MEF UNIVERSITY

Forecasting with Ensemble Methods: An Application Using Fashion Retail Sales Data

Capstone Project

Orkun Berk Yüzbaşıoğlu

İSTANBUL, 2019

MEF UNIVERSITY

Forecasting with Ensemble Methods: An Application Using Fashion Retail Sales Data

Capstone Project

Orkun Berk Yüzbaşıoğlu

Advisor: Asst. Prof. Hande Küçükaydın

İSTANBUL, 2019

MEF UNIVERSITY

Name of the project: Forecasting with Ensemble Methods: An Application Using Fashion Retail Sales Data
Name/Last Name of the Student: Orkun Berk Yüzbaşıoğlu
Date of Thesis Defense: 09/09/2019

I hereby state that the graduation project prepared by Orkun Berk Yüzbaşıoğlu has been completed under my supervision. I accept this work as a "Graduation Project".

09/09/2019
Asst. Prof. Hande Küçükaydın

I hereby state that I have examined this graduation project by Orkun Berk Yüzbaşıoğlu which is accepted by his supervisor. This work is acceptable as a graduation project and the student is eligible to take the graduation project examination.

09/09/2019

Director
of
Big Data Analytics Program

We hereby state that we have held the graduation examination of _____ and agree that the student has satisfied all requirements.

THE EXAMINATION COMMITTEE

Committee Member	Signature
1. Asst. Prof. Hande Küçükaydın	……..
2. ……..	……..

Academic Honesty Pledge

I promise not to collaborate with anyone, not to seek or accept any outside help, and not to give any help to others.

I understand that all resources in print or on the web must be explicitly cited.

In keeping with MEF University's ideals, I pledge that this work is my own and that I have neither given nor received inappropriate assistance in preparing it.

Name	Date	Signature
Orkun Berk Yüzbaşıoğlu	09/09/2019	

EXECUTIVE SUMMARY

Forecasting with Ensemble Methods: An Application Using Fashion Retail Sales Data

Orkun Berk Yüzbaşıoğlu

Advisor: Asst. Prof. Hande Küçükaydın

SEPTEMBER, 2019, 29 pages

In this project, ensemble methods of machine learning are used to predict short term store sales of a fashion retailer. Sales forecasts of various products at different stores are generated for a span of three months with bagging tree regressor, random forest regressor, and gradient boosting regressor algorithm. Algorithms are trained and evaluated with real past sales data of a Turkish fashion retailer. The predictive performance of the models is compared with linear regression. The results of the study show that random forest regressor shows the best performance.

Key Words: Time Series Analysis, Sales Forecasting, Ensemble Methods, Bagging Tree Regressor, Random Forest Regressor, Gradient Boosted Regression Tree, Linear Regression

ÖZET

Topluluk Metotları ile Tahmin: Hazır Giyim Satış Verilerini Kullanan Bir Uygulama

Orkun Berk Yüzbaşıoğlu

Tez Danışmanı: Yrd. Doç. Dr. Hande Küçükaydın

EYLÜL, 2019, 35 sayfa

Bu projede topluluk metotları ile bir hazır giyim şirketininin mağazalarının satışı tahmin edilmiştir. Çeşitli ürünlerin farklı mağazalardaki satışının tahminleri, sonraki üç ay için torbalama-regresyon ağaçları, rassal orman regresyonu ve gradyan artırma regresyon ağaçları algoritmaları kullanarak üretilmiştir. Algoritmalar gerçek geçmiş satış verileri kullanılarak eğitilip, performansları değerlendirilmiştir. Algoritmaların tahmin performansı doğrusal regresyonla karşılaştırılmıştır. Çalışmanın sonuçlarına göre rassal orman regresyonu en yüksek performansı göstermiştir.

Anahtar Kelimeler: Zaman Serisi Analizi, Satış Tahmini, Torbalama-Regresyon Ağaçları, Rassal Orman Regresyonu, Gradyan Artırma Regresyon Ağaçları, Doğrusal Regresyon

TABLE OF CONTENTS

.

1. INTRODUCTION

Forecasting is the activity of predicting what will happen in the future based on past and present information. Accurate forecasts are critical for many sectors. In this project, the aim is to forecast short term sales of the brick and mortar stores of an apparel retailer. For this purpose, three different predictive machine learning models are developed, bagging tree regressor, random forest regressor and gradient boosting regressor using a real-world data set. Each model generates forecasts of the sales of different products at each store. Models' performances are evaluated, and results obtained by these models are compared with linear regression results.

1.1. Forecasting

Forecasting is commonly used in business life to aid effective decision making in various areas. In general, a business organization needs forecasts at three different time horizons:

1) Short-term forecasts: are needed for tactical decisions such as scheduling of production and scheduling of employees.

2) Medium-term forecasts: are needed for minor strategic decisions such as determining the staffing levels for the sales goal and.

3) Long-term forecasts: are needed for major strategic decisions such determining resource requirements [6].

Due to these requirements to forecasts, a business organization needs to develop and utilize various forecasting models to predict uncertain events [5]. Hyndman [4] states that a good forecasting system should capture the genuine patterns and relationships in the historical data. To produce accurate forecasts, various forecasting techniques have been developed, and they fall into two broad categories: quantitative techniques and qualitative techniques

Quantitative techniques depend purely on data. Such models are built on the numerical past data [5]. Quantitative techniques consist of time series models and explanatory models. Time series is defined as a sequence of observations of a variable over time [14]. Time series models are based on the past values of the variable of interest to predict its future values. So, they assume that nothing is known about the factors which are believed to influence the future values of the response variable [24]. In time series models,

behavior of the response variable over time is analyzed to make predictions about its future behavior.

On the other hand, explanatory models try to unveil relations among the explanatory/dependent variables and variable of interest/independent variable. Thus, they assume that there is an explanatory relationship between the explanatory variables and the independent variable [5]. These models try to incorporate information from other variables [4]. Various statistical and artificial intelligence-based models such as the Linear Regression, Neural Networks and Decision Tree Regressors fall under this category.

Qualitative forecasting models do not require numerical past data. Instead, they depend on qualitative information. They are based on judgement and accumulated knowledge about the variable of interest [5]. These forecasts are generally generated with expert knowledge. Qualitative methods are generally used for medium-range and long-range forecasting when there is no past data is available.

1.2. Literature Review on Forecasting Applied for Fashion Industry and Fashion Retail Sales

Fashion apparel retailers are the main actors of the fashion/clothing industry's supply chain network because they both create order for the apparel manufacturers and supply the customers with the finished products [1]. To decrease costs, nowadays clothing production takes place in low cost Far East countries which increases order lead times. Due to the considerably long order lead times, fashion supply chain is vulnerable to bullwhip effects. An accurate long-term demand forecasting system/model may assist reducing the fluctuations in the fashion supply chain and the bullwhip effect [2]. Accuracy is also important for short-term sales forecasting, because accurate forecasts reduce i) excess stocks at the stores which increases utilization of floor space and ii) cost of stockouts (loss of goodwill) and thus improved forecast quality ultimately leads to overall cost savings [11].

Thus, textile–apparel companies operating in a competitive environment rely on/need accurate and reliable forecasting systems to stay competitive [3]. Lack of forecasting competency brings about either low inventory levels (stockout) and low service levels or high inventory levels (overstock) and obsolescence of products [7]. Thus, forecasting is crucial for a successful apparel retail inventory management, because a good

forecasting system can help to avoid understocking and overstocking in retail inventory planning [2].

However, Lieu et al. [2] remark that fashion retail sales forecasting is a complex task due to demand volatility. Thomassey et al. [8] signify in their study various factors affecting demand such as:

-the item (e.g. features such as color and selling price, lack of historical sales data due to the short life cycle of products),

- the distributor (e.g. number of stores, merchandizing),

- customers Choice (e.g. fashion trends),

- external/exogenous factors (e.g. weather, holiday periods, sales promotion, end of season sales, purchasing power of consumers).

Nenni at al. [7] remark that customer demand for the fashion products is highly volatile and difficult to predict and conclude that the demand of the fashion products can be classified as lumpy. A forecasting model should tackle with these features of fashion sales.

On the other hand, fashion retailing industry requires sales forecasts both to make decisions on the budget of the next season and to determine the correct number of products to ship to the stores. Therefore, sales forecasts are produced at two different horizons in textile-apparel sector. Thomassey [1] defines these two different forecasting horizons as:

- a long-range sales forecast (mean term), i.e. 1 year to plan and schedule the production and sourcing of the next season
- a short-range sales forecast (short term), i.e. a few weeks to replenish stores with products and to adjust deliveries

Although time series forecasting methods are widely used for retail sales predictions, Liu et al. [2] state that they are not leading to promising results for fashion retail sales forecasting and Thomassey [3] points out that they are neither efficient nor easily applicable for fashion and apparel sales. Time series methods are designed for smooth demand and do not work with lumpy demand due to their lack of capturing nonlinear patterns in demand [7]. The forecasting models differ according to the forecasting horizon. In the literature for short term fashion retail forecasting is generally realized using artificial neural networks (ANN) based methods and hybrid methods. For mean/long term forecasting fuzzy logic, ANN and tree-based models are used. These

artificial intelligence (AI) models are known for their power of capturing nonlinear relationships in the data. Hybrid forecasting models combine different models so that they carry strengths of different models.

Thomassey et al. [8] use fuzzy logic to determine the effects of explanatory variables and then build an ANN model to perform short term forecasting for apparel sales. Thomassey and Fiordaliso [12] build a hybrid system using real empirical sales data of a French apparel retailer. They cluster different items into sales profiles and then a decision tree assigns each new item into one of these sales profiles. The forecast is generated for the sales profiles. Sun et al. [31] employ extreme learning machine to forecast retail sales. They consider the relationship between sales and various factors such as color, size of the products and price of the products.

1.3. Product and Sales Data Aggregation

Fashion products consist of different types such as apparel, footwear, sportswear and formal wear [8]. Apparel include clothing for women, men and children. According to Thomassey [3], apparel products fall into 3 categories:

i) Basic items: They are sold each season/year without a change.

ii) Fashion items: They are sold in a short period and generally not replenished in stores.

iii) Bestselling items: They are sold each year with small modifications according to the fashion trends and could be replenished during the season.

Fashion items are not considered in the traditional short-term forecasting of store demand, because they are either not replenished or there is no past information available. However, some models are built to forecast demand of new items [10]. Forecasting systems are generally built to predict sales of basic items and "best-selling" items [3]. However, bestselling items are modified each season and therefore historical sales data is not available at the SKU/product level. Due to the short life span of the products and huge product variety aggregate data is used to build forecasting models. The level of aggregation is chosen either at the lowest level, where historical data of several years is available or a level from the internal hierarchical classification of the apparel retailer company [3]. An example of hierarchical product classification taken from [3] for an apparel retailer is shown in Figure 1:

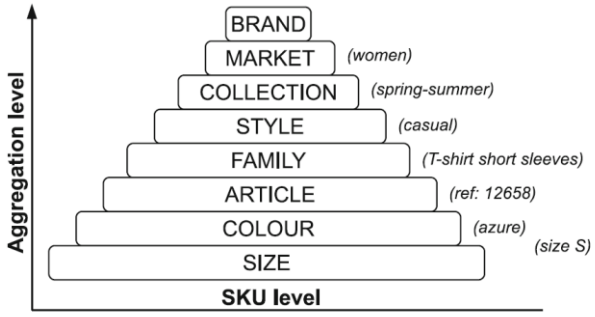

Figure 1. Product Hierarchy

In the literature, also some classification algorithms are used to group the SKUs [12]. With data aggregation, the main barriers of the forecasting fashion retail demand such as short selling season, lumpiness and lack of historical data are also alleviated [7].

2. Project Definition

In this capstone project, short term sales forecasts are generated for store chain of an apparel retailer company with machine learning techniques. The aim is to produce the demand forecasts of the multiple products at various stores with a single prediction model. To train the models and test their performance, real sales data of an important Turkish fashion retailer is used. This company operates nearly in 50 countries and has over 1000 stores globally. To manage the fullness of each store after the procurement lead time and to plan logistics and warehouse operations, accurate short-term sales forecasts are needed by this company. To address this requirements, three different ensemble learning algorithms are utilized to predict sales of the next three months. These are bagging tree regressor, random forest regressor and gradient boosting regressor. Thus, the models are used to generate short-term forecast for the lead time of each store. The output forecast figures determine the number of products required during the procurement lead time by each store for each product family.

3. About the Data

3.1. Data Description

In this project, sales data of a Turkish fashion retailer is used. The data set consists of daily sales of 5 different bestselling apparel product families at 20 different stores. Due to the huge product variety and the short life span of SKUs in apparel, aggregate data at family level is used. Sales are collected at daily level; therefore, this data set can be qualified as a time series data. The range of the data set is two years and seven months starting from January 2017 to the end of July 2019. The data set initially contains 94,163 observations of 15 variables. The target variable is the daily sales of different apparel product families at different stores.

3.2. Data Preparation

Two exogenous factors are removed from the data set, namely temperature forecast, and precipitation forecast since their forecast horizon is 5 days. This horizon is too short to use them as predictors because the lead time of the stores under consideration is 11 days. To incorporate effects of special days such as New Year, Christmas and Nowruz, some exogenous factors are added to the feature set which are expected to explain/impact the sales.

In time series analysis, the relationship of the current value/state of the response with its past values/states can increase the prediction power, too. To investigate this relationship, an autocorrelation plot (ACF) of sales is plotted at day level in Figure 2. Autocorrelation function (ACF) is the correlation coefficient of two observations in a times series [14]. So, autocorrelation represents the degree of similarity between a value and its lagged (backward shifted) versions over successive time intervals [15]. ACF plot, also known as correlogram, gives values of autocorrelation of any series with its lagged values [13]. When there is trend in data, the autocorrelations for small lags are positive and large and when there is seasonality the autocorrelations are large for the seasonal lags [4]. From the ACF plot it can be said that for the sales data there is weekly seasonality, because autocorrelations are relatively large for the lag of order 7 and the multiples of 7. Lead time of the stores is 11 day, so when making predictions for the next 11 days, 14 day lagged version of sales is known and it is included as a predictor to the data set.

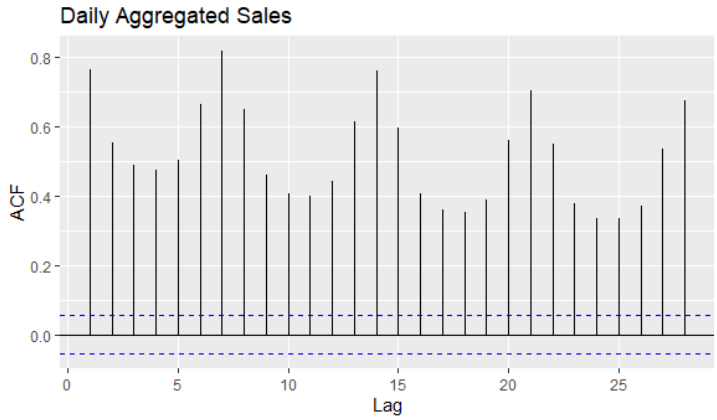

Figure 2. ACF plot of Daily Sales

3.3. Features

Each feature and its associated type are given in Table 1.

Variable	Type of Variable
Store	**Categorical**
Product	
Month	
Weekofyear	
Dayofweek	
City	
Season	
Outlet	**Binary**
Weekend	
Newyear	
Christmas	
Valentineday	
Womensday	
Nowruz	
Unityday	
Defenderday	
Victoryday	
Endofschool	
Constitutionday	
Backtoschoolday	
Blackfriday	
PastStock	**Numerical**
PastSales	
LaggedSales	

Table 1. Features

Categorical features are specified as “Factor” type in R. Store and product family is represented by two different categorical features. There are also 4 categorical features extracted from the date column of the initial data set:

1) Month: Month number of the observation. This variable’s possible values are from 1 to 12.

2) Week of Year: Week number of the observation. This variable’s possible values are 1 to 53

3) Day of Week: Weekday of the observation. Its possible values are from 1 to 7.

4) Day of Month: Day of the month for the observation. Its possible values are from 1 to 31.

To include the possible effects of weekends and seasons on sales, two categorical features are created:

1) Season: Season of the observation. It can take value from four different categories Winter, Spring, Summer, and Fall.

2) Weekend: If the date falls on a weekend or not. It is a binary variable.

Two of the categorical variables are related to the attributes of stores:

1) City: Location/City of the store. There are 12 different cities/values in the data set.

2) Outlet: Whether the store is inlet or outlet. It is a binary variable.

Other explanatory categorical features are related to special days. They take the value 1, if the date is a special date and 0 otherwise.

Numerical features are specified as “numeric” type in R. Numerical features in the data set are:

1. PastStock: stands for the past year’s stock level for the given store and product family pair.

2. PastSales: stands for the past year’s number of sales for the given store and product family pair.

3. LaggedSales: stands for the current year’s sales value (response variable) shifted back by 14 days.

3.4. Exploratory Data Analysis

The data is analyzed at different aggregation levels to gain insight into data. From the exploratory data analysis, it can be inferred that there is annual seasonality in the sales data.

3.4.1. Country Month Level

The auto correlation plot of monthly sales is given in Figure 4. The correlation is large for lag of 12 months and this shows that beside weekly seasonality, there is also annual seasonality in the data. Therefore, this time series exhibit a complex seasonal pattern known as multiple seasonality.

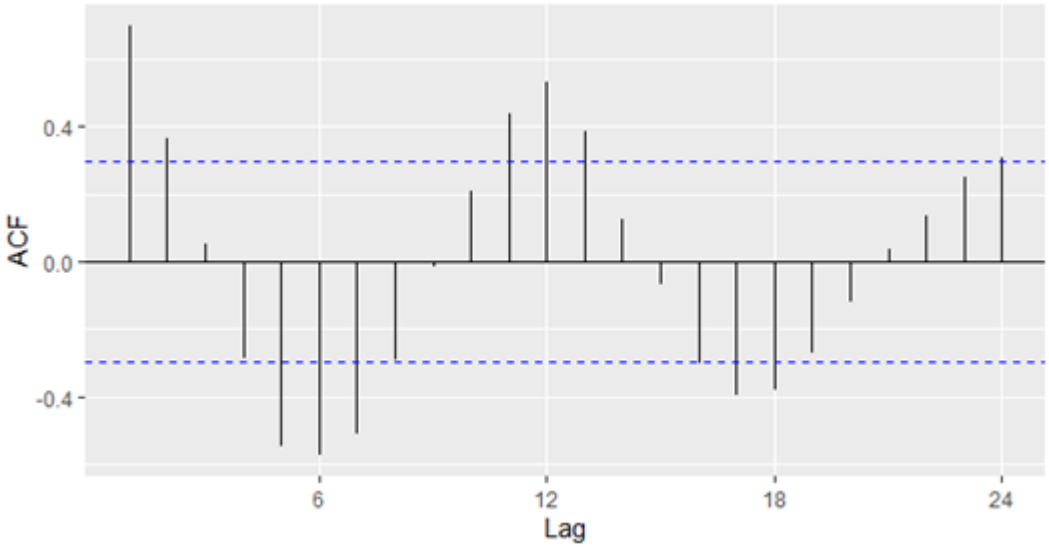

Figure 3. ACF plot of Monthly Sales

The seasonal plot with polar coordinates depicted by Figure 5 shows that sales are higher in June and July and are lower in November, December and January over the years.

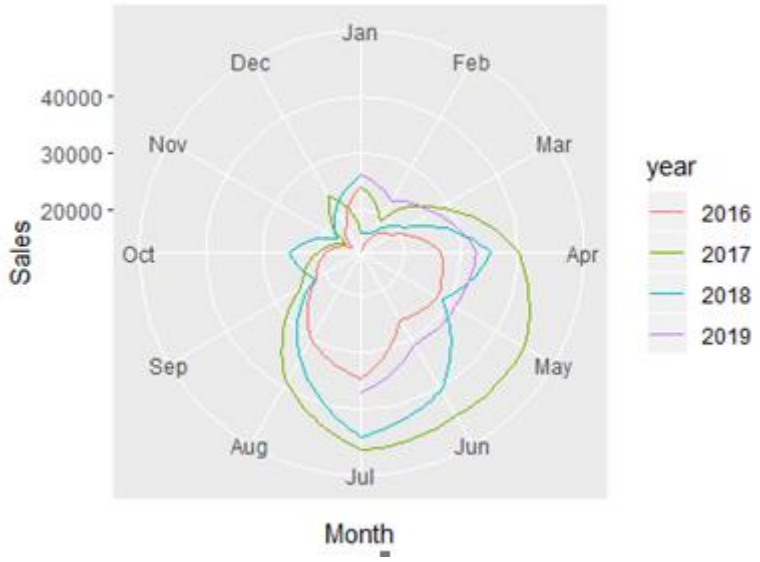

Figure 4. Seasonal Plot of Monthly Sales

3.4.2. Country Day Level

A line plot of total daily sales (sales over all the stores and products) for years 2016 to mid-2019 is given in Figure 6.

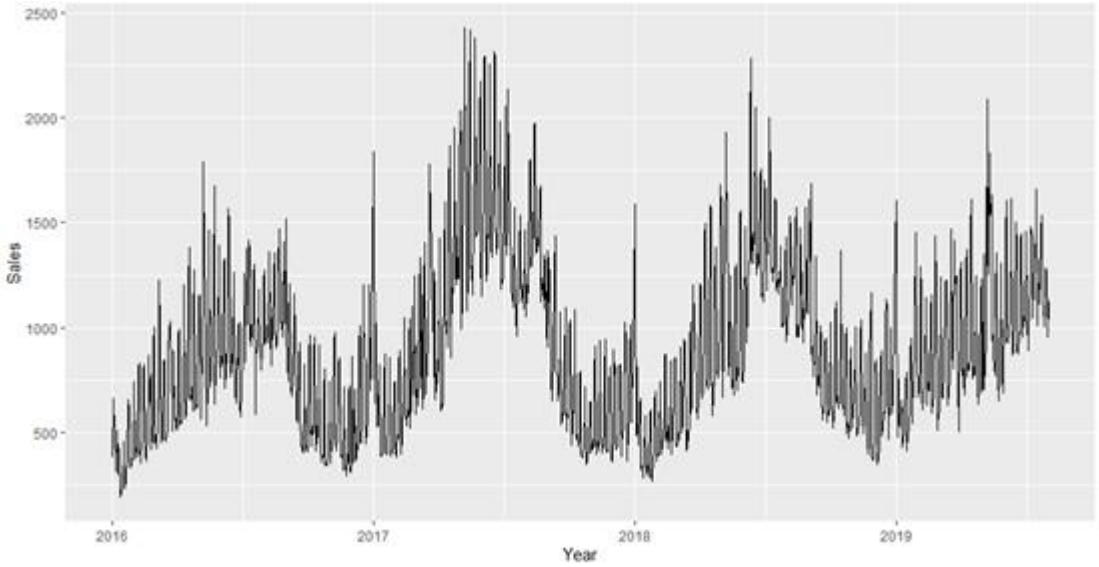

Figure 5. Line plot of total daily Sales

Another representation of the annual seasonality is shown by Figure 7, where each color represents a different year's daily sales data. The annual seasonality can be seen by the similar upward and downward movements occurring at the same time of each year.

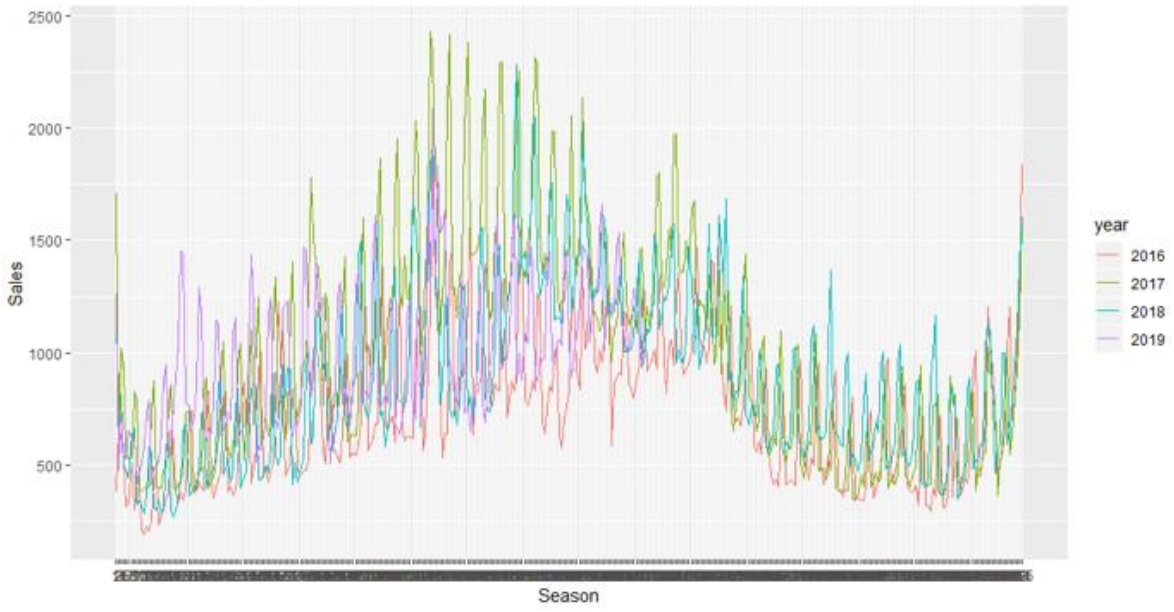

Figure 6. Seasonal plot of total daily Sales

3.4.3. Country Product Family Level

By eyeballing to the line plot of the total daily sales of different product families given by Figure 8, it can be said that seasonality and the trend are similar for these 5 product families. So, all of them have a similar sales profile over the years.

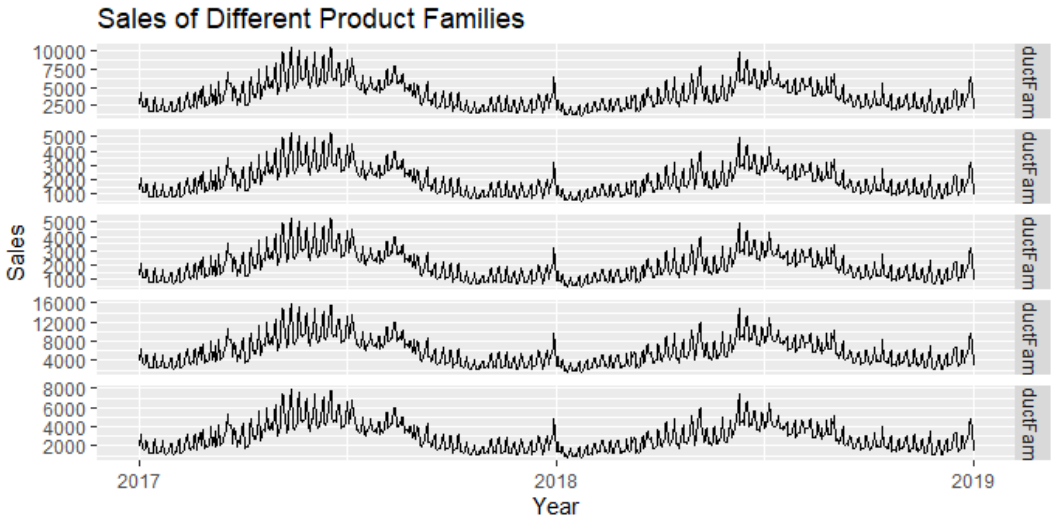

Figure 7. Line Plots of Product Families

4. Methodology

The machine learning models in this project include bagging tree regressor, random forest regressor, and gradient boosting regressor. The remainder of this section review these models briefly. Besides, it overviews scaling techniques for numerical data.

4.1. Feature Scaling

Due to the difference in the order of magnitudes of features, features should be transformed by using feature scaling methods to get them to the same scale before applying any algorithm. Feature scaling may improve the performance of the machine learning algorithms. It is only applied to features and not to the response variable. To this end, two feature scaling methods are employed, namely min-max scaling and standard scaling, which are two popular scaling algorithms.

Min-Max scaling is also called normalization. To apply min-max scaling, the minimum of a column/feature/sample is subtracted from every element of that column/sample and then divided by the difference of the maximum value minus the minimum value. After the min-max scaling the value of the transformed feature lie between 0 and 1 [16].

Standard scaling is also called standardization. In this method features are standardized by removing the mean value of the column from each value and then by dividing them by the standard deviation of the sample/column [17].

Features are transformed by both scaling algorithms initially, and the scaling algorithm yielding the lowest prediction error is preferred for further analysis such as fine tuning of the prediction models.

4.2. Forecasting Methods

Short term forecasts are generated with three three different ensemble learning algorithms, namely bagging tree regressor, random forest regressor and gradient boosting regressor. Ensemble methods combine decision trees as weak learners to achieve higher predictive performance [28]. Thus, decision trees are base estimators for ensemble methods.

Decision tree is a supervised learning method. It is a highly preferred machine learning method due to its simplicity in visualizing and interpreting results and they are also the building block of the random forests. However, decision trees typically suffer from high variance [25].

Decision tree regressors divide the predictor space into several regions and assign each response in the training set to a region [25]. To divide the feature space, a decision tree first splits the training set into two regions by choosing a single feature and determining a threshold value for that feature [16]. At each split, the feature to split on is chosen such that mean squared error (MSE) is minimized [16]. The splitting continues until the MSE is minimized or the maximum depth is reached. If the maximum depth of the decision tree regressor is not specified, it is prone to overfitting. To predict the value of a new observation, its region/leaf in the tree is found and typically mean or median of the training observations in that leaf is the prediction [25].

In machine learning, ensemble learning algorithms are meta-algorithms that combine several predictors into one predictive model [18]. An ensemble method can be either sequential, where base learners are trained iteratively (e.g. Gradient Boosting) or parallel, where base learners are trained parallelly (e.g. Bagging Tree, Random Forest). The aim of the ensemble methods is combining a group of weak learners to form a strong learner [20]. A classic weak learner used in ensemble models is decision tree.

Bagging stands for bootstrap aggregation. Bootstrapping is a sampling method where random samples are drawn from the population with replacement. A bagging tree regressor is an ensemble method that fits a number of weak learners to the bootstrapped random subsets of the data and combines their predictions into a single one. So, weak learners are combined to form a strong learner. In bagging, random samples are drawn with replacement. Generally, the decision trees are preferred as the weak learners [19]. In building of the individual regression trees all the features in the data set are considered. Bagging tree regressor aggregates the individual estimations from the decision trees by averaging to generate the final estimate [21]. Estimation from the bagging has a lower variance than the individual estimates due to combining/averaging estimates of different models [19]. By fitting a bagging regression tree, the importance of each predictor can be obtained by using the residual sum of squares (RSS) [16]. Bagging Tree Regressor's tuning parameter is only the number of decision trees to grow.

Random forest regressor is an ensemble method that fits a specified number of decision trees to the bootstrapped random subsets of the data [16]. To give an estimation, random forest regressor averages estimation from the individual decision trees and this improves both the decision accuracy and prevents over-fitting [22]. Unlike bagging tree regressor, in random forest only a subset of features is randomly selected to create each individual decision tree regressor/model. So, at each split, the random forest considers only a random sample of the predictors which causes uncorrelated trees [25]. In random forest both selecting a random sample of the data and a random subset of the features to build the decision tree results in a low variance model. Like bagging tree regressor, the importance of each predictor can be obtained from the random forest regressor by ordering the predictors according to the decrease in RSS due to splitting over the given predictor over all the trees [16]. Random forest's tuning parameters are the number of decision trees in the forest and the number of features considered by each tree at each split [26].

In boosting, weak learners are built sequentially. Boosting algorithms use weighted averages to turn weak learners into strong learners [19]. At each iteration the new model tries to correct the errors of its predecessor model [16]. In boosted regression trees, each decision tree is built using the information from the previously grown tree [25]. Each tree is fit to a modified version of the training set to explain the residuals of the previous model [25]. Boosted Regression Tree's tuning parameters are number of trees to grow, shrinkage parameter and the number of splits in each tree.

AdaBoost and Gradient Boosting are the popular implementations of boosting method. In gradient boosting, each model tries to explain the residuals of its predecessor model [16]. The advantage of gradient boosted regression tree is that it can handle both categorical and numerical features and robustness to the outliers in the response variable [23].

5. Results

To predict the sales during the replenishment lead time of the stores, bagging tree regressor, random forest regressor and gradient boosted regression trees are trained using the daily sales data from 2017 to 2019. The test set consists of the daily sales from January 2019 to August 2019. The accuracy scores are calculated for the test set. All algorithms are built in R version 3.6.1 (2019-07-05). Their results are compared with the traditional forecasting method linear regression.

3 different data sets are used to train the bagging tree regressor and random forest regressor models:

i) Unscaled Data: Numerical features are not scaled.

ii) Standardized Data: Numerical features are standard scaled.

iii) Normalized Data: Numerical features are min-max scaled.

After training each model with these 3 different data sets, the data set which yields higher train and test R^2 value is kept for hyperparameter tuning and sales prediction.

5.1. Computational Results of Bagging Regression Tree

Bagging regression tree is imported from the randomForest package of R. Version 4.6-14 of the package is installed on R. R^2's of bagging regression tree for scaled and unscaled training and testing data set are given in Table 1. Each model is trained by constructing 100 different trees.

Bagging Tree	Train R^2	Test R^2
Min Max Scaled	**79.0%**	**70.8%**
Standard Scaled	79.2%	70.7%
Unscaled	79.3%	70.5%

Table 2. R^2 for Bagging Regression Tree

Min-Max scaled data yields the highest test R^2. The importance of variables based on their effect on MSE is shown in Figure 9.

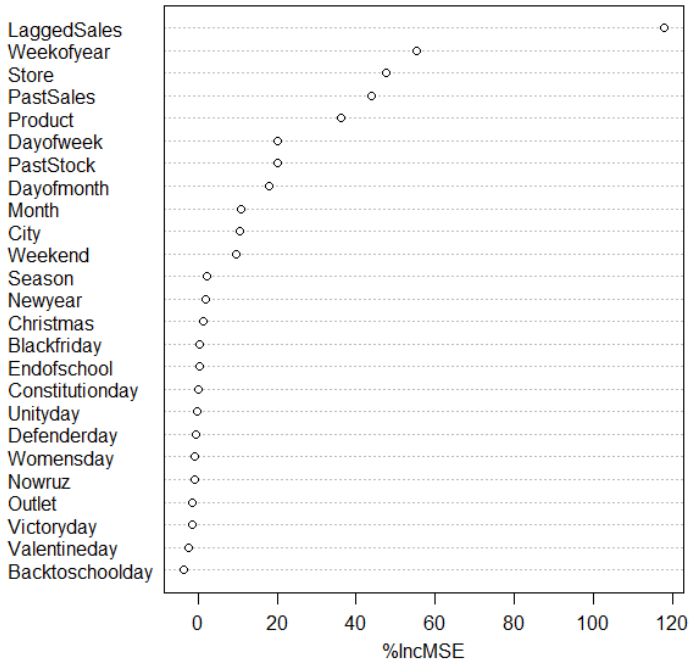

Figure 8. Variable Importance from Bagging Regression Tree

The features yielding an increase in MSE are listed in Table 2. Thus, increased MSE leads to an increased prediction error. According to the Table 2, the least important variables are Backtoschoolday, Valentineday and Victoryday.

As a result, Backtoschoolday, Valentineday and Victory day variables are removed from the feature set and the model is trained with the remaining features.

Backtoschoolday	Valentineday	Victoryday	Outlet	Nowruz
-3.5556434	-2.3970290	-1.4559029	-1.3704848	-0.9256290

Table 3. Variable Importance from Bagging Regression Tree

The R^2 of the resulting tree is 79.2 % on the training set and 71.1 % on the testing set.

5.2. Computational Results of Random Forest Regression

Two different implementations of random forest regression tree are trained. They are imported from the randomForest package and caret package of R. Version 4.6-14 of the

randomForest package and version 6.0-84 of caret is attached on R. First random forest regressor is built by combining 100 different decision trees to determine the minimum size of the terminal nodes. Table 3 shows R^2 values of different node sizes and bold text indicated the best model with the highest test R^2 :

Node Size	Train R^2	Test R^2
10	81.34%	72.46%
20	81.51%	72.55%
30	81.55%	72.66%
40	81.35%	72.80%
50	**81.14%**	**72.88%**

Table 4. R^2 for Random Forest Regressor

Node size of the terminal nodes are selected as 50, because it leads to the highest R^2 on the test set. Features which decreases the MSE are removed from the feature set. Removed features are given in Table 4.

Backtoschoolday	Victoryday	Valentineday
-1.434460	-1.184043	-0.721627

Table 5. Variable Importance from Random Forest Regressor

A grid is built for hyperparameter tuning. The grid contains different values of the number of features randomly selected as candidates to split on (mtry). For regression models, (number of features)/3 is the default value of the mtry. The highest R^2 value is achieved when mtry is set to 8. After removing the features and tuning the number of features, the R^2 values for scaled and unscaled data are given in Table 5:

Random Forest				
Scaling	mtry	Node Size	Train R^2	Test R^2
Min Max Scaled	**8**	**40**	**81.30%**	**72.88%**
Standard Scaled	8	40	81.55%	72.86%
Unscaled	8	40	81.46%	72.77%

Table 6. R^2 for final Random Forest Regression

The variable Importance for the final random forest model is depicted in Figure 10. Store and lagged sales are especially important in predicting the sales.

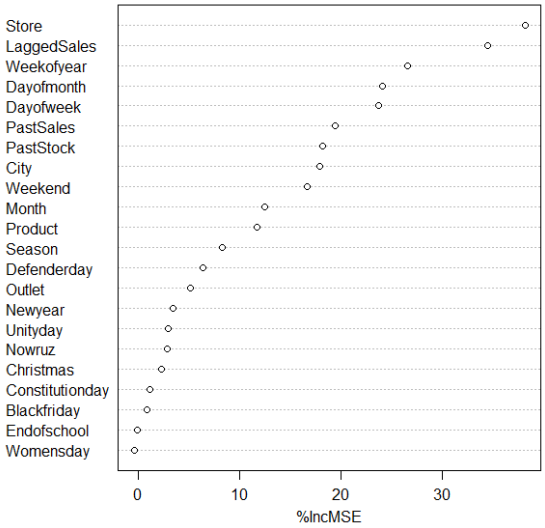

Figure 9. Variable Importance for the Final Random Forest Regression Tree

The final random forest regression tree yields a root mean square error (RMSE) of 7.71 on the test data set for the first 7 months of 2019. Target values versus predictions is plotted in Figure 11. From the graph it can be said that as the value of the sales increases over 100, deviations/errors tend to increase. For the lower target "sales" values, the predictions from the random forest regression tree are in line with the target.

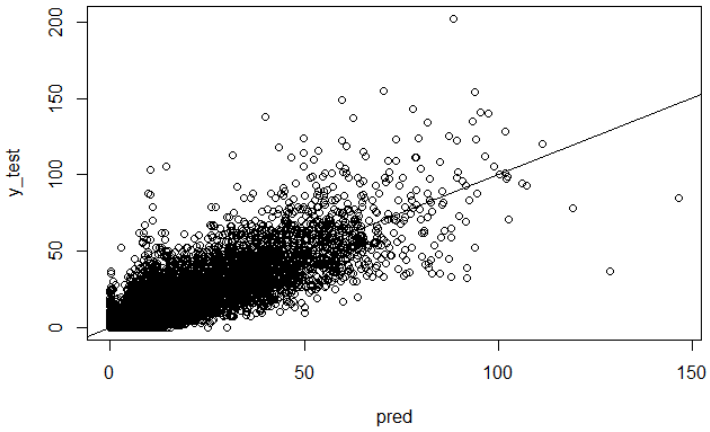

Figure 10. Predictions vs Test Response Value

Random forest regression tree also built by using the caret package with 10-fold cross validation. Variable importance from the resulting model is given in Figure 12. Store and Lagged Sales are the most influential variables. Victor Day and Women Day are the least influential variables on sales.

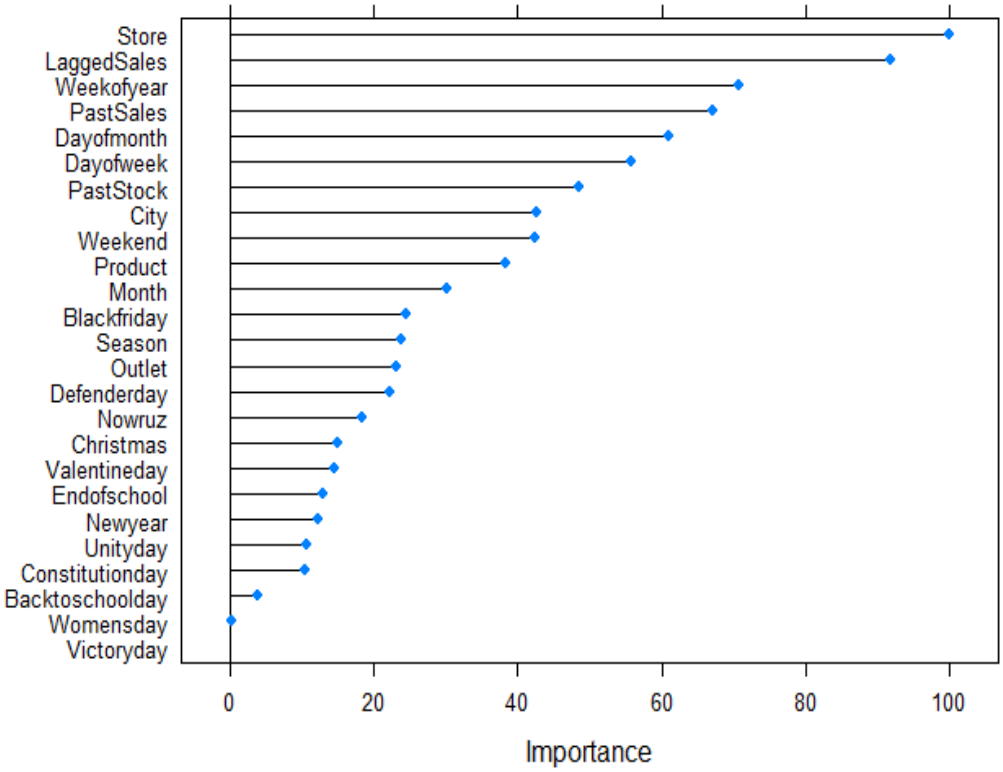

Figure 11. Variable Importance Plot for Random Forest Regressor

This version of random forest regressor yields an RMSE of 7.01, R^2 value of 81 % and MAE (mean absolute error) of 3.86 on the 10-fold cross validated data. Thus, the error of the final random forest model is less than the error obtained by the linear regression model used as a benchmark.

5.3. Computational Results of Gradient Boosted Regression Tree

Gradient boosting machine (GBM) is imported from the caret package by wrapping the gbm package of R. Version 2.1.5 of the gbm package and version 6.0-84 of caret are attached on R.

For hyperparameter tuning, a grid is constructed containing various values of the

i) the number of trees to grow

ii) the shrinkage parameter

iii) the number of splits in each tree

After constructing several grids consisting of different values for the hyperparameters, the lowest RMSE is achieved when the number of trees is 1000, the shrinkage parameter is 9 and the interaction depth is 9 by 10-fold cross validation. Importance of features resulting from gbm is given in Figure 13. The most important feature is Lagged Sales and Season, whereas Valentineday, Christmas, Endofschool and Womenday are the least important features.

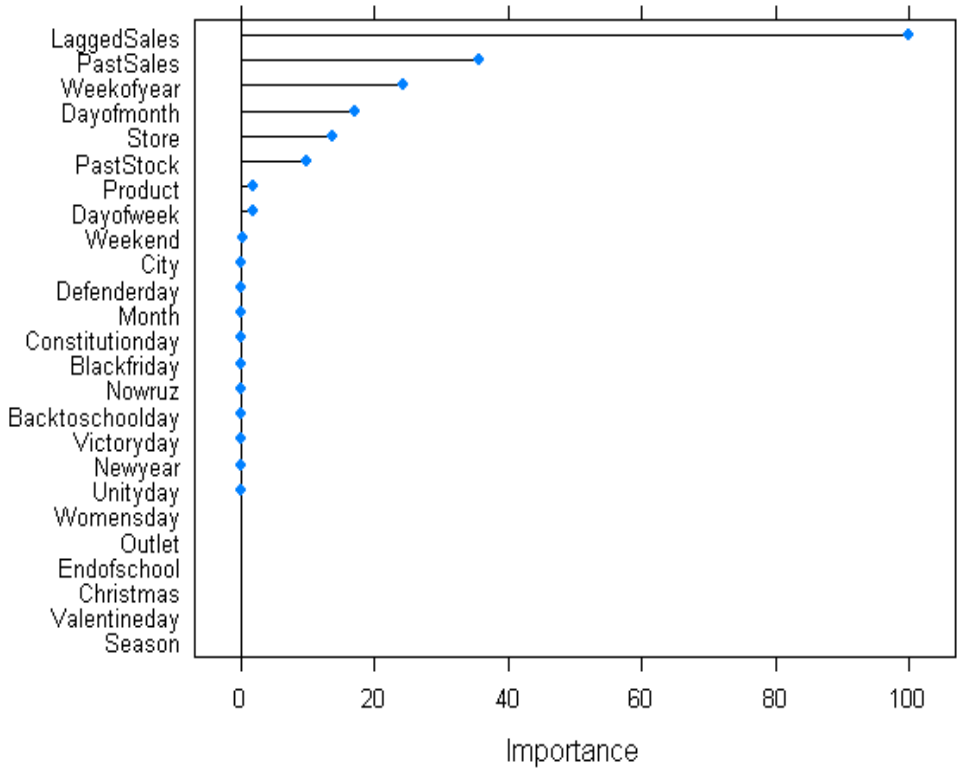

Figure 12. Variable Importance from Gradient Boosted Regression Tree

R^2 value for the training data set is 83.0 %. R^2 value of the gbm on the test data set is 71 % and RMSE score is 8.01. Finally, predictions from gbm vs actual sales values are plotted in Figure 14 for training data set. Again, as sales values increase, the errors tend to increase which signals that some outlier check could improve the performance of the regressors.

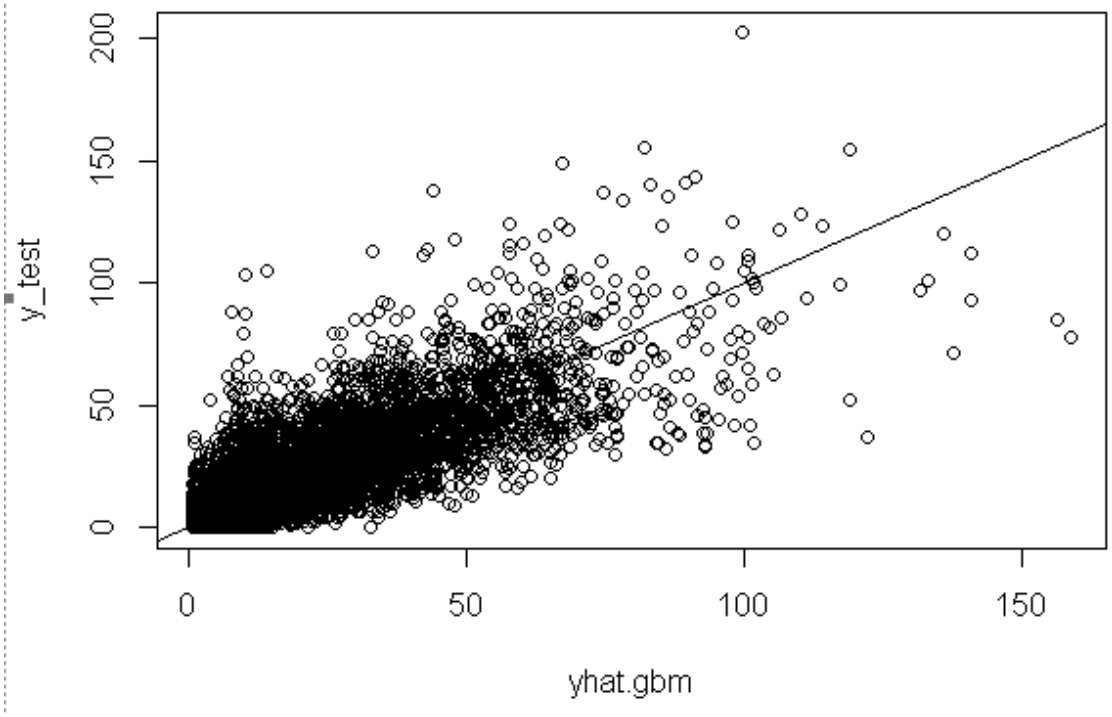

Figure 13. Predictions vs Test Response Value

5.4. Comparison of Results

The comparison of results for different machine learning models and linear regression are presented in Table 8. Random Forest Regressor has the highest explanatory power (R^2) on future sales and the lowest prediction error (RMSE) on the test set, whereas benchmark model, linear regression, has both the lowest explanatory power (R^2). Overall, the performance of the ensemble methods is better than the linear regression in terms of both the explanatory power and the prediction error.

	Test Set R^2	**RMSE**
Bagging Regression Tree	70.8 %	8.05
Random Forest Regressor	72.88 %	7.71
Gradient Boosted Regressor	71.07%	8.01
Linear Regression	70.03%	8.06

Table 7. Comparison of R^2 and RMSE for Different Forecasting Techniques

6. Conclusion

The objective of this project is to study the effectiveness of ensemble regression methods on forecasting the demand of various products at different stores. Three different ensemble models are built to predict the in-store sales of the next three months for multi-product multi-store data. However, ultimately sales forecasts are generated by using a single model.

Although showing better results both in terms of explanatory power on the target variable (R^2) and prediction error (MSE), ensemble methods do not provide a huge improvement over the linear regression with the available feature set. As the ACF plot suggests, interaction between the observations are the most important feature for multi-product, multi-store retail sales. Thus, future research can be directed towards feature engineering on the lagged values of sales or building autoregressive neural networks.

APPENDIX A

```
countryts <- readRDS(paste0(here(),"/input/countryts.rds"))
data <- readRDS(paste0(here(),"/input/data.rds"))
#line plot
autoplot(countryts) +
ggtitle("Sales of BUN") +
xlab("Year") +
ylab("Sales")
#Integer to Factor/Categorical
data$Store <- as.factor(data$Store)
data$Product <- as.factor(data$Product)
data$Year <- as.factor(data$Year)
data$Month <- as.factor(data$Month)
data$Weekofyear <- as.factor(data$Weekofyear)
data$Dayofweek <- as.factor(data$Dayofweek)
data$Dayofmonth <- as.factor(data$Dayofmonth)
data$Outlet <- as.factor(data$Outlet)
data$City <- as.factor(data$City)
data$Season <- as.factor(data$Season)
data$Weekend <- as.factor(data$Weekend)
# data$Holiday <- as.factor(data$Holiday)
data$Newyear <- as.factor(data$Newyear)
data$Christmas <- as.factor(data$Christmas )
data$Valentineday <- as.factor(data$Valentineday)
data$Womensday <- as.factor(data$Womensday)
data$Nowruz <- as.factor(data$Nowruz)
data$Unityday <- as.factor(data$Unityday)
data$Defenderday <- as.factor(data$Defenderday)
data$Victoryday <- as.factor(data$Victoryday)
data$Endofschool <- as.factor(data$Endofschool)
data$Constitutionday <- as.factor(data$Constitutionday)
```

```
data$Backtoschoolday <- as.factor(data$Backtoschoolday)
data$Blackfriday <- as.factor(data$Blackfriday)
# data <- data %>% select(-c(Percipitation), everything())
# data <- data %>% select(-c(Temperature), everything())
data <- data %>% select(-c(PastStock), everything())
data <- data %>% select(-c(PastSales), everything())
data <- data %>% select(-c(LaggedSales), everything())
data <- data %>% select(-c(Sales), everything())
data %<>% select(-Year)
# Train Test Split
train_len <- which.min(data$Date<ymd("2019-01-01"))-1
data_len <- nrow(data) - train_len
training <- slice(data,1:train_len),
testing <- slice(data,train_len+1:data_len)
y_train <- training[,27]
y_test <- testing[,27]
X_train <- training[,2:26]
X_test <- testing[,2:26]
# Bagged Tree
start.time <- Sys.time()
bagFinal <- randomForest(y = y_train, x = X_train, ytest = y_test, xtest = X_test, mtry=length(X_train), ntree=100, nodesize = 20, keep.forest = TRUE, importance=TRUE)
end.time <- Sys.time()
time.taken <- end.time - start.time
# randomForest ,train-test split ntree=200, mtry=8, nodesize= 40 Unscaled
start.time <- Sys.time()
rfFinal <- randomForest(y = y_train, x = X_train, ytest = y_test, xtest = X_test, mtry=8, ntree=200, nodesize = 40, importance=TRUE, keep.forest = TRUE)
end.time <- Sys.time()
time.taken <- end.time - start.time
## gbm
training$weekend <- as.factor(training$weekend)
```

```
boost.KZ = gbm(value~.-time, data=training, distribution="gaussian", n.trees=1000, interaction.depth=4, shrinkage = 0.001, n.cores = 16)
summary(boost.KZ)
#predicting using validation set
yhat.boost=predict(boost.KZ,newdata=testing,n.trees=1000)
mean((yhat.boost-testing$value)^2)
```

REFERENCES

[1] Thomassey S. (2014) Sales Forecasting in Apparel and Fashion Industry: A Review. In: Choi TM., Hui CL., Yu Y. (eds) Intelligent Fashion Forecasting Systems: Models and Applications. Springer, Berlin, Heidelberg

[2] Na Liu, Shuyun Ren, Tsan-Ming Choi, Chi-Leung Hui, and Sau-Fun Ng, "Sales Forecasting for Fashion Retailing Service Industry: A Review," Mathematical Problems in Engineering, vol. 2013, Article ID 738675, 9 pages, 2013. https://doi.org/10.1155/2013/738675.

[3] Thomassey, S. (2010). Sales forecast in clothing industry: The key success factor of the supply chain management. Int. J. Production Economics 128 ,470–483.

[4] Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. Accessed on 18/05/2019

[5] Makridakis, Spyros, Steven C. Wheelwright, and Rob J. Hyndman, Forecasting: Methods and Applications, Third edition. John Wiley and Sons, 1998.

[6] Wisdomjobs.com, 'SHORT, MEDIUM AND LONG-TERM FORECASTING – MARKETING MANAGEMENT', 2014. [Online]. Available: https://www.wisdomjobs.com/e-university/marketing-management-tutorial-294/short-medium-and-long-term-forecasting-9586.html. [Accessed: 3- Aug- 2019].

[7] Giustiniano, Luca & Nenni, Maria & Pirolo, Luca. (2013). Demand Forecasting in the Fashion Industry: A Review. International Journal of Engineering Business Management. 5. 10.5772/56840.

[8]Thomassey, S.,Happiette,M.,Castelain,J.M.,(2005). A short and mean-term automatic forecasting system—application to textile logistics. European Journal of Operational Research161(1),275–284.

[9] J. Spacey, "12 Types of Fashion Product," *Simplicable*, 21-Feb-2017. [Online]. Available: https://simplicable.com/new/fashion-products. [Accessed: 15-Aug-2019].

[10]S. Thomassey and M. Happiette, "A neural clustering and classification system for sales forecasting of new apparel items," Applied Soft Computing Journal, vol. 7, no. 4, pp. 1177–1187, 2007

[11] Carbonneau R, Laframboise K and Vahidov R (2008), "Application of Machine Learning Techniques for Supply Chain Demand Forecasting", European Journal of Operational Research, Vol. 184, pp. 1140-1154.

[12] Thomassey, S., Fiordaliso, A. (2006). A hybrid sales forecasting system based on clustering and decision trees. Decision Support Systems 42 (1), 408–421.

[13] J. Salvi, "Significance of ACF and PACF Plots In Time Series Analysis," *Medium*, 27-Mar-2019. [Online]. Available: https://towardsdatascience.com/significance-of-acf-and-pacf-plots-in-time-series-analysis-2fa11a5d10a8. [Accessed: 15-Aug-2019].

[14] I. Pardoe, "10.2 - Autocorrelation and Time Series Methods," *10.2 - Autocorrelation and Time Series Methods | STAT 462*. [Online]. Available: https://newonlinecourses.science.psu.edu/stat462/node/188/. [Accessed: 15-Aug-2019].

[15] "What is lag in a time series". [Online]. Available: https://math.stackexchange.com/questions/2548314/what-is-lag-in-a-time-series [Accessed: 10- Aug- 2019].

[16] Géron, A, Hands-On Machine Learning with Scikit-Learn & TensorFlow, 1st Edition, O'Reilly Media, 2017

[17] "sklearn.preprocessing.StandardScaler" *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html. [Accessed: 15-Aug-2019].

[18] V. Smolyakov, "Ensemble Learning to Improve Machine Learning Results" *Medium*, 22-Aug-2017. [Online]. Available: https://blog.statsbot.co/ensemble-learning-d1dcd548e936. [Accessed: 15-Aug-2019].

[19] J. Shubham, "Ensemble Learning - Bagging and Boosting" *Medium*, 06-Jul-2018. [Online]. Available: https://becominghuman.ai/ensemble-learning-bagging-and-boosting-d20f38be9b1e. [Accessed: 15-Aug-2019].

[20] R. Garg, "A Primer to Ensemble Learning – Bagging and Boosting" *Analytics India Magazine*, 19-Feb-2018. [Online]. Available: https://www.analyticsindiamag.com/primer-ensemble-learning-bagging-boosting/. [Accessed: 15-Aug-2019].

[21] "sklearn.ensemble.BaggingRegressor" *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingRegressor.html#sklearn.ensemble.BaggingRegressor. [Accessed: 15-Aug-2019].

[22] "sklearn.ensemble.RandomForestRegressor" *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html. [Accessed: 15-Aug-2019].

[23] "1.11. Ensemble methods" *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/ensemble.html. [Accessed: 15-Aug-2019].

[24] H. Arsham, "Time-Critical Decision Making for Business Administration," *Time Series Analysis for Business Forecasting*. [Online]. Available: http://home.ubalt.edu/ntsbarsh/Business-stat/stat-data/Forecast.htm. [Accessed: 15-Aug-2019].

[25] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. New York: Springer.

[26] W. Koehrsen, "Hyperparameter Tuning the Random Forest in Python" *Towards Data Science*, 10-Jan-2018. [Online]. Available: https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74. [Accessed: 15-Aug-2019].

[27] H. Kandan, "Bagging the skill of Bagging (Bootstrap aggregating)" *Medium*, 14-Feb-2018. [Online]. Available: https://medium.com/@harishkandan95/bagging-the-skill-of-bagging-bootstrap-aggregating-83c18dcabdf1. [Accessed: 15-Aug-2019].

[28] A. Nagpal, "Decision Tree Ensembles- Bagging and Boosting" *Towards Data Science*, 17-Oct-2017. [Online]. Available: https://towardsdatascience.com/decision-tree-ensembles-bagging-and-boosting-266a8ba60fd9. [Accessed: 07-Sep-2019].