P$ represents the probability that the customer actives the rule

The results are related to the rule (1) 82202 => 89655. In figure 4.13 a graph shows the weights of the most significant attributes used to classify the clients by the $C4.5$. The most important attributes are features that separate better the dataset, so how useful or valuable the features are in the construction of the tree. The more occurs the attribute the more it is important. In figure 4.15 there is the obtained decision tree graph where in each node there is a little histogram of the number of Y and N. Some wide used parameters to evaluate the result are the sensitivity and specificity values defined as:

$$sensitivity = P(\hat{y} = Y|y = Y) \tag{4.7}$$

$$specificity = P(\hat{y} = N|y = N) \tag{4.8}$$

where y is the target function real value and $\hat{y}$ is the predicted one. The sensitivity is also called true positive value, because it defines the probability that a value is

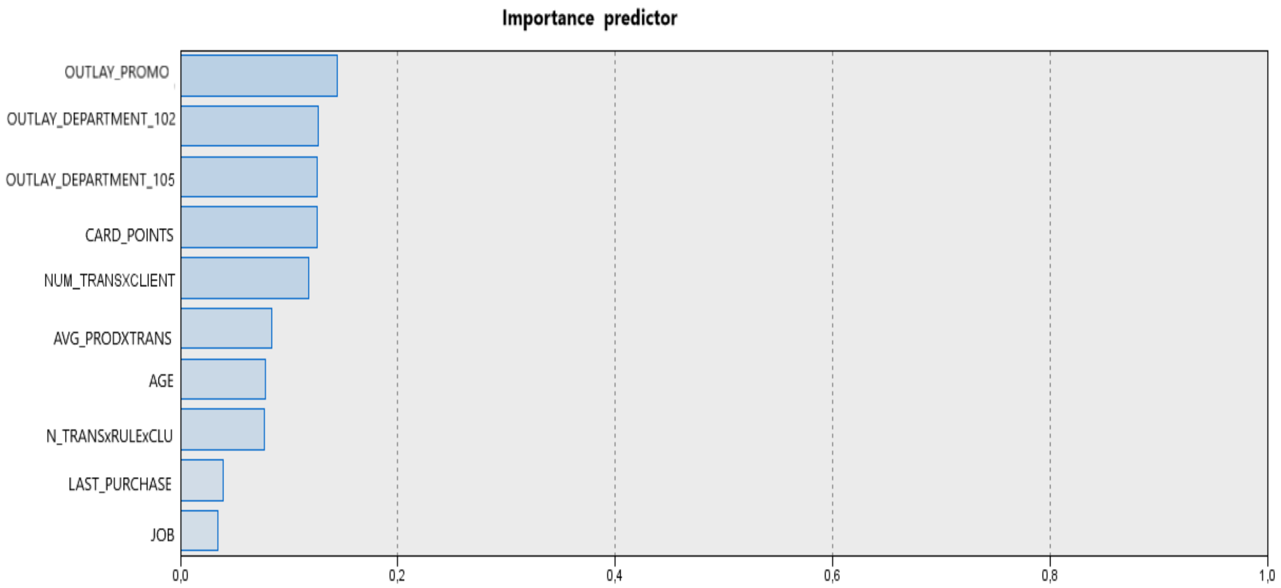


Figure 4.13. Most significant attributes for the rule (1)

estimated Y given a real Y value. Instead the specificity is called false negative because it represents the values estimated N given real N values.

The obtained value for the case of the rule (1) are: $sensitivity = 0.80$ and $specificity = 0.69$. The desired values of sensitivity and a specificity should be near 1 because this means that the algorithm is able to guess all the Y value as Y and all the N one as N. The given results let some doubt about the specificity, anyway in this particular use case the interested parameter to take in consideration is the score probability that a customer applies the rule.

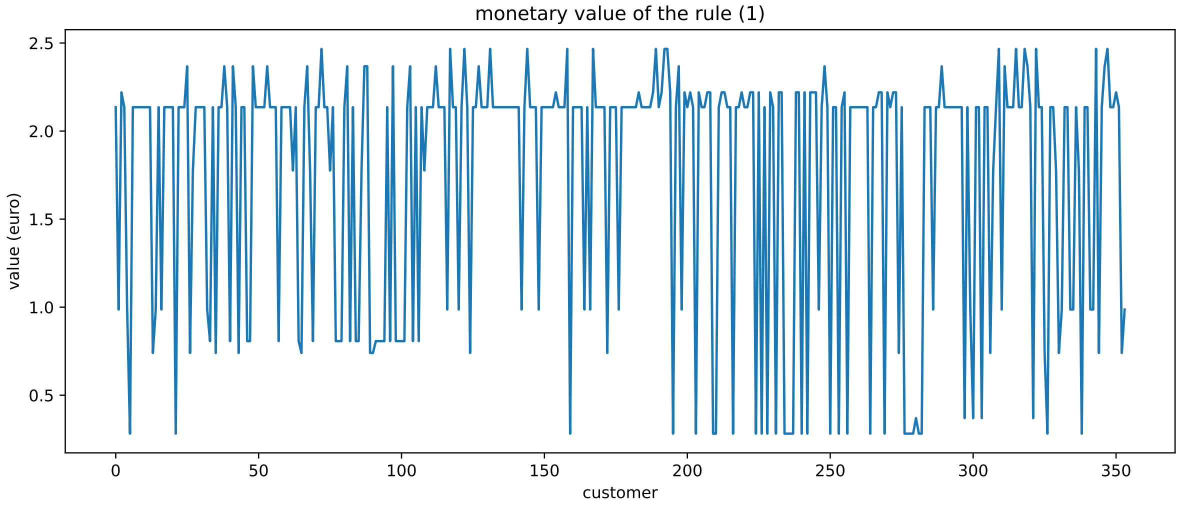


Figure 4.14. Monetary value of the rule for each customer

In figure 4.14 there is a plot of the monetary value of the rule for each customer (m_i^r), this value is obtained as:

$$m_i^r = v^r \cdot p_i^r \tag{4.9}$$

where v^r is the price value of the r rule and p_i^r is the probability that the customer i activate the rule r. In particular, p_i^r is a good indicator for the suggestion of a new product for the customer: if the probability value is high it means that the company can invest for that client, for example doing targeted advertising.
The main metric that can be extracted through this procedure is the rule monetary value estimation defined as:

$$M_r = \sum_{i \in T} m_i^r = \sum_{i \in T} v^r \cdot p_i^r \tag{4.10}$$

where T is the set of all the customer and M_r is the monetary value of the rule r. This value makes in evidence the more profitable rules: the higher the M_r the higher is the profit that the r rule carries for the company.

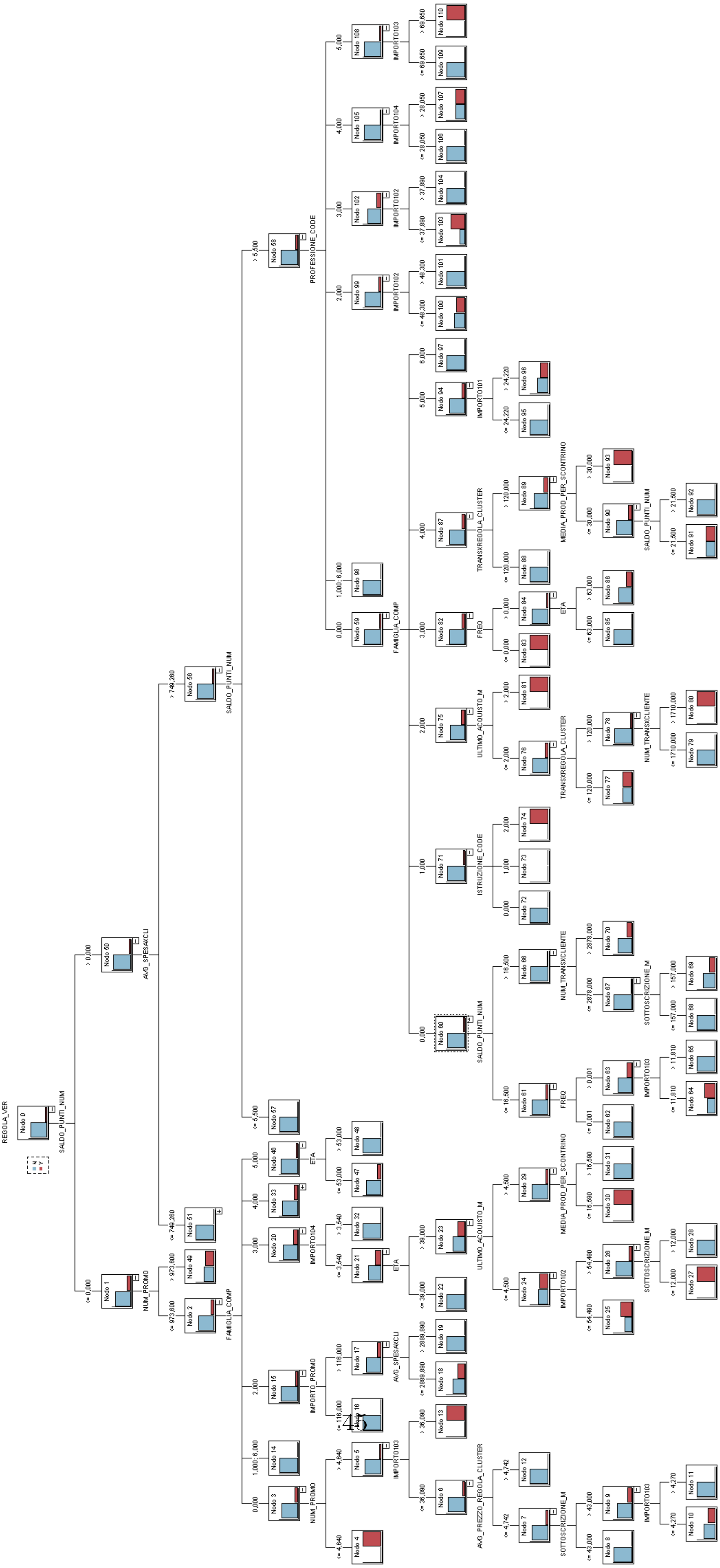


Figure 4.15. Decision three for the rule (1)

4.4 Forecast performances and results

After the clusters creation, it is tried to predict the future sales through the techniques previously explained: Linear predictor and LSTM network.
The input data is a sequence of the daily purchases done by each cluster from the last three year, that can be seen in figure 4.16. It is evident a certain periodic trend in the year, moreover zooming the graph as can be shown in figure 4.17, it can be seen also a weekly periodic trend: the higher peak sales was always on Saturday. Also remember that to cluster the client, it was used only the last year sales information, so obtaining a similar trend looking at the three years is not so obvious.

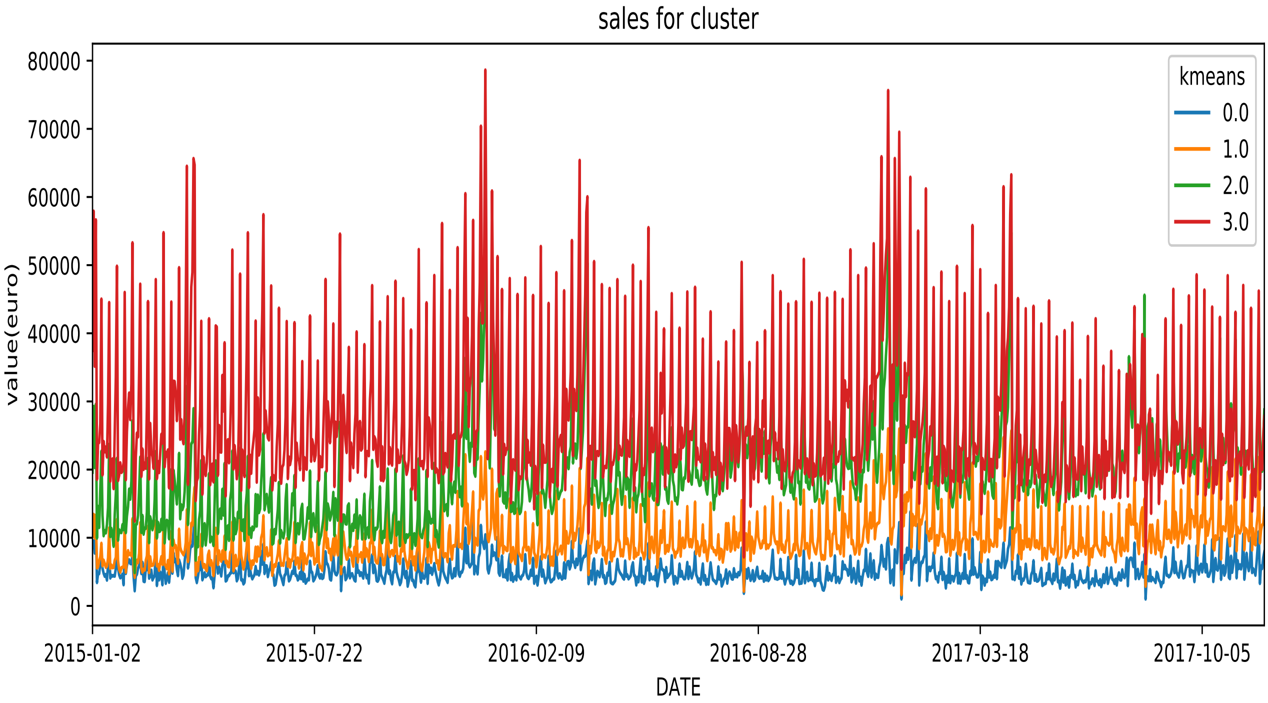


Figure 4.16. Daily sales for cluster during three years

Now, taking each cluster trend, it is tried to predict the sales for the next month. To implement the LSTM network the python library tensorflow [33] is used. Tensorflow allows to create a variety of machine learning algorithm using a data flow graph where the node are the mathematical operations and the edge are the *tensors*. Tensors are “arbitrary dimensionality arrays where the underlying element type is specified or inferred at graph-construction time” [33]
The tensorflow LSTM network require some parameter as input:

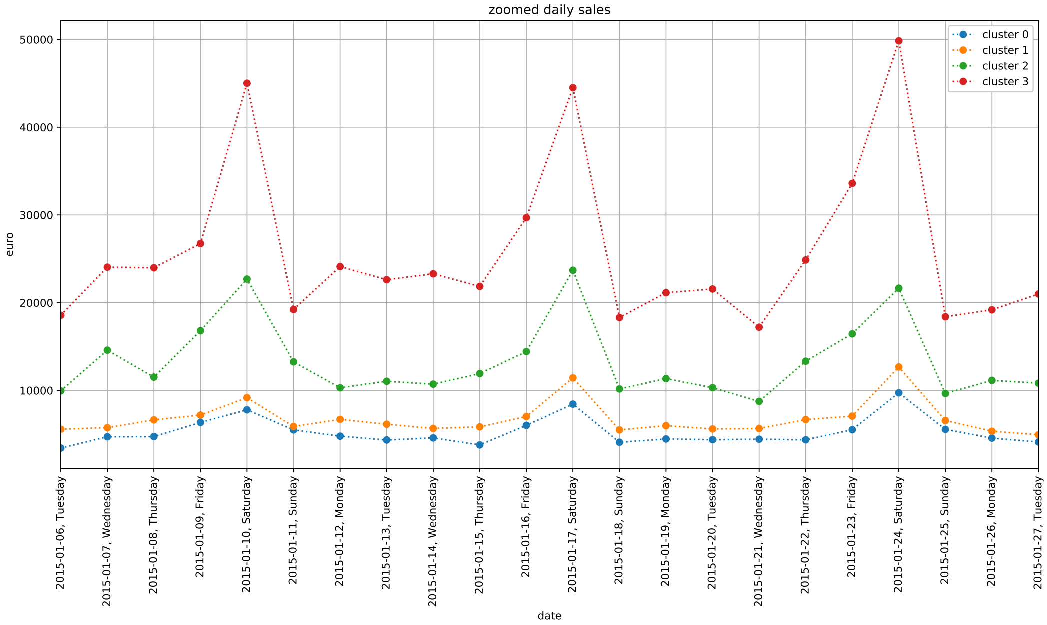


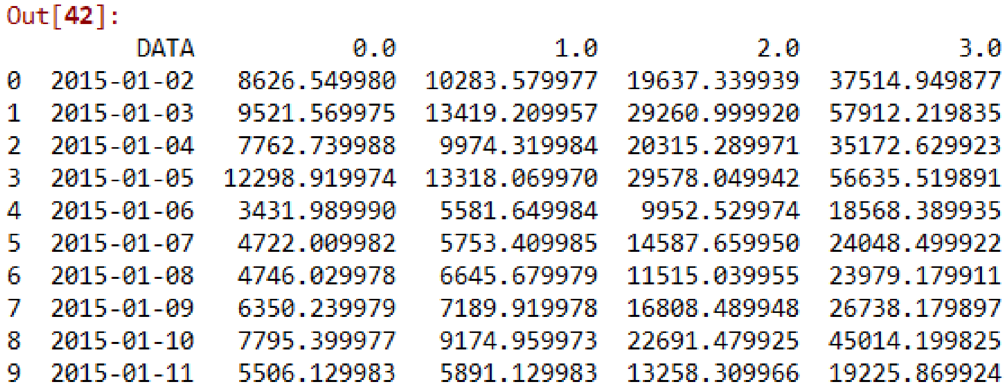
Figure 4.17. Zoomed daily sales for cluster

- *optimizer*: is a *tensorflow* container for the iterative function used to minimize the error or square difference between the output and the desired value, in this case it is chosen the Adam function, but it can be Gradient Descendent for example. The error function to minimize is the SSE (Sum square error).
- *learning rate*: is the learning rate for the iterative function in the optimizer, usually it is a value much smaller than 1.
- *number of hidden layers*: is the number of the hidden layer of the network
- *iterations number*: is the number of times that the network runs

The last three listed parameters are changed until a desired result is obtained; the procedure is to run the network, look at the results and at the SSE value and if there is an improvement, then change the parameter and re-run.

To reach a drastic minimization of the SSE value does not mean to obtain a good result because the LSTM network, like the NN in general, tend to suffer of *overfitting*, as previously we explained. The network must be stopped earlier (the technique is called early stopping). The input data of the network is a matrix in which the $i-th$ column corresponds to the sales sequence of the $i-th$ cluster, the figure 4.18 shows the first 10 rows.

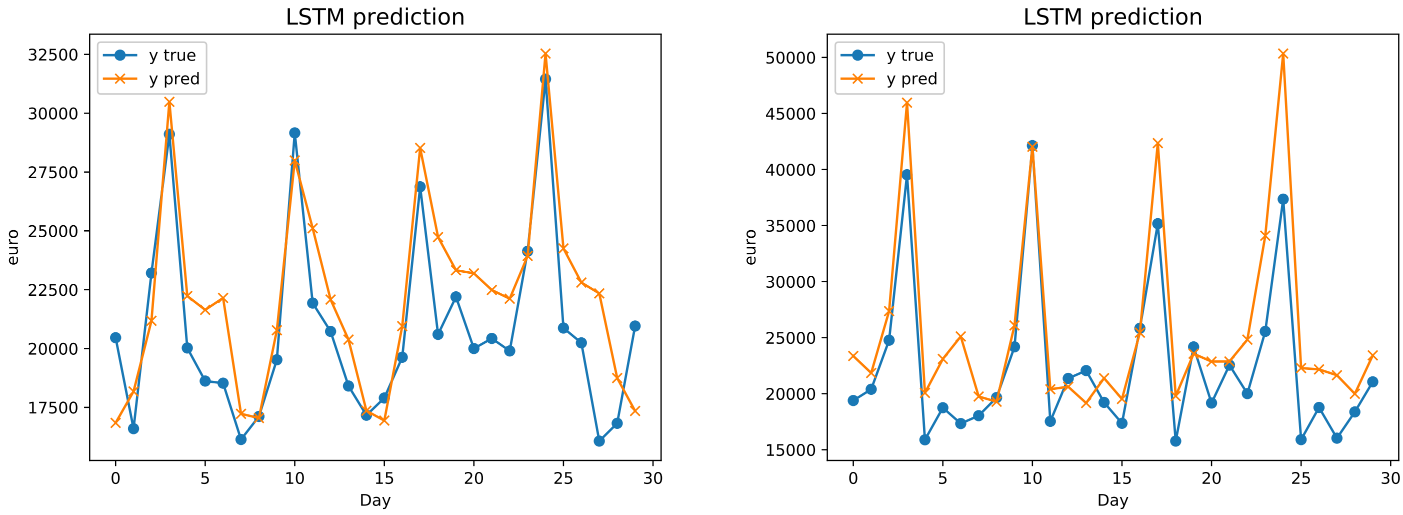
In figure 4.19 it can be seen the result of the LSTM network for the prediction of the sales of the cluster 3 and cluster 2 for the next 30 days (using as training 900 days).



```
Out[42]:
         DATA          0.0           1.0           2.0           3.0
0  2015-01-02   8626.549980  10283.579977  19637.339939  37514.949877
1  2015-01-03   9521.569975  13419.209957  29260.999920  57912.219835
2  2015-01-04   7762.739988   9974.319984  20315.289971  35172.629923
3  2015-01-05  12298.919974  13318.069970  29578.049942  56635.519891
4  2015-01-06   3431.989990   5581.649984   9952.529974  18568.389935
5  2015-01-07   4722.009982   5753.409985  14587.659950  24048.499922
6  2015-01-08   4746.029978   6645.679979  11515.039955  23979.179911
7  2015-01-09   6350.239979   7189.919978  16808.489948  26738.179897
8  2015-01-10   7795.399977   9174.959973  22691.479925  45014.199825
9  2015-01-11   5506.129983   5891.129983  13258.309966  19225.869924
```

Figure 4.18. Input matrix for the prediction

In blue there is the true value, in orange the predicted one. In figure 4.20 there is a plot of the true sequence *y_true* versus the LSTM prediction *y_pred*: higher the distance between the points and the black line higher the error of estimation.

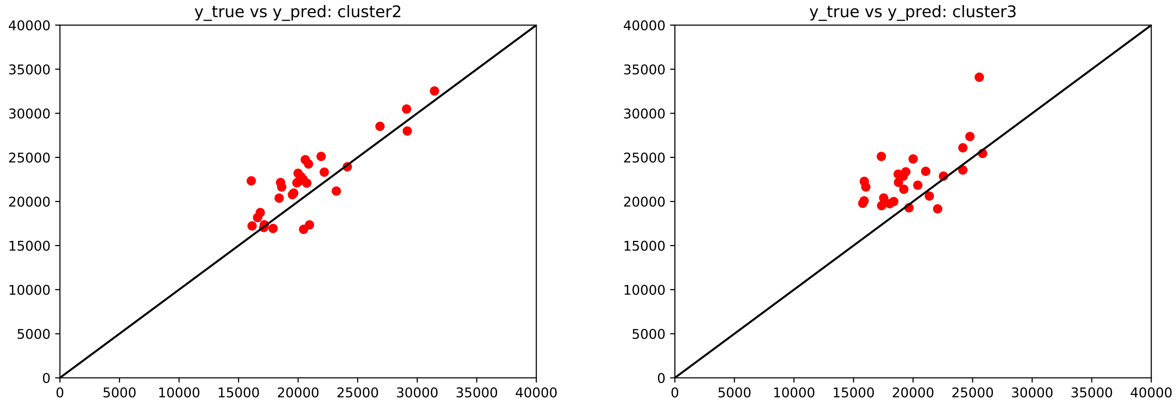


(a) cluster 2

(b) cluster 3

Figure 4.19. LSTM network predictions

The Linear Predictor is implemented through the pyhton *numpy* library. The only parameter to be set in input is the p order of predictor and optionally the updating coefficient for the estimated autocorrelation. Here the input data is the vector sequence of sales of each single cluster. As previously for the LSTM, in figure 4.21



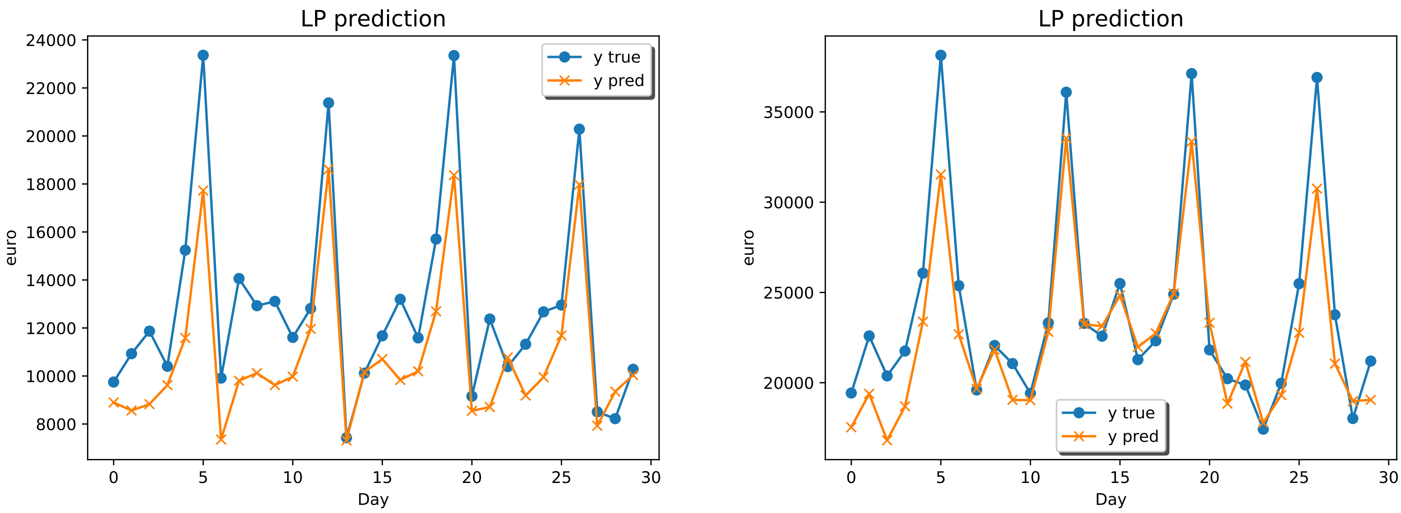
(a) cluster 2

(b) cluster 3

Figure 4.20. Plot of the true sequence *y_true* versus the predicted one *y_pred* of the LSTM netwotk

are depicted the Linear Predictor results for the cluster 0 and cluster 3 for the next 30 days: in blue there is the true value and in orange the estimated one. Instead, in figure 4.22 it can be seen the values of the true sequence *y_true* versus the LP prediction *y_pred* as for the LSTM network.

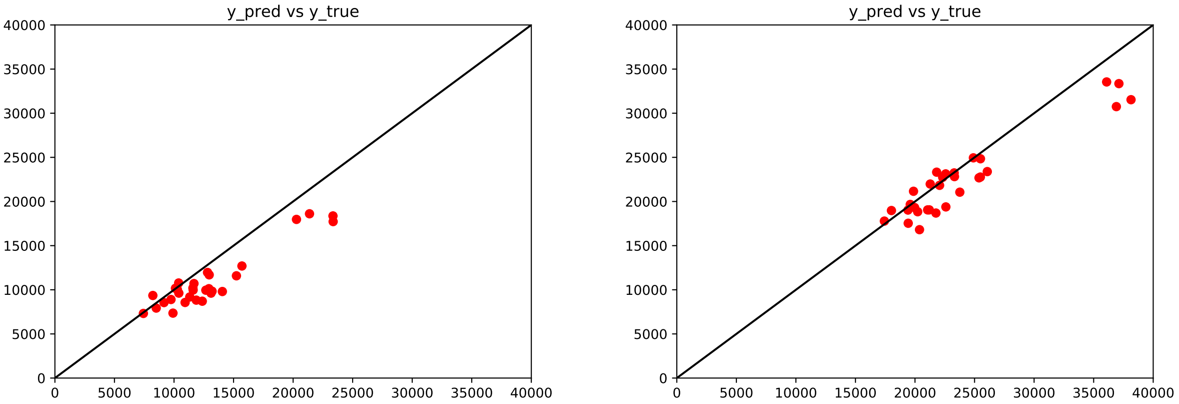
A variant of the Linear Predictor algorithm is the multi-LP that take as input data all the cluster sequence together as in figure 4.18. The results for the 30 days estimation, as for the previous case, can be seen in the figures 4.23 and 4.24



(a) cluster 2

(b) cluster 3

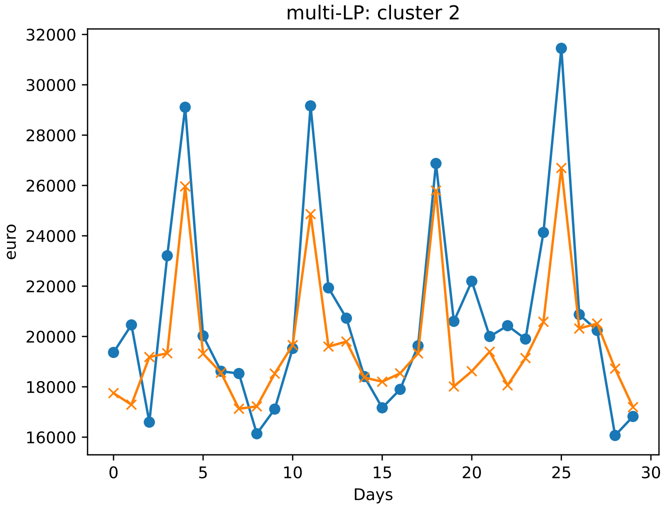
Figure 4.21. Forecast with Linear Predictor



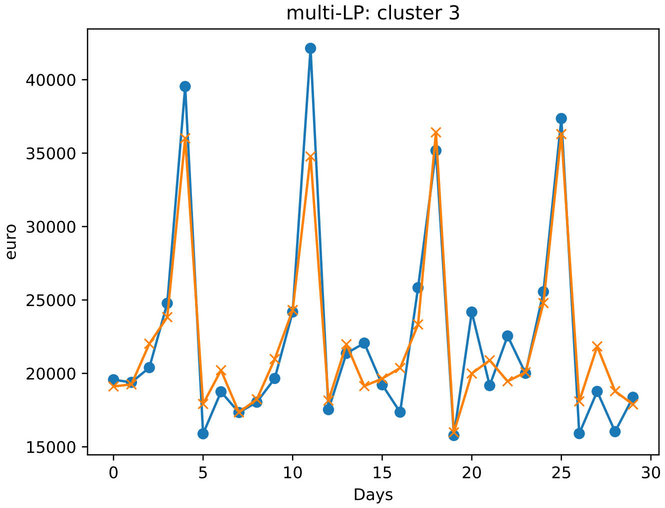
(a) cluster 2

(b) cluster 3

Figure 4.22. Plot of the true sequence *y_true* versus the predicted one *y_pred* of the Linear Predictor



(a) cluster 2. In blue the true value, in orange the predicted one

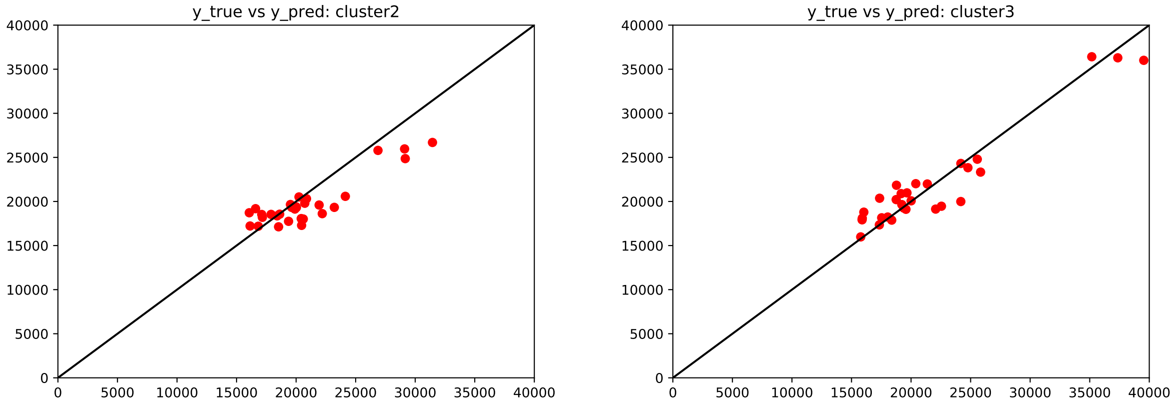


(b) cluster 3. In blue the true value, in orange the predicted one

Figure 4.23. Forecast with multi-LP

Table 4.2. $\log_{10}$ of the SSE

	cluster 0	***cluster 1***	***cluster 2***	***cluster 3***
LP	2.83	3.06	3.31	3.37
LSTM	3.08	3.17	3.39	3.66
multi-LP	2.76	3.04	3.34	3.36



(a) cluster 2

(b) cluster 3

Figure 4.24. Plot of the true sequence *y_true* versus the predicted one *y_pred* of the multi-LP

In table 4.2 there are the value of the $\log_{10}$ of the sum square error of the algorithms. The LP and the multi-LP perform the best. Also the LSTM network is the most expensive in term of time and computation, because it needs to estimate in background all the parameter each time and it must be run at least 1000 times to reach a good value. On the contrary the Linear Predictor is computationally cheaper.

In figure 4.25 there is the plot of the absolute value of the error of the Linear Predictor between the true value of the sales sequence *y_true* and the predicted one *y_pred*. In blue it is plotted the error when it is tried to estimate the next 200 days, in orange the error of first 30 days is point out. As expected, the figure evidences an increasing of the error considering a large prediction window because the farthest the points the smaller the correlation. However, looking at the first 30 days, the error does not diverge rapidly, this is due to the periodic nature of the initial signal, moreover the signal is a time sequence in which the value of two consecutive points are strongly correlated, this correlation persist, but when the distance between the observed points increase the correlation weaken and the prediction error rapidly increases.

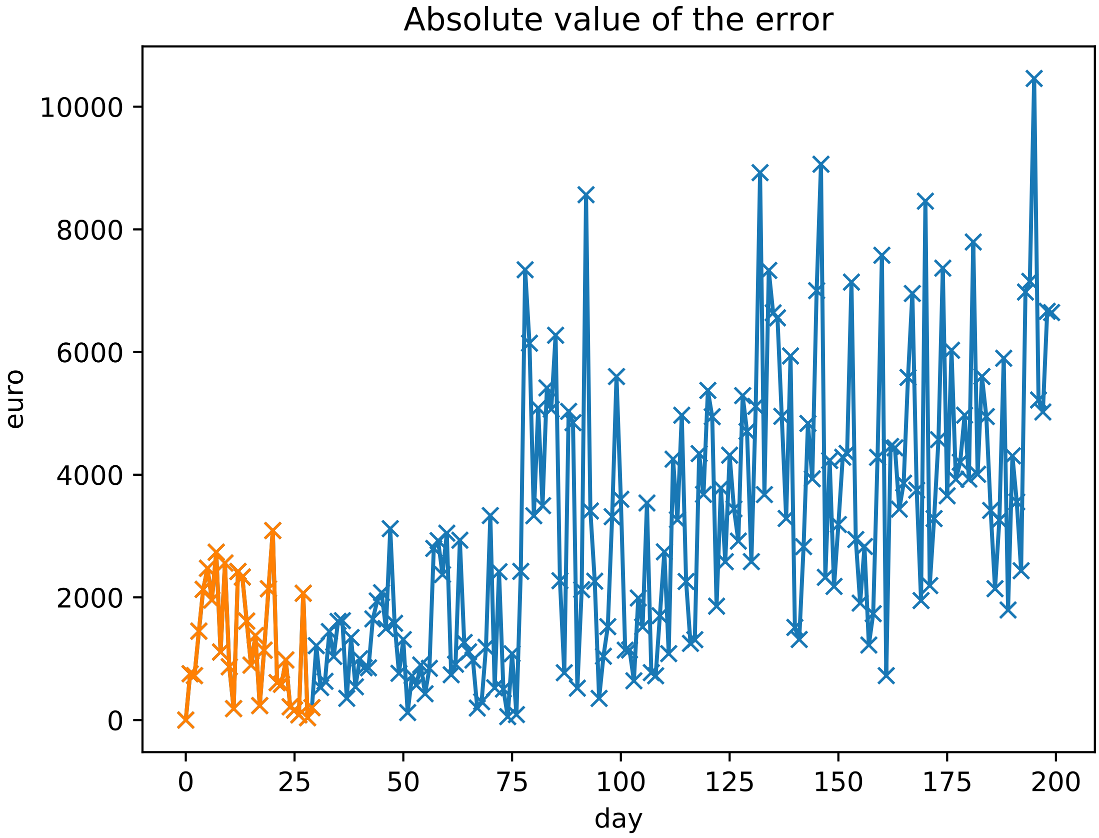


Figure 4.25. Absolute value of the error for the LP estimation: in blue the error for 200 days estimation, in orange the error for the first 30 days estimation

Joining the price of the rule and enforcing the forecast previously method describe to predict the daily number of clients in each cluster, another important quantity that can be extracted is the expected future profit taken by a rule.

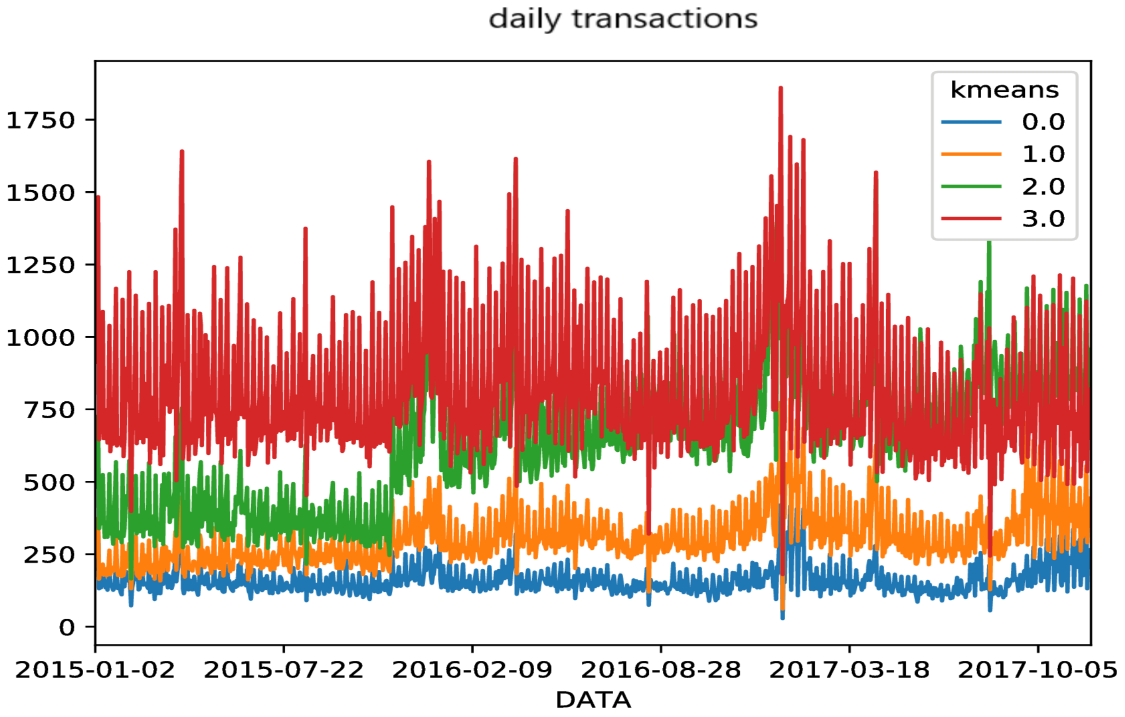
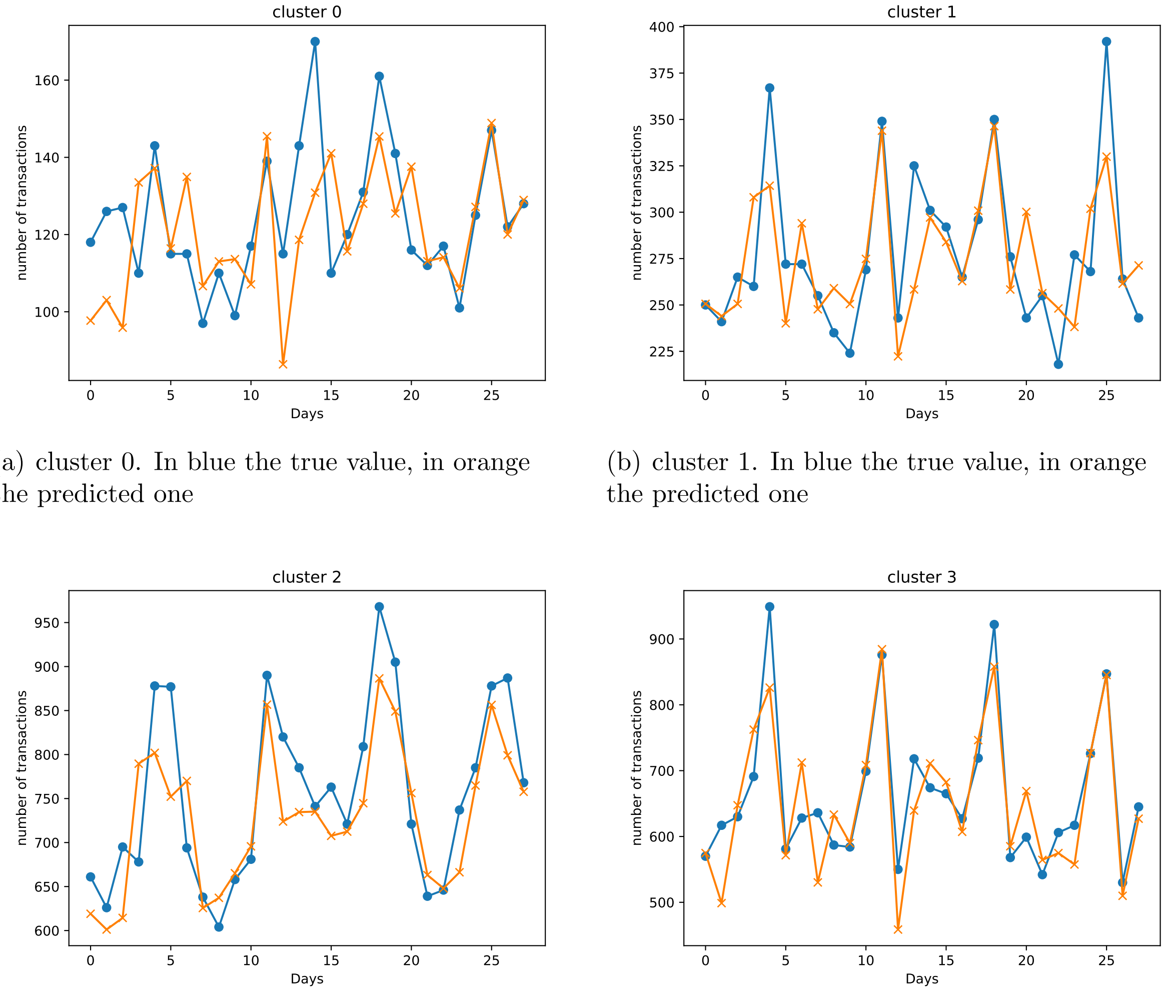


Figure 4.26. Daily transaction for cluster in three years

We can forecast the number of daily transactions. The values of the daily transaction, grouped for each cluster j, can be seen in figure 4.26.

Because as previously discussed it is less expensive in computation, we adopted the Linear Predictor in order to obtain the trend of the two future weeks, in particular here it is used the multi-LP that is a version of the Linear Predictor able to deal multiple sequences, so we put as input a packed matrix having in each column a cluster sequence.

In figures 4.27 can be seen the future trends estimations of the daily number of clients and its real value for each cluster. The table 4.3 presents the values of the $\log_{10}$ of the sum of square errors (SSE) of the prediction and the sum square of clusters (SSC) which represents the average "power" of the clusters. The values in the table make in evidence an high value of the SNR (Signal Noise Ratio), meaning that the final prediction is of good quality.



(a) cluster 0. In blue the true value, in orange the predicted one

(b) cluster 1. In blue the true value, in orange the predicted one

(c) cluster 2. In blue the true value, in orange the predicted one

(d) cluster 3. In blue the true value, in orange the predicted one

Figure 4.27. Daily transactions predicted with multi-LP

Table 4.3. $\log_{10}$ of the SSE and the SSC for the prediction of the daily number of transactions with multi-LP

	cluster 0	***cluster 1***	***cluster 2***	***cluster 3***
SSE	1.35	1.46	1.78	1.75
SSC	3.73	4.02	4.32	4.47

The support value $s_{r,j}$ has been used before in the decision tree as attribute, here is defined in detail. First the initial transactional database is divided in $C = 4$

sub-datasets containing the transactions of each cluster j, the support $s_{r,j}$ of the rule $r : X \to Y$ in cluster j is defined as:

$$s_{r,j} = \frac{\sigma_j(X \cup Y)}{N_j} \tag{4.11}$$

where $\sigma_j(X \cup Y)$ is the number of occurrences of the rule r in the sub-dataset of the cluster j and the N_j is the total number of transactions in j subset. The expect profit taken by the rule can be now defined as:

$$E_r(N) = \sum_{j=1}^{C} \hat{x_j}(N) \cdot s_{r,j} \cdot v_r \tag{4.12}$$

where $E_r(N)$ is the expected profit of the rule r in the day N, $\hat{x_j}(N)$ is the estimated number of clients belonging to the j cluster in the day N, $s_{r,j}$ is the support of the rule r in the cluster j and v_r is the previous defined annual average price of the rule r. In figure 4.28 can be seen the expected profit for the next month of the rule (1).

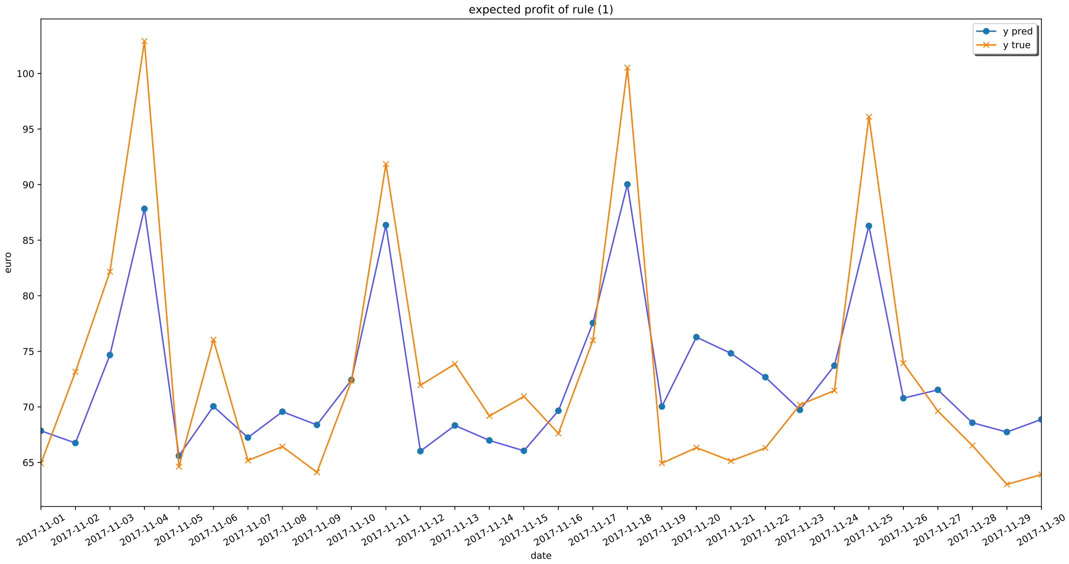


Figure 4.28. Expected profit for the rule (1)

At the contrary of the monetary value M_r, the expected profit of the rule $E_r(N)$ is a value that changed in time so this can be used for marketing decision as the decision of the period in which start advertising campaigns.

Conclusions

In this thesis various data mining and machine learning problems were analyzed through investigating various techniques in the scenario of a big retailer. The main problems to deal with are the need of the company to suggest product in a targeted way and predict future trend of the sales, at this purpose the initial dataset was exploited and the data mining methodology designed by the CRISP-DM [3] was applied.

The main problem was split in three subtasks: discovering possible future sales, searching for the profitable sales for the company and finally predicting future sales trend.

The possible future sales consists of the recommendation of products that were discovered applying a Cross-selling strategy scanning the transactional database of the clients and extracting the association between products in the market basket. The algorithm used, that perform market basket analysis, is the Apriori [4]: this algorithm extracts products association rules of the form $X \rightarrow Y$ where X is the subset of products that drags Y that is the possible recommended product. The rule was extracted considering the frequency of the products in the database, an higher number of the time the set X and Y are purchase together by the customers, if a customer bought X then with a strong probability it needs also Y.

A weakness in the association rules extraction of the previous step is that most of the times the rule are economically irrelevant for the retailer, moreover not always the customer is inclined to buy the suggested product, it depends on his shopping routine and habit. So it rise the need to give an economical weight to the rules and detect the set of customer more prone to buy the recommended product.

Analyzing the clients lists, it can be discovered sets of clients that take the same shopping attitude, in this way the initial dataset can be reduced in dimensionality. At this purpose clustering technique are used; in particular in this thesis to find significant sets, the analyzed clustering techniques are K-means, Mini Batch K-means, K-medoids and Agglomerative Clustering. For all of that, it was done a performance evaluation considering intra-cluster and inter-cluster parameters. The algorithms perform almost the same, the final choice depends on the number of clusters that would extracted and, in the use case considered of 4 clusters, the best performance was reached by the K-means algorithm.

To overcome the problem of suggesting profitable products, the proposed method is to create a Decision Tree with the C4.5 [30] for each rule in order to solve the binary decision problem consisting in the activation or not of the rule by the customers. Also the C 4.5 returns a score probability p_i^r representing the probability that customer i actives the rules r, so the retailer, sorting the customer by the p_i^r, can easily know to which customers to recommend the product Y of the rule.
Then multiplying the probability p_i^r for the cost of the rule it can be evaluated the total monetary value of the rule as:

$$M_r = \sum_{i \in Customer} m_i^r = \sum_{i \in Customer} v^r \cdot p_i^r$$

where M_r is the total monetary value of the rule r and v_r is the average price of the rule.
Then the forecast of the future sales trend for each cluster was performed investigating two methods. The first method adopted is the Recurrent Neural Network, this type of neural network is widely used and widespread in the language modeling and speech recognition [19] because it is designed to handle dependent sequential inputs. Here it is exploited a particular type of RNN, called Long-Short Term memory network that overcomes the vanished gradient problem. This problem consists in the inability of the RNN cell to manage long-term dependencies, as example dependencies between points that are far apart; the LSTM works essentially like the classical RNN with some difference inside the network cell, that allow to keep in count all the past points and decide which to discard. The second method used is the Linear Predictor that consists in the estimation of the future trend as a linear function of the data. Comparing the performance of the two algorithm, they reach a similar result, but the LSTM network is a more complex and expensive in term of computation respect of the Linear Predictor.
Enforcing the adopted forecast method to predict the daily number of clients for each cluster and joining it with the price value of a rule v_r, it can be defined the daily expected profit for each rule as:

$$E_r(N) = \sum_{j=1}^{C} \hat{x_j}(N) \cdot s_{r,j} \cdot v^r$$

where $s_{r,j}$ is the support of the rule in the cluster j and $x_j(N)$ is the predicted number of client of the cluster j in the day N.
The thesis suggest a complete framework for the analysis of the data available to the big retailers with which is possible to analyze and make strategic planning based on data mining and machine learning techniques.

Bibliography

[1] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, and ZhaoHui Tang. Introduction to data mining.

[2] Atul Parvatiyar and Jagdish N Sheth. Customer relationship management: Emerging practice, process, and discipline. *Journal of Economic & Social Research*, 3(2), 2001.

[3] Kenneth Jensen. `https://commons.wikimedia.org/w/index.php?curid=24930610`. ftp://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/18.0/en/ModelerCRISPDM.pdf (Figure 1), CC BY-SA 3.0.

[4] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM, 1993.

[5] Bin Liu, Shu Gui Cao, and Wu He. Distributed data mining for e-business. *Information Technology and Management*, 12(2):67–79, 2011.

[6] Margaret Rouse. advanced analytics. `http://searchbusinessanalytics.techtarget.com/definition/advanced-analytics`, 2017.

[7] Dalla business intelligence ai sistemi di predictive analytics. `http://www.dataskills.it/dalla-business-intelligence-ai-sistemi-di-predictive-analytics/`, 2015.

[8] `https:www.olap.com`.

[9] Chris Rygielski, Jyun-Cheng Wang, and David C Yen. Data mining techniques for customer relationship management. *Technology in society*, 24(4):483–502, 2002.

[10] Atul Parvatiyar and Jagdish N Sheth. Customer relationship management: Emerging practice, process, and discipline. *Journal of Economic & Social Research*, 3(2), 2001.

[11] G Bracchi-C Francalanci-G Motta. Sistemi informativi per l'impresa digitale, 2005.

[12] Gordon S. Linoff and Michael J. A. Berry. *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management.* Wiley Publishing, 3rd edition, 2011.

[13] Colin Shearer. The crisp-dm model: the new blueprint for data mining. *Journal of data warehousing*, 5(4):13–22, 2000.

[14] Cross-industry standard process for data mining. Cross-industry standard process for data mining — Wikipedia, the free encyclopedia. `"https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining"`. [Online; accessed 23-February-2018].

[15] Foster Provost and Tom Fawcett. *Data Science for Business: What you need to know about data mining and data-analytic thinking.* " O'Reilly Media, Inc.", 2013.

[16] Pete Chapman Ncr, Julian Clinton, Randy Kerber Ncr, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rüdiger Wirth. Crisp-dm 1.0. 1999.

[17] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques.* Elsevier, 2011.

[18] 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*, 12:2825–2830, 2011.

[19] Ken-ichi Kamijo and Tetsuji Tanigawa. Stock price pattern recognition-a recurrent neural network approach. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, pages 215–221. IEEE, 1990.

[20] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. `http://www.deeplearningbook.org`.

[21] Understanding lstm networks - colah's blog, http://colah.github.io/posts/2015-08-understanding-lstms/.

[22] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

[23] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

[24] Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024, 2011.

[25] Enrico Palumbo, Giuseppe Rizzo, Raphaël Troncy, and Elena Maria Baralis. Predicting your next stop-over from location-based social network data with recurrent neural networks. 2017.

[26] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[27] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[28] Letizia Lo Presti and Fabrizio Sellone. Fondamenti di analisi statistica dei segnali.

[29] Yoon Ho Cho, Jae Kyeong Kim, and Soung Hie Kim. A personalized recommender system based on web usage mining and decision tree induction. *Expert systems with Applications*, 23(3):329–342, 2002.

[30] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[31] Michael Hahsler, Bettina Gruen, and Kurt Hornik. arules – A computational environment for mining association rules and frequent item sets. *Journal of Statistical Software*, 14(15):1–25, October 2005.

[32] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 28-02-2018].

[33] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.