Project Report

Time-Series Forecasting of Walmart Sales Data

Shiraz Wasim (Team 4)

Abstract: In this work, forecasting models are developed for the time-series analysis of Walmart sales data. The target variables are the quantity of sales in three distinct categories: Hobbies, Foods and Household. Two statistical (moving average and Holt's Winter method) and two machine learning (gradient boosted trees and random forests) models are developed. A 7-day rolling window was implemented for the moving average. An additive variation of the Holt's Winter method, with seasonal periods of 7 days, was used to account for the seasonality and trend discovered in the data during exploratory data analysis. For the ML models, features were prepared using one-hot-encoding. Extra features were also generated by applying lags to the sales data. For model optimization, an increasing window rolling *k*-folds validation and a feature importance study were implemented; hyperparameter tuning using grid search was also performed. The performance of the models was evaluated on three time-horizons (1-day, 7-days and 28-days) using the sMAPE metric. The results showed that the ML models outperformed the statistical models for all three categories and time-horizons. The random forests approach achieved the lowest error in all cases and also achieved the lowest error overall with a value of 2.88%. Future work would be the addition of window features or developing a model that uses a combination of statistical and machine learning methods.

Keywords: Time-series; forecasting; seasonality; statistical models; machine learning; one-hot-encoding, feature generation; rolling *k*-folds validation; model optimization; sMAPE

1. Introduction

Several technological advances in the areas of computing, such as database systems, machine learning and cloud computing have resulted from the recent conversion of data into useful information to support decision making. Advances, in these areas, have also led to the development of computer systems that are capable of storing, analyzing and managing large and increasing amounts of data. In particular, temporal information, where data is organized in a chronological order, is a data representation that has attracted a lot of interest and has driven the creation of these large databases [1]. Data, that is ordered in this way, is known as a time-series.

A time-series is a series of data points indexed (or graphed) in time order and often taken at successive equally spaced points in time. Examples of time-series include the opening and closing prices of a stock, the infection rates of the COVID-19 disease or the change in global temperature levels. The objective of time-series analysis is to extract meaningful statistics and other characteristics from a set of data. Specifically, time-series forecasting is the development of a model to predict future values based on previously observed values. Thus, time-series forecasting methods rely on the idea that historical data includes intrinsic patterns which convey useful information for the future description of the phenomenon investigated. Obtaining these patterns is often non-trivial and their discovery is one of the primary goals of time-series analysis [1].

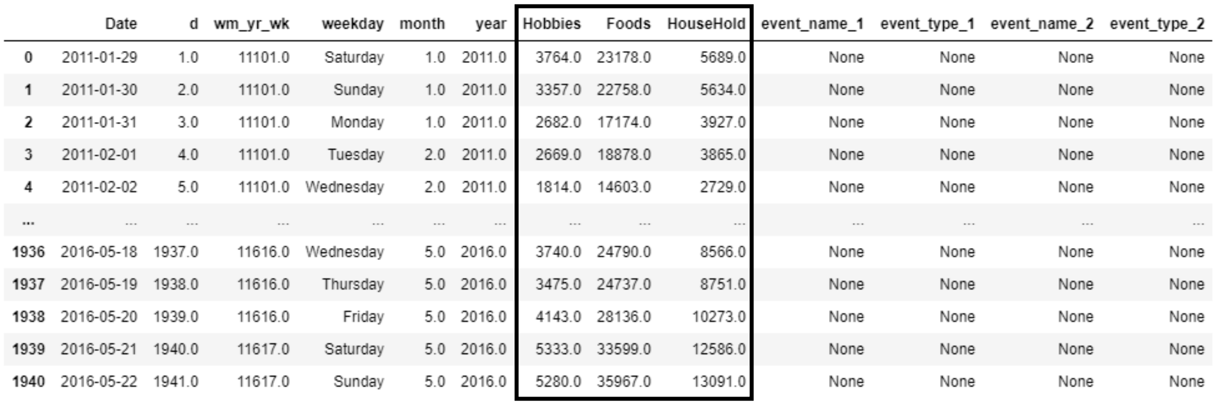
There exists extensive research into developing forecasting models. There are two main approaches: using statistical or machine learning (ML) methods. Statistical methods attempt to fit a mathematical model to the dataset, often through the use of recursive formulas. A common approach is to forecast future values by applying weights to past observations, such as an equal weight in a simple moving average or an exponentially decreasing weight in exponential smoothing techniques. In addition, a large variety of ML approaches have been proposed in the literature. These include linear regression, k-nearest neighbor regression, multilayer perceptron's, recurrent neural networks, ensemble methods (gradient boosted trees and random forests) and several others. In fact, a large portion of the literature compares the performance of statistical and ML approaches on time-series analysis using empirical results [2]. From this literature, two conclusions using empirical results are repeatedly made: (1) purely statistical approaches generally outperform purely ML approaches (in the presence of limited data) and (2) advanced combination methods, where one or more statistical or ML approaches are combined, outperform purely statistical or purely ML models. In this report, we do not consider combination methods and limit our analysis to two purely statistical methods: a simple moving average (MA) and Holt's Winter Method (Triple Exponential Smoothing) and two purely ML methods: gradient boosted trees (GBT) and random forests (RF).

2. Data Preprocessing and Evaluation Criteria

The data utilized in this work was obtained from the M5 Kaggle competition [3]. The M-Series is a series of competitions aimed at developing the state-of-the-art in forecasting models for time-series data. M5, the fifth iteration of this competition, provides contestants with Walmart sales data for a period of 1941 days (approx. 5.4 years). The objective is to develop forecasting models that are optimized for accuracy (i.e. minimizing the error between the prediction and the target). The dataset, the initial data processing, the evaluation metric and the time horizons considered are presented in this section.

2.1. Dataset, Aggregation and Target Variable

The original Walmart dataset, consisted of the unit sales of various products sold in the USA, organized in the form of grouped time-series. More specifically, the dataset included the unit sales of 3,048 products, classified into 3 product categories (Hobbies, Foods, and Household) and 7 product departments, in which the mentioned categories are disaggregated. The products are sold across ten stores, located in three States (CA, TX, and WI). This results in a total of 30,490 time-series on the product level alone. Herein, to reduce the complexity of this project and the number of time-series being considered, the sales data was aggregated over each product category, resulting in 3 time-series: one for Hobbies, one for Foods and one for Household. The aggregated values, on each day for the 1941 days, are shown in Figure 1. Additional information about each specific day such as the day of the week it is or the month it's in, etc., is also included. Thus, the target variable, in this work, is the expected total sales, within a particular category, on a given day. The extra details provided about each day serve as the features in the ML models.



	Date	d	wm_yr_wk	weekday	month	year	Hobbies	Foods	HouseHold	event_name_1	event_type_1	event_name_2	event_type_2
0	2011-01-29	1.0	11101.0	Saturday	1.0	2011.0	3764.0	23178.0	5689.0	None	None	None	None
1	2011-01-30	2.0	11101.0	Sunday	1.0	2011.0	3357.0	22758.0	5634.0	None	None	None	None
2	2011-01-31	3.0	11101.0	Monday	1.0	2011.0	2682.0	17174.0	3927.0	None	None	None	None
3	2011-02-01	4.0	11101.0	Tuesday	2.0	2011.0	2669.0	18878.0	3865.0	None	None	None	None
4	2011-02-02	5.0	11101.0	Wednesday	2.0	2011.0	1814.0	14603.0	2729.0	None	None	None	None
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1936	2016-05-18	1937.0	11616.0	Wednesday	5.0	2016.0	3740.0	24790.0	8566.0	None	None	None	None
1937	2016-05-19	1938.0	11616.0	Thursday	5.0	2016.0	3475.0	24737.0	8751.0	None	None	None	None
1938	2016-05-20	1939.0	11616.0	Friday	5.0	2016.0	4143.0	28136.0	10273.0	None	None	None	None
1939	2016-05-21	1940.0	11617.0	Saturday	5.0	2016.0	5333.0	33599.0	12586.0	None	None	None	None
1940	2016-05-22	1941.0	11617.0	Sunday	5.0	2016.0	5280.0	35967.0	13091.0	None	None	None	None

Figure 1. The example dataset with the target variables highlighted

From a business perspective, the expected quantity of items sold is a useful metric for a company as it allows the pre-ordering of stock to match expected demand. Furthermore, estimates for the minimum and maximum future revenue, for each category, could be calculated by multiplying the predictions for the total quantity sold with the price of the cheapest and the most expensive item, in each category, respectively.

2.2. Evaluation Metric and Time Horizons

The evaluation metric used was the symmetric Mean Absolute Percentage Error (sMAPE). The formula for this is

$$sMAPE = \frac{100\%}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{(|A_t| + |F_t|)/2}, \tag{1}$$

where A_t is the actual/target value, F_t is the forecasted/predicted value and n is the number of target-prediction pairs being considered. The advantage of sMAPE is that it is scale-independent, so it can be used to compare forecast performance between various datasets.

In addition, three time horizons were used in this work: a 1-day, 7-day (week) and 28-day (month) time horizon. The different time horizons were used as products are often ordered in advance and how early they are ordered depends on the nature of the product. For example, some food items are ordered at most a day or a week in advance because they have a short shelf life. Furthermore, Walmart sells fast-moving consumer goods (FMCG). This means that the customer choices are dynamic so short time horizons are beneficial as they allow the necessary adaptation to any observed short-term changes to customer behaviour.

3. Stochastic/Statistical Models

3.1. Exploratory Data Analysis

A time series Z of size *n* can be formulated as an ordered sequence of observations, i.e. , $Z = (z_1, z_2, \ldots, z_n)$ where z denotes an observation at time *t* [1]. For the sales data, *z* represents the quantity of sales in any particular category and *t* represents a particular day. Thus, the Walmart data is an example of a time-series that is discrete and uniformly sampled over time. The three major components of a time-series are trend, seasonality and residue. Trend (T) is the long-term increase or decrease in the data. Seasonality (S) is the occurrence of cyclic patterns of variation that repeat, at relatively constant time intervals, with the trend component. An example of a seasonal pattern is the increased purchases of warm clothing in Winter or the peak in retail sales seen during the Christmas season. Residue (R) is the short-term random fluctuations that are neither systematic nor predictable. The trend component determines if the time series is stationary or not. A stationary series develops randomly around a constant average, reflecting some stable equilibrium [1]. Namely, if the series is stationary, then the mean and variance of the data do not change over time. The 'Trend' plot in Figure 2 shows that the time series for the Hobbies category displays an increasing trend. Similar increasing trends were seen for the Foods and Household categories (although they are not shown here).

Furthermore, the sales data contained seasonality as shown by the 'Seasonal' plot in Figure 2. This was also seen for the Foods and Household categories. From this exploratory data analysis, it is clear that an effective statistical model is one that incorporates seasonality and an increasing trend. The statistical models considered, in this work, are described in more detail below.

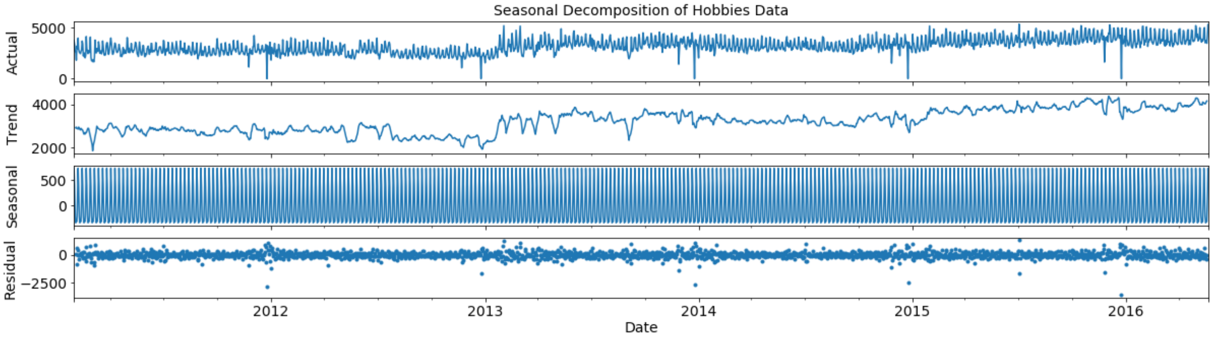


Figure 2. Seasonal decomposition of the aggregated Hobbies data

3.2. Moving Average

The Moving Average (MA) model is a simple technique where an arithmetic average of the last m values of the time series is used to predict the future value. The expression for this is

$$Z_{t+m} = \frac{Z_t + Z_{t-1} + \cdots + Z_{t-r+1}}{r}, \qquad (2)$$

where Z_t is the observation value at time t, Z_{t+m} is the prediction for m periods ahead (i.e. $m = 1$ for a 1-day time horizon) and r is the number of past observations to average on (i.e. the size of the rolling window). The higher the value of r, the more uniform (smoothed) the predicted behavior will be. The limitations of the MA model are its low accuracy and its inability to deal with trend and seasonality, which, as we have seen from the previous section, exists in the sales data. Furthermore, the weights assigned to the r observations are all equal. Therefore, there is no emphasis on the most recent observations.

Practically, the MA was applied using the *rolling* window function with $r = 7$. The value of r was not optimized as the MA model was only considered as a baseline from which to compare the more complex models, which are described later.

The same average, obtained from a particular set of r observations, was used as the prediction for the 1-day, 7-day and 28-day time horizons. For example, if for a series of 50 observations the average of the first 7 values is equal to a value x, then x is used as the prediction for the 8th observation (for a 1-day horizon), the 14th observation (for the 7-day horizon) and the 35th observation (for the 28-day horizon) in that series. Subsequent predictions followed the same process after a shift in the set of r values by 1. Namely, the oldest observation in the set is dropped and the observation at the next time step is added. Thus, for our example, the second set of 7 observations would be the 2nd to the 8th observations and the average of these values would be the predictions for the 9th observation (for a 1-day horizon), the 15th observation (for the 7-day horizon) and the 36th observation for the (28-day horizon).

3.3. Holt's Winter Method (Triple Exponential Smoothing)

Triple exponential smoothing was used as the second statistical model to improve on the limitations of the MA model (i.e. the inability to deal with trend and seasonality as well as the use of equal weights on past observations). Exponential smoothing techniques involve applying exponentially decreasing weights to past observations to predict future values. Namely, the weights increase exponentially over time so that the most recent observations exert more influence on the calculation of future predictions. The HW method, is an advanced exponential smoothing technique that also considers trend and seasonality. It consists of a forecasting equation and three smoothing equations – one for an overall smoothing S_t, one for trend smoothing b_t and one for seasonal smoothing I_t, with corresponding parameters α, β and γ. Another constant, m, is used to denote the frequency of the seasonality; for example, for quarterly data, $m = 4$. There are also two variations of the HW method, known as the additive and multiplicative method. An additive model is used when the seasonal variations (i.e. the amplitude of the seasonality curve) are constant throughout the series, while the multiplicative method is preferred when the seasonal variations are changing in proportion with the trend of the series. As can be seen from the 'Seasonal' plot in Figure 2, the amplitude of the curve remains constant with an increasing time. As a result, an additive model was used in this work. For reference, the equations for the additive model can be seen in Appendix A.

Practically, the HW method was implemented using the *ExponentialSmoothing* function from the *statsmodels* library. This function fits and optimizes the parameters of the HW model to inputted data based on user-defined criteria: these include the number of seasonal periods ($m = 7$ for our dataset), the type of trend (additive for our dataset) and the type of seasonality (additive for our dataset). The value for the seasonal periods was determined using autocorrelation and partial autocorrelations plots, shown for the Hobbies category in Figures 3a and 3b. From the plots, it can be seen that there is a correlation between observations that are separated by 7 days. The same correlation period was seen for the Foods and Household categories.

The predictions for the three-time horizons were handled in a similar way to the MA model. The difference was that the entire dataset up until one day before the target day (for the 1-day horizon), 7 days before the target day (for the 7-day time horizon) and 28 days before the target day for the (28-day time horizon) was used to make the prediction instead of the fixed number of observations, r, used in the MA model. For example, for a total dataset of 50 observations, the predictions for the 50th observation (target value) would be made by inputting observations 1-49 (for the 1-day time horizon), 1-43 (for the 7-day time horizon) and 1-22 (for the 28-day time horizon) into the HW model. The output, of the *ExponentialSmoothing* function, were the forecasts for the next day, next 7 days and next 28 days for the three time-horizons respectively. As a result, for the 7-day and 28-day horizon, only the 7th and 28th outputted forecast would be used as the prediction.

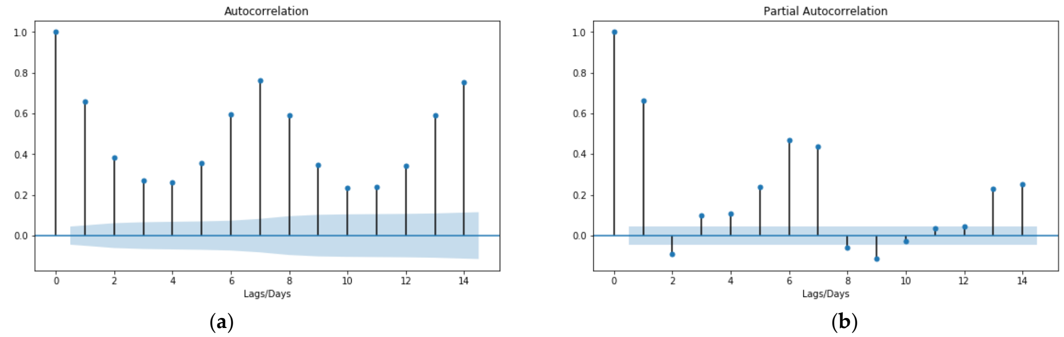


Figure 3. The (**a**) autocorrelation and (**b**) partial autocorrelation plots of the Hobbies sales data

4. Machine Learning Models

Two supervised ML models are considered in this work: Gradient Boosted Trees (GBT) and Random Forests (RF). These are both examples of ensemble methods. An ensemble method is where multiple models, known as base learners, are trained on the dataset and combined in some way. This is often applied for tree-based models, such as regression trees, and there are two main variations known as bagging and boosting. In bagging, the base learners are generated in parallel; the benefit of this is that it exploits the independence between the base learners since the error is dramatically reduced by averaging. An example of this is RF. Boosting, is when the base learners are generated sequentially; this exploits the dependence of the base learners and the overall performance can be boosted by training subsequent learners on the residuals (i.e. the difference between the predicted and true values). An example of this approach is GBT. The base learners were regression trees because the target is continuous. Ensemble methods are used, in this work, because they typically perform better than single regression trees. Also, ensemble methods have been shown to outperform other ML models in past Kaggle competitions. To apply the varying time horizons, the target values, in the dataset, were lagged while maintaining the position of the features. Namely, for the 1-day, 7-day and 28-day time horizons, the target values were lagged by 1, 7 and 28 values respectively. A diagram of the workflow followed for the ML models is shown in Figure 4 and each step is described in more detail in the subsequent sections.

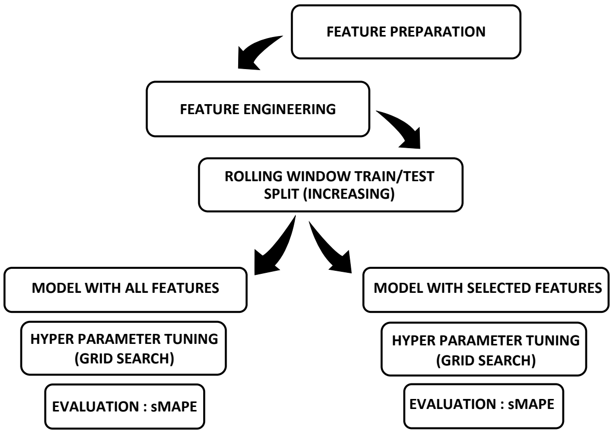

FEATURE PREPARATION
FEATURE ENGINEERING
ROLLING WINDOW TRAIN/TEST SPLIT (INCREASING)
MODEL WITH ALL FEATURES
HYPER PARAMETER TUNING (GRID SEARCH)
EVALUATION : sMAPE
MODEL WITH SELECTED FEATURES
HYPER PARAMETER TUNING (GRID SEARCH)
EVALUATION : sMAPE

Figure 4. Workflow of the ML models

4.1. Feature Preparation

The features in the original dataset did not require extensive preprocessing. Null values were only present for the categorical features pertaining to event name and event type. As a result, these were all changed to the 'None' category to reflect that these were regular days with no special events. Otherwise, no data cleaning was required. The features provided in the original dataset and examples of each are shown in Table 1.

Table 1. Original dataset features

Feature Type	Example
Day of the Week	Saturday, Sunday, etc.
Month	January, February, etc.
Year	2011, 2012, etc.
Event Name	Christmas, Super Bowl, etc.
Event Type	National, Sporting, etc.

Table 2. Examples of the OHE Features

weekday_OHE	month_OHE	year_OHE	event_type_1_OHE
[0, 7, [1], [1]]	[0, 12, [3], [1]]	[0, 6, [4], [1]]	[0, 6, [0, 1], [1, 1]]
[0, 7, [0], [1]]	[0, 12, [3], [1]]	[0, 6, [4], [1]]	[0, 6, [0, 1], [1, 1]]
[0, 7, [2], [1]]	[0, 12, [3], [1]]	[0, 6, [4], [1]]	[0, 6, [0, 1], [1, 1]]
[0, 7, [6], [1]]	[0, 12, [0], [1]]	[0, 6, [4], [1]]	[0, 6, [0, 1], [1, 1]]

As can be seen, a limited number of features were available and they all took the form of categorical data. As a result, One-Hot-Encoding (OHE) was applied to all the features. The transformation for some of the features is shown in Table 2, for the first four observations/days, in sparse vector form.

4.2. Feature Engineering

A number of features were engineered for use in the ML models. These were created by applying day lags to the data and are summarized in Table 3.

Table 3. The generated features and their description

Feature Name (in code)	Description
Pre_d1	The sales quantity one day before the target
Pre_d2	The sales quantity two days before the target
Pre_d3	The sales quantity three days before the target
Pre_d4	The sales quantity four days before the target
Pre_d5	The sales quantity five days before the target
Pre_d6	The sales quantity six days before the target
Pre_d7	The sales quantity seven days before the target
Pre_d10	The sales quantity ten days before the target
Pre_d14	The sales quantity fourteen days before the target
Pre_d28	The sales quantity twenty-eight days before the target
No of events	The number of events on a given day

Lag features are commonly used as the generated features for time-series analysis as adjacent observations often show some correlation. This is especially true for the cyclical data considered in this work. It should be noted that the original dataset was reduced from 1941 observations to 1913 observations as a result of generating these lagged features. Namely, the dataset had to be reduced by a maximum of 28 days to account for the null values created by the lag for the *Pre_d28* feature.

4.3. Model Optimization

Several techniques were used to perform the model optimization. Firstly, a rolling *k*-folds validation was implemented as this is the best validation approach for time-series data. This is because the splits have to retain chronological order and cannot be random. The original dataset was split into four sets for use in a three-step increasing window. An example of how the dataset was split is shown in Figure 5 for a 1-day horizon. It should be noted that the 7-day and 28-day horizon's had slightly different values for their splits as the size of the original dataset was reduced when the prediction window was applied.

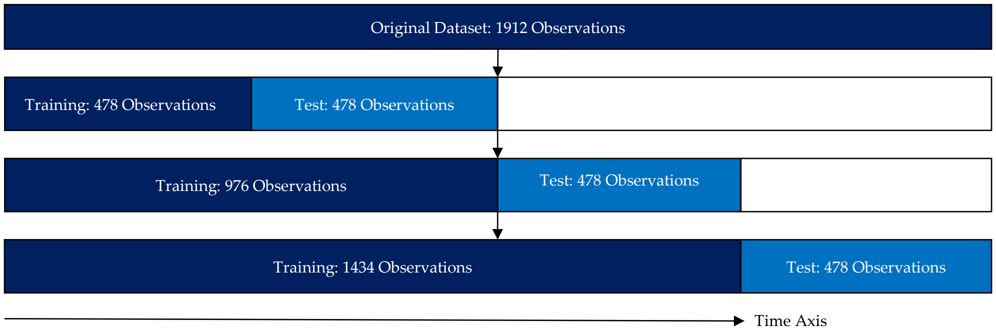

Original Dataset: 1912 Observations
Training: 478 Observations
Test: 478 Observations
Training: 976 Observations
Test: 478 Observations
Training: 1434 Observations
Test: 478 Observations
Time Axis

Figure 5. Example values of the rolling *k*-folds validation used for the day horizon

Secondly, a feature selection/importance study was implemented. This determined which features have the most significant effect on the performance of the model. Practically, the importance score was estimated by initially training the model with all the features. The features were then ranked, and the top 10 features were subsequently used in the optimized model.

Lastly, a set of discrete values for the hyperparameters were defined for each model to reduce computational complexity. These are shown in Table 4. A grid search was used to determine the performance of every possible combination of the hyperparameters on the training and test set. These were then ranked, by evaluating against the sMAPE error, and then the optimal combination (i.e. the one that gave the lowest error) was chosen for each model.

Table 4. The hyperparameters of the two ML models

Model	Hyperparameter Name (in code)	Hyperparameter Description	Hyperparameter Options
GBT	*maxdepth*	The maximum depth of each tree	2, 5
	maxIter	The maximum number of iterations	10, 20
Random Forests	*maxDepth*	The maximum depth of each tree	5, 10, 15
	numTrees	The maximum number of trees used	5, 10, 30, 50

5. Model Summary

The important details, pertaining to all the models that were implemented, are summarized in Table 5.

Table 5. Summary of all model details

Model	Used Data	Feature Transformation	Train/Test Split	Overfitting?	Model Optimization
Moving Average	Only daily sales data	N/A	Multiple	N/A	N/A
Holt's Winter	Only daily sales data	N/A	Multiple	N/A	N/A
GBT	Multiple	One-Hot Encoding	Rolling *k*-fold cross-validation	No	Feature Selection, Tuning using Grid Search
RF	Multiple	One-Hot Encoding	Rolling *k*-fold cross-validation	No	Feature Selection, Tuning using Grid Search

6. Results and Discussion

As mentioned, the two ML models, GBT and RF, were tested with and without feature selection/importance. The results of this can be seen in Table 6. For both models and all cases, the use of feature selection has resulted in the reduction of the sMAPE error.

Table 6. The sMAPE score achieved by ML model's with and without feature selection (all values are a %)

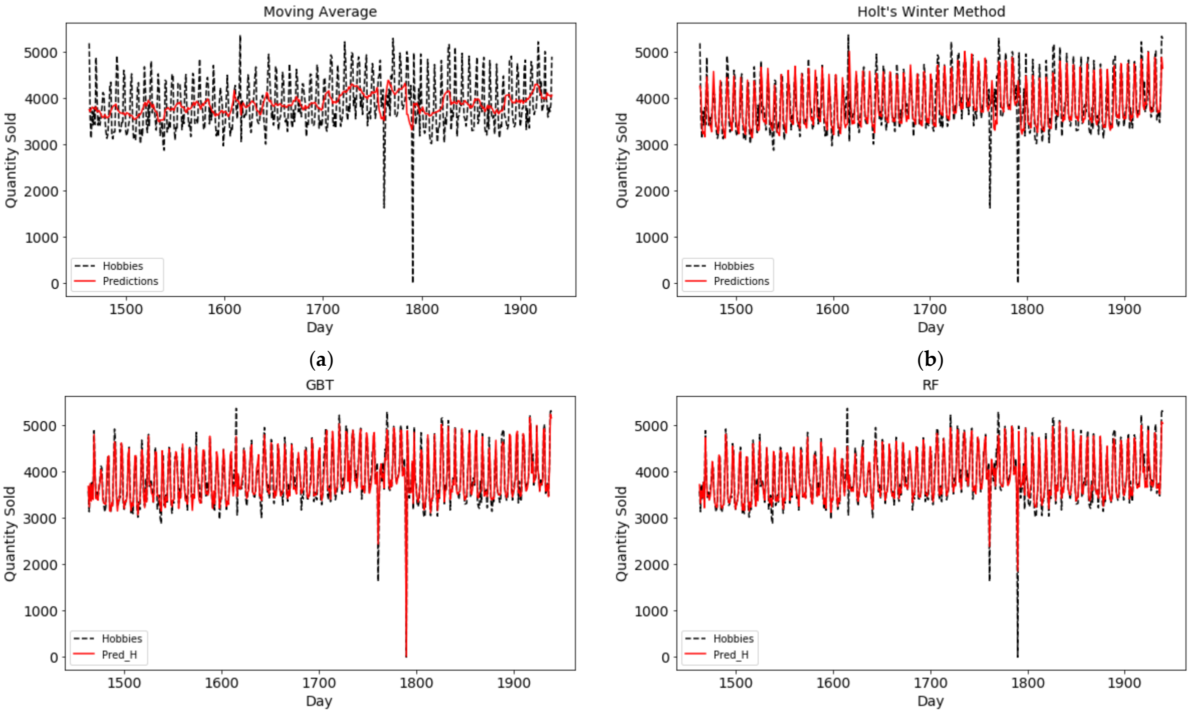
Category	Time Horizon	GBT (all features)	GBT (with feature selection)	RF (all features)	RF (with feature selection)
Hobbies	1 Day	9.00	***3.34***	7.77	***2.92***
	7 Days	9.62	***3.17***	8.29	***2.96***
	28 Days	9.80	***3.50***	8.34	***2.99***
Foods	1 Day	8.50	***3.52***	7.21	***3.16***
	7 Days	9.66	***3.62***	8.34	***3.38***
	28 Days	8.93	***3.22***	7.86	***3.20***
HouseHold	1 Day	9.56	***3.43***	8.42	***2.94***
	7 Days	9.63	***3.40***	8.57	***3.02***
	28 Days	10.24	***3.32***	9.45	***2.88***

The improved values of the ML models, with feature selection, are compared to the performance of the statistical models, on all three categories and time horizon's in Table 7. The ML models significantly outperform the stochastic methods with RF achieving the best performance in all cases. The lowest error, with a value of 2.88% is achieved by the RF model on the Household category for the 28-day time horizon. The largest error, with a value of 15.87%, is achieved by the MA model on the Foods category for the 7-day time horizon. In addition, it is generally expected that the error should increase with an increasing time horizon. Namely, forecasting further into the future is expected to be less accurate. However, the error does not show an obvious pattern with an increasing time horizon; in fact, in most cases it remains relatively constant. A possible explanation for this is that the data follows weekly cycles as seen in the autocorrelation plot in Figure 3. Therefore, the prediction value for 7-days or 28-days into the near future might not change too much and consequently the error would be similar for the varying horizons. However, as the data showed an increasing trend, if longer time-horizons were to be considered (i.e. two or four months) then it is expected that a more obvious decrease in performance, with an increasing horizon, would be seen.

Table 7. The sMAPE score for all model's with forecasts on each category and time-horizon (all values are a %)

Category	Time Horizon	MA	Holt's Winter	GBT	RF
Hobbies	1 Day	11.55	6.26	3.34	***2.92***
	7 Days	11.97	6.56	3.17	***2.96***
	28 Days	12.21	7.15	3.50	***2.99***
Foods	1 Day	12.81	6.85	3.52	***3.16***
	7 Days	15.87	10.89	3.62	***3.38***
	28 Days	13.37	7.37	3.22	***3.20***
HouseHold	1 Day	14.50	6.47	3.43	***2.94***
	7 Days	15.25	7.16	3.40	***3.02***
	28 Days	15.35	7.84	3.32	***2.88***

Example plots of the predicted values versus the target values are shown in Figure 6 for all four models on the Hobbies category and for a 1-day time horizon. Similar plots for the Foods and Household category can be seen in Figures 7 and 8 in Appendix B. It can be seen that the fit of the prediction curve progressively gets closer to the targets' curve as you move from Figures 6a – 6d. This reflects the improvement in performance seen in the error values, when moving from the statistical models to the ML models, in Table 7. The prediction curves for both ML models closely resembles the target curves. Interestingly, GBT was able to more accurately model the sudden dip in sales on day 1791 than the RF model, despite the fact that RF performs better overall. A reason for this might be that RF averages the value of predictions for multiple trees (bagging) whereas GBT sequentially improves on the performance of the previous tree (boosting) which would make the latter more accurate in modelling sudden fluctuations. The dip in sales was because stores were closed on that day for Christmas.



(**a**) (**b**)

(**c**) (**d**)

Figure 6. Plots showing the predictions vs the target values for the (**a**) MA model, (**b**) Holt's Winter model, (**c**) GBT model and (**d**) RF model on the Hobbies category for a 1-day horizon.

The ML methods outperformed the statistical models because they consider a larger set of data, by including features, to make predictions. In particular, including whether or not there is an event on a given day is an important feature as there is a strong correlation with the sales data. For example, sales of foods on a Super Bowl day were significantly higher than regular days and there were no sales on Christmas day because stores were closed. Furthermore, neither statistical model was able to model the sharp fluctuations, as can be seen in Figure 6a and 6b, and this would have contributed to the higher error values. RF performed slightly better than GBT. This might be because RF is better able to reduce the high variance in the data and is less likely to overfit to the training data, by averaging the predictions of the ensemble, than GBT. However, it should be emphasized that the performance of the two models was not significantly different.

7. Conclusions

In this work, two statistical models and two ML models were used to forecast sales of products in three distinct categories: Hobbies, Foods, and Household. Stationarity and seasonality tests were used to justify the choice of statistical models: moving average and Holt's Winter method. Ensemble methods were used in the ML models as these produce more accurate results than a single regression tree by averaging or improving the predictions of multiple trees. The features were prepared using one-hot-encoding and additional lag features were generated. The models were optimized by performing a feature importance study and a rolling k-folds validation; grid search was used for hyperparameter tuning. The performance of the models was evaluated on three time-horizons (1-day, 7-days and 28-days) using the sMAPE metric. The results showed that the ML models outperformed the statistical models for all three categories and time-horizons. The RF approach performed the best overall and achieved the lowest overall error value of 2.88%. The ML models could be improved by generating more features; these could include window features which are averages of past observations over a defined window. A *hv* cross-validation could also be applied where a gap is added between the training and test sets, in the rolling window *k*-folds validation, to mitigate the effects of the dependence of adjacent observations. Future work could also include the use of a combination of statistical and ML methods which have been empirically shown to outperform purely statistical or purely ML models, as mentioned in the introduction [1].

References

[1] A. R. S. Parmezan, V. M. A. Souza, and G. E. A. P. A. Batista, "Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model," *Inf. Sci.*, vol. 484, pp. 302–337, May 2019, doi: 10.1016/j.ins.2019.01.076.

[2] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M4 Competition: 100,000 time series and 61 forecasting methods," *Int. J. Forecast.*, vol. 36, no. 1, pp. 54–74, 2020.

[3] "M5 Forecasting - Accuracy." https://kaggle.com/c/m5-forecasting-accuracy (accessed Aug. 27, 2020).

Appendices

Appendix A

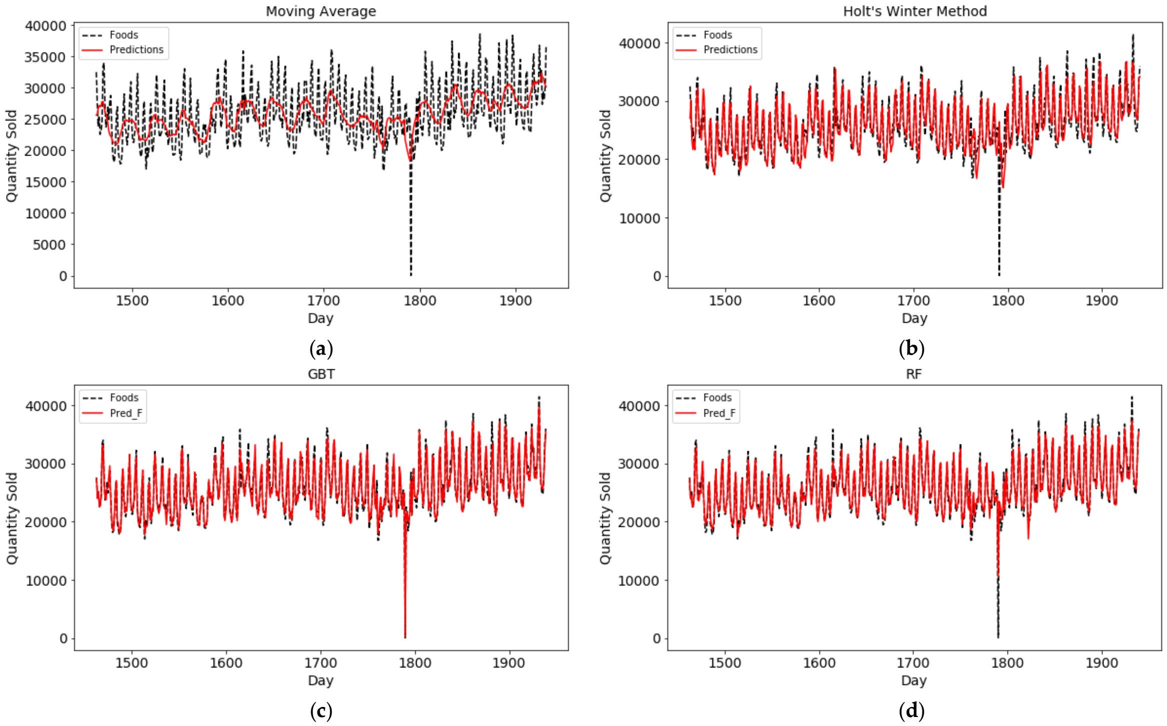
The equations for the Holt's Winter method with an additive model are

$$\begin{aligned} S_t &= \alpha \frac{y_t}{I_{t-L}} + (1-\alpha)(S_{t-1} + b_{t-1}) \\ b_t &= \beta(S_t - S_{t-1}) + (1-\beta)b_{t-1} \\ I_t &= \gamma \frac{y_t}{S_t} + (1-\gamma)I_{t-L} \\ F_{t+m} &= (S_t + mb_t)I_{t-L+m} \end{aligned} \tag{3}$$

where y is an observation, S is the smoothed observation, b is the trend factor, I is the seasonal index, F is the forecast at m periods ahead and t is the time index [1].

Appendix B

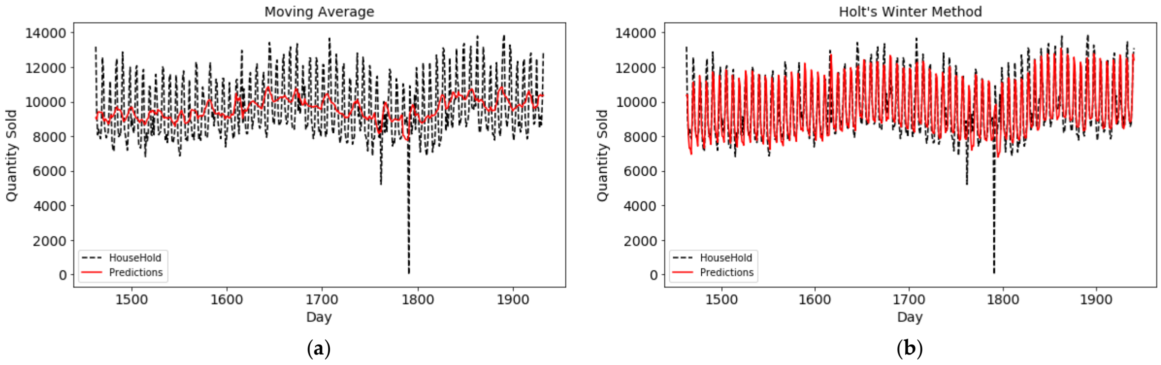
Plots showing the performance of the models on the Foods category for the 1-day prediction horizon are shown in Figure 7.



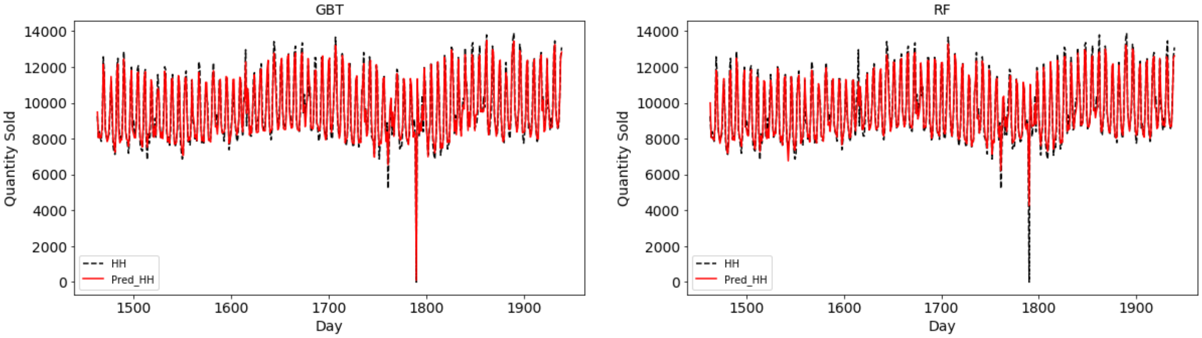
(**a**) (**b**) (**c**) (**d**)

Figure 7. Plots showing the predictions vs the target values for the (**a**) MA model, (**b**) Holt's Winter model, (**c**) GBT model and (**d**) RF model on the Foods category for a 1-day horizon.

Similarly, the plots showing the performance of the models on the HouseHold category for the 1-day prediction horizon are shown in Figure 8.



(**a**) (**b**)



(**c**) (**d**)

Figure 8. Plots showing the predictions vs the target values for the (**a**) MA model, (**b**) Holt's Winter model, (**c**) GBT model and (**d**) RF model on the Household category for a 1-day horizon.