FORECASTING of WALMART SALES using MACHINE LEARNING ALGORITHMS

[1]Nikhil Sunil Elias, [2]Seema Singh

[1] Student, Department of Electronics and Communication, BMS Institute of Technology and Management

[2] Professor, Department of Electronics and Communication, BMS Institute of Technology and Management

nikhil@renaissanceind.com, seemasingh@bmsit.in

Abstract — The ability to predict data accurately is extremely valuable in a vast array of domains such as stocks, sales, weather or even sports. Presented here is the study and implementation of several ensemble classification algorithms employed on sales data, consisting of weekly retail sales numbers from different departments in Walmart retail outlets all over the United States of America. The models implemented for prediction are Random Forest, Gradient Boosting and Extremely Randomized Trees (Extra Trees) Classifiers. The hyperparameters of each model were varied to obtain the best Mean Absolute Error (MAE) value and R^2 score. The number of estimators hyperparameter, which specifies the number of decision trees used in the model, plays a particularly important role in the evaluation of the MAE value and R^2 score and is dealt with in an attentive manner. A comparative analysis of the three algorithms is performed to indicate the best algorithm and the hyperparameter values at which the best results are obtained.

***Keywords* — Random Forests, Gradient Boosting, Extra Trees, Sales Forecasting, Mean Absolute Error, Mean Squared Error, R^2 score.**

I. INTRODUCTION

IN today's world where competition is cut-throat and making business decisions is increasingly difficult, the propensity to accurately make predictions is of extreme relevance. For example, it would be exceptionally beneficial to be able to predict the ups and downs of a country's economy or the fluctuations of its stock market prices. Forecasting has been done across a wide array of domains and spheres including environmental fields such as weather or even in sports performance due to the advantageous nature of prediction. The basis of this paper is sales prediction which is a more established yet still profoundly captivating application of forecasting. When organizations spread their capital and customers possess a deluge of options, even the slightest upper hand will have a significant impact on the fortunes of the organization. Sales forecasting uses trends identified from historical data to predict future sales, enabling educated decisions including assigning or redirecting current inventory, or effectively managing future production. This study in the application of sales forecasting explores the results of a range of models such as Random Forest [1], which forests is an ensemble learning method for classification, regression and other tasks, that functions by building a large number of decision trees at training time and producing the value that is the mean of the values (regression) of the individual trees at training time and producing the value that is the mean of the values (regression) of the individual trees, Gradient Boosting [2], which is also an ensemble learning method for regression, in that it minimizes the loss function by adding regression trees using the gradient descent procedure, and Extra Trees [3], which essentially consists of randomizing vigorously both attribute and cut-point selection while splitting the node of a tree.

Related Work—Popular approaches to the prediction problem are machine learning approaches using Artificial Neural Networks (ANNs) [4] and statistical approaches using Autoregressive Integrated Moving Average (ARIMA). Both methods when applied to stock predictions and sales forecasting have had moderate success [6]. ANNs have typically displayed good performance in forecasting due to their capacity to characterize non-linear data with very good accuracy [5]. However, their results could be further improved through better selection of hyperparameters. A combination of Seasonal Trend Decomposition using Loess and Autoregressive Integrated Moving Average (STL + ARIMA) has also been used in time series forecasting and generated good results [7]. The work of Pao *et al,* proved to be a significant stepping stone to the use of machine learning algorithms on which this paper is based. Another solid approach to sales prediction employed an extreme learning machine [8] that forecasted sales in the fashion retail industry. The results of this work, published by Sun *et al,* contributed to the hyperparameter selection proposed by this paper.

The paper entails three algorithms namely, Random Forest, Gradient Boosting, and Extra Trees, that are executed on the Walmart dataset. The algorithms were implemented using Python 3.7 running on Jupyter Notebooks in the Anaconda distribution. The performance of each algorithm was compared to highlight the best results.

II. DATASET

The dataset comes from the Kaggle platform and consists of data from an American retail organization, Walmart Inc. The dataset was used for a machine learning competition in 2014 [10]. It comprises data from 45 Walmart department stores mainly centered around their sales on a weekly basis. The dataset has 282,452 entries that will be used for training the models. Each entry has attributes as follows: the associated store (recorded as a number), the corresponding department (81 departments, each entered as a number), the date of the starting day in that week, departmental weekly sales, the store size, and a Boolean value specifying if there is a major holiday in the week. The major holidays being one of Thanksgiving, Labor Day, Christmas or Easter. Along with the aforementioned attributes is a parallel set of features for each entry including Consumer Price Index, unemployment rate, temperature, fuel price, and promotional markdowns. Since there is no test-set provided, they are generated from the given training data for cross-validation, and final testing.

III. METHODS

Three forecasting models were constructed in this research on the following algorithms: Random Forest, Gradient Boosting, and Extremely Randomized Trees (Extra Trees). Other algorithms such as Naïve Bayes and Adaptive Boosting were scrutinized, but their performances were not up to the mark and insights were trivial, so they will not be considered herein. All models were implemented in Python 3.7. on the Anaconda distribution using Jupyter Notebooks. The code used for the implementation has been uploaded to GitHub [12].

A. Random Forest

The Random Forest architecture is best described by Figure 1 [13]. As more trees are grown, the Random Forest algorithm adds more randomness to the model. It searches for the best feature amidst a random subset of features in place of searching for the most relevant feature while splitting a node. This results in more accurate model as it leads to a much greater diversity. Thus, in Random Forest, only a random subset of the features is considered by the algorithm for diverging a node. Trees can be made more random by using random thresholds for each feature instead of searching for the best thresholds (like a normal decision tree does).

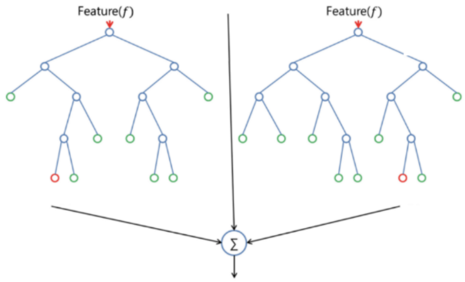

Figure 1. Random Forest Architecture

The features used for training the model were week number, store number, department number, the holiday flag, Consumer Price Index, unemployment rate, temperature, fuel price and store size. The algorithm was carried out using Python's RandomForestRegressor function present in the scikit-learn class. In the Python implementation, Mean Absolute Error (MAE), mean-squared error (MSE) and R^2 score are calculated for the predicted values.

Figure 2 shows the comparison of the predicted values and the actual values of the weekly sales with the hyperparameters set at the optimized values.

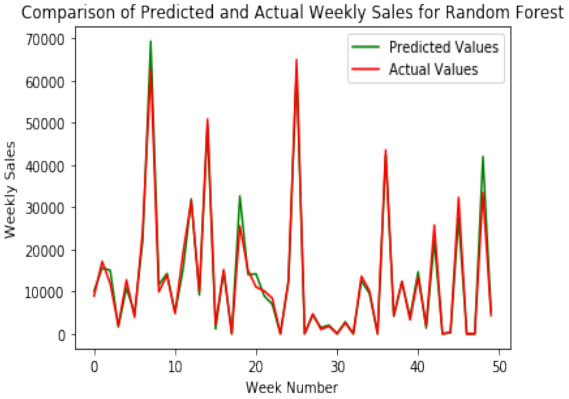

Figure 2. Performance Visual of Random Forest

B. Gradient Boosting

According to Friedman *et al,* Gradient Boosting sequentially fits a simple parameterized function to current pseudo-residuals by least-squares at each iteration by building additive regression models. The gradient of the loss function being minimized with respect to the model values at each training point are referred to as the pseudo-residuals in the model. Figure 4 illustrates the operation of the Gradient Boosting algorithm at various iterations.

The features used for training the model were the same as the ones used in the Random Forest classification. The algorithm was implemented using Python's GradientBoostingRegressor function from the scikit-learn class, and the mean absolute error, mean squared error and R^2 score were calculated for the predicted values.

Figure 3 shows the comparison of the predicted values and the actual values of the weekly sales with the hyperparameters set at the optimized values.

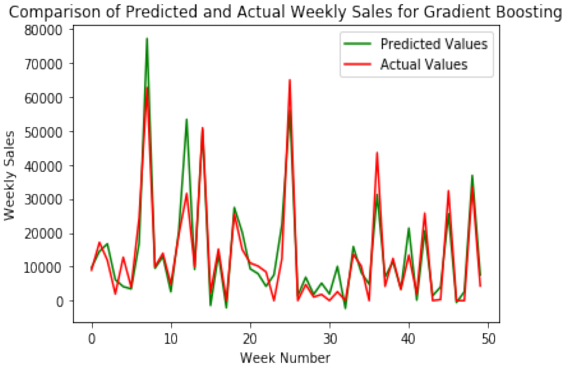

Figure 3. Performance Visual of Gradient Boosting

C. Extra Trees

The Extra Trees and Random Forest algorithms are almost the same. In the Random Forest algorithm, the tree splitting phenomenon is deterministic in nature whereas in the case of Extremely Randomized Trees, the split of the trees is completely random. In other words, during the process of splitting, the algorithm chooses the best split among random splits in the selected variable for the current decision tree.

The features employed are like the ones used in the previous algorithms. Python's ExtraTreesRegressor function from the scikit-learn class was used to execute the algorithm, and the various performance metrics calculated for the previous methods are evaluated and reported.

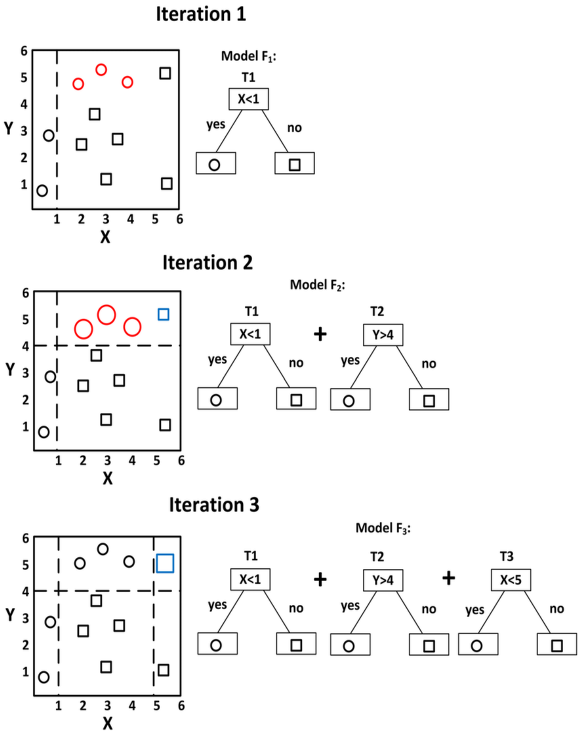

Figure 4. Operation of Gradient Boosting

Figure 5 shows the comparison of the predicted values and the actual values of the weekly sales with the hyperparameters set at the optimized values.

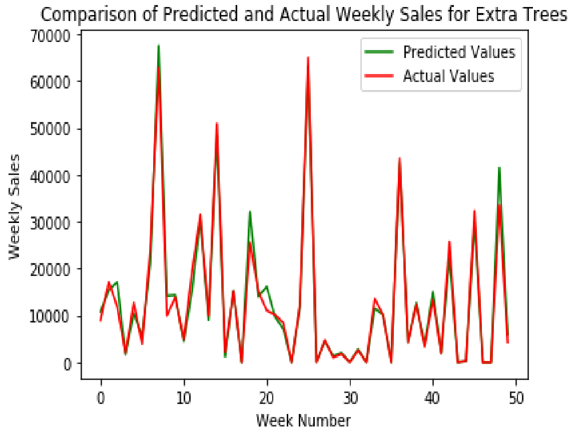

Figure 5. Performance Visual of Extra Trees

IV. RESULTS

The winning submission for the Kaggle competition had a Mean Absolute Error (MAE) of around 2130 [11]. As a reference, a submission where all the predicted values of weekly sales are 0's, the MAE is found to be approximately 21000. In this study, the last 20% of the training dataset was used as the local test-set.

The Gradient Boosting algorithm was taken as a baseline and the MAE was found to be 5771.5, with a R^2 score of 0.80 that implies that 80% of the predicted values were accurate. These were the best results obtained with the n_estimators hyperparameter, which refers to the number of decision trees that are used for regression, set at 200. The other hyperparameters were set to their default values. Table 1 refers to different values given to the parameters and the results that followed.

Table 1: Gradient Boosting performance

No. of estimators	Min sample split parameter	Min sample leaf parameter	MAE	R^2 score
101	2	1	6721.5	0.75
150	3	5	6134.1	0.78
200	3	1	5771.5	0.80

The Random Forest algorithm performs much better than Gradient Boosting in that it's MAE was calculated as 1979.4, with a R^2 score of 0.94. These performance metrics were the best achieved with the n_estimators hyperparameter set at 150, while the min_samples_split parameter, which specifies the minimum number of samples required to split an internal node, and the min_samples_leaf parameter which specifies the minimum number of samples required to be at a leaf node, are set at 2 and 1 respectively. Table 2 refers to different values given to the parameters and the results that followed.

Table 2: Random Forest performance

No. of estimators	Min sample split parameter	Min sample leaf parameter	MAE	R^2 score
51	3	5	2051.5	0.93
150	3	5	2047.3	0.93
150	2	1	1979.4	0.94

The Extremely Randomized Trees algorithm works slightly better than the Random Forest. This increase in performance may be attributed to higher randomization in the training process. The n_estimators parameter was set to 150, while the min_samples_split and min_samples_leaf parameters were placed at 2 and 1 respectively, to obtain the best results wherein the MAE was 1965.5 and R^2 score was 0.94. Table 3 refers to different values given to the parameters and the results that followed.

Table 3: Extra Trees performance

No. of estimators	Min sample split parameter	Min sample leaf parameter	MAE	R^2 score
71	3	5	1976.8	0.94
115	3	4	2039.2	0.94
150	2	1	1965.5	0.94

It was noted that as the value of the n_estimators hyperparameter is increased beyond the values provided in the tables above, it was found that the MAE was increasing instead of decreasing, possibly implying that the models were overfitted. Also, increasing the number of regression trees indiscriminately is not advised as it leads to increased computational intensity resulting in a larger amount of time spent in training the model without benefitting it's accuracy.

Table 4 presents the best results obtained from each machine learning algorithm applied to the dataset.

Table 4: Comparison of ML Algorithms

Algorithms	Mean Absolute Error	Mean Squared Error	R^2 score
Gradient Boosting	5771.5	9.87E+07	0.80
Random Forest	1979.4	3.00E+07	0.94
Extra Trees	1965.5	2.96E+07	0.94

Figure 6 represents the comparative performance of each of the algorithms implemented with the hyperparameter, n_estimators, set at 150.

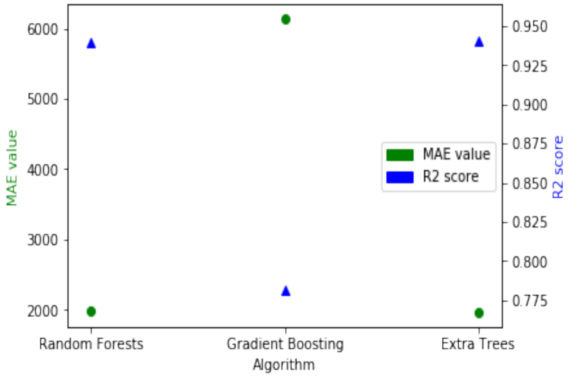

Figure 6. Comparison of MAE values and R^2 scores

V. CONCLUSION

This paper dealt with the implementation of three algorithms namely, Random Forest, Gradient Boosting, and Extra Trees, on the Walmart dataset and a comparative analysis was carried out to determine the best algorithm.

Random Trees was confirmed to be a very effective model in forecasting sales data. Extra Trees, an extension of Random Forest, also showed very good accuracy for the best implementations. These algorithms could possibly produce even better results if they are provided with better hardware electronics like Graphics Processing Units (GPUs).

Future work would include the Extra Trees model being developed to consider sparse promotional markdown data and moving holidays. It would also involve the fine-tuning of the hyperparameters of the models to improve the accuracy of prediction. Future work could also entail combining the models to produce an ensemble training model that could represent even the tiniest details present in the data. With the development of deep learning techniques, the results of this research could be further improved in the near future through the use of more complex and multilayer ANNs.

This work shows that there are highly efficient algorithms to forecast sales in big, medium or small organizations, and their use would be beneficial in providing valuable insight, thus leading to better decision-making.

VI. ACKNOWLEDGMENT

The authors would like to thank the management of BMS Institute of Technology and Management for their encouragement in pursuing research in diverse fields of study.

VII. REFERENCES

[1] Kulkarni, Vrushali Y., and Pradeep K. Sinha. "Random forest classifiers: a survey and future research directions." *Int J Adv Comput* 36.1 (2013): 1144-53.

[2] Friedman, Jerome H. "Stochastic gradient boosting." *Computational Statistics & Data Analysis* 38.4 (2002): 367-378.

[3] Geurts, Pierre, Damien Ernst, and Louis Wehenkel. "Extremely randomized trees." *Machine learning* 63.1 (2006): 3-42.

[4] Zhang, Guoqiang, B. Eddy Patuwo, and Michael Y. Hu. "Forecasting with artificial neural networks: The state of the art." *International journal of forecasting* 14.1 (1998): 35-62.

[5] Allende, Héctor, Claudio Moraga, and Rodrigo Salas. "Artificial neural networks in time series forecasting: A comparative analysis." *Kybernetika* 38.6 (2002): 685-707.

[6] Adebiyi, Ayodele Ariyo, Aderemi Oluyinka Adewumi, and Charles Korede Ayo. "Comparison of ARIMA and artificial neural networks models for stock price prediction." *Journal of Applied Mathematics* (2014), Article ID 614342, 7 pages, 2014. doi:10.1155/2014/614342.

[7] James J. Pao, Danielle S. Sullivan, “Time Series Sales Forecasting”, Final Year Project, 2017. Accessed at http://cs229.stanford.edu/proj2017/final-reports/5244336.pdf

[8] Sun, Zhan-Li, et al. "Sales forecasting using extreme learning machine with applications in fashion retailing." *Decision Support Systems* 46.1 (2008): 411-419.

[9] Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and Regression Trees. Boca Raton, FL: Chapman & Hall, 1984.

[10] Kaggle. “Walmart Sales Forecasting Data”. https://www.kaggle.com/c/walmart-sales-forecasting/data

[11] Kaggle. “Walmart Sales Forecasting Leaderboard”. https://www.kaggle.com/c/walmart-sales-forecasting/leaderboard

[12] Nikhil Elias, Sales Forecasting using ML algorithms (2018), GitHub repository. https://github.com/NikhilElias/Sales-Forecasting-using-ML-algorithms/blob/master/Code.py

[13] Niklas Donges, “The Random Forest Algorithm”, Towards Data Science. https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd