NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SC4000: Machine Learning
Project Report

Project: Store Item Demand Forecasting Challenge

Team 48

Name	Matriculation Number	Contribution
Lee Zong Yu	U2040660C	Prophet Model
Chen Gordon Tian Xiao	U2140820A	XGBoost & Extrapolation Model
Min Khant Htoo	U2140545E	ARIMA Model
Bryan Wu Jiahe	U2122993K	Neural Network Model
Chua Ming Ru	U2140945D	Writing & Compilation

1 Project Overview

1.1 Task and Dataset

This project focuses on predicting future sales over three months for a retail dataset spanning five years. The dataset includes essential details such as date, store information, item specifics, and daily sales.

Our task is to predict the sales of a particular item on a specific date in a given store, with the assumption that each store operates daily and public holidays are not considered. The project prioritizes model accuracy on test data as the primary evaluation metric. Through various rigorous model evaluations and experimentations, we delivered a concise and effective predictive model capable of anticipating sales trends within the specified timeframe.

1.2 Evaluation Criteria

$$\text{SMAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{(A_t + F_t)/2}$$

Figure 1: SMAPE formula

Submissions in this project will be assessed based on the SMAPE metric, a widely used accuracy measure in forecasting. SMAPE compares forecasted values (Ft) with actual values (At) at time t. If both Ft and At are zero, the summand is defined as zero. A lower SMAPE value is indicative of better accuracy in predictions. The goal is to minimize SMAPE, emphasizing the precision of forecasts relative to actual values.

1.3 Model Performances

Our best model achieved a SMAPE score of 13.83774 in the public leaderboard for this competition (figure 2), which places us in **fourth position among 461 submissions.** The top-performing SMAPE value in the public leaderboard is 13.83614. In this project, we have put forth the following models. Detailed discussions for each model can be found in the subsequent sections.

1. XGBoost
2. ARIMA
3. Prophet
4. Neural Network
5. Extrapolation (Best Model)

Figure 2: Screenshot of the best model performance

Prior to model training, we conducted an exploratory data analysis to identify potential correlations among the provided inputs.

2 Data Exploration

2.1 Total sales by store and item

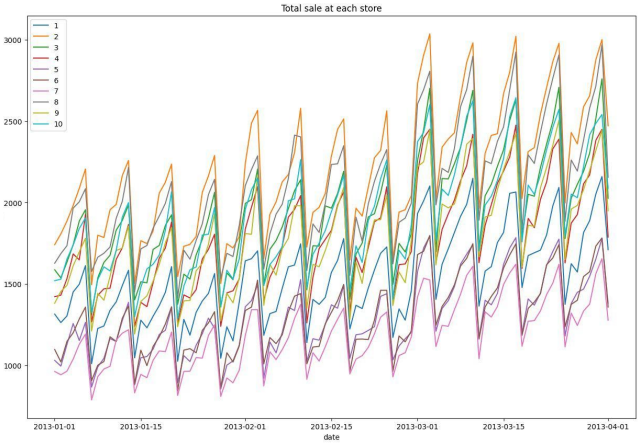

Figure 3: Distribution of total sales of items in different stores across 3 months

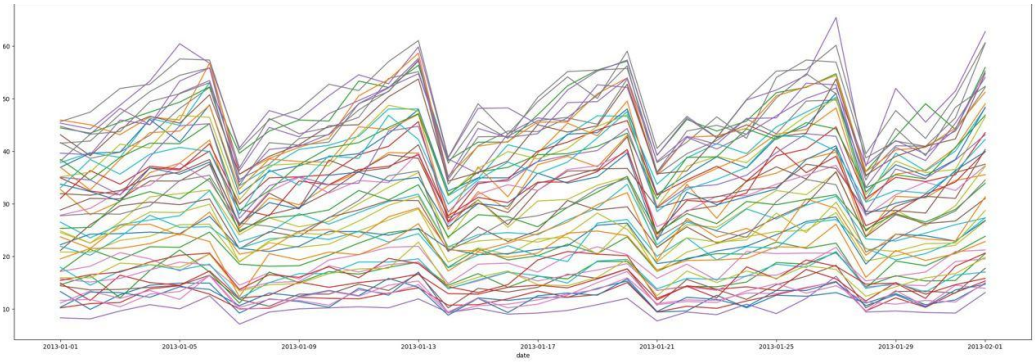

Figure 4: Distribution of total sales of items per item across a month

We can clearly observe a consistent cycle of peaks and valleys indicating the presence of regular and predictable patterns in the data. From figure 3, we can make a few key observations. All stores (figure 3) and items (figure 4) exhibit a shared cyclical and patterned behavior in their sales distributions. While the overall shapes of the plots remain consistent, individual stores and individual items showcase different sales volumes within this common cyclical pattern (High Correlation and High Variance). This proves that **item and store are both significant forecast predictors**.

The aggregate store sales display a recurring weekly cycle, indicating a possible significance of the ‘day of the week’ as a predictor. While the amplitude of store sales remains relatively constant throughout each month, variations across different months

imply that the month of the year is another possible predictor**.** These observations are to be analyzed through statistical analysis.

2.2 Time Series Model

The simplest models to explain the phenomenon in section 2.1 are the additive and multiplicative models as defined below:

- Additive: $f(t) = trend(t) + \sum seasonal(t) + error$
- Multiplicative: $f(t) = trend(t) \times \prod seasonal(t) \times error$

$trend(t)$ refers to a general long-term movement of the time series while $seasonal(t)$ refers to patterns that repeat at regular intervals. We assume that $error$ follows a normal distribution. Next, we will explore the trend and seasonality of the data.

2.2.1 Yearly Trend

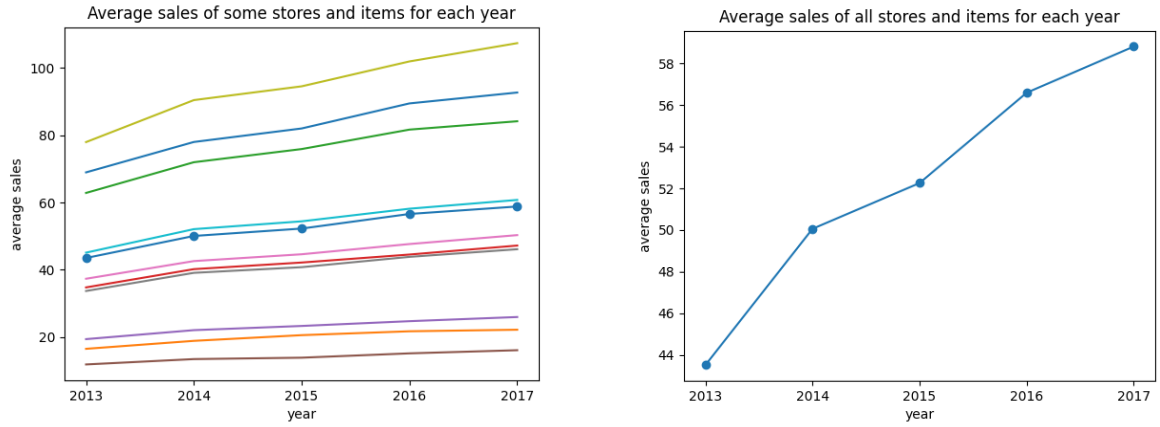

Figure 5: Annual Trend per store per item

In Figure 5, an increasing yearly trend is observed for average sales across all stores and items. To show that the average trend is an aggregation of consistent store-item yearly trends, we calculated the pairwise correlation between the yearly average sales of each store-item pair and yearly average sales across all stores and items.

	Minimum	Maximum
Correlation Coefficient	0.98109	0.99999

Table 1: Correlation results

From Table 1, we observe a high correlation coefficient, which implies that the yearly trend of average sales for each store-item pair is highly similar. **This proves that the 'year' is a significant predictor.**

2.2.2 Yearly Seasonality

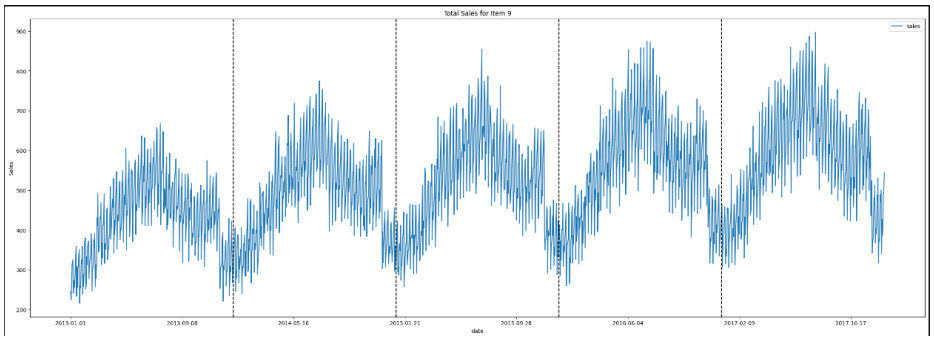

Figure 6: Yearly Seasonality of Total Sales of Item 9

Figure 6 reveals a distinct yearly cycle, with sales peaking around June and declining until January. This observation suggests that the dataset exhibits a consistent yearly seasonality with a one-year period.

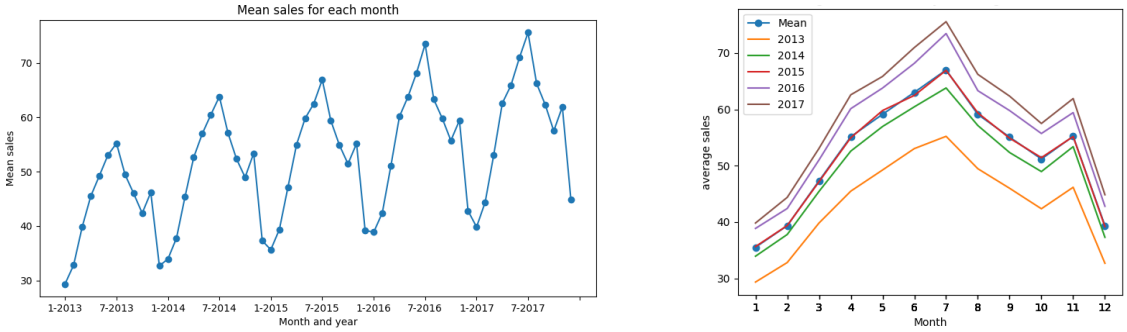

Figure 7: Mean sales per month from 2013 to 2017 (Left) vs mean sales per month of each year from 2013 to 2017 (Right)

In Figure 7, we calculated the behaviour of mean sales across the months for each year (right plot). To show that this trend is an aggregation of consistent yearly seasonalities, we calculated the pairwise correlation coefficient between each year's mean monthly sales and the overall mean monthly sales.

	Minimum	Maximum
Correlation Coefficient	0.91261	0.99902

Table 2: Correlation results

From Table 2, we observe a high correlation coefficient, which implies that each year's cycle across the months is highly similar. **This proves that the 'month' is a significant predictor.**

2.2.3 Monthly seasonality

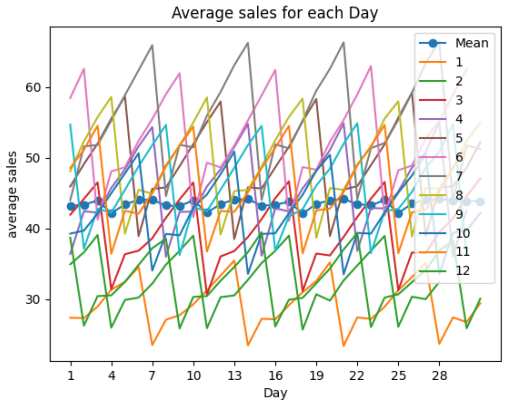

Figure 8: Average sales for each day of the month

From Figure 8, we observe that there is no clear seasonality within each month. Hence, it is likely that the **day of the month is not a significant predictor**.

2.2.4 Weekly seasonality

Figure 3 highlights a possible presence of weekly seasonalities, which appear to be consistent across each month. To explore this pattern further, we plotted the average sales for each day of the week, from Monday (Day 0) to Sunday (Day 6), over the year 2013.

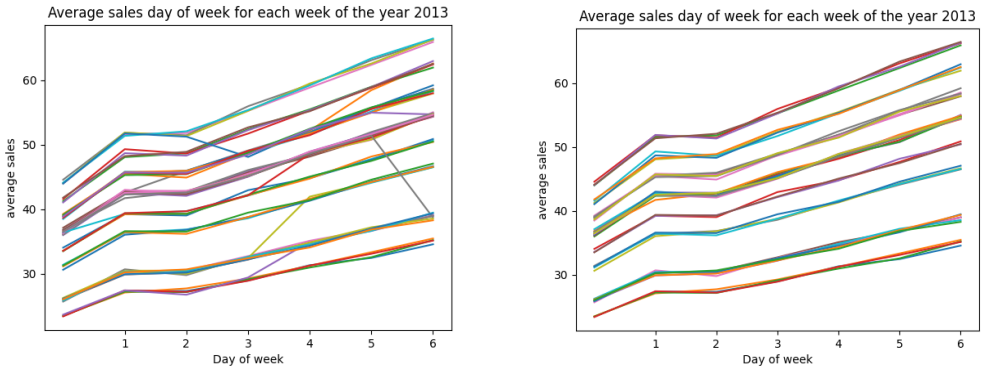

Figure 9a and 9b: Before and After removal of weeks that span across 2 months

From Figure 9a we can observe a clear seasonality trend with minor discrepancies. However, this is addressed by removing weeks that span across two separate months (figure 9b). Hence this discrepancy is likely due to the effect of yearly seasonalities.

	Minimum	Maximum
Correlation Coefficient	0.99742	0.99992

Table 3: Correlation results

From Table 3, we observe a high correlation coefficient, which implies that each week's cycle across the days is highly similar. **This proves that the 'day of the week' is a significant predictor.**

2.2.5 Additive vs Multiplicative Relationship Between Mean Sales

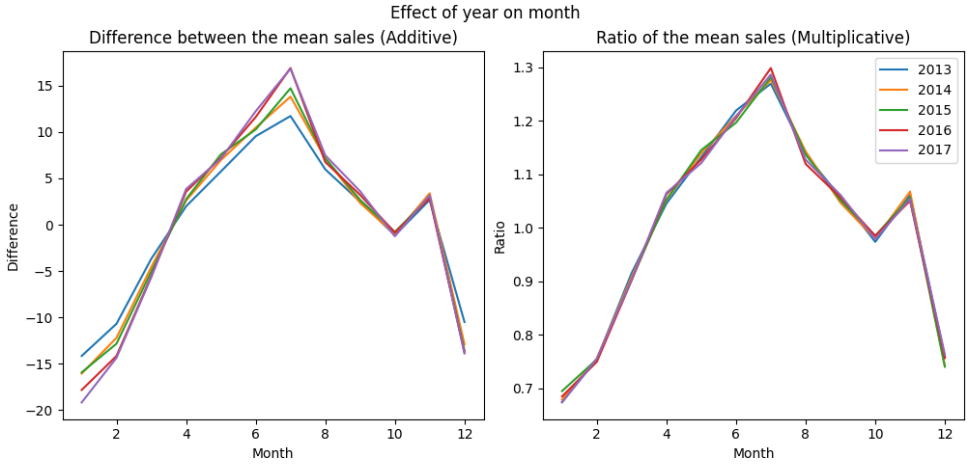

Figure 10: The Difference and the ratio between mean sales of different years

To determine whether the trends and seasonality are additive or multiplicative, we plotted 2 different graphs - the difference between the mean sales and the ratio of mean sales in order to represent additive and multiplicative seasonality respectively.

Figure 10 shows the graphs obtained for yearly seasonality. We can see that for the right graph, the lines are more compact than the left graph, suggesting the variance is smaller, as evidenced by the table 4 below.

Repeating this for the other trends and seasonalities observed, we can determine that the yearly trend, weekly and yearly seasonalities are all multiplicative.

Variances of	Minimum	Maximum
The Difference	0.038283	4.7505
The Ratio	6.2261e-06	0.00012221

Table 4

2.3 Relation between features

In this section, we will study the relationship between store, item and time-related features - day of week, month and year. Based on the observations in section 2.2.4, we experiment with the effects of cleaning the data on its correlation as well. All the following tables show the correlation on the cleaned data with the value in the brackets representing the difference from the original data.

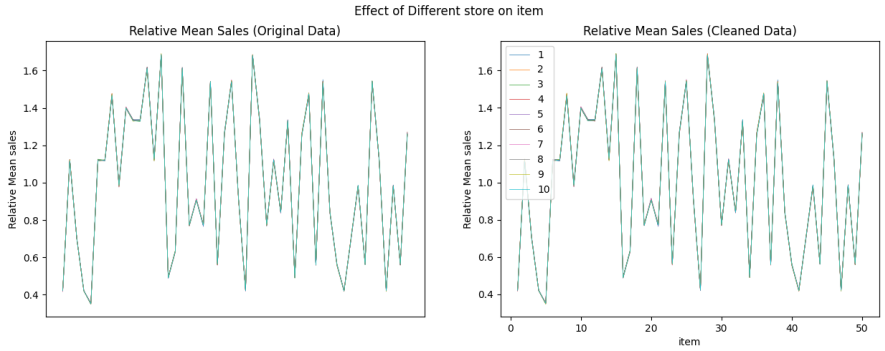

Figure 11: Relative Mean sales of different Stores with Items before and after cleaning data

Feature	Minimum	Maximum
Store	0.99994 (-6.6691e-06)	0.99998 (-5.3806e-06)

Table 5: Correlation of items with stores

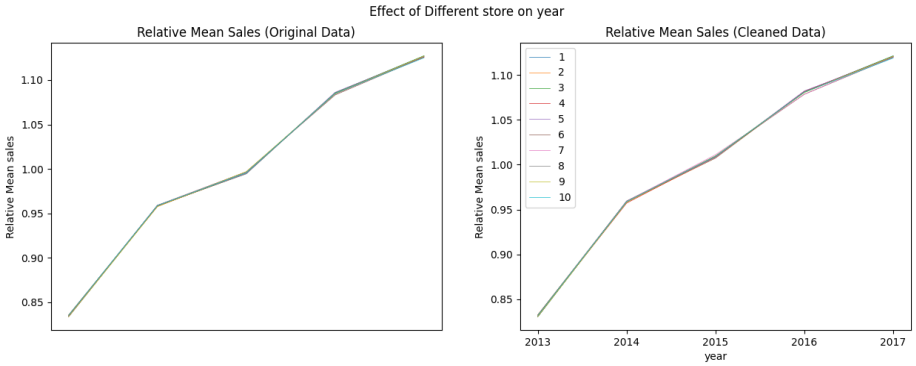

Figure 12: Relative Mean sales of different Stores with year before and after cleaning data

Feature	Minimum	Maximum
Year	0.9999 (-7.6406e-05)	1.0 (+3.6384e-07)
Month	0.99993 (-7.8931e-06)	0.99999 (-3.3981e-06)
Day of Week	0.99991 (-2.8947e-05)	0.99999 (-6.1542e-06)

Table 6: Correlation of store with time-related features

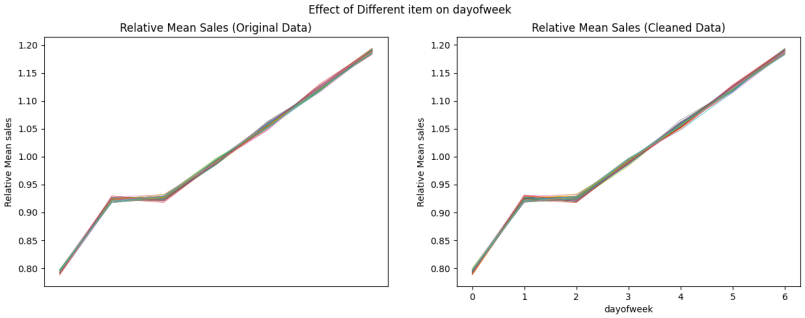

Figure 13: Relative Mean sales of different items with day of week before and after cleaning data

Feature	Minimum	Maximum
Year	0.99916 (-1.7161e-05)	0.99999 (-5.3291e-06)
Month	0.99926 (-2.8586e-05)	0.99994 (-1.6156e-05)
Day of Week	0.99904 (-0.00042816)	0.99997 (-1.1849e-06)

Table 7: Correlation of item with time-related features

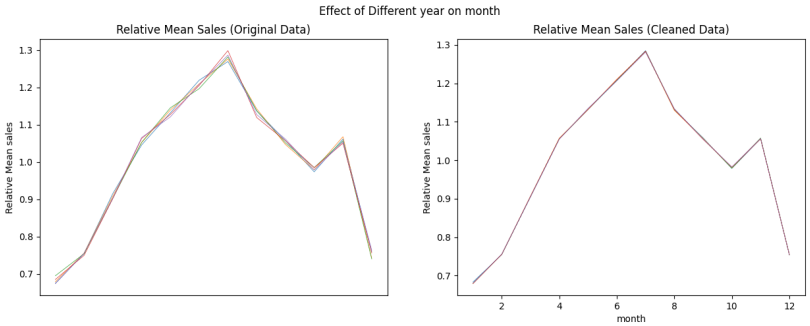

Figure 14: Relative Mean sales of different year with month before and after cleaning data

Feature	Minimum	Maximum
Month	0.99997 (+0.00087268)	0.99999 (+0.00060877)
Day of Week	0.999748 (+0.00027751)	0.9999645 (+0.00012487)

Table 8: Correlation of year with time-related features

Observation 1: Store, Item, Day of Week, Month and Year are highly correlated. From the sections above, all of the features correlate well with each other.

Observation 2: Store has a better relation with time features than item. We can see Figure 12 and Figure 13 in this section that there is a slight fluctuation in the graph.

Observation 3: Time features (Year, Month, Day of Week) are better correlated under the cleaned dataset while store and item are not necessarily.
Based on the brackets from Table 5 to Table 8, the correlation improves when comparing the Month and Day of the week with the year.

3 Experimental Framework

3.1 Dealing with seasonality

For many machine learning models, handling seasonality in a non-stationary dataset can be challenging. To address this, we need to preprocess the inputs via feature engineering to eliminate the inherent seasonality, making the dataset stationary. This not only simplifies the dataset's complexity but also enhances the models' ability to learn, preventing overfitting.

3.2 Postprocessing

Predicted sales were rounded to the nearest integer. This is because the actual forecast is an integer and rounding would likely reduce the error. For example, by rounding the forecasted sales from 4.9 to 5 when the ground truth is 5, the error is likely reduced. To validate our hypothesis, we executed our model to predict the training dataset and tallied the number of residuals (forecast - actual) with a residual fraction greater than or equal to 0.5. The results from Table 9 below affirm the accuracy of our hypothesis. Moreover, we observed a marginal increase of 0.02 in the SMAPE score.

	SMAPE		Residual Fractional	Number of forecasts
Before Rounding	12.57		< 0.5	470878
After Rounding	12.55		>= 0.5	442122

Table 9: SMAPE score on training data before and after rounding

4 Regression Modelling (Best Model)

In section 2.2.5, we demonstrated that the relationship between the seasonality of different weeks, months, and years is multiplicative. Furthermore, in section 2.2, we established a high level of correlation among the seasonality of each week, month, and year. Consequently, in our proposed model, we aimed to extrapolate future sales from the training data using the following formula:

$$sales(x) = f(x) \times baseSale(s, i)$$

$$f(x) = Trend(Year) \times Seasonality(Month) \times Seasonality(dayofweek)$$

$$baseSale(s,i) = the\ mean\ sale\ for\ store\ s\ and\ item\ i$$
$$Seasonality(Month) = AvgSales(Month)/Grand\ Average$$
$$Seasonality(dayofweek) = AvgSales(dayofweek)/Grand\ Average$$
$$Trend(Year) = linear\ regression\ of\ training\ data$$

$$Grand\ Average = Average\ sale\ across\ all\ items\ across\ all\ stores$$

Here, we calculate the base sale by the mean sale of an item at a particular store. The base sale is then multiplied by a multiplicative factor f(x) computed from the average Sales per day of week, month, and year. Trend(Year) then models the training data on a linear function. From this simple model, we obtained a public score of **14.22319.**

4.1 Improvement in regression technique

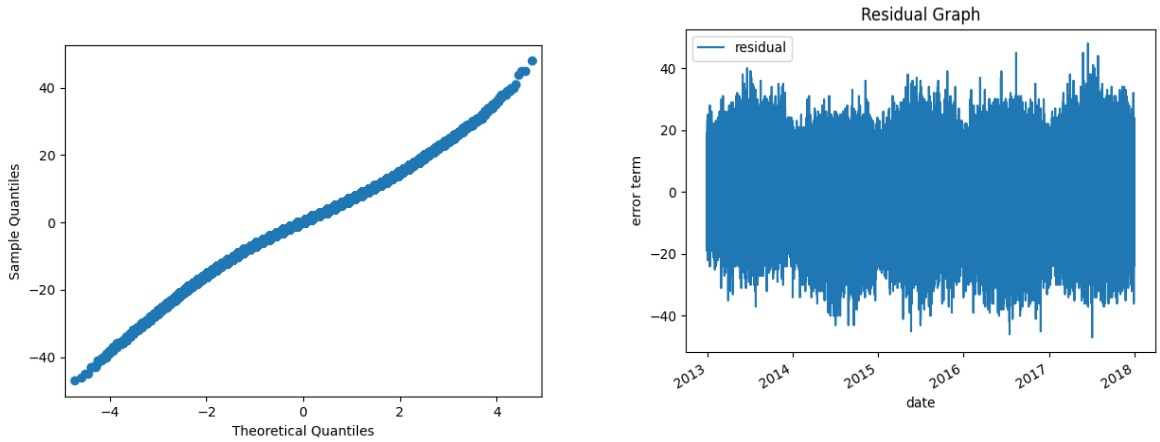

Figure 15: qqplot and residual plot of extrapolation model

From Figure 15, we observe that the error terms follow a normal distribution with constant variance. This shows that regression modelling is a valid approach to this question.

year	2013	2014	2015	2016	2017
Mean of Residual	1.818782	-1.183632	0.095439	-0.747214	0.602137

Table 10: Analysis of the mean of the residual graph of each year

The fact that the means are not centred at 0 suggests that the trend with respect to the year may not be linear. Hence, we experimented with various other functions.

Degree of Polynomial	Relative Sales in 2018	Public Score
1 (linear)	1.2043	14.22319
2 (quadratic)	1.1606	**13.85972 (Second Best)**
3 (cubic)	1.1882	14.01512
1* (logarithm)	1.1574	**13.85602 (Best)**

* $trend(year) = log_e(year - 2012) * m + c$ *where m and c are gradient and intercept of the graph*

Table 11: Relative Sales and public score of each experiment's function

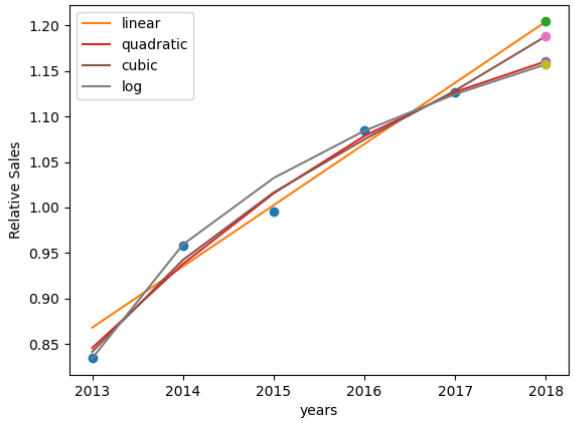

Figure 16: The plot of the extrapolation of the yearly trend function

year	2013	2014	2015	2016	2017
Mean of Residual	0.646865	-1.1042423	0.0769111	-0.283055	0.065874

Table 12: Analysis of mean residual for Quadratic function

year	2013	2014	2015	2016	2017
Mean of Residual	0.057901	0.076456	1.640253	0.011912	-0.060396

Table 13: Analysis of mean residual for Logarithmic function

4.2 Improvement in the calculation of Base Sale

Based on our initial observations, we found that the sale of an item can fluctuate (figure 13) with time series data, hence representing Base Sale as the mean of all its stores would be an inaccurate representation of a particular item. To reduce the variance, we calculated the mean sale of an item on a particular day of the week.

$$sales(x) = baseSale(dayofweek, item) \times f(x)$$
$$f(x) = storeFactor(store) \times seasonality(month) \times trend(year)$$

$$storeFactor(store) = AvgSales(Store)/GrandAverage$$

	Quadratic	Logarithm
Public Score	**13.84298**	13.84482

Table 14: Public Score obtained by applying new equation

4.3 Minor Improvement with slight variation in each week of a month

Motivation:
According to the model equation, within the same month, the forecasted sales by the model would remain the same for every week. Although we observed that there is no clear monthly seasonality in section 2.2, there is still a subtle fluctuation that could potentially contribute to making each week within the month slightly different.

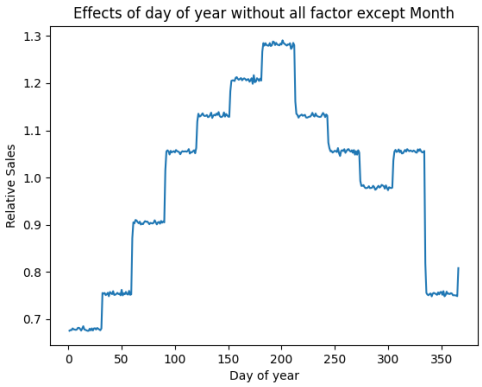

Figure 17: Relative sales of each day of the year after removing weekly seasonality

Experiment & Results:
We modelled the relative sales in Figure 17 for each day of the month and multiplied it by the model equation. Concerned about potential overfitting, we calculate the mean of the relative sales within a 7-day window (limited to the same month). This approach ensures the additional multiplicative terms consider a one-week interval, preventing excessive influence on both weekly and yearly seasonality.

Min	Average	Max
0.9943194541591215	0.9990537217926146	1.0014900255836867

Table 15: The range of the minor factors mentioned.

With this, we obtained a public score of **13.83774,** which is ranked 4th in the public leaderboard with the following mathematical model.

The equation of our final model is as such:

$$sales(x) = baseSale(dayofweek, item) \times f(x)$$

$$f(x) = storeFactor(store) \times seasonality(month) \times trend(year) \times term(month,\ day)$$

where,

$$storeFactor(store) = AvgSales(Store)/GrandAverage$$

$$seasonality(month) = AvgSales(month)/GrandAverage$$

$$trend(year) = quadratic\ regression\ of\ means\ sales\ by\ years$$

$$term(month,\ day) = the\ terms\ mentioned\ at\ 4.3$$

$$Grand\ Average = Average\ sale\ across\ all\ items\ across\ all\ stores$$

To calculate $term(month,\ day)$,

$temp(date) = ActualSales(day)\ /\ AvgSales(dayofweek)$, remove weekly seasonality

$temp2(month,\ day) = temp(day)\ /\ Avg(temp(month))$, remove yearly seasonality

$$term(month,\ day) = 1/7 \times \sum_{i\ =\ day\ -\ 3}^{day\ +\ 3} temp2(month,\ i)\ .$$

For temp2(month,i), the summation is cutoff such that the value is within the same month.

5 Other Models

In this section, we explore different machine learning models that deal with time series data and compare their effectiveness and performance.

5.1 Model A: ARIMA

ARIMA is a popular model used for time series forecasting [1] and is a combination of the autoregressive ($AR(p)$) and moving average models ($MA(p)$). It is commonly referred to as $ARIMA(p, d, q)$, where p is the autoregressive parameter, q is the moving average parameter and d is the degree of differencing. The differencing, d, is done to eliminate linear trends from the time series, however, ARIMA, by itself, does not account for seasonality in the time series, which is why SARIMA was created. SARIMA extends ARIMA by adding on a seasonal model with similar terms P, D and Q for the seasons and is referred to as $SARIMA(p,\ d,\ q,\ P,\ D,\ Q)$.

It should be noted that there are some issues with ARIMA/SARIMA:

1) ARIMA is a single response/predictor model, meaning that we would have to generate 10x50 store-item models.
2) ARIMA is intended for short-term prediction and is unsuitable for the competition's forecast interval (90 days).

3) SARIMA model generation runs slow for large seasonal lengths (365 days), which, when combined with point 1, makes this approach extremely slow. Moreover, the model generation often results in excessive memory usage.
4) SARIMA was designed to handle only a single seasonality. The time series that we are predicting has multiple seasonality.
5) ARIMA does not handle the exogenous factors, such as the price of other stores, which would mean we are unable to model the interactions between stores, and between items.

As such, SARIMA is not a suitable model for the competition requirements. We tried a workaround of using a Fourier series to model and extrapolate multiple seasonality quickly, but this also failed as the time-series is *heteroscedastic* i.e. the time-series (and its seasons) does not have constant variance. The figure 18 shows our ARIMA, with Fourier Approximation, trying to predict the test interval (test sample). It is clear that the Fourier Approximation understands that there is a seasonality, but is unable to correctly discern it, due to the aforementioned heteroscedastic property of the time-series.

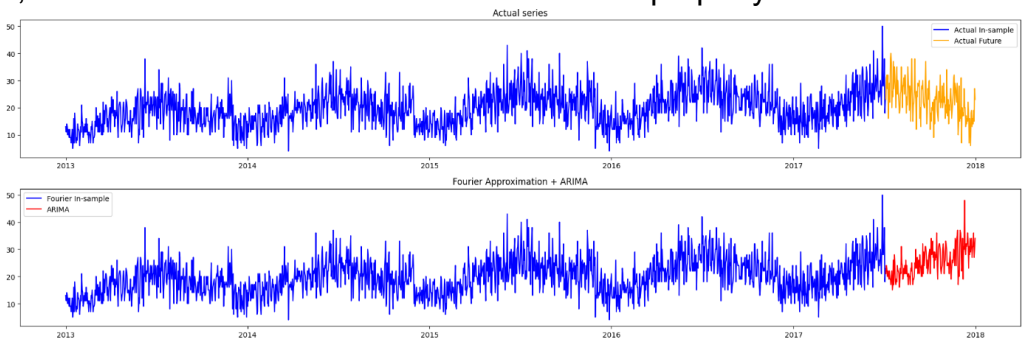

Figure 18: ARIMA prediction with Fourier Approximation

Due to the points mentioned above, we do not believe that ARIMA/SARIMA is a suitable model for this competition.

5.2 Model B: Prophet

Prophet is a forecasting model developed by Meta [2]. This model makes use of the observation in 2.2 by finding the trend function and seasonality function.

5.2.1 How each component function is calculated

Prophet determines the trend function by finding the linear piecewise function through automatic changepoint selection which differs from ARIMA. It finds the changepoint by observing the possible changing trend with Laplace distribution with variance known as changepoint prior scale. This changepoint prior to scale can be hypertuned.

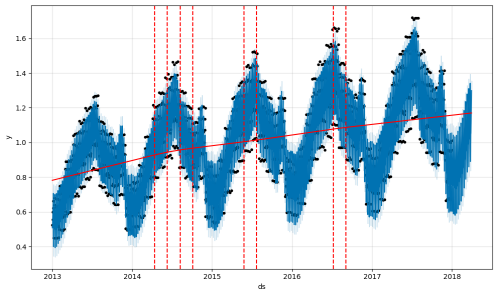

Figure 19: Trend function of prophet models with changepoints (Red dotted Line shows the changepoint)

A Fourier series is used, which is a common way to deal with each seasonality.

$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right)$, P refers to the period which is 365.25 for years, 7 for day of the week etc. Therefore, the seasonality is found by constructing the vector of examples and multiplying it with a smoothing factor known as seasonality prior scale which can be hypertuned.

5.2.2 Experiments & Results

We first fit the data into 500 separate models corresponding to each (store, item) pair. Secondly, as the store_id and item_id correlate highly with time-related features, we can fit one multiplicative prophet model and multiply them with the base sale calculated from the mean. Thirdly, we hypertuned the parameter changepoint prior scale and seasonality prior scale mentioned in 4.3.1 by using cross-validation of SMAPE error rate. Table 16 below is the models' performance.

Model	Public Score	Private Score
500 models for each store and item	**15.11501**	13.52896
1 multiplicative model for all store and item	**14.94652**	13.44298
Hypertuning Parameter	**14.97298**	13.34434

Table 16: Results of Prophet model

The results are not as good as expected even after hypertuning. The possible reason may be that Prophet plots seasonality as a curve (Fourier series) which is not what is observed in the dataset, being a multiplicative seasonality.

5.3 Model C: Neural Network

Given the relative simplicity of training data, where seasonality trends can be clearly modelled by the year, month and day of the week, we can train a simple neural network to model the seasonality trends and make sales predictions.

5.3.1 Feature Engineering

As the sales were observed to be correlated to different stores and items, we added a new feature of ‘base_sales’, where the average sales of a (store, item) pair are assigned to each data record, corresponding to its store and item. We then removed the ‘store’ and ‘item’ features. This effectively encodes the correlation of sales between each store and item. The final features used to train the network are ‘base_sales’, ‘year’, ‘month’, and ‘dayofweek’.

5.3.2 Model Architecture

A simple feedforward neural network was used, with one hidden layer of 256 linear neurons and ReLU activation, and one output layer of one linear neuron that gives the sales prediction (figure 20).

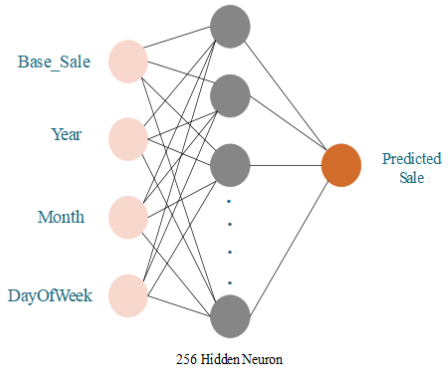

Figure 20: Architecture of the neural network

5.3.3 Experiments & Results

The model was trained for 200 epochs at a learning rate of 0.001. We first trained the model on the full dataset. We observed that the test data only required the first 3 months of the year, hence we trained a second model only using the first 3 months of each year in the training data. The table 17 below shows the model’s performance.

Model	Public Score	Private Score
Neural network (full year)	**13.89389**	12.64152
Neural network (first 3 months)	**13.87677**	12.6143

Table 17: Results of the neural network model

Training only on the first 3 months of training data seems to have improved the model performance. The model achieved a rank of 68 in the public leaderboard, and performed the best compared to other machine learning models we have tried thus far.

5.4 Model D: XGBoost

XGBoost (Extreme Gradient Boosting) applies a boosting technique in forming a random forest classifier based on the input features.

5.4.1 Feature Engineering

From section 2.2, we observed that there is a yearly, monthly and weekly seasonality. Hence, for XGboost, we provide the 3 additional input categorical features namely, dayOfWeek, month and year. Additionally, we also add in the sin of the month and the cosine of the month as a possible feature because the yearly seasonality in section 2.2.2 is somewhat like a sin or cosine graph.

5.4.2 Feature Importance

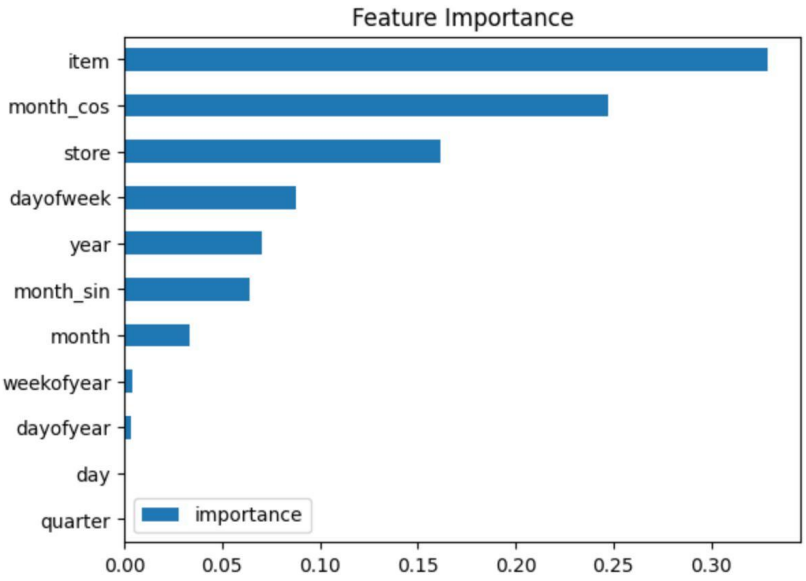

Figure 21: Feature Importance graph obtained from XGBoost

The feature importance shows that item, store, day of week, year and month are important features which support our observation from exploration analysis (Fig 21).

5.4.3 Results

We fit the model with the data and train the model with a maximum depth of 6 and 1000 estimators. We obtained a public score of **14.0853**

6 Conclusion

In summary, we learn that, contrary to what we believe at first, the best model is the extrapolation model. This may be because the dataset we have is a very simple dataset, hence the simplest solution is the best solution. Additionally, we learn that working with time-series data can be quite complex, with us needing a lot of feature engineering such as seasonality and trend for our model to be able to produce accurate results.

7 References

[1]: Hyndman, R.J., & Athanasopoulos, G. (2018) *Forecasting: principles and practice*, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. Accessed on 23 Nov 2023

[2]: Taylor SJ, Letham B. 2017. Forecasting at scale. *PeerJ Preprints* 5:e3190v2https://doi.org/10.7287/peerj.preprints.3190v2