International Journal of Computer (IJC)
ISSN 2307-4523 (Print & Online)
http://ijcjournal.org/

# Performance Analysis of Machine Learning Models for Sales Forecast

Omogbhemhe Izah Mike[a*], Odegua Rising[b]

[a,b]*Department of Computer Science, Ambrose Alli University, Ekpoma, Nigeria*
[a]*Email: mikeizah@g.com*

**Abstract**

Many supermarkets today do not have a strong forecast of their yearly sales. This is mostly due to the lack of the skills, resources and knowledge to make sales estimation. At best, most supermarket and chain store use adhoc tools and processes to analyze and predict sales for the coming year. The use of traditional statistical method to forecast supermarket sales has met a lot of challenges unaddressed and mostly results in the creation of predictive models that perform poorly. The era of big data coupled with access to massive compute power has made machine learning model the best for sales forecast. In this paper, we investigated the forecasting of sales with three machine learning algorithms and compare their predictive ability. Three different methods used are K-Nearest Neighbor, Gradient Boosting and Random forest. The data used to train the machine learning models are data provided by Data Science Nigeria on the Zindi platform, the data were collected from a supermarket chain called "Chukwudi Supermarkets". The results show that the Random Forest algorithm performs slightly better than the other two models, we saw that Gradient Boosting models were prone to over-fitting easily and that K-Nearest Neighbor even though fast, performs poorest among the three.

***Keywords:*** Machine Learning; Sale Forecast; Models; Performance.

## 1. Introduction

The goal of every supermarket is to make profit. This is achieved when more goods are sold and the turnover is high. A major challenge to increasing sales of a supermarket lies in the ability of the manager to forecast sales pattern and know readily before hand when to order and replenish inventories as well as plan for manpower and staffs. The amount of sales data has steadily be on the increase in recent years and the ability to leverage this gold of data separates high performing supermarket from the others. One of the most valuable assets a supermarket can have is data generated by customers as they interact with various supermarkets. Within these data, lies important patterns and variables that can be modeled using a machine learning algorithm; and this can to a very high degree of accuracy correctly forecast sales [1, 2]. There exist several techniques to forecasting supermarket sales and historically, many supermarkets have relied on these traditional statistical models [3].

---

* Corresponding author.

However, machine learning has grown to be an important area of data science that has gained ground due to its high predictive and forecasting powers and as such as become the go-to for highly accurate sales forecasting as well as other important areas [3, 4, 5]. To correctly forecast a future event, a machine learning model is trained on data from which it learns patterns that are used to predict future instances. An accurate forecasting model can greatly increase supermarket revenue and is generally of great importance to the organization as it improves profit as well as provides insights into the way customers can be better served [3]. The main goal of this paper is to evaluate new machine learning techniques for sales forecasting to simple traditional methods.

### *2.1 K-Nearest Neighbor*

K- Nearest Neighbor (KNN) is one of the simplest type of machine learning algorithm [13]. The idea behind KNN is that given a sample of instances in a sample space, a new instance is similar if it belongs to the same class as already existing sample. The idea is to first select k nearest neighbor to the sample whose class we want to predict. In that sense, KNN does not need any training and is seen as a memorization based techniques. KNNs are good and fast for small data set, but becomes less efficient when the data set increases.

**K-Nearest Neighbor Algorithm**

1. Load the data set
2. Initialize K with a value
3. For 1 to total number of data points:
1. Calculate the distance between test data point and each row of the training data points.
2. Sort the calculated distances in ascending order based on distance.
3. Get the top k rows from the sorted values.
4. Return the class of the top k rows as the predicted class.

### *2.2 Gradient Boosting Model*

Boosting is a popular machine learning algorithm that falls under the umbrella of ensembles. Boosting was introduced in answer to the question whether a "weak learner" could be made better by using some form of modification. This was discovered to be possible and the first boosting algorithm Adaptive Boosting (AdaBoost) was created by [10]. The concept of boosting is to correct the mistakes made by previous learners and improving on those areas [11]. Boosting can also be seen as a kind of stage wise "additive modeling" in that it is an additive combination of a simple base estimator. Gradient Boosting [12] is a type of boosting where the objective is treated as optimization problem and training is done using weight updates by gradient descent.

**Gradient Boosting Algorithm**

1. Specify the following as input:
I. Input data N

II. Number of iterations M
III. A base-learner h
IV. A loss function l
2. Initialize l0 to a constant
3. for t = 1 to M: compute the negative gradient
4. fit a new base-learner function hi
5. Find the best gradient descent step-size p
6. update the function estimate

### *2.3 Random Forest Model*

Random forest is a tree-based machine learning algorithm introduced by [6]. In random forest, multiple decision trees are constructed and trained on a bootstrap sample drawn from the original dataset. The final result in the case of regression task, is an average of the individual predictions from each decision tree, and a majority class vote in a classification task. Breiman [7] defined Random Forest as a classifier consisting of a collection of trees structured classifiers {h(**x**,Θk ), k=1, ...} where the {Θk} are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x.

Numerous empirical studies [6,8,9], shows that random forests are serious competitors to state-of-the-art methods such as boosting [10]. Most industry practitioners consider it to be one of the most accurate general-purpose learning techniques. Random Forest are fast and easy to implement, produce highly accurate predictions and can handle a very large number of input features without the risk of overfitting.

**Random Forest Algorithm**

The random forest algorithm for both classification and regression task is shown below:

- Draw n bootstrap samples from the original dataset.
- For each ni bootstrap sample, grow a classification or regression tree, by choosing the best split among *m* randomly selected variables.
- Predict new data by aggregating the predictions of the n trees using average for regression and majority voting for classification.

## 3.Methodology

The models compared in this study (K-Nearest Neighbor, Gradient Boosting and Random Forest) have been used for numerous problems in different forecasting task and have been chosen based on their popularity in the industry. In addition, the data used in this study is provided by Data Science Nigeria, a Data Science and Artificial Intelligence Hub as part of their machine learning competitions.

### *3.1 The Data*

The data consist of numerous supermarket variables like opening year, product prices, supermarket location etc. The data set contains a sample of 4990 instances with 13 features/variables. The description of the data is shown in table 1.

**Table 1:** Data Description

| **Data Feature** | **Description** | **Feature Type** |
|---|---|---|
| Product_Identifier | Unique identifier for each product | String |
| Supermarket_Identifier | Unique identifier for each supermarket | String |
| Product_Supermarket_Identifier | Combination of product and supermarket identifiers | String |
| Product_Weight | The weight of a product | Numeric |
| Product_Fat_Content | The fat content contained in a product. | Categorical |
| Product_Shelf_Visibility | A numeric value that captures the visibilty of a product. | Numeric |
| Product_Type | The type of product. | Categorical |
| Product_Price | The selling price of a product. | Numeric |
| Supermarket_Opening_Year | The year the supermarket was opened. | Numeric |
| Supermarket_Size | The size of a supermarket | Categorical |
| Supermarket_Location_Type | The location of a supermarket | Categorical |
| Supermarket_Type | The type of the supermarket | Categorical |
| Product_Supermarket_Sales | The sales made by the supermarket (**Target Feature)** | Numeric |

### *3.2 Data Processing and Engineering*

After extensive data cleaning and processing, three features (Product_Identifier, Supermarket_Identifier, Product_Supermarket_Identifier) were removed as they are unique IDs and add little or no effect to our model's performance. Further exploration of the dataset showed the need to create new features from the existing ones. This process is termed feature engineering; The new features created are:

1. is_normal_fat: Groups the feature *Product_Fat_Content* into two groups 0 and 1
2. open_in_the_2000s: Groups the feature S*upermarket_Opening_Year* into two classes.
3. Product_type_cluster Clusters the *Product_Type* into two classes

Next, we used one-hot-encoding scheme to encode all categorical variables, filled missing instances in Product_Weight feature with the mean and finally standardize our data set by subtracting the mean and then dividing by the standard deviation. These three models where trained on the data set and a 10 fold cross validation strategy was used since the data set was limited. The mean absolute error was recorded as performance metrics.

### *3.3 K-Nearest Neighbor*

The KNN model is implemented in sklearn, a Python machine learning library. The important parameter here is the k-number of neighbors to use in voting - which we set to 50. The other parameters where left as default.

### *3.4 Gradient Boosting Model*

For gradient boosting, we set the number of boosted trees (n_estimators) to 200, *max_depths* to 6, *max_features* as square root and the mini_sample_split to 4. All other parameters where left as default.

### *3.5 Random Forest Model*

For random forest model, we specify the number of trees (n_estimators) to 100 and the *max_depth* to 5. All other parameters where left as default.

### *3.6 Performance Metric*

We use the mean absolute error (MAE) in model evaluation. This means that a lower MAE results in a better model. The choice of performance metric is based on the fact that the task is a regression task and the MAE is a tested and trusted metric that gives a good measure of model performance.

#### *3.6.1 Mean Absolute Error*

Mean absolute error (MAE) is a measure of difference between two continuous variables. Assume *X* and *Y* are variables from an observations, say X is the known value and Y is the predicted value from a machine learning model, the Mean Absolute Error (MAE) is the average vertical distance between each observed point and the predicted point.

The mean absolute error is given by:

$$\text{MAE} = \sum_{i=1}^{n} \frac{|y_i - x_i|}{n}$$

## 4. Results

In this section the results of the three models is presented. The results were obtained by applying the three models; KNN, GB and RF on a 10-fold cross validation dataset.

### *4.1 Error Measures & Standard Deviations*

In table 2, the MAE and standard deviations are shown respectively. Taking the average prediction of the 10 fold cross validation, we observe that the Random Forest algorithm does best among all three with a MAE of 0.409178. The Gradient Boosting model has a close MAE to the KNN but with a much lower standard deviation. Figure 2, 3 and 4 shows the learning curves of the three models. I.e the plot of MAE against training size of the dataset over a cross validation of 10 folds, Figure 5 side by side comparison, while Figure 6 and 7 shows the important features that contributed the most to our predictions.

**Table 2:** Comparison of MAE

| **Models** | **MAE** | **SD** |
|---|---|---|
| **KNN** | 0.425103 | 0.018008 |
| **RF** | 0.409178 | 0.014420 |
| **GB** | 0.428260 | 0.014420 |

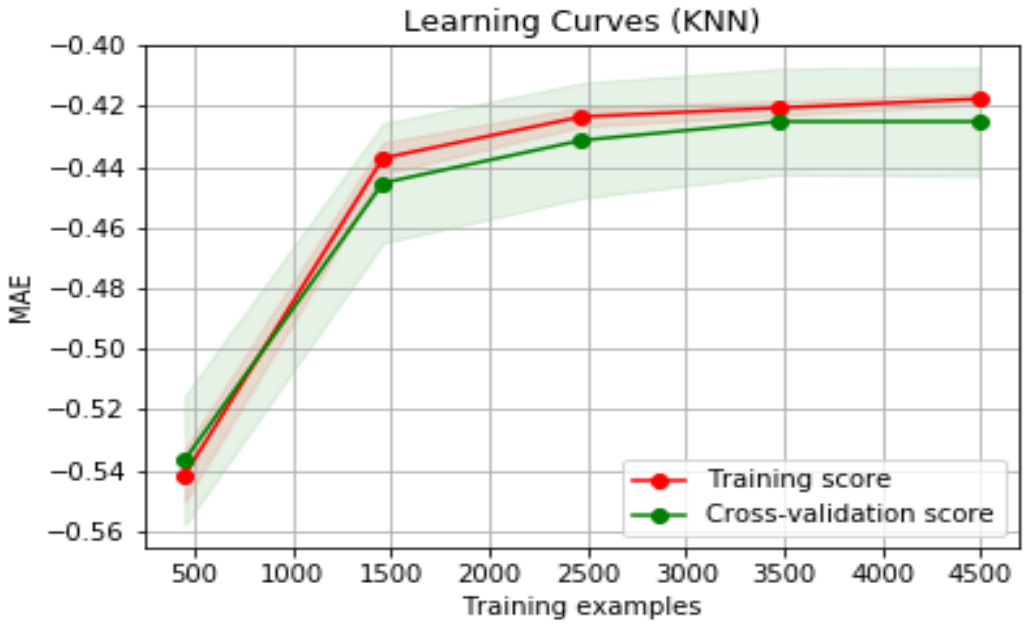


**Figure 2:** Learning curve plot for KNN

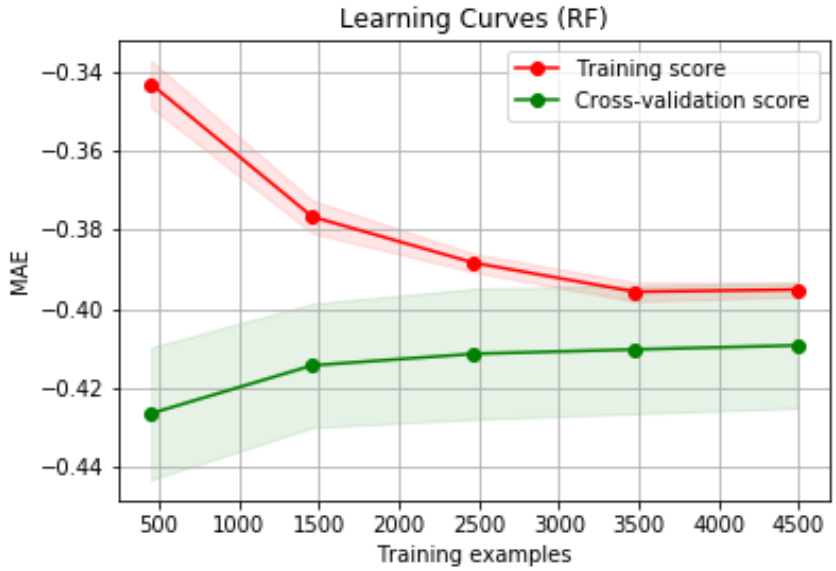


**Figure 3:** Learning curve plot for RF

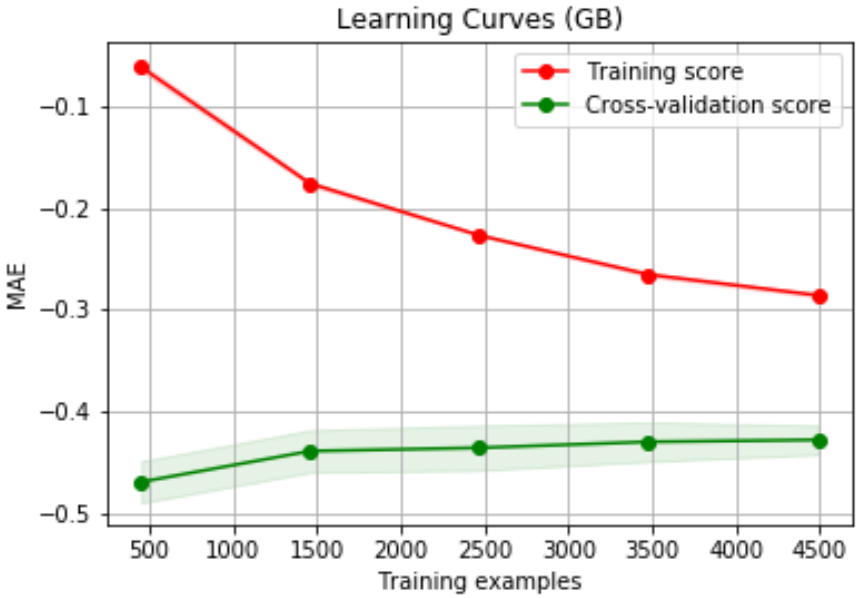


**Figure 4:** Learning curve plot for GB

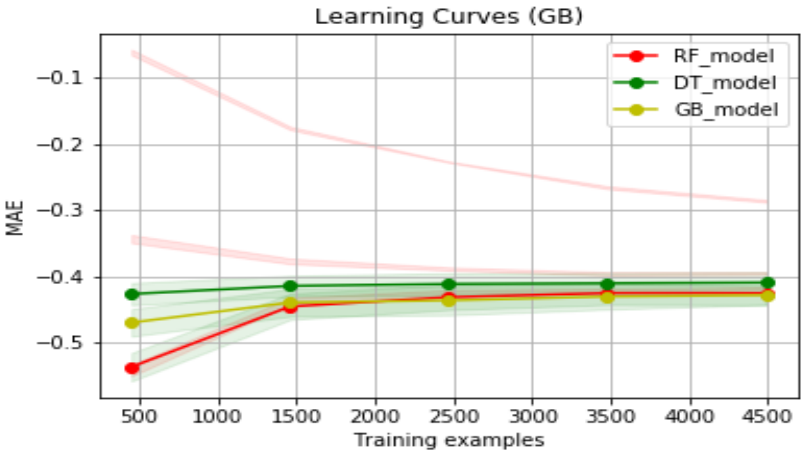


**Figure 5:** Learning curve plot for the three models

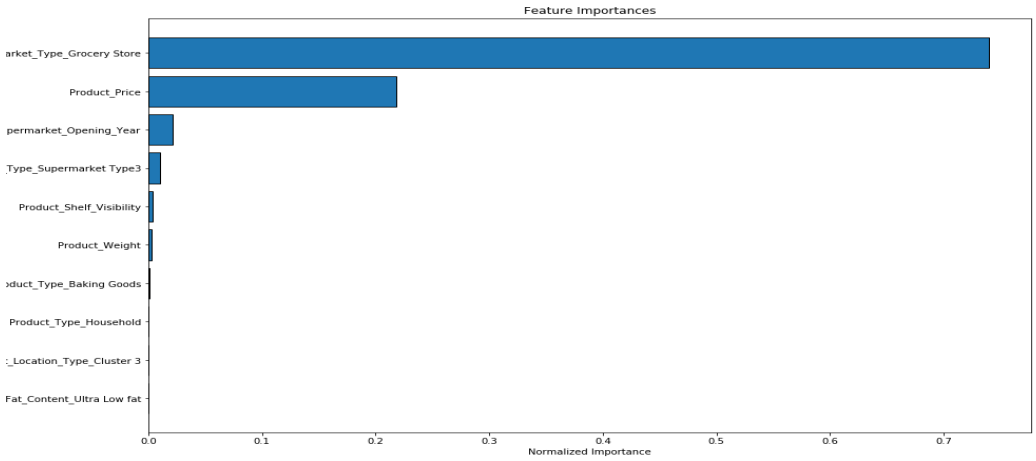


**Figure 6:** Feature importance chart for RF

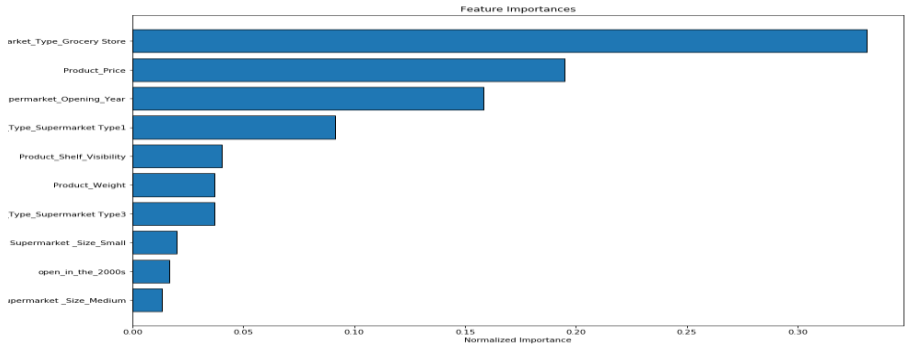


**Figure 7:** Feature importance chart for GB

## 5. Discussion

In order to verify our problem statement we chose to use the MAE as error measures. We saw that the RF scored a lower average MAE than the KNN and the GB. However, for all three models, the training score and the cross-validation score are both not good at the start, but becomes better as the training size increases. This is a common phenomenon in a complex high dimension data set. Moreover, we notice that the MAE for KNN is on the increase, and that introducing more data set would definitely make it perform better. Also the models in this study were trained on a small data set which might be a contributing factor to the generally high errors. Multiple studies have shown GB usage in sales forecasting, hence its poor performance was unexpected. The GB shows surprisingly high MAE compared to the other models. However our result could be a consequence of the parameter settings used since there are multiple ways to choose initial parameters. We also identified the most important features used by the RF and GB models. These features are Supermarket_Type_Grocery_Store, Product_Price, Supermarket_opening_year. The Grocery store types seem to sell more and generally have higher sales than the other types of stores. The Product Price also affects sales as higher prices of products that sell more generally contributes to higher sales and finally another important feature is the Supermarket_opening_year, where newer stores sell higher than older stores.

## 6. Conclusion

Sales forecasting is very crucial for every company, especially for big ones. This process is very complex because there are a lot of factors that should be taken into consideration. In order to implement achievable goals and successfully implement them. Supermarkets chains always want to forecast sales in other to help them plan. In this study of three machine learning algorithms (KNN, RF & GB) for sales forecasting, RF was observed to do better, as this method generally have a lower MAE, our choice of performance measurement in this task. We also observe that getting more data would generally increase the predictive power of our models.

## Reference

[1] Kim Brynjolfsson Hitt. “Strength in Numbers: How Does DataDriven Decisionmaking Affect Firm Performance”. In: (2011). URL: http://ebusiness.mit.edu/research/papers

[2] Orinna Cortes and Vladimir Vapnik. “Support-vector networks”. In: Machine Learning 20(3) (1995), pp. 273–297.

[3] Nari Sivanandam Arunraj and Diane Ahrens. “A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting”. In: Internation Journal Production Economics 170 (2015), pp. 321–335.

[4] Philip Doganis et al. “Time series sales forecasting for short shelflife food products based on artificial neural networks and evolutionary computing”. In: Journal of Food Engineering 75 (2006), pp. 196–20.

[5] Maike Krause-Traudes et al. Spatial data mining for retail sales forecasting. Tech. rep. Fraunhofer-Institut Intelligente Analyse- und Informationssysteme (IAIS), 2008.

[6] L. Breiman. Consistency For a Simple Model of Random Forests. Technical Report 670, UC Berkeley, 2004. URL http://www.stat.berkeley.edu/˜breiman.

[7] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. Classification and Regression Trees. Chapman & Hall, New York, 1984.

[8] V. Svetnik, A. Liaw, C. Tong, J. Culberson, R. Sheridan, and B. Feuston. Random forest: A classification and regression tool for compound classification and QSAR modeling. Journal of Chemical Information and Computer Sciences, 43:1947–1958, 2003.

[9] Diaz-Uriarte and S.A. de Andres. Gene selection and classification of microarray data using ´ random forest. BMC Bioinformatics, 7:1471–2105, 2006.

[10] Y. Freund and R. Shapire. Experiments with a new boosting algorithm. In L. Saitta, editor, Machine Learning: Proceedings of the 13th International Conference, pages 148–156, San Francisco, 1996. Morgan Kaufmann

[11] Z.-H. Zhou and M. Li. Ensemble Methods.(2012). Foundations and Algorithms, -13: 978-1-4398-3005 -5.

[12] Jerome H. Friedman . (1999). Greedy Function Approximation: A Gradient Boosting Machine, IMS 1999 Reitz Lecture.

[13] Cover T., Hart P., (1967), Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13(1), 21–27