Analysis of Market Demand and Forecasting for Eatables using Machine Learning

[1]Isha Madan, [2]Disha Sharma, [3]Ashar Ali, [4]Vikas Kumar, [5]Lal Pratap Verma,

[1,2,3,4,5]Moradabad Institute of technology Moradabad, India

Abstract— **The food industry plays a crucial role in human's life. Due to increasing population there is a huge demand of increase in food as well. So through our website (predicts the sale of food) we are reducing the wastage so that the storage cost is reduced and more investment can be done on the advertisement indirectly giving the benefit to the company and increasing its customers. 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 retail outlets all over the United States of America. The models implemented for prediction are Linear Regression, KNN, Decision tree, Random Forest, and Extremely Randomized Trees (Extra Trees). A comparative analysis of the five algorithms is performed to indicate the best algorithm and the hyperparameter values at which the best results are obtained.**

Keywords—Linear Regression, Random Forests, Extra Trees, Decision tree, Mean Absolute Error, R2 score.

I. Introduction

Usually the production which is done if it is produced in more than required amount then it is wasted as well as the consumers are not satisfied because they do not get the fresh product. The investments of industry goes on storage rather than advertisement. The idea of our project is that the demand planning truly matches what resellers and customers want, and cut the amount of over-production and material waste to the absolute minimum possible. 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.

Sales Forecasting is done to maximize sales, minimize waste and optimize efficiency. It ensures that freshness and quality of food products are maintained, since the company will produce only the required amount of food so the quality will not degrade and customer will be satisfied. With the upholding of customer's satisfaction it will also have the brand loyalty. It keeps the cost down and maintains the margin, the investment will be applied where required such as advertisement rather than storage of products.

It is basically a tool for making better decisions and happier customers by looking at the past data in order to predict the future needs. It uses the historical sales data to use as the basis of forecasting future demand, allowing future calculated trends to be analyzed and managed, to give a logical view of how demand may change over time. For e.g. at the time of festivals the demand of milk increases as compared to its usual demand in order to make sweets etc. or the demand of eggs increases in winters as compared to summers. So these seasonal demands make the prediction vary.

We are developing a website as a business analytical tool where you can not only predict the future sales but also can see the historical data, growth chart for each store and filter your view. The historical data will allow us to see the historical data, there are various types of store the website will display growth chart for each store showing the actual result as well as the predicted result. The filter option will allow to view the content according to the search result in filter.

This study in the application of sales forecasting explores the results of a range of models such as Random Forest, 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, decision tree], which is also an ensemble learning method for regression.

The paper entails five algorithms namely, Linear Regression, KNN, Decision tree, Random Forest, and Extra Trees, that are executed on the kaggle dataset. The algorithms were implemented using Python 3.4 running on Jupyter Notebooks in the Anaconda distribution. The performance of each algorithm was compared to highlight the best results.

II. Overview of Project

There are two modes- Company and Store Head, Through Company the company can access the website, and through the Store Head the store can access the website. First the user will register to our website if the account is not created or login if the account is already created. Then you can upload your sales data after that you can see the forecasting of the data.

A. *Company Mode*

You have to login to your account, then the company user can overview the sales of each store and predicted sales value for next 3 dates.

B. *Store Head Mode*

You have to register/login to your account, then upload the sales data of the store, make predictive sales forecast, customize your prediction, analyze your entire sales history, track your performance of result. There are historical sales data for 45 stores located in different region; each store contains a no. of department.

III. DATASET

The dataset comes from the Kaggle platform and consists of data from an American retail organization. The dataset was used for a machine learning competition in 2014. It comprises data from 45 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 (99) 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. Since there is no test-set provided, they are generated from the given training data for cross-validation, and final testing.

IV. METHODS

Five forecasting models were constructed in this research on the following algorithms: Linear Regression, KNN, Decision tree, Random Forest, and Extremely Randomized Trees (Extra Trees). The algorithms Linear Regression, and KNN were scrutinized, but their performances were not up to the mark and insights were trivial. All models were implemented in Python 3.7. on the Anaconda distribution using Jupyter Notebooks.

A. Linear Regression

Linear Regression assumes linear relation between ξ and ψ. The hypothesis function for linear regression is-

$$\psi=\mu1.\xi1+\mu2.\xi2+\mu3.\xi3+.......\mu\nu.\xi\nu+\chi \qquad (1)$$

where $\mu1,\mu2,\mu3$ are called the parameters and χ is the intercept of the line. The motive of the linear regression algorithm is to find the best values for $\mu1$, $\mu2$, $\mu3....\chi$. We used the gradient descent method to implement linear regression algorithm.

Fig. 1 shows the comparison of the predicted values and the actual values of the weekly sales with the hyper parameters set at the optimized values.

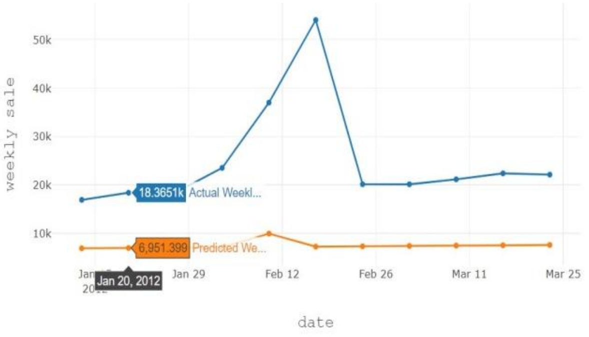

Fig. 1. Performance visual of Linear Regression

B. k-nearest neighbors

KNN is a simple algorithm that stores all available cases and predicts the numerical target based on a similarity measure (e.g., distance functions).

Fig. 2 shows the comparison of the predicted values and the actual values of the weekly sales with the hyper parameters set at the optimized values.

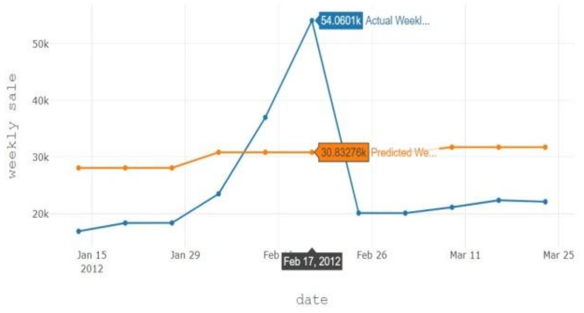

Fig. 2. Performance visual of KNN

C. Decision Tree

As a baseline method, a decision tree utilizing the features provided in the dataset was implemented. This model was chosen as a baseline since it is easily implemented and leveraged the way the provided data was organized.

The splitting attributes chosen were week number, store number, department number, the holiday flag, and the store size. The tree was implemented using sklearn ensemble, which follows the Classification and Regression Tree (CART) algorithm, choosing splits to maximize the chosen split-criterion gain. In the implementation using CART, mean-squared error is calculated for the responses and splits among the data are done to maximize mean-squared error reduction.

Fig. 3 shows the comparison of the predicted values and the actual values of the weekly sales with the hyper parameters set at the optimized values.

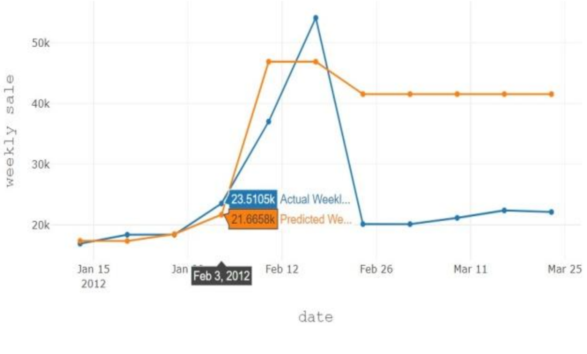

Fig. 3. Performance visual of Decision tree

D. Random Forest tree

The Random Forest tree architecture is best described by Fig. 4. 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 result is 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).

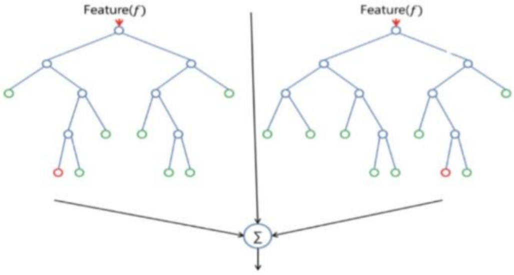

Fig. 4. Random forest architechture

The features used for training the model were week number, store number, department number, the holiday flag, 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 R2 score are calculated for the predicted values.

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

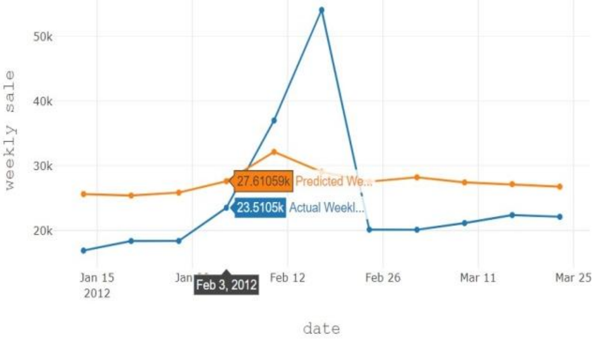

Fig. 5. Performance visual of Random forest tree

E. *Extra Forest tree*

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 scikitlearn class was used to execute the algorithm, and the various performance metrics calculated for the previous methods are evaluated and reported.

Fig. 6 shows the comparison of the predicted values and the actual values of the weekly sales with the hyperparameters set at the optimized values.

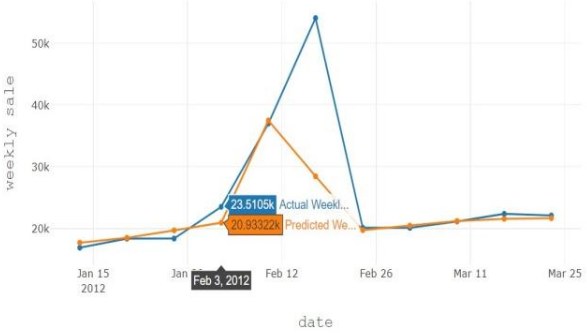

Fig. 6. Performance visual of Extra forest tree

V. RESULTS

Linear Regression algorithm was taken as a baseline and its MAE was calculated as 21272.6 with a R2 score of 0.13. These were the very bad results.

The k-nearest neighbors algorithm works slightly better than Linear regression algorithm and the MAE was found to be 13575.85, with a R2 score of 0.58 that implies that 58% of the predicted values were accurate. These were the best results obtained with the n neighbors hyperparameter, which refers to the number of neighbors to use, set at 5. 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 I. KNN PERFORMANCE

n neighbors	Mean absolute error	Mean Squared Error	R2 score
5	13575.85	3.39E+08	0.5851
7	13723	3.83E+08	0.5316
10	13297.41	4.75E+08	0.4188

The Decision tree algorithm was taken as first ensemble learning algorithm and the MAE was found to be 4268.345, with a R2 score of 0.93 that implies that 93% of the predicted values were accurate. These were the best results obtained with the max depth and min sample split hyperparameter, which refers to the maximum depth of the tree and the minimum number of samples required to split an internal node, set at 8 and 2 respectively. The other hyperparameters were set to their default values. Table 2 refers to different values given to the parameters and the results that followed.

TABLE II. DECISION TREE PERFORMANCE

Max depth parameter	Min sample split	Mean absolute error	Mean Squared Error	R2 score
4	2	9619.988	2.05E+08	0.7493
8	2	4268.345	49756050	0.9392
8	3	4328.303	50068830	0.951

The Random Forest algorithm performs much better than decision tree in that its MAE was calculated as 2627.581, with a R2 score of 0.97. 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 3 refers to different values given to the parameters and the results that followed.

TABLE III. RANDOM FOREST TREE PERFORMANCE

No. of estimations	Min sample split	Min sample leaf	Mean absolute error	Mean Squared Error	R2 score
51	3	5	2973.294	19280882	0.9764
150	3	5	2972.125	18475365	0.9774
150	2	1	2627.581	21149447	0.9741

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 2669.405 and R2 score was 0.98. Table 4 refers to different values given to the parameters and the results that followed.

TABLE IV. EXTRA FOREST TREE PERFORMANCE

No. of estimations	Min sample split	Min sample leaf	Mean absolute error	Mean Squared Error	R2 score
50	3	5	3252.306	23735392	0.971
115	3	4	3138.567	22127725	0.973
150	2	1	2669.405	19287264	0.9764

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 its accuracy.

Table 5 presents the best results obtained from each machine learning algorithm applied to the dataset.

TABLE V. COMPARISON OF MACHINE LEARNING ALGORITHMS

Algorithms	Mean absolute error	Mean Squared Error	R2 score
Linear regression	21272.69	7.07E+08	1E+11
KNN	13575.85	3.39E+08	0.5851
Decision tree	4268.35	5E+07	0.9392
Random forest tree	2627.58	2.1E+07	0.974
Extra tree	2669.405	1.9E+07	0.9764

VI. CONCLUSIONS

The ability to predict data accurately is extremely valuable in a vast array of domains such as stocks, sales, weather or even sports. This Project is the study and implementation of several ensemble classification algorithms employed on sales data, consisting of weekly retail sales numbers from different departments in retail outlets all over the United States of America. In this project, we dealt with the implementation of five algorithms namely, Linear Regression, k-nearest Neighbors, Decision Tree, Random Forest and Extra Trees. The hyperparameters of each model were varied to obtain the best Mean Absolute Error (MAE) value and R2 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 R2 score and is dealt with in an attentive manner. A comparative analysis of the five algorithms is performed to indicate the best algorithm and the hyperparameter values at which the best results are obtained.

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.

Furthermore, this research could also be improved in the near future through the use of time series algorithms. (Aster GLM, Aster Streaming, Aster ARIMA, ARIMA Predictor).In addition, the developer module can be introduced, that developer can make algorithm dynamic because the pattern of the sales data could be changed and because of that the accuracy of algorithms could decrease or increase, so it is needed to supervise the procedure. Also new columns can be introduced according to market requirements.

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. "Decision tree regression." 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