Forecast analysis for sales in large-scale retail trade

M. Nanni and L. Spinsanti, KDD-Lab, ISTI-CNR Pisa, Italy

{mirco.nanni, laura.spinsanti} @ isti.cnr.it

Abstract

In large-scale retail trade, a very significant problem consists in analyzing the response of clients to product promotions. The aim of the project described in this work is the extraction of forecasting models able to estimate the volume of sales involving a product under promotion, together with a prediction of the risk of *out of stock* events, in which case the sales forecast should be considered potentially underestimated. Our approach consists in developing a multi-class classifier with ordinal classes (lower classes represent smaller numbers of items sold) as opposed to more traditional approaches that translate the problem to a binary-class classification. In order to do that, a proper discretization of sales values is studied, and ad hoc quality measures are provided in order to evaluate the accuracy of forecast models taking into consideration the order of classes. Finally, an overall system for end users is sketched, where the forecasting functionality are organized in an integrated dashboard.

1 Introduction

Important business decisions and organization require a scientific framework to make systematic analysis of alternatives, as recognized since Taylor's classic work “The Principles of Scientific Management” (Taylor, 1911), that essentially marks the beginning of the Decision Science field.

A fundamental task in decision science is forecasting, involved in most decision making processes, sometimes even at an unconscious level. The basic idea is that the known history of the market (global or limited to a single company organization) can help to induce reasonable guesses of the effects of an action, therefore providing a valuable support in evaluating the several alternatives business managers typically have to sift through, and in choosing the most promising one.

Forecasting in the business context, and sales forecasting in particular, can be studied and applied at several different levels:

- in global market analysis, also called the *macro level*, where offers and demands are studied in the general context of global market without a specific focus on single products or services;

- in sector- or product-specific market analysis, also called the *micro level*, where the above analysis is focused on single or families of products/services;
- in within-company market analysis, where the focus is in ascertaining the health status of the company activities, mainly in a evolutive perspective that allows to recognize trends and possible weak points (actual or future), and in evaluating the future overall effects of actions to be taken within the company;
- in within-company product-specific sales analysis, where a single product is put under the lens of a microscope and analyzed in detail, highlighting its performances and its reactions to various kinds of inter-company stimuli (e.g., promotions or change of exposition level) and external ones (e.g., the introduction in the market of a new competitor product).

The aim of this work is to analyze a real case study in the latter context, focusing on the effects of promotions on the sales of a single product, mainly aimed at optimizing its stoking. The closed world context of such analysis, on one hand simplifies the forecasting problem by omitting external factors that are difficult to handle and that, in some cases, might have a large uncertainty; on the other hand, it allows to work with a complete and detailed history of previous sales, precise measures directly collected by the company, and even permits (to an economically-limited extent) to empirically evaluate models and strategies on the ground, by replicating situations and later measuring the effects.

1.1 The BI-COOP context

Coop Italia is today the largest holding of large retailers and the largest organization of consumers of Italy. Overall, Coop Italia includes 163 consumer cooperatives with approximately 1261 stores, 6 million members and over 52,000 employees. As part of this large organization, **Unicoop Tirreno** is a great reality of organized distribution that is present in Tuscany, Lazio, Umbria and Campania with 112 stores, more than 770,000 members and approximately 6,300 employees. In 2007 exceeded 1.16 billion euro of total sales. To better meet the needs of different customer realities, Unicoop Tirreno manages different types of shops: small supermarket brand InCoop (64), the supermarkets (39), the hypermarkets brand Ipercoop (9).

In this context Unicoop Tirreno decided to develop Business Intelligenge solutions, reactive to market changes, and start the project *Business Intelligence and Data Warehouse (BI-Coop)*. The objectives of the BI-Coop project can be summarized as follows: (one) to create and populate a data warehouse from the operational data and to create interactive data reports (two) to develop forecasting models through the use of data mining technologies. In particular, data mining is used to predict customer defection and promotion sales previsions.

In this paper we describe the methodology and results obtained in order to obtain predictive models for promotion sales. This forecasting talk can use only future promotion features and sales data over the recent past. Moreover, in order to face this problem, we need to consider a side effect of promotions: the so-called “out of stock” phenomenon, i.e. the event a store is found out of products to sell before the promotion is finished, a signal of an incorrect storage estimate and a cause of lost income. Out of stocks are not currently tracked in the operational database, making it difficult to quantify the extent of the phenomenon.

The forecast models described in this work have been developed using SPSS Clementine (Clementine, 2009), and they will be used by Coop in marketing planning to optimize the storage of goods in shops and to develop new promotions.

2 Predictive data mining models for sales data

2.1 Choosing the optimal classification model

In general terms, the aim of Coop is to provide sales forecasts of promoted products. There exists a wide literature regarding sales forecasting. However, most approaches focus on time series analysis and prediction (for a survey, see for instance (Arsham, 1994)), which present two big drawbacks in our context:

- time series analysis is based on the extraction of trends and other behavioural models that are then matched to the current situation to forecast future values, implicitly assuming that a model that captured the past behaviour of the system is applicable to the present situation. However, in our context, we aim to predict what happens in response to an external event – a promotion – that naturally creates a discontinuity with the past behaviour of the series, therefore compromising the above mentioned regular evolution assumption;
- time series represent only part of the information we need to handle. In particular, in addition to the sales history of the promoted product, each promotion has its own characteristics which include, among the others, the promotion type, its size, its duration, and various descriptors of

the conditions under which the promotion takes place. An approach purely based on time series might not be able to take advantage of this important knowledge.

Other complex approaches like neural networks and regression-based models (linear and non-linear), can be appealing for what concerns potential accuracy, yet they usually yield models that are hard to inspect and interpret. On the opposite, in practice, the domain expert (which also plays the role of end user) requires that the resulting models can be understood, and possibly also amended, and therefore simplicity is a must. Moreover, another Coop requirement about the model is twofold: on one side, they wish to be able to quickly modify it in order to satisfy market department requests, and on the other side, they wish to be able to quickly recalculate it on order to consider new trends.

The most natural candidate satisfying all the requirements listed above is classification by means of decision trees, for instance computed by the standard C4.5 algorithm (Quinlan, 1992) or its variants. In particular, decision trees allow to:

- obtain a simple-to-read model, suitable for interaction with the domain expert analyst;
- take as input both the history of sales (for instance in the form of monthly, quarterly and yearly aggregates) and the context descriptors, whatever their nature (numerical, categorical);
- provide as output a set of classes, representing the sales bands that are forecast by the model. Determining the optimal number and size of such bands is a problem that is also tackled in this work.

2.2 Discretization-based classification

In this scenario, from the market department point of view, the goal is to predict a meaningful interval of sales, not just a number. Sales, in fact, are influenced also by largely unpredictable factors, such as social events, weather, traffic, an so on, making precise numeric predictions not meaningful. Hence, we have decided to discretize the objective function (sales amount) into a set of classes.

This step is critical and needs a good *trade-off* among:

- low number of classes (useful for classification algorithms and also for easy model evaluation)
- significance of the predicted value compared to storage choices
- distribution as uniform as possible between classes.

The particular contribution of our approach was to work directly with multi-class ordinals classifiers. In addition, the measures generally used, accuracy and standard deviation, do not perfectly fit our problem. Therefore, a specific measure for classifiers with ordinal classes is defined.

Concerning the out of stock phenomenon, the corporate data warehouse does not have data about the quantity of goods in storage. The occurrence of the out of stock event has been derived from an analysis of sales data. The definition was made in conjunction with experts, essentially by identifying sharp decreasing in sales quantity over each promotion day.

Once the models are built, the system is available through a simple web interface: it takes in input the main features of a promotion and outputs a prediction on the sales quantity or their percentage change. Finally, it also provides a risk factor for the out of stock event.

2.3 Related works on multi-class classification with ordinal classes

An ordinal quantity differs from a nominal one because it exhibits an order among the different values it can assume. An ordinal attribute could be, for example, a temperature measure represented by the values Hot, Mild and Cool. It is clear that there is an order among those values: Hot > Mild > Cool. Standard classification algorithms map a set of attribute values to a categorical target value. These algorithms generally are unable to use ordering information during the classification process and treat an ordinal target class attribute like a nominal class. However, some information is lost when this is done, information that can potentially improve the predictive performance of a classifier.

Real circumstances frequently involve situations exhibiting an order among different categories represented by the class attribute. There are many statistical approaches to this problem, but they are generally based on specific distributional assumptions for the class values (Herbrich, Graepel, & Obermayer, 1999). In recent years different approaches for ordinal classification were proposed using different approaches: decision trees (Potharst & Bioch, 2000; Frank & Hall, 2001), regression (Herbrich, Graepel, & Obermayer, 1999; Lin & Li, 2007; Rennie & Srebro, 2005), regression trees (Kramer et al., 2001), boosting (Freund, 2003), decision rules (Dembczyński, Kotłowski, & Słowiński, 2007). In each of these cases, the proposed algorithms or methodologies improve ordinary results by a marginal-to-moderate amount, usually not greater than a 5% gain in accuracy. Moreover, this is obtained through implementation costs and loss in flexibility or comprehension of results. In our applicative experience this kind of improvement is not critical – nor very significant – for the end user. Therefore, in this work we choose to use well know algorithm C4.5 (Quinlan, 1992), and define new measures of accuracy for evaluating the output models.

3 Data Mining on promotional sales

3.1 Data exploration

Coop has three distinct store formats, mainly characterized by their surface. The analysis focused on promotional sales of *hypermarket* stores (the largest format) using only promotions on food products. A first distribution analysis of volumes sales through a discretization in 25 equidistant (i.e., equal-size) bins is shown in Figure 1.

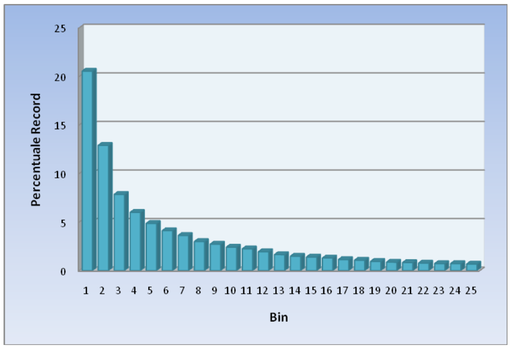


Figure 1: Sales volume distribution in hypermarkets with equal-size binning

This distribution should be interpreted as follows: 20.53% of the promotions in a single store sold between 0 and 24 items (the leftmost bar in the figure), the 12.89% sold between 25 and 49 items, and so on. The distribution shows a large number of promotions with a low volume of sales: in fact over 50% of the promoted products sold less than 125 pieces. This is an entirely unexpected result since these sales are calculated over a period of sales promotion for 15 days and for a large store.

A deeper analysis shows that there are many products with zero sales. Regardless of the possible reasons (incomplete population of the database, promotions never started in some store), for the purposes of our analysis it was decided to disregard promotions with less than 5 pieces sold. These promotions have been eliminated from the tables, obtaining the new distribution in Figure 2.

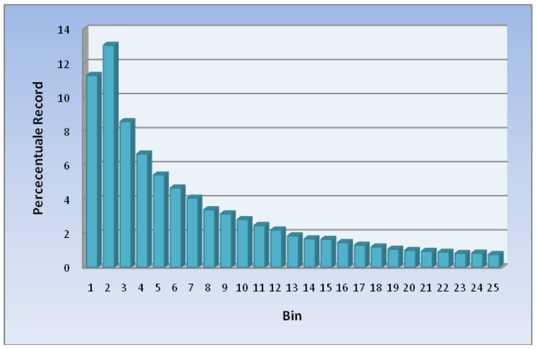


Figure 2: Sales volume distribution of promotions with al least 5 items sold

The classification models can be built following two alternative ways. The first option is to extract a single overall model for each promotion, calculating averages on sales and on *out of stock* over all stores involved in the promotion. With this more approximate approach, the output will provide the average values of sales over all stores, thus not providing an accurate information about how much goods have to be stocked in each store. The second option is to build an *ad hoc* model for each hypermarket: this solution requires to develop more models, nevertheless it produces closer previsions of actual sales for each store. Between these two possibilities we choose the second, since sales volumes on a single store basis are of greater interest for Coop.

3.2 Data preprocessing issues

The starting point for this work was the use of a corporate data warehouse, containing 1 year of sales in more than 100 stores. Among the several tables available, the most interesting for our purpose was the sales data table, characterized by a very large number of lines (926,774,117), since each line of each cash receipt is a record. The tables interest are 6 and are structured as in Figure 3.

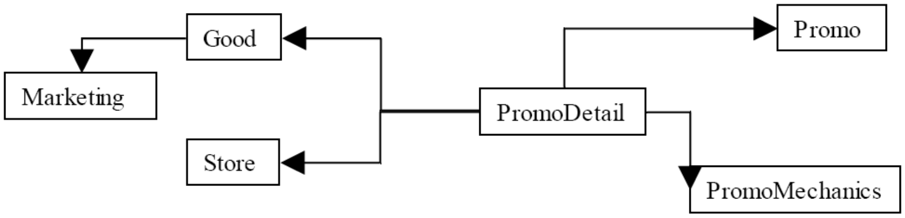


Figure 3: structure of relevant portion of the data warehouse

The selected information include the promotions, their type (or *mechanics*) and details, the goods involved, their position in the product classification taxonomy and, finally, the stores where promotions are performed.

Detail level options

Creating the *mining table*, i.e., the table that collects the information used in the model mining phase, needs to define the level of detail of the individual records and then determine the appropriate data aggregation. We have evaluated two different strategies: (a) one row for each promotion of each good, and (b) one row for each promo detail. In the first case it is expected to create a row of the table for each item that appears in a particular promotion. The advantage of this solution is the data storage on a global level, but it has the weak point that it can not verify if the promotion has been successful in each store. For example, if an article is promoted in a single Coop store, the sales data aggregation (and also of articles of the same marketing segment) does not highlight the real answer to the promotion. In the second case, a promo detail identifies an item that belongs to a promotion in a store. In this way we can get a detailed response to the promotion for every article in all the different shops where the same promotion was active. The final choice was for the second option.

Mining Table

We used data sales of 16 months in 134 stores (522,541,764 records). The data were aggregated into 4 time slots, collecting total sales information of an item for each store in a single row of the table. The use of aggregation creates a mining table of 240,059 rows. The fields of the table are the union of the fields present in the 6 separate tables shown in Figure 1, for a total of 62 fields, with the addition of 5 calculated fields that are described in Table 1.

Field name	**Description**
Vend_Art_3_1	Sales of the article from 3 months to 1 month before the promotion
Vend_Seg_3_1	Sales of the segment from 3 months to 1 month before the promotion
Vend_Art_1_0	Sales of the article in the last month before the promotion
Vend_Seg_1_0	Sales of the segment in the last month before the promotion
Giorni_Promozione	Duration of the promotion in days

Table 1: Derived attributes and their description

The target variables used to train the models are: (1) the sales amount of the promoted item and (2) the number of *out of stock* that occurred during the promotion.

3.3 Discretization

The target variable is continuous and to be able to use many classification algorithms it is necessary a discretization. Moreover, it is strongly skewed to low values and the distribution of sales volume is very sparse: the values range between 0 and 105,650, yet 80% of promotions sold less than 500 items.

A first attempt to discretize the target variable has been an equal-size binning. This type of discretization has certainly the advantage of being very simple and precise, but in our case produces too many classes. Even increasing the size of the bin, the number of classes remains large (unless we adopt extremely large intervals) and data are heavily unbalanced: this leads to a decrease of prediction accuracy.

What follows are some examples of discretization:

Bin width	Resulting n. of classes	Population within first 3 classes
10	965	18,38%
25	572	32,63%
50	382	47,28%
100	249	64,28%

Table 2: Sample discretization with equal-size bins

Classification algorithms are unable to generate high accuracy models over such a great number of classes. As alternative, we tried to performed an equal-frequency binning, but with not satisfactory result. For instance, using 20 bins the result is not particularly significant: first, it is non interesting for market analysis to know whether a product will sell between 6 and 14 items or between 14 and 23, and it is an extremely useless knowledge to know that it will sell more than 1600 items.

We manually refined the discretization trying to satisfy the following issue:

- Low number of classes (maximum 20)
- Significance of the predicted value with respect to subsequent storage choices
- Uniform distribution among classes

The chosen discretization, which will be used for the forecast model construction, is shown in the following figure (Figure 4) compared to the result of an equal-frequency binning.

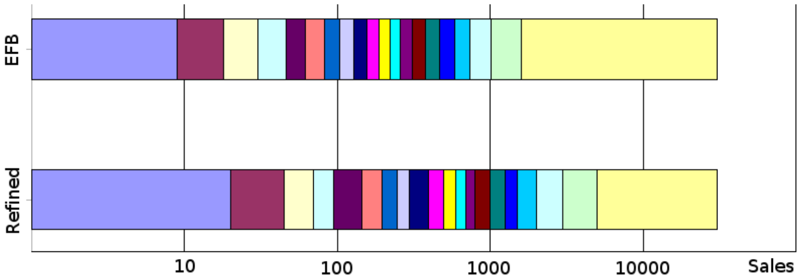


Figure 4: Discretization through equal-frequency binning (top) and its manual refinement (bottom)

The each bar represents the whole dataset, divided into 20 bins. The horizontal axis is in logarithmic scale, for better emphasizing lower value bins. As we can see, the refined discretization “moves” the bins towards higher values, thus providing a more detailed division for middle-high sales values. The same refined discretization is shown in Figure 5 as a bar plot.

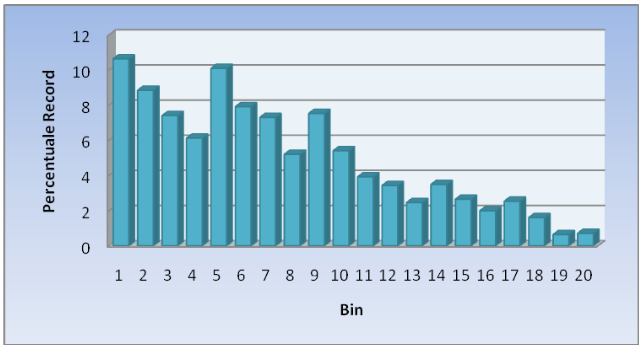


Figure 5: Distribution of sales volume discretized in 20 bins – refined discretization

In this work, we develope two separate models that differ on the variable to predict: *intervals in volume sales*, which used the discretization previously defined; and *response to the promotion*, which provides the change percentage in sales with respect to the previous time period. In both cases we are dealing with the creation of multi-class classifiers with ordinal classes.

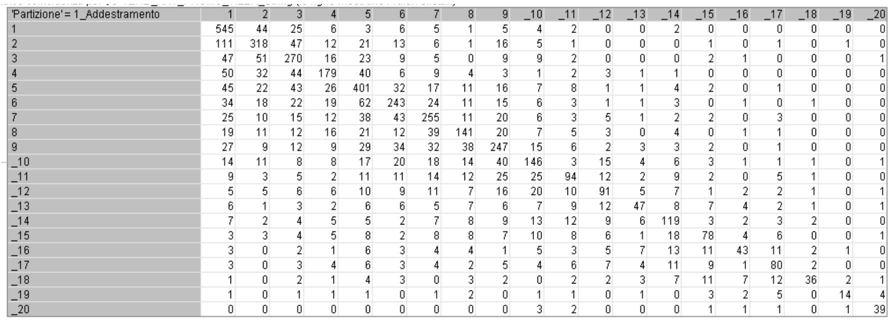
3.4 Predicting the sales volume class

Following a standard procedure, the input dataset was randomly partitioned into a *training set*, containing the 70% of available promotions, and a *test set*, containing the remaining 30%. The records of the former are used to build the classifier, while the records of the latter are exclusively used to validate the classifier.

The classification model was extracted by means of C4.5, a standard decision tree construction algorithm, applied with several different parameter settings, including various pruning strengths (how much the simplicity of the model is traded with its precision), minimum number of records that fall in each leaf of the tree (the larger it is, the smaller is tree), usage of boosting techniques, etc. The best performing model, which is described in the following, had moderate settings: pruning strength at 80%, a minimum of 3 records per leaf of the tree, and no boosting applied.

The resulting model reached an accuracy of 55,1% on the training set, which drops to 22,45% over the test set. The first percentage value indicates that the model extracted represents to a good level of detail the data distribution of the cases used to build it; the second value, correspondingly, confirms that the model is able to predict the behaviour of *new* promotions, whose class was unknown. We remark that, though the performances over the test set might appear low, it should be evaluated taking into consideration the high number of target classes (20), where a basic random classifier would score an accuracy of just 5%.

A much more detailed view of the performances is given by the confusion matrix in Table 3.



'Partizione'= 1_Addestramento	1	2	3	4	5	6	7	8	9	_10	_11	_12	_13	_14	_15	_16	_17	_18	_19	_20
1	545	44	25	6	3	6	5	1	5	4	2	0	0	2	0	0	0	0	0	0
2	111	318	47	12	21	13	6	1	16	5	1	0	0	0	1	0	1	0	1	0
3	47	51	270	16	23	9	5	0	9	9	2	0	0	0	2	1	0	0	0	1
4	50	32	44	179	40	6	9	4	3	1	2	3	1	1	0	0	0	0	0	0
5	45	22	43	26	401	32	17	11	16	7	8	1	1	4	2	0	1	0	0	0
6	34	18	22	19	62	243	24	11	15	6	3	1	1	3	0	1	0	1	0	0
7	25	10	15	12	38	43	255	11	20	6	3	5	1	2	2	0	3	0	0	0
8	19	11	12	16	21	12	39	141	20	7	5	3	0	4	0	1	1	0	0	0
9	27	9	12	9	29	34	32	38	247	15	6	2	3	3	2	0	1	0	0	0
_10	14	11	8	8	17	20	18	14	40	146	3	15	4	6	3	1	1	1	0	1
_11	9	3	5	2	11	11	14	12	25	25	94	12	2	9	2	0	5	1	0	0
_12	5	5	6	6	10	9	11	7	16	20	10	91	5	7	1	2	2	1	0	1
_13	6	1	3	2	6	6	5	7	6	7	9	12	47	8	7	4	2	1	0	1
_14	7	2	4	5	5	2	7	8	9	13	12	9	6	119	3	2	3	2	0	0
_15	3	3	4	5	8	2	8	8	7	10	8	6	1	18	78	4	6	0	0	1
_16	3	0	2	1	6	3	4	4	1	5	3	5	7	13	11	43	11	2	1	0
_17	3	0	3	4	6	3	4	2	5	4	6	7	4	11	9	1	80	2	0	0
_18	1	0	2	1	4	3	0	3	2	0	2	2	3	7	11	7	12	36	2	1
_19	1	0	1	1	1	0	1	2	0	1	1	0	1	0	3	2	5	0	14	4
_20	0	0	0	0	0	0	0	0	0	3	2	0	0	0	1	1	1	0	1	39

Table 3: Confusion matrix of the model over test set

Though difficult to interpret, the matrix shows a clear *diagonality*, which demonstrates a good quality of the model. A more understandable analysis of the results can be obtained by evaluating, for each prediction, the extent of mistakes, in terms of distance from the diagonal in the confusion matrix – equal to zero for correct predictions, and to the class displacement in other cases. This methodology will be introduced in detail in later sections of this chaper. The distribution of mistakes is shown in Figure 6.

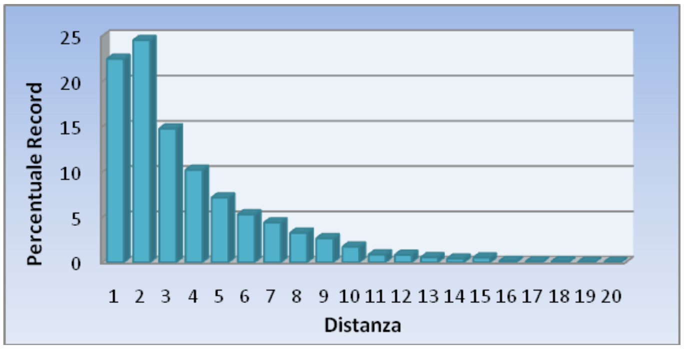


Figure 6: Class displacement distribution of predicted class vs. real class

Despite the apparently low precision of the classifier on the test set (22.45%), the chart above shows a strong concentration of records whose predicted class is very close to the real one. Indeed, most than 50% of records fall in the first three classes depicted in the chart.

3.5 Predicting the percentage variation of sales

In order to cope with the unbalancing of the sales data, we defined a new target function, aimed to forecast the percentage variation of sales during the promotion w.r.t. sales in the 15 days that preceded the promotion. This variation is able to effectively express the real response of customers to the promotions.

The following graph (Figure 7) provides a first rough summary of the distribution of sales variation during promotions – promotions where the promoted article was not sold at all during the preceding 15 days were not considered in the analysis.

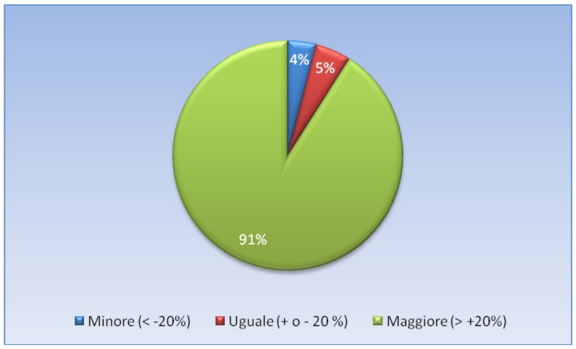


Figure 7: Percentage variation of sales under promotion

Beside confirming our obvious expectations – i.e., most articles sell more during promotions – the graph shows that usually such increase of sales is largely significant. It is also clear that the three classes used above are not informative enough for an effective goods stock planning. Further explorations of the data led to the definition of a set of percentage variation intervals, that reaches a trade-off among the following common properties:

- yield a precise information about the sales volume of each article, by means of small intervals;
- adopt a small number of classes, in order to ease the task of classification algorithms and to make the resulting predictive models easier to evaluate and to understand;
- obtain a class distribution as even as possible.

The discretized classes adopted in the successive model extraction phases were the result of both data inspection, consideration of the algorithms to be used, and the indications and practical requirements of the domain experts. The result consists of ten intervals of percentage sales variations, indicated in the following table:

Class	Contents and interpretation
1	Drop of sales (sales variation ≤ -20/%)
2	No variations (variation between +20% and -20%)
3	Small increase 1 (variation between + 20% and +100%)
4	Small increase 2 (variation between +100% and 200%)
5	Small increase 3 (variation between +200% and 300%)
6	Large increase 1 (variation between +300% and 500%)
7	Large increase 2 (variation between +500% and 1000%)
8	Large increase 3 (variation between +1000% and 1500%)
9	Extreme increase 1 (variation between +1500% and 2500%)
10	Extreme increase 2 (variation ≥ 2500%)

Table 4 - Classes for percentage variations of sales

Using the classes defined above, the three-classes distribution shown in Figure 6 is refined to the following one (Figure 8).

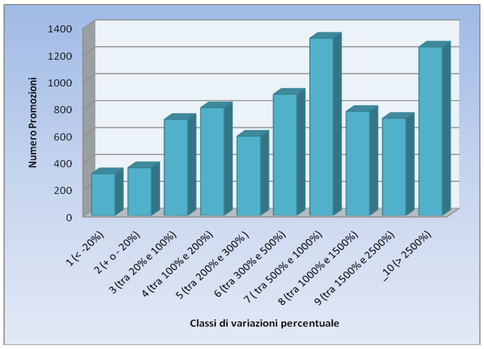


Figure 8 - Distribution of promotions along the percentage variation classes defined in Table 5

The figure shows peaks on classes 7 and 10, thus indicating that several promotions lead to a large or extreme increase of sales. At this point of the analysis, it is interesting to see which product categories are responsible of the peaks in Figure 7. A representative insight is depicted in the following graph (Figure 9), that represents the percentage coverage of the variation classes by the three sectors of the *food* category.

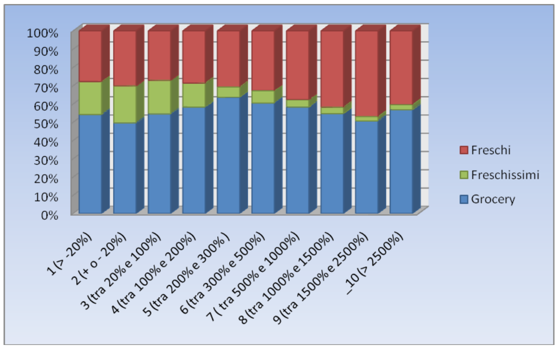
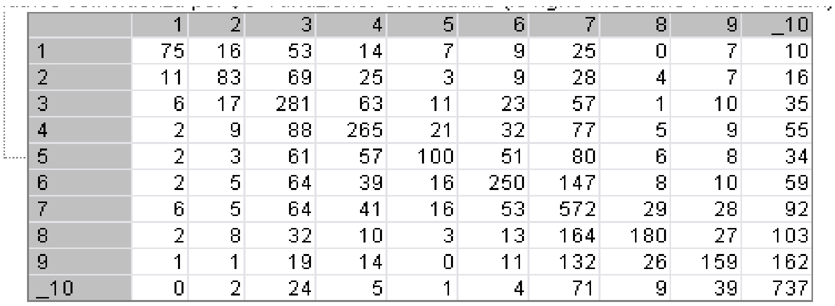


Figure 9: Sector distribution with classes of sales variations

This graph highlights some interesting facts. For instance, we can infer that promotions for the *freschissimi* sector (highly perishable products) have a relatively low response, since they are mostly concentrated over the first 4 classes, where the sales show only a small increase or even a drop. An opposite behaviour is that of the *freschi* sector (fresh products), whose promotions tend to fall in higher classes, in some cases reaching the same coverage of the (larger, in terms of promotions) *grocery* sector – e.g., in class 9.

The construction of this model follows the same steps and criteria as the model for sales volume. Indeed, the same partition into training and test set and the same predictive attributes are adopted. The main parameters used in the model construction include a small pruning factor (45) and a small minimum population for each leaf (3). The output model was produced without any kind of boosting, in the form of classification rules.

Accuracy reaches the 49.99% on the training set and 32.67% on the test set. Also in this case, such apparently low percentages are much larger than what a purely random classifier can achieve, due to the relatively high number of classes (10). A detailed view of the performances is provided in Table 5 as a confusion matrix, whose strongly diagonal structure attests the good quality of the output model.



	1	2	3	4	5	6	7	8	9	_10
1	75	16	53	14	7	9	25	0	7	10
2	11	83	69	25	3	9	28	4	7	16
3	6	17	281	63	11	23	57	1	10	35
4	2	9	88	265	21	32	77	5	9	55
5	2	3	61	57	100	51	80	6	8	34
6	2	5	64	39	16	250	147	8	10	59
7	6	5	64	41	16	53	572	29	28	92
8	2	8	32	10	3	13	164	180	27	103
9	1	1	19	14	0	11	132	26	159	162
_10	0	2	24	5	1	4	71	9	39	737

Table 5: Confusion matrix of the model

Anticipating again the more exhaustive discussion provided in Section 5, a useful summary of the confusion matrix is given in Figure 10, where the corresponding distribution of class prediction errors (i.e., the distances between predicted and actual classes) is plotted. As we can see, the distribution quickly drops after the first values, and indeed, ca. 76% of the records falls in the first three groups, i.e., 76% of predictions have error equal to or smaller than 2.

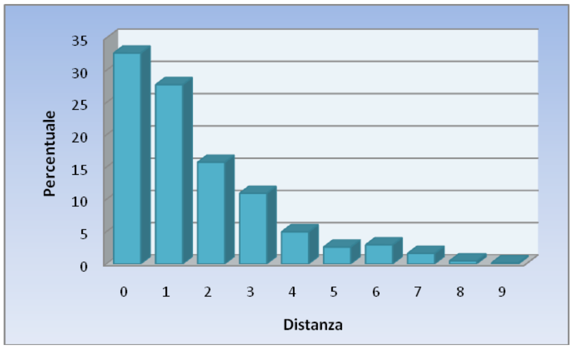


Figure 10: Distribution of class prediction error

3.6 Classification rules

Inspecting the rules that constitute the classification model obtained above, we can highlight some behaviours generally common to all shops. In particular, in Table 6 we provide a sample of such rules, also characterized by:

- support– number of cases where the rule applied
- confidence – accuracy of the rule

Rule	Support	Confidence	Confidence with error ≤ 1	Confidence with error ≤ 2
if CATEGORIA = ZUCCHERO E DOLCIFICANTI e FL_VOLANTINO = No e VEND_ART_1_0 > 37 **then** class = 2	47	23%	82%	93%
if CATEGORIA = 'ALIMENTI INFANZIA' e VEND_ART_1_0 > 275 **then** **class = 3**	138	50%	82%	97%
if CATEGORIA = CONSERVE DI FRUTTA e MESE = 8 **then** **class = 5**	113	24%	65%	86%
if CATEGORIA = YOGURT e DESCRIZ. = TAGLIO PREZZO e MESE = 9 e VEND_ART_1_0 > 54 e VEND_SEG_1_0 <= 4487 **then** **class = 6**	110	35%	65%	82%

if CATEGORIA = 'PASTA FRESCA' e
MESE = 10 e
VEND_ART_1_0 > 51
then class = 7
	42	38%	57%	78%

if FL_COOP = Si e CATEGORIA = BISCOTTI e FL_VOLANTINO = Si e VEND_ART_1_0 <= 275 **then** class = 8	52	25%	61%	78%

Table 6 - Classification rules with support and confidence, including limited tolerance to errors

Beside the basic confidence of rules, Table 6 reports the confidence values we obtain when a prediction error not larger than 1 (column 4) or 2 (column 5) is tolerated. In general, we can see that such small error tolerance increases the confidence considerably. Some of the rules above (which make use of the original names, in Italian) can be explained as follows:

- Rule 1: if more than 37 articles were sold in the last month before the promotion (vent_art_1_0 > 37) in the category "sugar" (categoria = zucchero e dolcificanti), and the promotion was not advertised in the advertising leaflets, the promoted item will sell the same or just a slightly higher amount than before the promotion (class = 2).
- Rule 2: similar to Rule 1, but for food for children and with an higher threshold (275) and a slightly higher gain in sales (class = 3).
- Rule 6: biscuits of the Coop brand that sold less than 25 pieces in the last month before the promotion and that was advertised in the leaflets will dramatically increase their sales (classe = 8, corresponding to a gain up to 1500%).

3.7 Comparing the two classification models

Both the classifiers generated, respectively aimed to predict the sales volume and its percentage variation w.r.t. recent past, have their pros and cons. In the first case, it is easy to find a satisfactory granularity of the classification classes, but, on the other hand, the resulting classes are strongly unbalanced. In the second case, the opposite happens, yielding balanced classes that, however, might gather largely different absolute values.

In this section we report the result of a comparison of the two models, aimed to measure the level of coherence of their predictions. The experiment was conducted on a single shop of the chain, and included the following steps, aimed to compare the forecast provided by each classifier for each promotion:

- the *percentage variation* model was applied to the whole dataset, thus associating an interval of percentage variations (e.g., [+100%,+200%]) to each promotion classified;
- for each prediction, a corresponding interval of (absolute) sales values is computed;
- since in general the interval obtained above cannot be traced back to a single class of the *absolute sales* model (discretization shown in Table 5), the set of the classes that overlap such interval is computed;
- the *absolute sales values* model is applied to the whole dataset;
- each prediction obtained above is checked against the corresponding result set derived from the predicted *percentage variation*. That is done by computing the distance between the first one and its closest element in the second result set.

The following figure (Figure 11) summarizes the results.

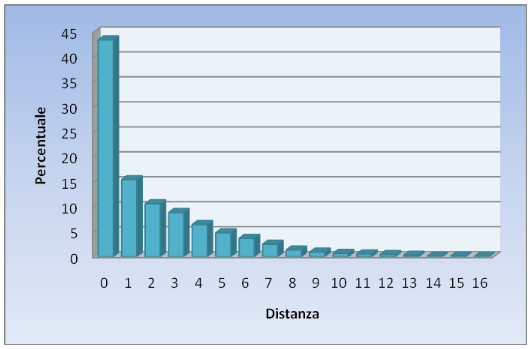


Figure 11: Coherence analysis between the two models

Almost half the records yield coherent results (i.e., they show a null distance), and moreover the distance distribution quickly decreases, showing that the two classifiers are strongly coherent.

4 Evaluation of multi-class classifiers

4.1 Problem definition

Evaluating the quality of the classifiers generated so far by means of a synthetic measure is a challenging problem. Indeed, most of them are defined over several *ordinal* classes, that is to say there is a natural order between classes, that should be considered in evaluating predictions. That is a direct consequence of the fact that such classes were obtain through discretization of a continuous variable.

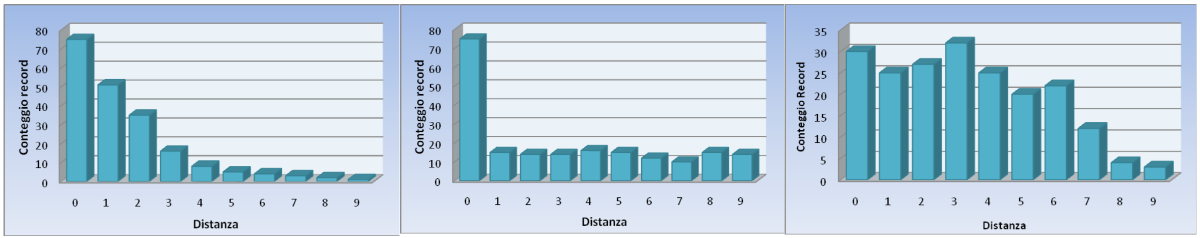
Accuracy, the most commonly used quality measure in model evaluation, only considers perfect matches between predicted and actual classes, counting all other cases simply as generic *misclassification errors*. On the contrary, our context suggests that the distance between the predicted class and the actual one should be part of the evaluation function, thus considering trade-offs between perfect matches and perfect mismatches. In the literature on classification models validation, there is a substantial lack of quality measures that solve these problems. Therefore, in the following we discuss some approaches that provide more precise evaluations of the quality of multi-class models with ordinal classes.

4.2 Distance from the diagonal

The basic step, already mentioned in the previous sections, consists in computing the distance between the predicted class and the actual class of each promotion in the dataset. Then, we plot the distribution of these values, which provides the means for easily checking the *diagonality* of the confusion matrix, and therefore to obtain a first qualitative assessment of the model under analysis.

Example 1.

In Figure 12 three sample distributions are plotted, corresponding to three fictitious classification models. In the following we present an example of how these distributions can help to infer the quality of the corresponding classifiers.



(a) a good classifier (b) a not so good classifier (c) a bad classifier

Figure 12 – Distribution of class error for three sample classifiers

With classifier (a) a large number of records falls in the first categories, corresponding to low distance values, and the distribution quickly drops for larger distances. That means that most records are correctly or quasi-correctly classified, and only a small number of them is associated to classes that are very different from the correct one. Therefore, (a) appears to be a good classifier.

Classifier (b) was built in such a way that its value for the traditional accuracy measure is the same as classifier (a). Indeed, leftmost bar on both the plots, corresponding to perfectly classified elements, show the same length. However, it is clear that classifier (a) should be preferred to (b), since the latter has an almost uniform error distribution for errors greater than zero, meaning that small and large errors are equally probable.

Finally, classifier (c) presents the same problems of (b) and, moreover, the peak on the zero-error bar is missing, meaning that the classifier does not guarantee a good accuracy (in the standard sense) nor a good percentage of quasi-correct classifications. This is a clear case of bad classifier.

4.3 Quantitative measures

Qualitative measures, though useful for general understanding of models and for spotting specific problems, do not provide an objective means for evaluating models or for comparing two of them. A measure should be defined that revises the traditional notion of accuracy in order to take into consideration the overall class errors distribution. In this section we provide two improvements of the standard definition of accuracy, that essentially compute an aggregate count of errors of the model, weighted on the basis of the gravity of each error.

Weights Vector-based approach

In this approach, we assume the user provides a vector *weights* of *N* positive (possibly null) values, where *N* is the number of classes in the classification problem, and such that each value *weights[i]* (*i=0, ..., N-1*) represents the weight associated to the errors of size *i*. For instance, *weights[3]* represents the weight associated to errors where the predicted class has a displacement of 3 classes w.r.t. the correct one, whichever is the direction of the displacement. Then, a vector *freq* of *N* elements is computed, each value *freq[i]* (*i=0, ..., N-1*) representing the number of promotions whose predicted class has distance *i* from the correct class. Then, we define the *vector-based accuracy* of the model as follows:

$$Accuracy^{vector} = \frac{\sum_{i=1}^{N} \left(freq[i] \cdot weights[i] \right)}{\sum_{i=1}^{N} freq[i]}$$

In principle, the values of *weights* should fall in the range *[0,1]*, in order to preserve the statistical meaning of accuracy, however, small negative values might be used to *penalize* some particular errors. Table 7 reports three examples of weight vectors, each putting a different emphasis to the different types of errors.

Distance	***Weights 1***	***Weights 2***	***Weights 3***
0	1	1	1
1	0	0,7	0,7
2	0	0,5	0,5
3	0	0,2	0,2
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	-0,2
8	0	0	-0,3
9	0	0	-0,5

Table 7 - Three sample weight vectors over 10 classes

- In *Weights 1*, only the records that are correctly classified are considered (distance zero). In this case, $Accuracy^{vector}$ = *Accuracy*, since the computed aggregate has the same value as the traditional accuracy.
- *Weights 2* considers as partially correct also the records that are classified *close* to the correct class. While a perfect prediction has weight 1, quasi-correct ones have a smaller one.
- *Weights 3* differs from the previous one because negative weights are associated to predictions that are very distant from the correct class. That strongly penalizes large errors.

The effects of these three different weight settings are reported in Table 8, which lists the corresponding values obtained for the three sample distributions shown in Figure 12 of Example 1.

	Distribution (a)	Distribution (b)	Distribution (c)
Weights 1	37,5%	37,5%	15,0%
Weights 2	65,7%	47,6%	33,7%
Weights 3	64,8%	40,9%	31,1%

Table 8 - Vector-based accuracy over distributions of Figure 12, with weights of Table 7.

The values obtained are coherent with the qualitative results discussed above:

- classifier (c) has a value of $Accuracy^{vector}$ smaller than classifiers (a) and (b), independently of the weights adopted;
- classifiers (a) and (b) have the same value of standard accuracy (equal to $Accuracy^{vector}$ computed with *Weights 1*) yet they differ significantly on the new accuracy measure. In particular, classifier (a) performs better and classifier (b), especially with *Weights 3,* which penalizes the significant amount of large errors yielded by classifier (b).

Limits of the vector-based approach

The simplicity of the vector-based model causes a few drawbacks that might limit its usefulness in some contexts:

- *there is no distinction between approximations by excess and by defect*. In some contexts it can be useful to stress the importance of one error against the other. For instance, in the case of stocking of articles, it might be convenient to prefer overestimates, that corresponds to the risk of having stock on hand after the end of a promotion, against underestimates, that would correspond to the risk of getting out of stock during the promotion.
- *the weight of an error does not directly depend on the actual class to be predicted.* Indeed, only the distance between the predicted class and the actual one is considered. In some cases, it might be useful to discriminate also w.r.t. the actual class. For instances, when the classes (as in our classification approach) are obtained through a discretization process that can yield discretization intervals of highly variable width, the gravity of an error could be dependant on such interval width, penalizing errors performed around larger bins.

Empirical experimentation on the field tells us that $Accuracy^{vector}$ yields a precise evaluation of the model quality in most practical cases. However, when the above mentioned limitations play a too strong role to be neglected, a generalization of the approach can be followed, which will be described later in this section.

Vector-based accuracy of the generated models

Following the procedure described above, we first choose a vector of weights, in order to decide how important should be each classification mistake. In our experiments, we selected two vectors, shown in Table 10, that mainly differ on the severity of penalties for errors of large extent.

Distance	Weights 1	Weights 2
0	1	1
1	0,9	0,85
2	0,8	0,7
3	0,7	0,6
4	0,6	0,5
5	0,5	0,4
6	0,3	0,3
7	0,2	0,2
8	0	0
9	0	0
10	0	0
11	0	-0,5
12	0	-0,5
13	0	-0,5
14	0	-0,5
15	0	-1
16	0	-1
17	0	-1
18	0	-1
19	0	-1

Table 9: Weights for vector-based accuracy of the sales forecasting model

Both vectors assign an almost linearly decreasing weight to errors from zero to 7. However, while in the first vector all other cases have a null weight, in the second one larger errors receive further penalties. A comparison of the results with the above weights is provided in Table 10, where also the standard accuracy is reported.

Traditional accuracy	Vector-based accuracy Weights 1	Vector-based accuracy Weights 2
22,45%	72,37%	65,83%

Table 10 - Accuracies for the sales forescasting model, using weights in Table 10.

Both vector-based accuracy are relatively high, therefore attesting the good quality of the model extracted. We can observe that the second measure, that penalizes large errors, yields a significant drop w.r.t. the first one, meaning that a significant amount of errors have a large extent, though not enough to seriously compromise the overall quality of the model.

A similar process was followed for the *percentage variation* model, for which two other vectors of weights are chosen, shown in Table 11.

Distance	Weights 1	Weights 2
0	1	1
1	0,8	0,8
2	0,5	0,5
3	0,3	0,3
4	0	0
5	0	0
6	0	-0,5
7	0	-0,8
8	0	-1
9	0	-1

Table 11: Weights for vector-based accuracy of the percentage variation model

The two vectors have a meaning similar to the previous case analyzed, the second vector stressing more the importance of errors of large extent. The corresponding vector-based accuracies are shown in Table 12, together with the value of traditional accuracy.

Traditional accuracy	Vector-based accuracy Weights 1	Vector-based accuracy Weights 2
32,70%	66,10%	62,60%

Table 12: Accuracies for the percentage variation model, using weights in Table 12.

In this case, while the accuracy is slightly lower than with the previous model, the performances are more *stable*. Indeed, the accuracy over *Weights 2* is very close to the one obtained over *Weights 1*, meaning that there were not many large errors to be penalized, and therefore the predicted classes are generally close to the true ones.

Matrix Vector-based approach

This solution is a generalization of the vector-based accuracy, that starts directly from the confusion matrix of a model and takes into consideration each single error type. In particular, a *NxN* matrix of weights, *mat_weights*, is provided by the user, such that each *mat_weights[i,j]* represents the weight associated with the cases (promotions, in our context) where the true class was *i* and the predicted class is *j*. The definition of the new accuracy is then computed by combining such matrix of weights with the confusion matrix (*mat_confusion*):

$$Accuracy^{matrix} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \left(mat\_confusion[i,j] \cdot mat\_weights[i,j] \right)}{\sum_{i=1}^{N} \sum_{j=1}^{N} mat\_confusion[i,j]}$$

Example 2.

Table 13 shows a sample confusion matrix that generated the first distribution discussed in Example 1.

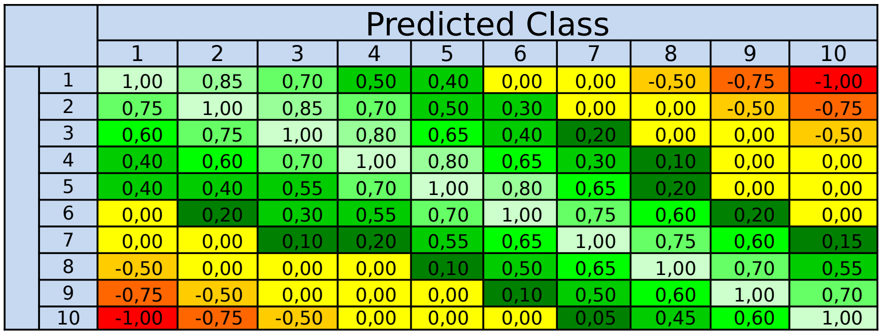
		Predicted class									
		1	2	3	4	5	6	7	8	9	10
	1	15	6	3	3	2	2	1	1	0	0
	2	8	14	4	3	1	1	0	0	1	0
	3	7	9	15	5	4	1	2	1	0	0
	4	3	4	6	16	3	4	2	3	2	2
	5	2	4	4	9	15	4	3	2	2	1
	6	4	3	2	5	8	19	3	4	0	1
	7	3	0	2	2	3	7	15	4	2	2
	8	1	0	4	3	3	5	8	18	2	1
	9	0	0	1	2	2	3	6	4	9	4
	10	1	0	1	0	1	2	3	7	6	14

Table 13 - Sample confusion matrix

As we can see, the matrix has a predominance of values along the diagonal, and moreover there is a higher density right below the diagonal, meaning that the model tends to predict values lower than the real ones. We apply the approach with two, slightly different matrices of weights, shown respectively in Table 14 and in Table 15.

		Predicted Class									
		1	2	3	4	5	6	7	8	9	10
	1	1,00	0,85	0,70	0,50	0,40	0,00	0,00	-0,50	-0,75	-1,00
	2	0,85	1,00	0,85	0,70	0,50	0,30	0,00	0,00	-0,50	-0,75
	3	0,70	0,85	1,00	0,80	0,65	0,40	0,20	0,00	0,00	-0,50
	4	0,50	0,70	0,80	1,00	0,80	0,65	0,30	0,10	0,00	0,00
	5	0,40	0,50	0,65	0,80	1,00	0,80	0,65	0,20	0,00	0,00
	6	0,00	0,30	0,40	0,65	0,80	1,00	0,75	0,60	0,20	0,00
	7	0,00	0,00	0,20	0,30	0,65	0,75	1,00	0,75	0,60	0,15
	8	-0,50	0,00	0,00	0,10	0,20	0,60	0,75	1,00	0,70	0,55
	9	-0,75	-0,50	0,00	0,00	0,00	0,20	0,60	0,70	1,00	0,70
	10	-1,00	-0,75	-0,50	0,00	0,00	0,00	0,15	0,55	0,70	1,00

Table 14: Matrix of weights 1



	Predicted Class									
	1	2	3	4	5	6	7	8	9	10
1	1,00	0,85	0,70	0,50	0,40	0,00	0,00	-0,50	-0,75	-1,00
2	0,75	1,00	0,85	0,70	0,50	0,30	0,00	0,00	-0,50	-0,75
3	0,60	0,75	1,00	0,80	0,65	0,40	0,20	0,00	0,00	-0,50
4	0,40	0,60	0,70	1,00	0,80	0,65	0,30	0,10	0,00	0,00
5	0,40	0,40	0,55	0,70	1,00	0,80	0,65	0,20	0,00	0,00
6	0,00	0,20	0,30	0,55	0,70	1,00	0,75	0,60	0,20	0,00
7	0,00	0,00	0,10	0,20	0,55	0,65	1,00	0,75	0,60	0,15
8	-0,50	0,00	0,00	0,00	0,10	0,50	0,65	1,00	0,70	0,55
9	-0,75	-0,50	0,00	0,00	0,00	0,10	0,50	0,60	1,00	0,70
10	-1,00	-0,75	-0,50	0,00	0,00	0,00	0,05	0,45	0,60	1,00

Table 15: Matrix of weights 2

The first matrix penalizes more the errors of larger extent, since the weights decrease as the distance from the main diagonal increases. Moreover, the extreme cells of the matrix (lower left and upper right corners) contain negative values, since these cases of very large errors highly degrade the usability of a predictive model. The second matrix presents a similar structure, but the records below the main diagonal are given stronger penalties, meaning that overestimates of the predicted class are preferred to underestimates. In the context of sales forecasting for stocking purposes, that corresponds to prefer avoiding out of stock situations that might compromise the effectiveness of a promotion.

The matrix-based accuracy values corresponding to the two weight matrices presented above are summarized in Table 16. The results show that the second set of weights introduces a loss of accuracy, due to the fact that the classification model analyzed, as mentioned above, tends to predict values lower than the actual classes, which are particularly penalized by the weight matrix.

Traditional accuracy	**Matrix-based accuracy Weights 1**	**Matrix-based accuracy Weights 2**
37,50 %	70,38%	66,50%

Table 16: Accuracies for the sales prediction model, using weights in Tables 15 and 16.

5 Data mining for ‘out of stock’ event

Every time the number of items available in a shop is less then its customer request, an out of stock event occurs. A consequence is that the good is not on shelf. Each out of stock is an income failure and sometimes it could be for significant amount. More, this could be a source of customer discontent that could lead to the shop abandonment.

This event is often connected to promotional occurrence and could derived from different causes. Most frequent cause is a wrong esteem of future sells that have, as consequence, a lower stoking number of items with respect to customer needs. Others causes could be a delay delivery from general warehouse to the shop or could be a delay delivery from the local warehouse to the shelves. For these reasons is it possible to define two different out of stock typology: warehouse level or shop level. In the first case the warehouse can not delivery items to the shop during the promotion days. This case could be treated only using stock data and is not possible in our scenario. In the second case, the shop is not furnished during a single day. This case could be treated starting from sell data and it is the analysis we propose.

Few models analyze this phenomenon and they discover out of stock making sell analysis during the all promotional period. In Coop usually warehouse provide with goods the shops everyday, so out of stock could be only during a single day. So, it is possible to have more than one out of stock during promotions days. A model *ad hoc* is required.

5.1 Out of Stock Model Definition

The model was made trying to capture all possible scenarios in which out of stock could occur. We model this phenomenon at the shop granularity using a division of a single day into four time slots: morning, lunch, afternoon and evening.

The model, at first approximation, is directed towards the detection of abrupt declines in sales between two contiguous time slots. The percentage change between contiguous time slots is analyzed and if this change exceeds a fixed threshold (default -90%) we assume an out of stock event took place (condition 1). In the sample table an out of stock occurs between *Lunch* and *Afternoon*.

Morning	Lunch	Afternoon	Evening
40	30	2	1

Table 17: Out of stock scenario

Considering only the percentage changes between the time slots, however, this first formulation of the model does not capture two possible anomalies: the resumption of sales and the existence of products with very low sales.

(Condition 2) If there is a sharp fall in sales in a intermediate slot (such as in condition 1), but then the sales increase, it is clear that this is not an out of stock event. To take in account this aspect, we need to verify that the sales at the time slot right after the fall stops below a threshold, in other words there is not a significant upturn in sales. In order to provide greater flexibility, this threshold value is calculated dynamically using the following formula:

$$OutOfStock^{threshold}=min(max(2,criticalValue),10)$$

The *criticalValue* is the sales value that caused the out of stock.

(Condition 3) The model with only the percentage changes (condition 1) recognizes an out of stock if a product sells 1 unit in all time slots except one in which it sells 0 (there is a percentage change of -100%). This is clearly a product that has very low sales and therefore not a case of rupture of stock. To properly handle such situations we introduce a new threshold that determines the minimum number of units sold which must precede an out of stock (Default value is 5).

(Condition 4) Condition 3 could mask some cases. In the scenario of Table 18 there is a strong decrease percentage between lunch and afternoon which satisfies the condition 1, but would not be regarded as out of stock as it has not checked the condition 3. In these cases the percentage decline is gradual and involves several time slots. To consider this case, too, a second threshold of percentage variation is introduced in the model and it is set to a lower value with respect of Condition 1 (default -70%).

Morning	Lunch	Afternoon	Evening
25	4	0	0

Table 18: Scenario with gradually decreasing sales

Now, in case there is a high percentage change that checks the condition 1 but not condition 3, the lowest threshold it is used, to identify decline in sales spanning several time slots. Choices and parameters for the thresholds was validated by a Coop marketing manager.

5.2 Model construction

Data Analysis of food sector in the *super* stores produces the following distribution for the number of out of stock events calculated as previously defined.

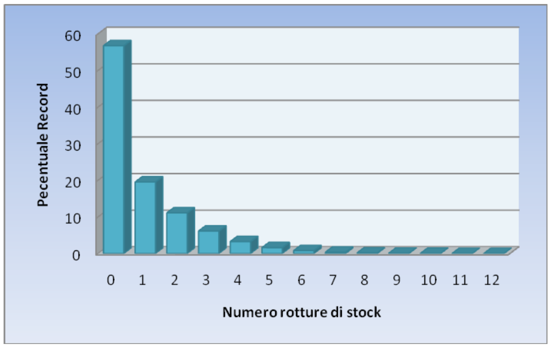


Figure 13: Distribution of out of stocks in the *Super* stores

The out of stock events occurs in 44% of cases. Moreover, for half of these cases it happens more than once. Whereas a promotion extends during 15 days and since the out of stock is a daily event, these numbers are surprising. Because of the strong unbalance of the distribution, it is not difficult to build a classifier using the number of out of stock in the promotion as objective function. Also in this case, it is necessary to use a discretization to allow the classifier to achieve good results. Two possible discretizations of the variable representing the number of out of stocks are possible: (1) in three classes (zero vs. one vs. more than one) or (2) in two classes (zero vs. at least one). The class distribution for both cases is provided in Figure 14.

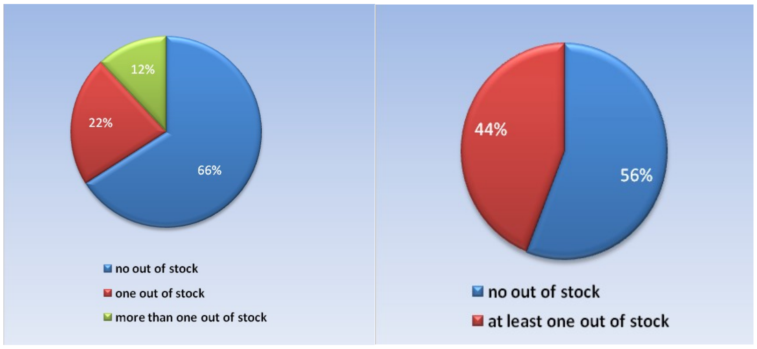


Figure 14 - Discretization for out of stock events with three (left) and two classes (right)

The first solution choice is certainly more convenient for the significance of the forecast: we have three different values that identify the degree of risk of out of stock event. The second solution, on the other hand, has a more balanced distribution (56% - 44%) and therefore it is more suitable for classification.

Both as regards the division of records between training set and test set, both for the choice of predictors used, the same considerations outlined in the previous chapter in connection with the construction of models for forecasting sales volumes are applied.

The main parameters chosen for the construction of this model are similar to those used for sales forecast: pruning severity = 45, min. number of cases per leaf = 3, no boosting, and output model in form of rules. The accuracy on the training set is found to be 75.14% and on the test set of 72.61%.

Generalizing the analysis of data for the food sector in the *hyper* stores, the following distribution results (Figure 15).

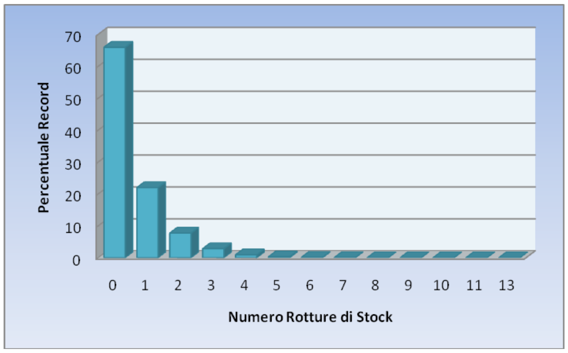


Figure 15: Percentage of out of stock for promotions in hypermarkets

As we can see in Figure 15, out of stock (65.9%) are 10% less and the number decreases more quickly. In this case too, the discretization was binary. The parameters used for the construction were the same as for the previous case, excepted for the pruning severity, set to 75 instead of 45. The accuracy found on the training set is 71.65% and on the test set of 67.58%. This accuracy decrease is an expected result, given the greater variety of goods and promotions taking place in *hypermarket* stores.

Creating the classifier as a set of association rules permits to verify the existence of some interesting phenomena. The following table shows some general rules that fits well, in which support and confidence values are calculated on all *hyper* stores data.

Rule	Support	Confidence
if CATEGORIA = GUSCIAME **then** class = 1	171	70%
if FL_VOLANTINO = Si **and** CATEGORIA = ACQUE **and** VEND_SEG_1_0 <= 78583 **then** class = 1	219	58%
if FL_VOLANTINO = Si **and** CATEGORIA = ALIMENTI INFANZIA **and** VEND_ART_3_1 > 142 **and** VEND_ART_1_0 > 96 **then** class = 1	677	65%
if FL_VOLANTINO = Si **and** CATEGORIA = ALIMENTI PER GATTI **and** VEND_SEG_1_0 > 4369 **then** class = 1	121	76%
if MESE = 12 **and** FL_VOLANTINO = Si **then** class = 1	3671	64%
if CATEGORIA = ZUCCHERO E DOLCIFICANTI **then** class = 0	127	86%
if RILEVANZA = IGIENE **and** CATEGORIA = ALIMENTI INFANZIA **and** VEND_ART_1_0 <= 45 **then** class = 0	1509	73%

Table 19: Classification rules for out of stock prediction in hypermarkets

- Rule 3 states that promotions involving food for children that were advertised in the leaflets, such that the product sold more than 96 pieces in the last month and more than 142 in the last three months before the promotion, are likely to go out of stock. We can notice that when the conditions of this rule are satisfied, also Rule 2 of Table 6 is, in which case we can expect that the sales prediction of the latter rule represents an underestimate of the real need of stocking for the promotion.
- Rule 6 states that the promotions involving sugar almost never go out of stock. In particular, we can see that this rule applies whenever Rule 1 of Table 6 does, in which cases we can expect that the latter rule provides a sufficient (possibly overestimated) forecast of sales.

To complete the analysis we have analyzed the rules created by the classifier for the super stores. Some examples are in Table 20, a sample of which are explained below.

Rules	Support	Confidence
if PRES_MKT = LEADER **and** VendSeg_1_0 > 479 **and** CATEGORIA = CAFFE' **then** class = 1	560	86 %
if TIPO_PROMO = Nazionale **and** CATEGORIA = ACQUE **then** class = 1	1731	87%
if TIPO_PROMO = Nazionale **and** VendArt_1_0> 59 **and** CATEGORIA = VINI DA TAVOLA **then** class = 1	798	87%
if TIPO_PROMO = Nazionale **and** FL_VOLANTINO = Si **and** CATEGORIA = BIRRA **then** class = 1	724	75%
if VendArt_1_0> 61 **then** class = 1	38826	79%
if MESE = 12 **and** CATEGORIA = GELATI **then** class = 0	379	95%
if PRES_MKT = COLEADER **and** CATEGORIA = YOGURT **then** class=0	359	88%

Table 20: Classification rules for out of stock prediction in *supermarkets*

- Rule 1 states that promoted coffee of a leader brand that sold more than 479 pieces in the last month will most likely go out of stock. Therefore, the estimates provided by the sales forecast models should generally be interpreted as underestimates.
- Rule 6 states that ice creams promoted during December almost never (95% of confidence) go out of stock, which was expected since the consumption and request of this kind of food is usually very limited during the winter season.

6 Deployment in a end-user application

The project ends with a first deployment of the system for end users, generally marketing staff managers. The application offers them a dedicated area for forecast sales analysis and for consulting historic promotions trend.

Users can connect to a *prevision* web page, shown in Figure 16 through an example, in which they can query the system about a sales forecast. They can choose the good to promote, the store, starting date, the duration and the mechanics of the promotion, and then launch the prediction step (bottom right button in Figure 16). The sample parameters in our example are the following:

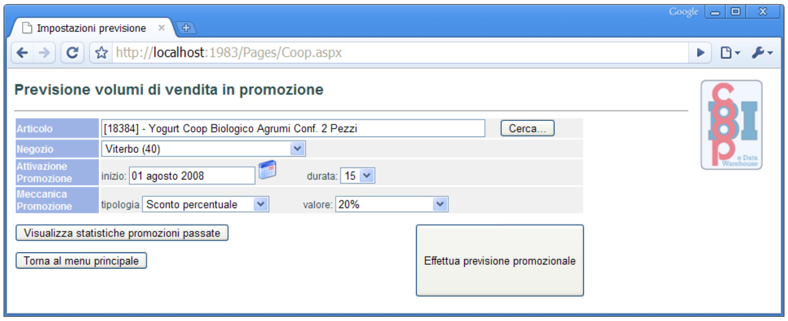
Good = yogurt [product id 18384]

Store = Viterbo (near Rome) [id store 40]

Starting date = 2008, August, 1st

Duration of promotion = 15 days

Promotion mechanics = 20% discount



Impostazioni previsione
http://localhost:1983/Pages/Coop.aspx
Google
Cerca...
Visualizza statistiche promozioni passate
Torna al menu principale
Effettua previsione promozionale
e Data Warehouse

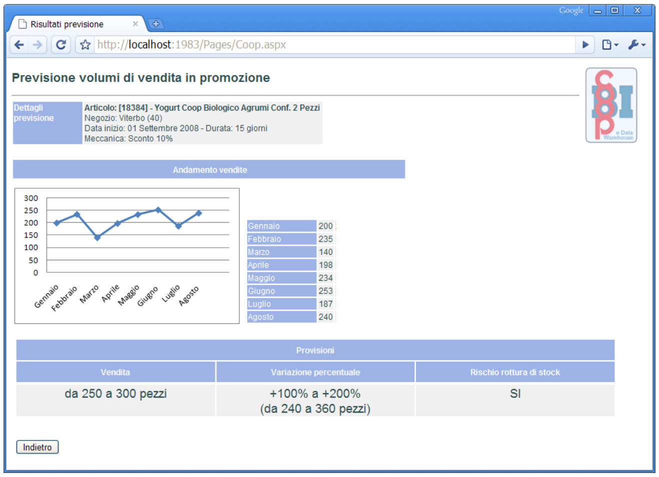
Previsione volumi di vendita in promozione

Articolo	[18384] - Yogurt Coop Biologico Agrumi Conf. 2 Pezzi
Negozio	Viterbo (40)
Attivazione Promozione	inizio: 01 agosto 2008 durata: 15
Meccanica Promozione	tipologia Sconto percentuale valore: 20%

Figure 16 - previsional Web page

At the end of the computation, results are shown in a trend of sales graph, as in figure 17. The time period analyzed covers several months preceeding the promotion. Forecasting of sales is expressed in terms of number and percentage range change over the previous period. It gives also an estimate of possible out of stock in the promotion time period.

The example output shows that the promoted yogurt will sell from 240 to 360 items (sales forecasting model), or it will increase its sell from 100% to 200% (percentage variation model). Moreover, the item results positive to out of stock prevision. This additional information could lead to understand that the forecast is underestimated.



Previsione volumi di vendita in promozione

Dettagli previsione	**Articolo: [18384] - Yogurt Coop Biologico Agrumi Conf. 2 Pezzi** Negozio: Viterbo (40) Data inizio: 01 Settembre 2008 - Durata: 15 giorni Meccanica: Sconto 10%

Andamento vendite

Gennaio	200
Febbraio	235
Marzo	140
Aprile	198
Maggio	234
Giugno	253
Luglio	187
Agosto	240

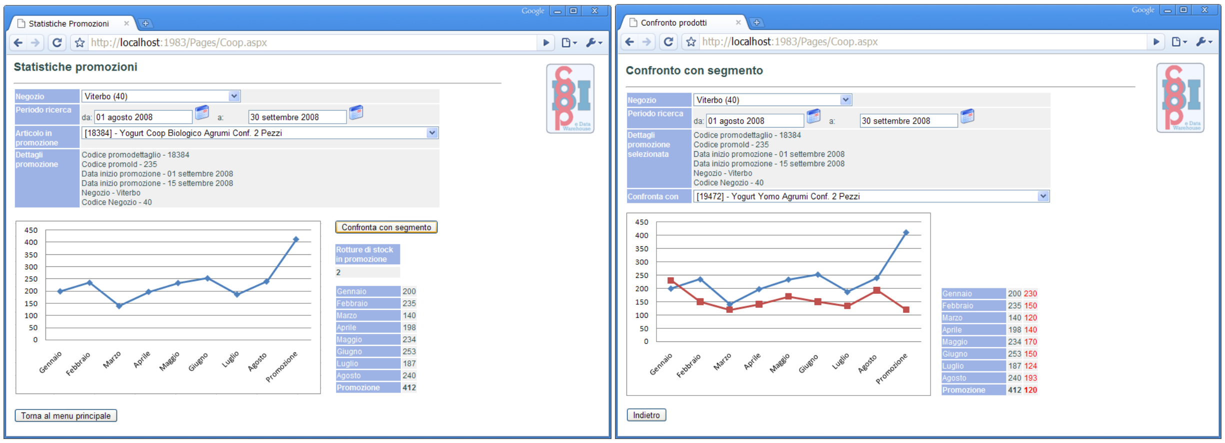
Provisioni

Vendita	Variazione percentuale	Rischio rottura di stock
da 250 a 300 pezzi	+100% a +200% (da 240 a 360 pezzi)	SI

Figure 17 – previsional web page – forecast prevision

User scan also access a statistical trend page in which to choose a the store, a time period and a good. The output is the good sales trend per month (in blue in the figure 18). Using the *compare* button a new trend is drawn (the red on the right side of the figure), representing the average sell of other goods in the same marketing segment.

In this way, marketing end users could browse the promotion data warehouse in a convenient analysis.



Statistiche promozioni

Negozio	Viterbo (40)
Periodo ricerca	da: 01 agosto 2008 a: 30 settembre 2008
Articolo in promozione	[18384] - Yogurt Coop Biologico Agrumi Conf. 2 Pezzi
Dettagli promozione	Codice promodettaglio - 18384 Codice promoId - 235 Data inizio promozione - 01 settembre 2008 Data inizio promozione - 15 settembre 2008 Negozio - Viterbo Codice Negozio - 40

Rotture di stock in promozione

2

Gennaio	200
Febbraio	235
Marzo	140
Aprile	198
Maggio	234
Giugno	253
Luglio	187
Agosto	240
Promozione	**412**

Confronto con segmento

Negozio	Viterbo (40)
Periodo ricerca	da: 01 agosto 2008 a: 30 settembre 2008
Dettagli promozione selezionata	Codice promodettaglio - 18384 Codice promoId - 235 Data inizio promozione - 01 settembre 2008 Data inizio promozione - 15 settembre 2008 Negozio - Viterbo Codice Negozio - 40
Confronta con	[19472] - Yogurt Yomo Agrumi Conf. 2 Pezzi

Gennaio	200	230
Febbraio	235	150
Marzo	140	120
Aprile	198	140
Maggio	234	170
Giugno	253	150
Luglio	187	124
Agosto	240	193
Promozione	**412**	**120**

Figura 18 – previsional web page – Statistics (left) and comparison (right)

7 Conclusion

The work described in this chapter is part of the BI-Coop project conducted by the KDD Laboratory at ISTI-CNR (Pisa, Italy) in collaboration with Unicoop Tirreno and is related to large retails. The goal was to analyze the trends in sales of articles offered for promotion to improve the quality of storage of such products. Our contribution is twofold. On the project/application side for the prediction of sales (both absolute value and in percentage value) and for the definition, analysis and prediction of out of stocks we track a methodology that allows to switch from business intelligence to data mining. On the research side, we defined a methodology for the qualitative and quantitative analysis of multi-class classifiers with ordinal classes. This seems to be essentially an open problem: literature does not explore convincing and consolidated solutions, although such cases are particularly frequent in the analysis of social phenomena, in which discretization of continuous values if often applied.

Despite the difficulty of the problem in terms of quantity and quality of data, excellent results were achieved both regarding the prevision of promotional sales, and the analysis of the out of stock phenomenon.

As future developments it is possible to refine the work considering the correlations that may exist between the various products in promotion and the others not included in a *what-if* scenario.

References

Arsham, H. (1994). Time-Critical Decision Making for Business Administration. (Web site checked on February 2nd, 2009) http://home.ubalt.edu/ntsbarsh/stat-data/Forecast.htm.

CRISP-DM (2009). http://www.crisp-dm.org/

Dembczyński, K., Kotłowski, W., & Słowiński, R. (2007). Ordinal Classification with Decision Rules. Mining Complex Data – ECML/PKDD Third International Workshop, MCD 2007, Warsaw, Poland, September 17-21, 2007, Revised Selected Papers. LNCS 4944. Springer.

Frank, E. & Hall, M. (2001). A simple approach to ordinal classification. 12th European Conference on Machine Learning Freiburg, Germany, September 5–7, 2001. LNCS 2167. Springer.

Freund, Y., Iyer, R., Schapire, R., Singer, Y. (2003). An efficient boosting algorithm for combining preferences. J. of Machine Learning Research 4, 933–969.

Herbrich, R., Graepel, T. & Obermayer, K. (1999) Regression models for ordinal data: A machine learning approach. Technical report, TU Berlin.

Kramer, S. , Widmer, G., Pfahringer, B., & DeGroeve, M. (2001). Prediction of ordinal classes using regression trees. Fundamenta Informaticae.

Larose, D.T. (2006). Discovering Knowledge in Data: An Introduction to Data Mining. Wiley.

Lin, H.T., Li, L. (2007): Ordinal regression by extended binary classifications. Advances in Neural Information Processing Systems 19, 865–872 (2007).

Oracle Warehouse Builder (2009). http://www.oracle.com/technology/products/warehouse/index.html

Oracle Sql Developer (2009). Web site (checked on March 4, 2009): http://www.oracle.com/ /technology/products/database/sql_developer/index.html

Potharst, R., & Bioch, J.C. (2000). Decision trees for ordinal classification. Intelligent Data Analysis, 4(2) 97–112.

Quinlan, J. R. (1992). C4.5 Programs for Machine Learning. Morgan Kaufmann.

Rennie, J., & Srebro, N. (2005). Loss functions for preference levels: Regression with discrete ordered labels. In: Proc. of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling.

SPSS Clementine (2009). http://www.spss.com/clementine/

Tan, P.-N., Steinbach, M., & Kumar, V. (2006). Introduction to Data Mining. Addison-Wesley.

Taylor, F.W. (1911). The Principles of Scientific Management. Harper. Digital version available on: http://melbecon.unimelb.edu.au/het/taylor/sciman.htm (Link checked on February 2nd, 2009).

Key Terms and Definitions

Classification Model: Classification is a procedure in which individual items are placed into groups based on quantitative information on one or more characteristics inherent in the items (referred to as traits, variables, characters, etc) and based on a training set of previously labeled items.

Classification Rule: Classification Rule is a popular and well researched method for discovering interesting relations between variables in large databases.

Data Mining: Data mining is the process of extracting hidden patterns from data. Data mining identifies trends within data that go beyond simple data analysis, through the use of sophisticated algorithms.

Discretization: Discretization concerns the process of transferring continuous models and equations into discrete counterparts.

Multi-class Classification: Multi-class Classification is a kind of Classification in which there are strictly more than two groups.

Out of Stock: Out of Stock is the event a store is found out of products to sell before the promotion is finished.

Sales Forecasting: Forecasting is the process of estimation in unknown situations. Sales Forecasting is used in the practice of Customer Demand Planning in every day business forecasting for manufacturing companies.