



Article Augmented Data and XGBoost Improvement for Sales Forecasting in the Large-Scale Retail Sector

Alessandro Massaro 🔍, Antonio Panarese *🔍, Daniele Giannone and Angelo Galiano 🔍

Dyrecta Lab, IT Research Laboratory, Vescovo Simplicio 45, 70014 Conversano, Italy; alessandro.massaro@dyrecta.com (A.M.); daniele.giannone@dyrecta.com (D.G.); maurizio.galiano@dyrecta.com (A.G.)

* Correspondence: antonio.panarese@dyrecta.com

Abstract: The organized large-scale retail sector has been gradually establishing itself around the world, and has increased activities exponentially in the pandemic period. This modern sales system uses Data Mining technologies processing precious information to increase profit. In this direction, the extreme gradient boosting (XGBoost) algorithm was applied in an industrial project as a supervised learning algorithm to predict product sales including promotion condition and a multiparametric analysis. The implemented XGBoost model was trained and tested by the use of the Augmented Data (AD) technique in the event that the available data are not sufficient to achieve the desired accuracy, as for many practical cases of artificial intelligence data processing, where a large dataset is not available. The prediction was applied to a grid of segmented customers by allowing personalized services according to their purchasing behavior. The AD technique conferred a good accuracy if compared with results adopting the initial dataset with few records. An improvement of the prediction error, such as the Root Mean Square Error (RMSE) and Mean Square Error (MSE), which decreases by about an order of magnitude, was achieved. The AD technique formulated for large-scale retail sector also represents a good way to calibrate the training model.

Keywords: extreme gradient boosting (Xgboost); regression; sales forecasting; data preprocessing; augmented data

1. Introduction

The large-scale retail trade has now established itself internationally for decades. The spread of this modern retail system is associated with a society characterized by a strong economy. Despite this, when a country's economy slows, this sector immediately suffers as sales drop. In this scenario, it is of great importance to have an innovative method that makes it possible to predict sales whether the trend of the economy is negative, steady, or even positive. Machine learning has tools capable of modeling business strategies taking into account consumer purchasing patterns. Machine learning is that branch of artificial intelligence that deals with algorithms capable of extracting useful information from the shapeless mass of data [1]. Machine learning algorithms learn from data in the training phase and are thus able to make predictions on experimental data [2,3]. Data mining (DM) combines the methodologies of machine learning and statistics in order to extract useful information from large amounts of data, such as databases and data warehouses [4]. DM uses artificial neural networks, classification trees, k-Nearest neighbor, Naive Bayes, and logistic regression to analyze data and generate classifications [5–7].

Data mining tools are also used by Business Intelligence (BI), which includes various processes and technologies used to collect, analyze, and make data usable in order to make better decisions [8]. BI also makes use of frontal analysis (dashboard), Executive Information System (EIS), and IT tools intended to support management activities, i.e., Decision Support System (DSS). BI performs a support function for business management with the aim of contributing to the qualitative improvement and greater speed of the organization's



Citation: Massaro, A.; Panarese, A.; Giannone, D.; Galiano, A. Augmented Data and XGBoost Improvement for Sales Forecasting in the Large-Scale Retail Sector. *Appl. Sci.* 2021, *11*, 7793. https://doi.org/ 10.3390/app11177793

Academic Editor: Alfonso Monaco

Received: 29 June 2021 Accepted: 20 August 2021 Published: 24 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). decision-making processes which can, therefore, increase its competitiveness [9,10]. The ultimate goal of BI is to provide support for decision making. Its functions allow information to be collected, processed, and used to support decision making [11]. In particular, in large-scale retail sector, the main activities of BI are: analysis of store layout efficiency, customer loyalty, warehouse management, and sales prediction.

The analysis of sales data allows for the prediction of market trends and helps companies to adopt adequate strategies to increase sales by acquiring greater competitiveness. A very powerful tool that is gaining ground in several areas is extreme gradient boosting (XGBoost). In fact, based on historical data, it is able to make precise predictions for diverse applications.

In [12] XGBoost algorithm was used in order to obtain weather predictions, such as medium-term precipitation prediction, while the authors of [13] used the XGBoost approach for predicting the short-term power consumption in rural and residential areas. The research studies [14,15] use an XGBoost-based model for deterministic wind power forecasting. The study [16] proposed a forecasting model based on a XGBoost—Long Short Term Memory (LSTM) algorithm in order to predict short-term and long-term sales changes. Similarly, the authors of [17] used XGBoost algorithms within a model they developed for Sales Time Series Forecasting. The authors of [18,19] combine cluster analysis and an XGboost algorithm to calculate the probability of different symptoms in young hypertensive patients and to predict heart disease, respectively. Finally, the research studies [20–22] propose different approaches based on an XGBoost technique for predicting, respectively, the onset of diabetes, crude oil prices, and urban fire accidents, which is useful for public safety. Different studies have been performed in Global Distribution System (GDS) and food factories, defining by DM algorithms Key Performance Indicators (KPIs) [23], product facing BI [9], quality processes [24,25], and sales prediction [26]. Concerning sales prediction, different DM algorithms can be adopted [27], including weather effect [28] and social trends [29]. The choice of the XGBoost algorithm is mainly due to the typically good performance in product classification [30] and sales prediction [31]. Following the experience of the authors in the large-retail sector and in sales prediction estimation, a research industry project focused on a supermarket store DSS was developed by defining multiple attributes by means of collaboration of the industry managers. The work is structured as follows:

- the DSS platform main functions are defined;
- the multiple attributes influencing the supermarket sales predictions are defined;
- the customer grid (customer segmentation) is structured;
- the AD approach able to increase the DSS performance is defined;
- the platform is implemented;
- the XGBoost algorithm is tested by proving the correct choice of the AD approach for the DSS.

Lack of data is a problem that machine learning algorithms face in many fields, especially when there are privacy concerns, such as in the medical field [32,33]. One method used in this case is the construction of synthetic or surrogate data using the copula method [34].

There are numerous research studies that have focused on data mining means in the field of sales forecasting [17,35–39]. In particular, the XGBoost [16,17], ref. [40] method is widely used for its performances, which also finds applications in e-commerce [40–42]. However, this powerful method requires a sufficient amount of data to be applied profitably. In our case the data were not enough, so we used the augmented data technique. This new technique, which has been tested in different sectors [43–45], is being applied for the first time for sales forecasting.

2. Methodology: Platform Design and Implementation and AD Approach

2.1. Platform Design and Implementation

Figure 1 illustrates the unified Modeling Language (UML) functional scheme of the implemented system applied for an industry working in the large-scale retail sector. The

main objective of the present research work was the implementation of an innovative BI platform. In detail, Figure 1 shows the Use Case Diagram (UCD) of the project. This diagram outlines the main functionalities that the intelligent platform is equipped with. Here, we focus on the functionality outlined in red, which is the prediction of sales through XGBoost algorithms. Figure 2 shows the UCD of the intelligent platform. The operator carries out the data entry. Then, the data is deposited in a data warehouse (DW) that allows for depositing the data coming from multiple sources and at the same time simplifies the data processing. A web app allows the customer to receive some personalized purchase information made possible by the customer profiling carried out by K-means algorithms. The various branches of the store send data in real time to a central database (DB). This is a Structured Query Language (SQL) DB which is not suitable for archiving historical data. Therefore, the massive data are sent on the Vertica DB. The resulting DW architecture is shown in Figure 3.



Figure 1. UML UCD of the implemented system. The paper discusses the implementation and results of the XGBoost algorithms used for sales prediction.



Figure 2. UML UCD of the implemented prototype platform with focus on data processing.



Figure 3. Data Warehouse Architecture.

We used the XGBoost algorithm in order to predict the future purchases of customers who have made a number of purchases equal to at least 30% of the number of days corresponding to the training period. On the basis of these predictions and the analysis of the association rules of the products purchased by the total customers of the supermarket, it is possible to formulate personalized offers for each customer. Customers were grouped into clusters, calculated on the basis of recency (distance from the last entry into the supermarket) and frequency (frequency of entry in the last 90 days). The prediction algorithm calculates the product that will be purchased by a customer who belongs to a cluster or to the entire clientele.

Figure 4 shows the flow chart of the implemented XGBoost algorithm. The algorithm takes as input the cluster on which to make the forecast, the training period, the number of days for which it carries out the prediction, and the code of the product in question. Finally, a check box permits one to select which of the following parameters must be used

by the algorithm to make the prediction. The multiple parameters used for the prediction estimation were the following:

- day of purchase of the product;
- week number;
- promotion status;
- weather conditions;
- perceived temperature;
- ratio between the number of receipts containing the specific product code and the number of receipts sold in total during the day;
- ratio between the number of customer receipts with the cluster identified and the total number of receipts for the day;
- ratio between the total receipts issued by the department of the specific product code and the number of total receipts issued during the day;
- unit price;
- ratio between the number of customers who bought that product and the total number of customers of the day;
- total number of customers of the day;
- product in promotion at the competitor;
- holiday week;
- week before a holiday week;
- holiday;
- day after a holiday;
- pre-holiday;
- closing day of the financial year.



Figure 4. Flow chart of XGBoost algorithm for cluster sales forecasting.

After building the dataset, a data pre-processing activity was carried out in order to apply Xgboost. Specifically, the techniques used were the following:

 one hot encoding, which is a mechanism that consists of transforming the input data into binary code; clustering, in order to obtain a relevant statistic, the clientele was divided into groups of customers with similar habits.

Unless the customer makes a large number of purchases in the training period, it is difficult to acquire an accurate prediction of the products they will buy. To overcome the lack of statistics on customers who are not very loyal, customers were grouped into clusters. Clusters are segments or groups of people with similar habits of recency and frequency. This segmentation allows for the increase of the statistics available as the data of the whole group are analyzed to predict what the whole group will buy. Subsequently, other algorithms could be used to distribute the predicted sales among the customers belonging to that cluster. Clustering is based on the calculation of the quintiles (20%, 40%, ... 100%) of the cumulative frequency of recency (number of days since the last order) and frequency (total number of orders in the last 90 days) for each customer who made purchases on the analysis day.

The quantiles were calculated over the last 90 days. On the basis of the recency and frequency values with respect to the quintile values, the following scores were defined for the recency (R) and for the frequency (F) which are associated with each customer:

$$R = \begin{cases} 5 \text{ for first quintile } (0 - 20\%) \\ 4 \text{ for second quintile } (20 - 40\%) \\ 3 \text{ for third quintile } (40 - 60\%) \\ 2 \text{ for fourth quintile } (60 - 80\%) \\ 1 \text{ for fifth quintile } (80 - 100\%) \end{cases}$$
$$F = \begin{cases} 1 \text{ for first quintile } (0 - 20\%) \\ 2 \text{ for second quintile } (20 - 40\%) \\ 3 \text{ for third quintile } (40 - 60\%) \\ 4 \text{ for fourth quintile } (60 - 80\%) \\ 5 \text{ for fifth quintile } (80 - 100\%) \end{cases}$$

Once the two scores have been calculated, the client's cluster was evaluated using the segmentation map (Figure 5). The segment is given by the Cartesian point (R, F) in the graph.



Figure 5. Segmentation map of customer clustering.

Furthermore, various optimization techniques of the predictive algorithm were elaborated, such as the Grid search technique and the Feature Selection.

Grid search, also called parameter sweep, is useful for finding the best combination of hyperparameters for the algorithm, which specifically are: max_depth, min_child_weight, learning_rate (eta), n_estimators, subsample, colsample_bytree, reg_lambda, and reg_alpha

in order to minimize the cost function as much as possible. The most-used error estimation parameters are: root mean square error (RMSE), mean square error (MSE), mean absolute error (MAE), mean square error on prediction (RMSEP), cross correlation coefficient (R), and standard deviation (SD) [46–49]. The reliability of the algorithms depends on the estimates of the errors that are made during data processing. The errors may also depend on the creation of inefficient training models. The cost function chosen to evaluate the accuracy of the implemented algorithm was RMSE.

The other technique used to increase the accuracy of the algorithm was the Feature Selection, which consists in the selection of the features with the highest relevance for the provided training set. In fact, there is no unique set of features suitable for predicting the quantities that are sold of all products. The paper concludes with a detailed paragraph on the testing phase of the algorithm and the results obtained.

We used the Python programming language to implement the XGboost algorithm. In particular, Python provides the XGBoost and scikit-learn open-source libraries. Using ScikitLearn Wrapper interface [50–52] for implementing XGBoost provides the opportunity to run the algorithm with the maximum number of parallel threads via the optional n_jobs parameter. The n_jobs = -1 option, permitting one to use all CPU cores, saves time, especially for trees that use a large number of nodes. Parallel computing is widely used in various sectors [12,53–55] because it optimizes computational costs.

We used the algorithm to be able to predict the future purchases of customers who had made at least 15 purchases in the last 90 days in order to create personalized offers for each customer. The offers are suggested by the analysis of the products purchased by the total customers of the supermarket. This analysis was performed using the association rules.

Appendix A lists the Python code of the adopted model, namely the piece of code where the XGBoost model is declared in terms of the hyperparameters: eta (alias learning_rate), max_depth, gamma, min_child_weight, subsample, colsample_bytree, alpha, reg_lambda, and n_estimators. For this purpose, the XGBRegressor and GridSearchCV classes of the scikit-learn library were used [56–58].

GridSearchCV allows the model to automatically optimize the value of the hyperparameters. In fact, initially, the value of the parameters was not defined, with the exception of the target 'reg: squarederror' and the value of n_j obs, which was set to -1 to parallelize the code. During the construction of GridSearchCV class, it is necessary to provide a dictionary of hyperparameters to evaluate. This class assigns the best combination of the hyperparameters used, as it builds and evaluates a model for each combination of parameters. In particular, the correct calibration of the values of the eta and subsample hyperparameters avoids any overfitting problems. This is achieved by choosing eta so that the update step is not too large. Furthermore, only a fraction of randomly extracted data were sampled. This fraction is decided by the subsample value. Finally, cross-validation was used to evaluate each individual model.

2.2. XGBoost Algorithm

The XGBoost algorithm is a machine-learning algorithm that is composed of a sequence of weak predictors. This algorithm was introduced by Chen [59] and perfects the gradient boosting algorithm. Gradient boosting is based on the iterative estimation of trees on the residuals obtained at each step and on adaptive updating of the estimates. Gradient boosting uses the gradient descent technique, for which the split that favors the approach to the minimum point of the objective function is chosen.

The optimizations of XGBoost, compared with gradient boosting, are due to its scalability, the use of parallel and distributed computing, tree pruning operations, management of missing values, and regularization to avoid overfitting and bias.

The input data are the set of values of the variables x_i used to make a prediction on the variable y_i :

$$\{(x_i - y_i)\}_{i=1}^n$$

This constitutes the training dataset. The model predicts the value of the variable y_i starting from the variables x_i , which are characterized by multiple features. In a linear regression problem, the predicted value is $\hat{y}_i = \sum_j \theta_j x_{ij}$, where θ_j is the weight of x_j . In a generic problem, θ denotes the parameters of the model.

The objective function, which measures the model's ability to fit the training data, consists of two terms:

$$Obj(\theta) = L(\theta) + \Omega(\theta)$$

where $L(\theta)$ is training loss function and $\Omega(\theta)$ is the regularization term. The loss function is a differentiable function that evaluates the prediction. The regularization term allows one to control the model's complexity and avoid overfitting.

XGBoost uses Taylor expansion of the loss function to write the objective function in the following form:

$$Obj(\theta) = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t)$$

where $g_i = \partial_{\hat{y}_i^{(t-1)}} L(y_i, \hat{y}_i^{(t-1)})$, while $h_i = \partial_{\hat{y}_i^{(t-1)}}^2 L(y_i, \hat{y}_i^{(t-1)})$. By defining the following quantities:

$$G_j = \sum_{i \in I_j} g_i$$
$$H_j = \sum_{i \in I_j} h_i$$
$$I_j = \{i | q(x_i) = j\}$$

we can write the optimal weight of the *j*-th leaf as $\theta_j = -\frac{G_j}{H_j + \lambda}$, where I_j is the instance set of the *j*-th leaf and $q(x_i)$ is a function that maps the data instance into a leaf of the tree, returning the index of the leaf itself. The objective function that optimizes the model is

$$Obj = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \gamma T$$

Here, *T* is the number of leaves that characterize the tree.

The computational cost of the algorithm is optimized due to the simultaneous training of all trees. Finally, the gain function, which allows one to evaluate the split candidates, is

$$Gain = \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{G_P^2}{H_P + \lambda}$$

where the first term is the contribution relative to the left nodes (subscript L), the second term is relative to the right nodes (subscript R), and the last term is the contribution of the parent leaf node (subscript P). The split condition that generates the greatest gain is chosen. This is a pruning strategy that optimizes a tree level to avoid overfitting.

For the implementation of the code in Python, the following open-source libraries were used:

- Numpy, which facilitates the use of large matrices and multidimensional arrays to
 operate effectively on data structures by means of high-level mathematical functions;
- Pandas, which allows the manipulation and analysis of data;
- XGBoost, which provides a gradient boosting framework;
- Sklearn, which provides various supervised and unsupervised learning algorithms.

A seed was awarded to initialize the pseudorandom number generator. The random module uses the seed value as the basis for generating a random number. This initialization allows for the replication of the same experiment, thus always obtaining the same results even if the process consists of random functions.

The training dataset was created using a shuffle function that allows for writing the lines in a random order.

2.3. AD Approach Applied for Large-Scale Retail Sector

In order to further improve the performance of the model, the augmented data technique was used. This approach was already validated in the case of LSTM neural networks [43–45], where data were artificially created to feed a data-poor training dataset. In [44], the training dataset was increased from 768 records to 10,000 records. In the present case, the dataset consisted of 7,212,348 records (instances or rows) relating to 897 sampling days. Each of this record refers to the quantity, which is expressed in pieces or weight according to the type of article, sold out of a single item in a store branch. For each article, the data are not sufficient to reach the desired accuracy, and for this reason increased data have been used. The following 14 parameters were the dataset attributes (variables or columns):

- 1. receipt identification code
- 2. receipt date
- 3. customer cluster
- 4. plu code
- 5. day of the week
- 6. week
- 7. quantity sold (number of items)
- 8. eventual promotion
- 9. store branch identifier
- 10. unit amount
- 11. measured quantity (kg)
- 12. product description
- 13. number of receipts
- 14. shop department identifier

Each of these observations refers to transactions relating to a specific product, sold on a specific day in a specific branch, by customers belonging to a specific cluster. Table 1 summarizes the statistics of the data.

Table 1. Statistics of the data.

Datum	Number
Number of records	7,212,348
number of daily customers	~1000
Sampling days	897
Number of products	30,159

Figure 6 shows the structure of the records that make up the original dataset and the methodology used to create artificial data. The technique consists of generating a certain number of receipts for each product in a sequential manner for each day. The identification number of the first artificial record is equal to the number of the last record of the original dataset increased by 1. For this reason, a reordering of the data was carried out based on the receipt date and not on the identification number.



Figure 6. Schematic of the Augmented Data (AD) generation mechanism.

The number of receipts generated for a certain product in a given day is chosen randomly by setting an upper and lower threshold. In a similar way to the technique used in [23], we used the random method random(a,b) to generate real numbers in the range (a,b). Furthermore, in generating the quantity sold and the number of receipts relating to a product, the average quantity sold for each item for that product was taken into account.

Figure 7 shows the statistic distribution of the number of receipts. The Log Scale option was used to make the histogram of this dataset attribute.



Figure 7. Statistic distribution of the attribute number of receipts.

3. Results and Discussion

The proposed methodological approach used in the development phase was validated and the functionalities implemented in the system were evaluated by comparing the results obtained from the prediction with the real results.

The test on the entire prototype system allowed the analysis of the results and the improvement of the prediction with the help of the Xgboost algorithm, whose behavior showed a very good performance both in training and prediction times and in the percentage of success in the predictions. In fact, the algorithm made it possible to create customized models, allowing any company to predict which customers will enter on the chosen day with a success rate of 70% and to predict what customers will buy, reaching a success rate of 35%.

Figure 8 shows an example of the test result of the predictive algorithm for micro clusters. The figure also shows the dashboard that was implemented to allow the manager to set up some parameters, which are useful for the prediction, such as PLU (Price look-up) code, training days, prediction days, etc. In this test, we predicted the quantities of the

product with code 3000055, which corresponds to a type of bread. The red plot shows the quantities actually sold and the blue plot represents the quantities foreseen by the algorithm.



Figure 8. Project platform dashboard implemented for setting the prediction parameters and for viewing the results.

The model was trained on 80% of the data and validated on the remaining 20%. The method was tested only for the item with plu equal to 3000055 in order to reduce the computational cost. This item appears on average in about 58 receipts with an average of 215 g of product sold per receipt. Respecting this average, 672 receipts for 30 consecutive days were created and added to the existing data.

Figure 9a,c shows the sales predictions of the item for two different periods of seven days each. The predictions were compared day by day with the real values recorded in these periods. The blue curve, which represents the predicted values, is very close to the red curve. As for the test, the prediction was made over a period of seven days following the period to which the data used for training and validation of the model refer. The model does not know what happened in the seven days and the comparison is only later.

Furthermore, Figure 9b,d show the same predictions made in Figure 9a,d, respectively, but this time performed with the AD technique. In this case, the blue prediction curve follows the real red curve very well. The improvement obtained with the use of the augmented data is confirmed by the RMSE value.

Figure 10 makes this enhancement tangible; in fact, the comparison of RMSE between the cases of Figure 9a,b is shown. In Figure 10, the blue curve refers to the case where training is performed using historical data, while the orange curve refers to the use of augmented data (AD). The AD technique improves the accuracy of the XGboost algorithm by about a factor between 2 and 3.



Figure 9. Predictions obtained for a given product for two different time intervals lasting 7 days, compared with the respective cases obtained with the augmented data technique: from 29 May to 4 June with training using the original data (**a**) and the augmented data (**b**); from 2 June to 8 June using the original data (**c**) and the augmented data (**d**).



Figure 10. Root Mean Square Error (RMSE) between predicted and actual values with reference to the predictions shown in Figure 8a,b.

The results of Figure 10 show that AD data provide a stable low RMSE output if compared with the oscillatory behavior obtained with the basic dataset. Table 2 reports the accuracy of the XGBoost model by comparing the values of RMSE and MSE between the original data and the augmented data for the test referred to in Figure 10. As for the

RMSE value reported in the first row of Table 2, this is calculated by averaging the seven values of Figure 10. Each of these values refers to a different run in which the prediction was made for a given day in the week shown in Figure 10. *MSE* values were calculated using the same runs. We computed the values shown in Table 2 as follows:

$$\overline{RMSE} = \frac{\sum_{i=1}^{7} (RMSE)_i}{7}$$
$$\overline{MSE} = \frac{\sum_{i=1}^{7} (MSE)_i}{7} = \frac{\sum_{i=1}^{7} (RMSE)_i^2}{7}$$

where the subscript *i* refers to the *i*-th day of the week.

Table 2. Accuracy results of the XGBoost model when trained with original or augmented data.

Accuracy	Original Data	Augmented Data
RMSE	0.63	0.28
MSE	0.93	0.092

When the model was trained by means of augmented data, the MSE improved by a factor of 10.

As in [60], we studied the accuracy of the model as the hyperparameters vary. The study confirmed that the optimal values are those obtained using GridSearchCV, however in the present case there was not a significant increase in performance as the value of the hyperparameters varied. Table 3 shows the values of the main hyperparameters obtained automatically for the prediction shown in Figure 9d and a brief description of their meaning (for a more detailed description, see [41]).

Table 3. Value of the main hyperparameters for the run of Figure 9d and their descriptions.

Hyperparameter	Value	Description ([41])
Eta (learning_rate)	0.1	Learning rate
<i>n</i> _estimators	175	Number of estimators
Max_depth	2	Maximum depth of the tree
Colsample_bytree	0.8	Subsample ratio of columns for each tree
Min_child_weight	1	Minimum sum of weights in a child
Alpha	0	Regularization term on weights
Lambda	1	Regularization term on weights

Finally, Figure 11 shows the Mean Square Error (MSE) obtained for the prediction, for the same product analyzed in the previous figures, as a function of the number of training days. The error had a minimum in the vicinity of 45 days. When the training took place over a very long period, i.e., over 120 days, the error grew as the products suffer the effect of seasonality and because the habits of customers can change over a long period. Moreover, other studies using other algorithms are oriented on the prediction of few days [61], thus confirming that there could be a lot of variables for long periods that cannot be controlled for the forecasting.

In order to fully validate the model, we carried out further tests on other inventory items from different categories. The corresponding departments were the following:

- 1. various foodstuffs (food and drinks, such as flour, pasta, rice, tomato sauce, biscuits, wine, vinegar, herbal teas, water, etc.);
- 2. delicatessen department (cured meats and dairy products);
- 3. fruit and vegetables (fresh fruit and vegetables, packaged vegetables);
- 4. bakery (bread, taralli, grated bread, bread sticks, etc.);
- 5. household products (napkins, handkerchiefs, toothpicks, shower gels, toothpaste, pet food, etc.);

- 6. frozen products (peas, minestrone, ice cream, etc.);
- 7. refrigerator packaged products (fresh milk, butter, dairy products, cheeses, packaged meats, etc.).





Department 1 is the largest as it contains most of the food and drinks, so we chose various products from this department. In detail, the items chosen were:

- 1. cream pack, flour pack, still water bottle, iodized sea salt pack;
- 2. mozzarella;
- 3. loose tomatoes;
- 4. bread;
- 5. paper towel;
- 6. frozen soup;
- 7. bottle of fresh milk, smoked scarmorze.

The implemented XGBoost algorithm predicts the quantity that will be sold in the selected period, which varies between 1 and 7 days. For packaged products, the number of sold pieces are predicted; for other products such as fresh products and delicatessen department products, however, the predicted quantities are expressed in Kg.

In all the presented cases, the improvement obtained with the use of the augmented data technique is evident.

Table 4 shows the *RMSE* values for the runs related to Figure 12. In the case of loose tomatoes and mozzarella, the quantities sold were very high, and therefore the *RMSE* values were quite high when compared with the other cases. In order to provide a parameter independent of the quantity sold, the *RMSEP* value was also calculated. This parameter normalizes each *i*-th error to the relative real value:

$$RMSEP = \sqrt[2]{\frac{\sum_{i=1}^{N} \left(\frac{y_i - \hat{y}_i}{y_i}\right)^2}{N}}$$
$$RMSE = \sqrt[2]{\frac{\sum_{i=1}^{N} \left(y_i - \hat{y}_i\right)^2}{N}}$$

where *N* = 7, while y_i and \hat{y}_i is the *i*-th real and predicted value, respectively.



Figure 12. Cont.



Figure 12. Cont.



Figure 12. Predictions obtained for different items in the inventory for time intervals lasting 7 days by using original data (left) and augmented data (right): cream pack (**a**,**b**), paper towel (**c**,**d**), flour pack (**e**,**f**), still water bottle (**g**,**h**), iodized sea salt pack (**i**,**j**), loose tomatoes (**k**,**l**), mozzarella in the delicatessen department (**m**,**n**), frozen soup (**o**,**p**), bottle of fresh milk (**q**,**r**), and smoked scarmorze (**s**,**t**).

Item	Original Data RMSE (RMSEP)	Augmented Data RMSE (RMSEP)
cream (Figure 12a,b)	1.56 (1.05)	0.53 (0.16)
paper towel (Figure 12c,d)	0.93 (0.65)	0.38 (0.19)
flour (Figure 12e,f)	2.10 (1.62)	0.84 (0.66)
still water (Figure 12g,h)	1.25 (0.89)	1.20 (0.88)
sea salt (Figure 12i,j)	1.00 (0.89)	0.38 (0.19)
tomatoes (Figure 12k,l)	4.00 (0.79)	1.70 (0.28)
mozzarella (Figure 12m,n)	2.13 (0.24)	0.72 (0.072)
frozen soup (Figure 120,p)	0.76 (0.76)	0.53 (0.53)
fresh milk (Figure 12q,r)	2.04 (0.44)	0.85 (0.13)
scarmorze (Figure 12s,t)	0.62 (1.96)	0.33 (0.40)

Table 4. Accuracy (RMSE and RMSEP) results of the XGBoost model when trained with original or augmented data for various items.

The computational cost of a single prediction run was on the order of one minute. Table 5 shows the execution time of the runs whose predictions are shown in Figures 9 and 12.

Table 5. Computational cost of the XGBoost model when trained with original or augmented data for various items.

Item	Original Data Run Time (s)	Augmented Data Run Time (s)
bread (Figure 9a,b)	84.54	69.77
bread (Figure 9c,d)	67.61	95.28
cream (Figure 12a,b)	55.69	69.65
paper towel (Figure 12c,d)	70.67	68.02
flour (Figure 12e,f)	79.33	62.17
still water (Figure 12g,h)	57.06	61.98
sea salt (Figure 12i,j)	57.94	59.19
tomatoes (Figure 12k,l)	109.36	84.59
mozzarella (Figure 12m,n)	55.77	58.65
frozen soup (Figure 120,p)	56.87	60.55
fresh milk (Figure 12q,r)	76.95	118.38
scarmorze (Figure 12s,t)	60.62	66.53

4. Conclusions

In this paper, we discussed the implementation of a predictive model, based on XGBoost algorithms, that was applied for forecasting sales in the large-scale retail sector. It is a multi-parameters model that allows for the consideration of various factors, such as, for example, the weather conditions that are very important for fresh products. However, the use of many parameters makes the method very precise, but it requires an adequate number of historical data to train the algorithm. To make up for the lack of data, the AD technique was used. This technique of enrichment of the training dataset has already been applied with excellent results to LSTM neural network models, but it was for the first time used for an XGBoost algorithm. Here the method, was tested on only one product, which is characterized by the 3000055 PLU code, with very promising results. The approach will be tested on the other products in future works. The test carried out for the prediction of the chosen product allowed us to validate the method through the calculation of MSE and RMSE errors. The results reported in Table 2 confirm that for both errors the accuracy improved due to the use of the technique. The improvement in the case of MSE is particularly advantageous; in fact, it decreased by an order of magnitude. Finally, the use of XGBoost algorithms corroborated by the AD technique allowed us to implement a sophisticated model, as it is based on the use of many parameters, but is also robust because it is capable of enriching the training dataset and obtaining an excellent accuracy measured in terms of MSE.

Author Contributions: Conceptualization, A.M., A.P. and D.G.; methodology, A.M.; software, D.G., A.P. and A.M.; validation, A.M. and A.G.; formal analysis, A.M.; investigation, A.M.; resources, A.G.; data curation, D.G. and A.M.; writing—original draft preparation, A.P. and A.M.; writing—review and editing, A.M., A.G. and A.P.; visualization, A.M. and A.P.; supervision, A.M.; project administration A.G.; funding acquisition, A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not Applicable, the study does not report any data.

Acknowledgments: The proposed work has been developed within the framework of the industry project titled: Piattaforma BigData/B.I. per il volantino dinamico profilato della clientela e per il supporto alle decisioni, basato su analisi predittive, analisi/utilizzo "social", e su ricerche di mercato in ambito GDO. 'GDO-DSS Dynamic Intelligence''' [BigData/B.I platform. for the dynamic flyer profiled for customers and suitable for decision support based on predictive "social" analysis and on market research in the GDO. 'GDO-DSS Dynamic Intelligence'].

Conflicts of Interest: There are no conflicts of Interest.

Appendix A

The following lines of code show the creation of the XGBoost model using the XGBRegressor and GridSearchCV classes of the scikit-learn library.

Table A1. XGBoost model.

```
model = XGBRegressor(booster = 'gbtree', n_jobs = -1,
learning_rate = gsearch234fin.best_params_['learning_rate'],
objective = 'reg:squarederror',
n_estimators = n_estimators_fit, max_depth = max_depth_fit,
min_child_weight = min_child_weight_fit, gamma = gamma_fit,
alpha = alpha_{fit}
reg_lambda = reg_lambda_fit, subsample = subsample_fit,
colsample_bytree = colsample_bytree_fit,
seed = 26
gsearch234 = GridSearchCV(
estimator = XGBRegressor(booster = 'gbtree', n_jobs = -1, objective = 'reg:squarederror',
learning_rate = learning_rate_fit,
n_{estimators} = n_{estimators_fit}
max_depth = max_depth_fit, subsample = subsample_fit,
colsample_bytree = colsample_bytree_fit,
min_child_weight = min_child_weight_fit,
gamma = gamma_fit, alpha = 0, reg_lambda = 1,
seed = 27),
param_grid = param_test234, iid = False, cv = 3, scoring = 'neg_mean_squared_error')
gsearch234.fit(selected_xtraintesttrain, selected_ytraintesttrain)
```

References

- 1. Raschka, S.; Mirjalili, V. Python Machine Learning, 3rd ed.; Packt: Birmingham, UK, 2019; p. 725, ISBN 978-1-78995-575-0.
- Zinoviev, D. Data Science Essentials in Python Collect → Organize → Explore → Predict → Value. The Pragmatic Bookshelf, 2016. Available online: https://pragprog.com/titles/dzpyds/data-science-essentials-in-python/ (accessed on 22 July 2021).
- Massaro, A.; Panarese, A.; Dipierro, G.; Cannella, E.; Galiano, A. Infrared Thermography and Image Processing applied on Weldings Quality Monitoring. In Proceedings of the IEEE International Workshop on Metrology for Industry 4.0 & IoT, Roma, Italy, 3–5 June 2020; pp. 559–564. [CrossRef]
- Palmer, A.; Jiménez, R.; Gervilla, E. Data Mining: Machine Learning and Statistical Techniques. In *Knowledge-Oriented Applications* in *Data Mining*; 2011. Available online: https://www.intechopen.com/books/1358 (accessed on 22 July 2021).

- Shmueli, G.; Patel, N.R.; Bruce, P.C. Data Mining for Business Intelligence. Concepts, Techniques, and Applications in Microsoft Office Excel with XLMiner; John Wiley & Sons: New Jersey, NJ, USA, 2007.
- Massaro, A.; Panarese, A.; Selicato, S.; Galiano, A. CNN-LSTM Neural Network Applied for Thermal Infrared Underground Water Leakage. In Proceedings of the IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0&IoT), Rome, Italy, 7–9 June 2021; pp. 219–224. [CrossRef]
- Massaro, A.; Panarese, A.; Galiano, A. Technological Platform for Hydrogeological Risk Computation and Water Leakage Detection based on a Convolutional Neural Network. In Proceedings of the IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0&IoT), Rome, Italy, 7–9 June 2021; pp. 225–230. [CrossRef]
- 8. Davenport, T.H. Competing on Analytics. Harvard Business Review. Harv. Bus. Rev. 2006, 84, 98–107. [PubMed]
- Massaro, A.; Galiano, A.; Barbuzzi, D.; Pellicani, L.; Birardi, G.; Romagno, D.D.; Frulli, L. Joint Activities of Market Basket Analysis and Product Facing for Business Intelligence oriented on Global Distribution Market: Examples of Data Mining Applications. *Int. J. Comput. Sci. Inform. Technol.* 2017, *8*, 178–183.
- Salonen, J.; Pirttimaki, V. Outsourcing a Business Intelligence Function. FeBR 2005, Frontiers of e-Business Research, 2005. Available online: https://researchportal.tuni.fi/en/publications/outsourcing-a-business-intelligence-function (accessed on 3 August 2021).
- 11. Turban, E.; Aronson, J.E. Decision Support Systems and Intelligent Systems, 6th ed.; Prentice-Hall: Hoboken, NJ, USA, 2001.
- 12. Aguasca-Colomo, R.; Castellanos-Nieves, D.; Méndez, M. Comparative Analysis of Rainfall Prediction Models Using Machine Learning in Islands with Complex Orography: Tenerife Island. *Appl. Sci.* **2019**, *9*, 4931. [CrossRef]
- 13. Liu, Z.; Yang, J.; Jiang, W.; Wei, C.; Zhang, P.; Xu, J. Research on Optimized Energy Scheduling of Rural Microgrid. *Appl. Sci.* 2019, 9, 4641. [CrossRef]
- 14. Phan, Q.; Wu, Y.K.; Phan, Q. A Hybrid Wind Power Forecasting Model with XGBoost, Data Preprocessing Considering Different NWPs. *Appl. Sci.* **2021**, *11*, 1100. [CrossRef]
- 15. Zheng, H.; Wu, Y. A XGBoost Model with Weather Similarity Analysis and Feature Engineering for Short-Term Wind Power Forecasting. *Appl. Sci.* **2019**, *9*, 3019. [CrossRef]
- 16. Wei, H.; Zeng, Q. Research on sales Forecast based on XGBoost-LSTM algorithm Model. J. Phys. Conf. Ser. 2021, 1754, 012191. [CrossRef]
- 17. Pavlyshenko, B.M. Machine-Learning Models for Sales Time Series Forecasting. Data 2019, 4, 15. [CrossRef]
- Chang, W.; Liu, Y.; Xiao, Y.; Xu, X.; Zhou, S.; Lu, X.; Cheng, Y. Probability Analysis of Hypertension-Related Symptoms Based on XGBoost and Clustering Algorithm. *Appl. Sci.* 2019, *9*, 1215. [CrossRef]
- 19. Yu, L.; Mu, Q. Heart Disease Prediction Based on Clustering and XGboost Algorithm. Comput. Syst. Appl. 2019, 28, 228–232.
- 20. Li, M.; Fu, X.; Li, D. Diabetes Prediction Based on XGBoost Algorithm. *IOP Conf. Ser. Mater. Sci. Eng.* 2020, 768. [CrossRef]
- Gumus, M.; Kıran, M.S. Crude Oil Price Forecasting Using XGBoost. In Proceedings of the International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey, 5–8 October 2017; pp. 1100–1103. [CrossRef]
- Shi, X.; Li, Q.; Qi, Y.; Huang, T.; Li, J. An accident prediction approach based on XGBoost. In Proceedings of the 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), Nanjing, China, 24–26 November 2017; pp. 1–7. [CrossRef]
- Massaro, A.; Dipierro, G.; Saponaro, A.; Galiano, A. Data Mining Applied in Food Trade Network. Int. J. Artif. Intell. Appl. 2020, 11, 15–35. [CrossRef]
- Massaro, A.; Galiano, A. Re-Engineering Process in a Food Factory: An Overview of Technologies and Approaches for the Design of Pasta Production Processes. *Prod. Manuf. Res.* 2020, *8*, 80–100. [CrossRef]
- Massaro, A.; Selicato, S.; Miraglia, R.; Panarese, A.; Calicchio, A.; Galiano, A. Production Optimization Monitoring System Implementing Artificial Intelligence and Big Data. In Proceedings of the IEEE International Workshop on Metrology for Industry 4.0 & IoT, Roma, Italy, 3–5 June 2020; pp. 570–575. [CrossRef]
- Galiano, A.; Massaro, A.; Barbuzzi, D.; Pellicani, L.; Birardi, G.; Boussahel, B.; De Carlo, F.; Calati, V.; Lofano, G.; Maffei, L.; et al. Machine to Machine (M2M) Open Data System for Business Intelligence in Products Massive Distribution oriented on Big Data. *Int. J. Comput. Sci. Inform. Technol.* 2016, 7, 1332–1336.
- Massaro, A.; Maritati, V.; Galiano, A. Data Mining Model Performance of Sales Predictive Algorithms Based on Rapidminer Workflows. Int. J. Comput. Sci. Inf. Technol. 2018, 10, 39–56. [CrossRef]
- Massaro, A.; Barbuzzi, D.; Vitti, V.; Galiano, A.; Aruci, M.; Pirlo, G. Predictive Sales Analysis According to the Effect of Weather. In Proceedings of the 2nd International Conference on Recent Trends and Applications in Computer Science and Information Technology, Tirana, Albania, 18–19 November 2016; pp. 53–55.
- 29. Massaro, A.; Vitti, V.; Galiano, A.; Morelli, A. Business Intelligence Improved by Data Mining Algorithms and Big Data Systems: An Overview of Different Tools Applied in Industrial Research. *Comput. Sci. Inf. Technol.* **2019**, *7*, 1–21. [CrossRef]
- Massaro, A.; Vitti, V.; Mustich, A.; Galiano, A. Intelligent Real-time 3D Configuration Platform for Customizing E-commerce Products. Int. J. Comput. Graph. Animat. 2019, 9, 13–28. [CrossRef]
- Masaro, A.; Mustich, A.; Galiano, A. Decision Support System for Multistore Online Sales Based on Priority Rules and Data Mining. *Comput. Sci. Inf. Technol.* 2020, *8*, 1–12. [CrossRef]
- 32. El-Bialy, R.; Salamay, M.A.; Karam, O.H.; Khalifa, M.E. Feature Analysis of Coronary Artery Heart Disease Data Sets. *Procedia Comput. Sci.* **2015**, *65*, 459–468. [CrossRef]

- Sabay, A.; Harris, L.; Bejugama, V.; Jaceldo-Siegl, K. Overcoming Small Data Limitations in Heart Disease Prediction by Using Surrogate Data. SMU Data Sci. Rev. 2018, 1. Available online: https://scholar.smu.edu/datasciencereview/vol1/iss3/12 (accessed on 22 July 2021).
- 34. Li, H.; Xiong, L.; Jiang, X. Differentially Private Synthesization of Multi-Dimensional Data using Copula Functions. *Adv Database Technol.* **2014**, 475–486. [CrossRef]
- Akshay, K.; Akhilesh, V.; Animikh, A.; Chetana, H. Sales-Forecasting of Retail Stores using Machine Learning Techniques. In Proceedings of the 3rd IEEE International Conference on Computational Systems and Information Technology for Sustainable Solutions, Bengaluru, India, 20–22 December 2018; pp. 160–166. [CrossRef]
- 36. Huang, W.; Zhang, Q.; Xu, W.; Fu, H.; Wang, M.; Liang, X. A Novel Trigger Model for Sales Prediction with Data Mining Techniques. *Data Sci. J.* 2015, 14. [CrossRef]
- Gao, M.; Xu, W.; Fu, H.; Wang, M.; Liang, X. A Novel Forecasting Method for Large-Scale Sales Prediction Using Extreme Learning Machine. In Proceedings of the Seventh International Joint Conference on Computational Sciences and Optimization, Beijing, China, 4–6 July 2014; pp. 602–606. [CrossRef]
- Kuo, R.; Xue, K. A Decision Support System for Sales Forecasting through Fuzzy Neural Networks with Asymmetric Fuzzy Weights. *Decis. Support Syst.* 1998, 24, 105–126. [CrossRef]
- 39. Hill, T.; Marquez, L.; O'Connor, M.; Remus, W. Artificial Neural Network Models for Forecasting and Decision Making. *Int. J. Forecast.* **1994**, *10*, 5–15. [CrossRef]
- Liu, C.-J.; Huang, T.-S.; Ho, P.-T.; Huang, J.-C.; Hsieh, C.-T. Machine Learning-Based E-Commerce Platform Repurchase Customer Prediction Model. PLoS ONE 2020, 15, e0243105. [CrossRef]
- Ji, S.; Wang, X.; Zhao, W.; Guo, D. An Application of a Three-Stage XGBoost-Based Model to Sales Forecasting of a Cross-Border E-Commerce Enterprise. *Math. Probl. Eng.* 2019, 2019, 1–15. [CrossRef]
- 42. Song, P.; Liu, Y. An XGBoost Algorithm for Predicting Purchasing Behaviour on E-Commerce Platforms. *Teh. Vjesn. Tech. Gaz.* 2020, 27, 1467–1471. [CrossRef]
- Massaro, A.; Panarese, A.; Gargaro, M.; Vitale, C.; Galiano, A.M. Implementation of a Decision Support System and Business Intelligence Algorithms for the Automated Management of Insurance Agents Activities. Int. J. Artif. Intell. Appl. 2021, 12, 1–13. [CrossRef]
- 44. Massaro, A.; Maritati, V.; Giannone, D.; Convertini, D.; Galiano, A. LSTM DSS Automatism and Dataset Optimization for Diabetes Prediction. *Appl. Sci.* **2019**, *9*, 3532. [CrossRef]
- 45. Massaro, A.; Panarese, A.; Gargaro, M.; Colonna, A.; Galiano, A. A Case Study of Innovation in the Implementation of a DSS System for Intelligent Insurance Hub Services. *Comput. Sci. Inform. Technol.* **2021**, *9*, 14–23. [CrossRef]
- Shcherbakov, M.V.; Brebels, A.; Shcherbakova, N.L.; Tyukov, A.P.; Janovsky, T.A.; Kamaev, V.A. A Survey of Forecast Error Measures. World Appl. Sci. J. 2013, 24, 171–176. [CrossRef]
- 47. Syntetos, A.A.; Boylan, J.E. The Accuracy of Intermittent Demand Estimates. Int. J. Forecast. 2004, 21, 303–314. [CrossRef]
- 48. Mishra, P.; Passos, D. A Synergistic Use of Chemometrics and Deep Learning Improved the Predictive Performance of near-Infrared Spectroscopy Models for Dry Matter Prediction in Mango Fruit. *Chemom. Intell. Lab. Syst.* **2021**, 212, 104287. [CrossRef]
- 49. Panarese, A.; Bruno, D.; Colonna, G.; Diomede, P.; Laricchiuta, A.; Longo, S.; Capitelli, M. A Monte Carlo Model for determination of binary diffusion coefficients in gases. *J. Comput. Phys.* **2011**, *230*, 5716–5721. [CrossRef]
- 50. Upadhyay, D.; Manero, J.; Zaman, M.; Sampalli, S. Gradient Boosting Feature Selection with Machine Learning Classifiers for Intrusion Detection on Power Grids. *IEEE Trans. Netw. Serv. Manag.* **2020**, *18*, 1104–1116. [CrossRef]
- 51. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Du-bourg, V.; et al. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2015**, *12*, 2825–2830.
- 52. Charoen-Ung, P.; Mittrapiyanuruk, P. Sugarcane Yield Grade Prediction using Random Forest and Gradient Boosting Tree Techniques. In Proceedings of the 15th International Joint Conference on Computer Science and Software Engineering (JCSSE), Nakhon Pathom, Thailand, 11–13 July 2018; pp. 1–6. [CrossRef]
- 53. Panarese, A.; Bruno, D.; Tolias, P.; Ratynskaia, S.; Longo, S.; De Angelis, U. Molecular Dynamics Calculation of the Spectral Densities of Plasma Fluctuations. *J. Plasma Phys.* **2018**, *84*, 905840308. [CrossRef]
- 54. Tolias, P.; Ratynskaia, S.; Panarese, A.; Longo, S.; De Angelis, U. Natural fluctuations in un-magnetized and magnetized plasmas. *J. Plasma Phys.* **2015**, *81*, 905810314. [CrossRef]
- 55. Twomey, J.; Smith, A. Performance Measures, Consistency, and Power for Artificial Neural Network Models. *Math. Comput. Model.* **1995**, *21*, 243–258. [CrossRef]
- Phan, Q.-T.; Wu, Y.-K. A Comparative Analysis of XGBoost and Temporal Convolutional Network Models for Wind Power Forecasting. In Proceedings of the International Symposium on Computer, Consumer and Control (IS3C), Taichung City, Taiwan, 13–16 November 2020; pp. 416–419. [CrossRef]
- Memon, N.; Patel, S.B.; Patel, D.P. Comparative Analysis of Artificial Neural Network and XGBoost Algorithm for PolSAR Image Classification. In *Pattern Recognition and Machine Intelligence*; PReMI 2019. Lecture Notes in Computer Science; Deka, B., Maji, P., Mitra, S., Bhattacharyya, D., Bora, P., Pal, S., Eds.; Springer: Basel, Switzerland, 2019; Volume 11941. [CrossRef]
- 58. Nelli, F. Machine Learning with Scikit-Learn. Python Data Anal. 2015, 237–264. [CrossRef]
- 59. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [CrossRef]

- 60. Massaro, A.; Maritati, V.; Savino, N.; Galiano, A.; Convertini, D.; De Fonte, E.; Di Muro, M. A Study of a Health Resources Management Platform Integrating Neural Networks and DSS Telemedicine for Homecare Assistance. *Information* **2018**, *9*, 176. [CrossRef]
- 61. Massaro, A.; Vitti, V.; Galiano, A. Model of Multiple Artificial Neural Networks Oriented on Sales Prediction and Product Shelf Design. *Int. J. Soft Comput. Artif. Intell. Appl.* **2018**, *7*, 1–19. [CrossRef]