Leveraging Machine Learning Algorithms to Improve Fashion Demand Forecasting

MSc Thesis, Business Analytics

By

Quinten van der Kaaij (2557876)

July 26, 2024

Supervised By

Guszti Eiben (VU)

Acknowledgement

I would like to thank Guszti Eiben for his guidance during the writing of this thesis. I am grateful Guszti's flexibility and open attitude to suggested adaptations.

Furthermore, I would like to express my gratitude to Annemieke van Goor for her involvedness and well needed regular check-ups during the writing of this thesis. I am grateful for all the help and advice I was given during a tough personal time.

Finally, I would like to thank my parents for their patience and continuous trust during the writing of this thesis and all the years before that. I am proud to have finished this thesis and thereby show their trust and patience was not misplaced.

Quinten van der Kaaij

July 26, 2024

ABSTRACT

In this research, an inquiry was made into the viability of using machine learning algorithms to improve demand forecasting for fast-fashion retailers. Specifically, retail sales from an Italian fast-fashion retailer as well as various exogenous variables were used to construct multiple machine learning models based on the eXtreme Gradient Boosting (XGBoost) and Random Forest (RF) algorithms. Using assumptions about opportunity and storage costs, along with self-constructed stock replenishment algorithms, the costs incurred by forecast errors could be quantified and a conclusive answer to the research question *"Can machine learning algorithms in the field of fashion sales forecasting be of added business value to a fast-fashion retailer?"* was given. Although a conclusive answer was given, assessing viability based on costs is heavily contingent on cost assumptions. Therefore, the aim of this research was more to propose a framework to come to such a conclusion, rather than giving a definitive verdict on the viability of using machine learning models in fashion demand forecasting.

SKU-level sales numbers were aggregated on categorical level. The three categories that are the most sold in the three most popular stores were investigated. Using weather indicators, macroeconomic features, Google trends data and various derived sales features, 9 datasets were created. Utilizing hyper-parameter tuning and k-fold temporal cross validation, the performance of the models was optimized. Looking at both the MAE and RMSE performance metrics, the RF model outperformed the XGBoost with an overall average MAPE of 22.8%, compared to 24.4% for the XGBoost models. While the models performed well in capturing demand variability, high demand peaks were often predicted too late, resulting in disproportionally high opportunity costs. Total costs due to mis-predictions exceeded a pre-established cost threshold. Therefore, based on the current analysis, it was concluded that machine learning models in the field of fast-fashion do not provide added business value to a fast-fashion retailer under the given assumptions.

Contents

1 Introduction and Preliminaries **1**
- 1.1 Introduction . . . 1
- 1.2 Problem statement and research goals . . . 3
- 1.3 Cost assumptions . . . 5
- 1.4 Practical considerations for forecasting . . . 5

2 Literature review **7**
- 2.1 Forecasting models in general sales forecasting . . . 7
- 2.2 ML algorithms in fashion sales forecasting . . . 9
  - 2.2.1 New item forecasting . . . 12
- 2.3 Data . . . 13
  - 2.3.1 Weather data . . . 13
  - 2.3.2 Other data in sales forecasting . . . 14

3 Data **17**
- 3.1 Raw data description . . . 17
  - 3.1.1 sales.CSV . . . 17
  - 3.1.2 vis2_gtrends_data.CSV . . . 18
  - 3.1.3 vis2_weather_data.CSV . . . 18
  - 3.1.4 ec_indicators.CSV . . . 19
- 3.2 Data preprocessing . . . 20
- 3.3 Feature engineering and exploratory data analysis . . . 21
  - 3.3.1 Sales data . . . 21
  - 3.3.2 Google trends data . . . 24
  - 3.3.3 Weather data . . . 24
  - 3.3.4 Economic indicators . . . 26
  - 3.3.5 Feature selection . . . 28
  - 3.3.6 Final data . . . 30

4 Methodology **31**
- 4.1 Algorithms . . . 31
  - 4.1.1 eXtreme Gradient Boosting . . . 31
  - 4.1.2 Random Forest . . . 33
- 4.2 Performance metrics . . . 34
  - 4.2.1 Mean Absolute Error (MAE) . . . 34
  - 4.2.2 Mean Absolute Percentage Error (MAPE) . . . 35
  - 4.2.3 Root Mean Squared Error (RMSE) . . . 35
  - 4.2.4 Root Mean Squared Percentage Error (RMSPE) . . . 36
- 4.3 Hyper parameter tuning . . . 36
  - 4.3.1 XGBoost . . . 36
  - 4.3.2 Random Forest . . . 38
- 4.4 k-fold temporal cross-validation . . . 39
- 4.5 Cost analysis . . . 40

5 Results **42**
5.1 XGBoost . . . 42
5.2 Random Forest . . . 45
5.3 Model comparison . . . 48
5.4 Cost analysis . . . 49

6 Conclusion **51**
6.1 Sub-question 1 . . . 51
6.2 Sub-question 2 . . . 51
6.3 Sub-question 3 . . . 52
6.4 Research question . . . 53

7 Discussion **54**
7.1 Costs and replenishment . . . 54
7.2 Data . . . 55
7.3 Model performance . . . 55

Introduction and Preliminaries

1.1 Introduction

In the fast paced and increasingly competitive landscape of fashion retail, it is of paramount importance to remain able to predict customer demand effectively to stay ahead of the curve. Fashion trends that evolve quickly, continuous change in customer preferences, and an ever-increasing ease for consumers to switch to competitors, all contribute to the challenge that is modern-day demand forecasting. Additionally, in the field of fast-fashion products have extremely high throughput and little to no historical data, which makes forecasting demand challenging. In addition, these products may have relatively high demand volatility and may be especially susceptible to fads [28].

Furthermore, it is becoming increasingly clear that events outside of the fashion ecosystem may also have a significant impact on customer demand. For seasonal items like shorts and skirts, the weather is an obvious external variable with high predictive power in short term demand planning. Additionally, various other factors like social media trends, holidays and inflation may all have an impact on the way consumers spend their money on fashion (see Section 2.3.2).

While traditional methods of demand forecasting made use mostly of historical sales data combined with expert opinions, the aforementioned changes in the field of online fashion retail make it increasingly difficult to rely on these partially qualitative ways of forecasting. In order to prevent unnecessary storage costs and lost sales by being out of stock, retailers are to a growing extent looking for more quantitative methods to automate the forecast process and effectively measure and reduce forecasting errors.

The rapid proliferation of Artificial Intelligence (AI) and the availability of Big Data opens new doors for advancements in the field of forecasting. Machine Learning (ML) algorithms make it increasingly easy to find hidden patterns in a multitude of numerical datasets. The aim of this thesis is to leverage the power of these ML algorithms to find such patterns, and thereby increase the accuracy of forecasts through the use of external data, compared to solely examining historical

sales data. Furthermore, using assumptions on costs and a proposed stock replenishment strategy based on the forecasts, a framework is provided to assess the viability of using ML for forecasting demand in the fast-fashion industry.

1.2 Problem statement and research goals

While adopting AI may seem like an appealing idea, as it can dramatically reduce costs and increase efficiency, the use of AI is not reserved for everyone. In order for AI to be implemented in an efficient manner, sufficient quality data is needed together with the willingness of top level management to take this leap and the expertise of employees to lead this technology in the right direction. Besides business level considerations to the implementation of AI in the field of demand forecasting, technical drawbacks exist to the use of AI as well.

Data quality is one of the most important factors in making accurate forecasts. When data is inconsistent or biased, it will negatively influence the prediction performance and may in turn lead to misguided business decisions. In addition, there is the problem of over fitting. This is a fundamental problem in the field of ML where the model performs well on the training data but may not generalize well to the real world, preventing it from having any practical advantage. Causes for this problem are the presence of noise in the training data, limited training data and overly complex algorithms [43]. In this research, a multitude of precautions are taken to minimize this problem (see Section 4.4 and 3.3.5).

Another consideration is the "black box syndrome". This refers to the inherent lack of interpretability of ML models, particularly complex ones, which makes it challenging for users to understand how the model arrives at its predictions [7].

Additionally, loss of human control and intuition is a common fear of stakeholders. This is a legitimized fear: due to a fault in the automatic trading systems of the New York Stock Exchange on 6 May 2010, momentary losses and gains in three largest US markets of a thousand points were realized in a matter of minutes, corresponding to trillions of dollars. This is an extreme example of the consequences of giving away too much authority to computers. Although in sales forecasting, these consequences may not be as dire, a malfunctioning fully automated prediction and order-placing system may incur heavy unwanted costs for a business.

Furthermore, using ML for forecasting should be financially beneficial to a company. First of all, it should be at least as accurate as a manually made forecast. Thereafter, estimates should be made about the costs of over- and under-stocking. By doing this, one can quantify the efficiency of ML forecasts in terms of Euros. In addition, using ML for forecasting can be automated. Depending on the size of the company, a solid ML implementation - while initially costly - may save costs considerably in the long run by reducing man hours.

Bringing it together, there are a lot of advantages and drawbacks to the use of ML in fashion sales forecasting. Moreover, the extent to which AI is useful depends on a wide range of factors.

Most importantly, the amount of historical data is often directly correlated with a prediction's accuracy. Whereas a year of historical sales data will capture seasonality dependence of an item, multiple years of data may be able to capture changing fashion trends and consumer behavior. Finally, in order for a business to actually implement ML forecasting, a thorough estimate of costs and benefits should be made.
In this research, an inquiry will be made on the feasibility of using ML forecasting in a business setting, taking into account all aforementioned considerations. The research question posed is as follows:

Can ML algorithms in the field of fashion sales forecasting be of added business value to a fast-fashion retailer?

The research question has multiple angles of incidence. In order to specify which facets of this research question will be focused on, the following sub-questions are posed:

Which ML algorithms perform best on forecasting sales? The answer to this question will be based both on the literature study and the experimental results. In the literature review, an investigation will be conducted on the most used and best performing models in the field of sales forecasting. Thereafter, a selection of the best performing models will be tested using regression related performance metrics.

What kind of external data can have a positive impact on the accuracy of clothing sales forecasts? Following the same procedure, an analysis will be done on relevant literature to see which types of external data have the most impact on sales in the fashion industry. Then, after running the model with viable features, a feature importance plot will be used to conclude the answer for this particular case.

How well do ML forecasts approach real demand? In order to answer this question, the aforementioned sub-questions are answered first. Using selected ML algorithms and a variety of data not directly related to clothing sales, forecasts will be produced. Using multiple numerical performance metrics, the accuracy of these forecasts can be assessed based on an already known actual demand and will be collocated to findings from relevant literature.

1.3 Cost assumptions

To answer the research question, assumptions on the costs of over- and understocking have to be made. Once we address the third sub-question, we can quantify whether the model over- or underestimated the demand for each time step. Using our cost assumptions, we can then calculate the total missed revenue resulting from forecasting errors.

In a real-world scenario, the missed revenue from using ML forecasting can be compared to the revenue from previous forecasting strategies to assess the business value of a ML implementation. However, this research lacks access to historical forecasts or supply chain cost data from the companies involved. Therefore, to conclusively answer the research question, ML algorithms in fashion sales forecasting are proposed to be beneficial when total costs do not exceed 5% of total sales revenue.

In other words, we expect an automated forecast system to reduce costs by 5% of total sales revenue by minimizing the man-hours required for manual monthly forecasts. If forecasting models incur high errors, resulting in costs that exceed this threshold due to over- or under-predicted demand, an ML forecasting model would not be considered viable. Further details on cost assumptions are provided in Section 4.5. Finally, it is important to note that assumptions about costs and the proposed threshold value significantly impact the conclusions drawn. Therefore, the primary aim of this research is to provide a framework for conducting this type of study, rather than definitively assessing the viability of using ML for demand forecasting in this specific scenario.

1.4 Practical considerations for forecasting

Prior to constructing a forecasting model, it is essential to consider several practical aspects:

- **Forecasting Horizon:** The forecasting horizon is the future temporal range in which forecasts are made. This can be days, weeks, months or years and is dependent on both the business - in fast-fashion, forecasting horizons are often shorter due to short product lifespans - and the amount of historical data: when there is less than a year of historic sales, it is challenging to capture seasonality and thus yearly forecasts will likely turn out inaccurate. In addition, external data used can impact the horizon choice: when making demand forecasts using weather data, the forecasting horizon cannot exceed more than two weeks, as weather forecasts surpassing that window are too unreliable.

- **Product aggregation level:** In nearly every fashion business, a product hierarchy exists. Often, the category with the highest granularity in this hierarchy is the Stock Keeping Unit (SKU). The SKU is often a combination of category, color, and size. While forecasts on SKU level may be most helpful in a business perspective, it is often computationally expensive to run forecasts for each SKU. Therfore, forecasts are often made on product or even category level.

- **Sales channel aggregation level:** When items are sold in physical stores, an other consideration to be made is how to aggregate demand with respect to the sales channel. For example, demand can be forecasted on store level, which is useful for making sure stock in stores is replenished accurately. Aggregating and forecasting demand for subsets of stores that correspond to the same distribution center can give insights for higher-level supply chain operations, such as production planning and warehouse replenishment. Finally, aggregating and forecasting demand in all stores is useful for long-term production planning and insight into projected growth of a company.

Literature review

The purpose of this literature review is to investigate the usefulness and effectiveness of different ML algorithms, performance metrics, and external data to forecast precision. The structure of the literature review is as follows: first, an inquiry on the use and effectiveness of ML algorithms in general sales forecasting will be made. While the specificities of the forecasting in the fashion industry are not touched upon in this section, it is a useful way to evaluate the advantages and disadvantages of different algorithms of forecasting in general. Secondly, the scope is then narrowed to sales forecasting in the fashion industry to inquire on challenges and opportunities specific to this field of forecasting. Finally, we will touch on the use and effectiveness of external data in sales forecasting.

2.1 Forecasting models in general sales forecasting

While there is relevant literature on sales forecasting in the fashion industry, some relatively new models when dealing with time series data such as Prophet are not mentioned in the literature of fashion sales forecasting. Therefore, in order to investigate the usefulness of other models, the scope of the literature review was broadened a bit to include research on forecasting sales in any industry.

In a 2017 study by Gurnani et al. (2017), a variety of statistical and ML models were used to forecast sales of a drug store company [18]. Among others, Auto Regressive Integrated Moving Average (ARIMA) models, Auto Regressive Neural Network (ARNN), Support Vector Machines (SVM), and various hybrid models using ARIMA together with XGBoost, ARNN and SVM were evaluated. Finally, a so-called STL (Seasonal and Trend decomposition using Loess) decomposition was used. By using the Mean Absolute Error (MAE) between the actual and predicted sales, they concluded that ARIMA performed worst due to it not being able to capture the non-linear part of the data. STL decomposition had the lowest MAE, with XGBoost being a close second. The ARIMA-hybrid models performed worse than their singular counterpart, only the

ARIMA-ARNN hybrid performed better than ARNN on its own.

In 2017, Facebook (currently Meta) released a time-series forecasting tool called Prophet. In the original blog post where Prophet was announced as an open source project, the authors display the following characteristics of a forecasting problem where Prophet performs well [32]:

- "hourly, daily, or weekly observations with at least a few months (preferably a year) of history"
- "strong multiple "human-scale" seasonalities: day of week and time of year"
- "important holidays that occur at irregular intervals that are known in advance (e.g. the Super Bowl)"
- "a reasonable number of missing observations or large outliers"
- "historical trend changes, for instance due to product launches or logging changes"
- "trends that are non-linear growth curves, where a trend hits a natural limit or saturates"

These characteristics are similar to the nature of this research, indicating that Prophet might be an interesting algorithm to examine further.

Zunic et al. (2020) used Prophet to forecast monthly retail sales for a supermarket [45]. In order to measure model performance, they use an expanding window back testing strategy. This means that they used all available data up to month t_{-1} and used that to predict the sales of month t. Then, they used all data up to t to predict month t_{+1}. This is all historical data for which the actual sales are known. When implementing an expansion of the test strategy in data with a yearly seasonal pattern, researchers recommended taking at least a rolling window of 12 months. Using the Mean Absolute Percentage Error (MAPE) as a performance metric, the researchers were able to produce a monthly forecast error of 8%.

In their paper, called Time series forecast of sales volume based on XGBoost, Zhang et al. (2021) inquired about the effectiveness of using XGBoost in forecasting based on sales volume [44]. The data used included two data sets of sales orders from two stores that sell the same milk tea in Beijing. In addition, weather data was collected to try to improve forecast accuracy. Among others, weather features included are air quality, condition (rainy, sunny, or cloudy) and temperature. Moreover, multiple time-based features like *month*, *day_of_week*, *is_holiday* were constructed from the timestamps. XGBoost was compared to several other models such as Long Short-Term Memory (LSTM), ARIMA, Prophet, and Gradient Boosting Decision Tree (GBDT). The Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) were used

as performance metrics. In terms of RMSE and MAE, XGBoost performed best, with GBDT being a very close second, indicating that the boosting decision tree models are among the best performing models in such forecasting problems.

In his paper, Dairu (2021) researched the effectiveness of XGBoost in sales forecasting, using sales data from Walmart stores that span 1913 days [14]. While no external data was used, explanatory variables like price, promotions, day of the week, and special events were incorporated. The Root Mean Squared Scaled Error (RMSSE) was used as a performance measure. RMSSE is particularly useful when comparing models across different time series, as it scales the error in a way that allows for meaningful comparisons regardless of the scale of the original data. In addition to XGBoost, ridge regression and linear regression were examined. The XGBoost model outperformed ridge regression and linear regression with RMSSE scores of 0.665, 0.774 and 0.783 respectively. Moreover, the XGBoost model required less computing power and memory resources.

In a recent study by Ensafi et al. (2022), various models like RF, LSTM, ARIMA and Prophet variations were evaluated on furniture sales that have a strong seasonal component in their demand, using a real-world sales dataset [15]. The stacked LSTM model outperformed the other models, obtaining the lowest MAPE and RMSE values. The models' performances were evaluated using three metrics: MSE, RMSE, and MAPE. In several performance measurements, the SARIMA model demonstrated superior performance compared to the other classical techniques. Prophet was mentioned especially because its performance relative to its ease of use and quickness was highly favorable, making it a popular choice for practitioners seeking a balance between forecast accuracy and implementation efficiency.

2.2 ML algorithms in fashion sales forecasting

Since a lot of literature on sales prediction has been written, the decision was made to narrow the scope the literature review specifically on the use of ML to predict demand in the fashion industry. By doing so, predictions can be made more accurately as the fashion sector is especially sensitive to features like seasonality and weather conditions.

The implications of this seasonality effect are backed up by Thomassey (2010) [38]. More specifically, he laid out four fundamental considerations to make when forecasting sales in the fashion industry.

Seasonality

Since each item of clothing is related to a certain season and/or weather conditions, most items are seasonal. The degree to which items are seasonal may vary. Furthermore, while seasonality can give an idea of the trend, unpredictable variations in the weather can result in significant fluctuations in the demand, again depending on the garment.

Exogenous variables

Other variables that might have an influence on demand are end-of-season sale, sales promotion, consumer purchasing power, and competitor performance.

Fashion trends

Fashion trends are to a varying extent responsible for demand changes, especially over a multiple year period. These trends and their impact on consumer demand are often challenging to predict, especially in the fast-fashion industry.

Color and size

Generally, it is best to maintain an extensive collection in terms of varying sizes and colors. In this way, the collection will attract more customers. However, this makes forecasting challenging, since customer preferences and body type may also change over time.

Sen et al. (2008) described other challenges in the retail forecasting field like short product life cycles, unpredictable customer demands, a wide range of product choices, and lengthy supply processes [33]. In the case of Nuna Lie, these challenges are considerable: a high number of unique products with a shelf life of merely 12 weeks, together with a low demand in absolute numbers pose difficulties in making accurate predictions (see Section 4).

A study by Wong et al. (2010) inquired on the usefulness of various prediction models in the forecasting of medium-term fashion sales [42]. They proposed a Hybrid Intelligence (HI) model. This HI model is a combination of a neural network, an extreme learning machine, and a harmony search algorithm. They compared the performance of this model with an Evolutionary Neural Network (ENN), ARIMA, AR and AR2 using the MAPE as a performance metric. The HI model outperforms all the other models in monthly and annual forecasting in terms of the MAPE, with the AR2 and ARIMA models performing second and third best, respectively. It should be noted that only historical sales data were used to make this prediction. The reason that the classical Autoregressive models outperformed the ENN, is likely due to predictions being based purely on historical data: no exogneous variables were used. The exponential smoothing and ARIMA models fall under the category of linear methods, as they utilize a linear functional form to model time-series data. When time series data with strong non-linear trends is used to forecast, these

methods may fall short on precision [29].

Wong et al. (2010) implied that fashion sales data contains such non-linearities, so they reverted their focus to Extreme Learning Machines (ELM): an adaptation on the basic Neural Network (NN) that offers a solution to numerous challenges encountered by gradient learning algorithms [42]. It avoids problems related to the selection of stopping criteria, determining learning rates, and setting learning epochs, due to its unique learning mechanism [37].

Using ELMs, Sun et al. (2008) aimed to investigate the link between fashion product sales and demand-influencing characteristics such as design factors, using real data from a fashion retailer in Hong Kong [37]. Compared to gradient-based algorithms, ELMs learn faster with better generalization performance and avoid issues such as stopping criteria, learning rate, learning epochs, local minima, and over fitting. ELMs do come with a downside: the model's biases and weights are chosen at random, resulting in a different output each time the model is run. Consequently, numerous ELMs were run, and the average of those outputs was taken to minimize this problem. Three experiments were carried out to investigate the influence of demand-influencing characteristics. In these different experiments, researchers compared ELMs with two different gradient descent algorithms; the batch steepest descent backpropagation algorithm (GDA), in which a variable learning rate was used, and the gradient descent momentum and adaptive learning ratio backpropagation (GDX).

In all three experiments, a prediction for jeans, socks and jackets was made, respectively. All three contained the variables price, color and historic sales data, and the jeans and jackets forecast also incorporated size. For all three of these experiments, the ELM outperformed the GDA and GDX.

In a study done by Catal et al. (2019), various regression models as well as time series analysis models were compared to benchmark their effectiveness based on Walmart sales data [8]. Linear regression, Bayesian regression, neural network regression, decision forest regression, and boosted decision tree regression were used in this study. Furthermore, seasonal ARIMA, non-seasonal ARIMA, seasonal ETS, non-seasonal ETS, naive method, average method and drift method were examined. Besides historical sales data, external data like weather and economic indicator data was used. When predicting sales for a single store, the decision forest regression technique had the lowest RMSE and MAE. When taking into account all stores, the time-series techniques could not handle the higher dimensionality and were hence dropped from the analysis. For the multi store data, the Boosted Decision Tree regression model performed best in terms of RMSE and MAE.

Beheshti et al. (2015) also come to the conclusion that linear methods such as ARIMA, ex-

ponential smoothing, but also Box-Jenkins, are often not suitable with real-world data [6]. In their meta-analysis, they recommend the use of Artificial Neural Networks (ANN) for real world applications.

Lopez et al. (2008) inquired on the effectiveness of ENNs in fashion sales forecasting [23]. Using two years of historical sales data, ENN was compared to SARIMA. The researchers concluded that ENNs outperform SARIMA, especially when data is noisy. No exogenous variables were used.

In a study done by Vairagade et al. (2019), the effectiveness of ML models to predict groceries was investigated. In addition to historical sales, external data such as oil prices, holiday information and store information was used [40]. Using r^2, MSE and MAE as performance metrics, the researchers found the RF algorithm to perform best.

In an article from Ji et al. (2019), XGBoost was used to forecast sales for a cross-border e-Commerce company [21]. While time series models such as exponential smoothing and ARIMA are effective in capturing linear behavior with seasonality, they cannot capture external factors like price changes and promotions. Therefore, these models are often used as a benchmark. Furthermore, the researchers produced a so-called C-A-XGBoost model, which combined the power of clustering to obtain important sales features, ARIMA models for the linear part of the sales data, and XGBoost for the non-linear part of the data. This combined model outperformed the regular XGBoost model.

2.2.1 New item forecasting

A well known problem that fashion retailers face when making a forecast, especially with seasonal items, is the introduction of new items. These items do not have any historical sales data, making it challenging to predict sales accurately. A study done by Loureiro et al. (2018) focused on that particular problem [24]. Specifically, they predicted sales of women's bags over two seasons that had no historical data. The bags were sold in physical stores as opposed to online. The data set consisted of numerous input variables that they categorized into three different types:

Domain knowledge: This category contained variables such as *expectation_level* which provides insight into he expected sales and is based on expert opinion.

Physical characteristics: This category contained variables like *family color* and *price*.

Logistical and organizational aspects: This category contained variables like *store_type* and *fashion*, the latter being a categorical variable saying something about the overall fashion trend of this summer.

The researchers used Linear Regression, Decision Trees, RF, Support Vector Regression, Artificial Neural Networks and Deep Neural Networks to make predictions, and found that the Deep Neural Network performed best based on RMSE and MSE, whereas RF performed best in terms of R-squared, MAPE and MAE. Furthermore, Loureiro et al. (2018) looked at feature importance and found that *expectation_level*, *store_type* and *fashion* were among the variables with the highest predictive power [24].

In his research, Thomassey (2010) highlighted the dichotomy between long-term (one year) and short-term (a few weeks) forecasting [38]. His long-term forecasting method relied on using a fuzzy inference system for the influence of explanatory variables modeling. In his short-term forecasting method, Thomassey relied on neural networks, which allowed intermediate updates on the long-term forecast using the last known sales. An important note that Thomassey made is that it is imperative that both the short-term forecast and the long-term forecast have enough historical data. Therefore, he aggregated SKUs to the family of items; the lowest hierarchical aggregation level for which historical data existed.

2.3 Data

This section delves into the use and effectiveness of adding additional features to the input data. In order to use ML effectively, the data that enters the model has to be relevant to the prediction one wants to make. Furthermore, it is important to make sure that the number of features does not grow too large. This might result in the 'curse of dimensionality' [41]. The phenomenon known as the curse of dimensionality occurs in high-dimensional environments where the amount of data becomes sparser and less reliable as the number of dimensions (features) increases. Consequently, this results in the ML model becoming exponentially more complex with the increasing number of features. Therefore, it is imperative to ensure a balance between adding informative features and maintaining a manageable feature set. Selecting relevant features through methods such as feature selection or dimensionality reduction, can enhance model performance and generalizability.

2.3.1 Weather data

In a recent 2017 paper by Steinker et al. (2017), an inquiry was made on the use of weather data on sales forecasting for e-commerce operations in Germany [36]. Although future weather is unknown, weather forecasts are known. In their paper, they observed that including weather forecasts on a 1-day time horizon reduced the MSE of the model by 13.9% on average, while a 7-day horizon only reduced the MSE by 4.7% on average. This finding suggests weather data

can be an important predictor in fashion demand forecasting, but will lose predictive power as the forecasting horizon progresses. Another notable consideration posed in the research is the granularity of aggregation of weather forecast data. In case of a retail store, the process is relatively simple, and one can take the weather for that specific location. In case data is aggregated over multiple locations, the process becomes more complicated. Especially there are customers from different countries present in the data, it can be challenging to decide how one wants to aggregate the weather data on the locations that one sells in. In their paper, Steinker et al. performed a correlation analysis on weather stations throughout Germany to see if the differences throughout the country are significant. They found a high correlation of weather variables between stations in the region under scrutiny, which lead the researchers to take the weather variables of a single station as a proxy for the entire region. Absolute measures such as hours of sunshine, precipitation, and temperature were used by Steinker et al., along with relative measures such as derived weather scores that take into account the season. These scores acknowledge that multiple hours of sunshine are less common in certain months, thereby assigning a higher relative weather score to such occurrences during these periods.

In a recent study done by Lv et al. (2023), the added value of weather data to predict sales for retail stores using ML was measured [25]. Using sales data from 2018 to 2019, the performance of XGBoost, RF and GBDT was examined. In terms of MSE, the GBDT model scored best, followed by XGBoost and RF respectively. Items were aggregated on categorical level and the MSE for predicting dresses, down jackets, and shirts was reduced by 86.03%, 80.14%, and 41.49% on average over the models by incorporating weather data. Interestingly, the researchers found that adding weather variables for t-shirts, trousers and knitwear increased the MSE, indicating that weather data can be noise and of no predictive value depending on the garment under scrutiny.

In 2021, Rose et al. (2021) made use of RF models to incorporate various weather variables into their forecast of supermarket sales in England [30]. Using r^2 as a performance measure, the researchers found that the addition of weather variables to the model increased r^2 from 0.86 to 0.90, with the summer and spring months being the largest contributors to this increase.

2.3.2 Other data in sales forecasting

Ali et al. (2009) researched grocery sales at the SKU level and the use of promotion in these predictions to make a more accurate forecast [1]. They found that in times when there is no promotion, simple time-series models generally suffice to make accurate predictions. However, when looking at promotion time-series they could improve the accuracy by 65% by including

manually engineered features and more advanced models like regression trees.

Trapero et al. (2015) found that using Principal Component Analysis (PCA) solves collinearity and high-dimensional issues that arise when incorporating promotion features such as price drops in the train data [39]. Their models outperformed statistical methods like ARIMA and expert's forecast in weekly demand.

Ma et al. (2016) researched demand forecasting when taking into account inter- and intra-categorical promotions [26]. In their study, they found that including promotional data for SKU-level forecasting increases accuracy by 12.6%. 95% of that increase is due to the inclusion of inter-categorical promotion. This means that the promotion of an SKU within a certain category itself is a way more valuable predictor than promotion on SKU's from different categories.

Cui et al. (2017) researched the operational value of social media information [13]. Specifically, they used internal operations data and publicly available social media data to train various ML models. The internal operations data consisted of the sales, money spent on promotion and the sales forecast. Open social media data consisted of the number of posts, comments and likes on the page of the company in question. Different ML models were used, with RF having the highest relative improvement in the forecast with an increase of 52%, followed by the SVM and XGBoost model, obtaining 31% and 21% improvement respectively compared to the naive forecasting method.

In a study done by Allenby et al. (1996), an inquiry was made on how sales forecasts can be improved by taking customer confidence in the economy in to account [3]. 84 months of monthly sales data of five fortune 500 retailers were used as main input for the model. Furthermore, the researchers define *the ability to buy* by dividing Personal Disposable Income (PDI) by the Consumer Price Index for Apparel (CPIA). This means that the ability to buy increases, when PDI increases and/or CPIA decreases. Besides the ability to buy, customer confidence data from the University of Michigan was used. Using a Seemingly Unrelated Regression (SUR) and a Hierarchical Bayes model, the researchers discovered that ability to purchase and consumer confidence have a distinct influence on sales in the pre-season and in-season, with *confidence* being a better predictor of pre-season sales and *ability to buy* being a better predictor of in-season sales.

A more recent study done by Sageart et al. (2018) evaluated several techniques to improving tactical sales forecasting through the use of macroeconomic leading indicators [31]. Using data from a tire factory, they split the raw data into an autoregressive and a seasonality part, and added economic indicators. Using the FRED database, more than 68.000 indicators were considered. LASSO was used to drop irrelevant indicators, resulting in a final set of 1.082 indicators. Some

well-known indicators used were the inflation rate, employment rate, and CPI. Using economic indicators, the researchers found an increase in accuracy of 18.8%. The effectiveness of using macroeconmic leading indicator is backed up further by Guo et al. (2013) [17]. In 2013 study, they aimed to forecast retail sales for a large fashion retailer in Hong Kong. Among others, endogenous variables included price information, information about material, promotion strategy, life span of an item and release date of an item. Exogenous varibles included multiple weather variables, and macroeconomic indicators such as CCI, CPI, GDP, producer price index. Using the six month average, these economic variables were interpolated to match the granularity of the input data. The researches used a harmony search-wrapper-based variable selection (HWVS) module to find the optimal subset of features, which were consequently fed to a multivariate intelligent forecaster (MIF) module to produce forecasts. The most important features included original selling price, shop quantity, release date, life span, climate index and five economic indexes. Models using this subset of features outperformed models using all possible features in every performance metric, highlighting the importance of thorough variable selection.

Data

3.1 Raw data description

The historical sales data, weather data and Google trends data used in this research is part of the VISUELLE2.0 dataset [35]. This is an open source dataset containing real data of 5,355 unique fashion items from the Italian fast fashion retailer Nuna Lie. The dataset consists of multiple CSV files, each containing different information about sales, demand, and exogenous variables. In addition, macroeconomic indicators were added manually based on recommendations in the literature.

3.1.1 sales.CSV

This CSV file is a 108,651x22 dataset containing information about the 12-week demand for each individual product in each store where it is sold. The columns in the dataset are described in table 3.1

Table 3.1: Description of Product Data Columns

Column Name	**Data Type**	**Description**
external_code	Text	Code for identifying the product externally.
retail	Int	Store identifier.
season	String	Season of release for the product.
category	String	Category of the product.
color	String	Color of the product.
image_path	String	Path to the image of the product.
fabric	String	Fabric/material of the product.
release_date	Date	Date when the product was released.
0-11	Numeric	Demand of the product in weeks 0-11.

Each row in this CSV file represents a unique product/store combination, together with its 12-week demand in that store following the introduction date. Not every item is sold in every store. For example, item with external code 005, a grey long sleeve with acrylic fabric, is sold in twelve different stores. In addition, the release date is not the same in all stores. Whereas in store 36,

item 005 was introduced on 2016-11-28, the same item was introduced on 2016-12-12 in store 19. This is important to keep in mind, since the time series for each item/store combination is the same (12 weeks) and starts after its introduction date. That is, exogenous time series data should be matched accordingly.

An example of the demand curve for a single item following the twelve weeks after its introduction date (2019-11-28) can be seen in Figure 3.1.

Figure 3.1: Example demand curve of item 005

3.1.2 vis2_gtrends_data.CSV

This CSV file is a 220*97 time series data set that contains weekly aggregated Google trends data for each clothing category (e.g. long sleeve, miniskirt), each available color (10 in total) and fabric(e.g. fur, plush). The time series spans from 2015-10-05 to 2019-12-09. Values for each column represent the average Google search popularity of said category, color, or fabric and range from 0-100.

3.1.3 vis2_weather_data.CSV

This CSV file is a 89,071*14 time series data set that contains daily weather information for different store location. Data were retrieved from the Italian Weather Stats website Ilmeteo. The columns are described in Table 3.2.

Since all retail stores and weather locations have numerical values, a *shop_weather_pairs* dictionary is provided to match retail stores to the closest weather locations.

Column Name	Data Type	Description
locality	Float	Numeric location identifier
date	Date	Date of the recorded weather data.
avg temp °C	Float	Average temperature in degrees Celsius.
min temp °C	Float	Minimum temperature in degrees Celsius.
max temp °C	Float	Maximum temperature in degrees Celsius.
dew point °C	Float	Dew point temperature in degrees Celsius.
humidity %	Float	Humidity percentage.
visibility km	Float	Visibility in kilometers.
avg wind km/h	Float	Average wind speed in kilometers per hour.
max wind km/h	Float	Maximum wind speed in kilometers per hour.
gust km/h	Float	Gust speed in kilometers per hour.
slm pressure mb	Float	Sea level pressure in millibars.
avg pressure mb	Float	Average atmospheric pressure in millibars.
rain mm	Float	Amount of rainfall in millimeters.

Table 3.2: Description of weather data columns

3.1.4 ec_indicators.CSV

Using the FRED website, various economic indicators pertaining to Italy were derived based on the literature and the availability of the data. Adding macroeconomic data is largely inspired by the research of Sageart et al. (2018) who managed to increase the accuracy of the forecast by 18.8% by adding macroeconomic indicators to their model [31].

A compiled overview of all economic indicators used is shown in Table 3.3. A more detailed explanation about the variables can be found in Section 3.3.4.

Column Name	Data Type	Description
HICP	Float	Measure of inflation.
GDP	Date	Measure of a countries' economic growth.
Household debt to GDP	Float	Ratio of debt to GDP, viewed as a percentage.
Interest rate	Float	Risk free rate of 10-year government bonds.
Consumer Confidence	Float	Measure of confidence in economy and personal finances by consumers.

Table 3.3: Description of Economic Indicator columns

3.2 Data preprocessing

Before being able to use the data in the algorithms, the data needs to be in the right format. Prior to that, it is imperative to know what exactly will be researched. In section 1.4, three practical considerations were posed. Based on the structure of the data, the following choices were made regarding the considerations:

Forecasting horizon: Since we are working with items whose demand is heavily influenced by the weather and seasons, the choice was made to research the impact of weather data on forecast accuracy. Because weather forecasts do not reach more than two weeks in the future, the forecast horizon will be two weeks. To make this possible, the assumption was made that weather forecasts are perfect representations of the actual weather.

Product aggregation level: As mentioned in section 3.1.1, each unique item in the dataset has exactly 12 weeks of historic data. This is a problem for making forecasts: seasonal trends cannot be reflected well in a time span this short. Therefore, the choice was made to aggregate data to category level.

Sales channel aggregation level: Because the choice was made to forecast on a relatively short horizon, a logical consequence would be to forecast on store level. Having an accurate 2 week forecast can be highly beneficial at store level to optimize replenishment strategies, while being less useful on a total level.
Since more than 100 stores and 50 categories are available in the dataset, the following choices were made to constrain the width of the research.

1. Using the total number of items sold per store, the three top selling stores were selected.

2. For each of those stores, the three best selling categories were identified.

The stores and categories researched can be found in table 3.4

Store	**Category** 1	**Category** 2	**Category** 3
12	culottes	doll_dress	long_sleeve
30	culottes	doll_dress	long_sleeve
36	culottes	long_sleeve	sleeveless

Table 3.4: Most popular stores and corresponding categories

This means, that per algorithm used, nine different forecasts are be made. For consistency purposes, the choice was made to use the *store 12 culottes* forecast for most examples and images.

Any data transformation and feature engineering was done in a likewise manner for the remaining eight data sets.

3.3 Feature engineering and exploratory data analysis

3.3.1 Sales data

In order to obtain useful features from the raw sales data, various data transformations were performed. Firstly, since the the raw data contains weekly demand per unique store/item combination after the introduction date, total demand per store and category is obtained by grouping individual demand based on store and category. An example of a resulting data frame is displayed in Table 3.5.

release_date	category	retail	0	1	2	3	4	5	6	7	8	9	10	11
2017-01-30	culottes	12	1	0	1	1	1	1	1	0	0	1	1	0
2017-02-06	culottes	12	3	1	2	0	1	1	1	2	1	1	1	2
2017-02-13	culottes	12	2	3	6	1	6	3	1	3	3	0	0	0
2017-02-20	culottes	12	8	3	3	4	5	2	4	4	0	1	0	0
2017-02-27	culottes	12	10	7	6	15	12	4	6	2	8	2	0	0

Table 3.5: Demand aggregation for store 12

We observe the summed demand for each category in store 12 per introduction date. Since we want to end up with a single demand column, the next step is to match the weekly demand per category based on introduction week. This means that we are looking for the sum of the diagonals from left bottom to the top right. For example, the total demand at 2017-02-27 should be the demand in that week, plus the demand in the week before etc, going back a maximum of 12 weeks. This results in a data frame as displayed in Table 3.6.

release_date	category	retail	demand
2017-01-30	culottes	12	1
2017-02-06	culottes	12	3
2017-02-13	culottes	12	4
2017-02-20	culottes	12	14
2017-02-27	culottes	12	20

Table 3.6: Retail Data for Culottes

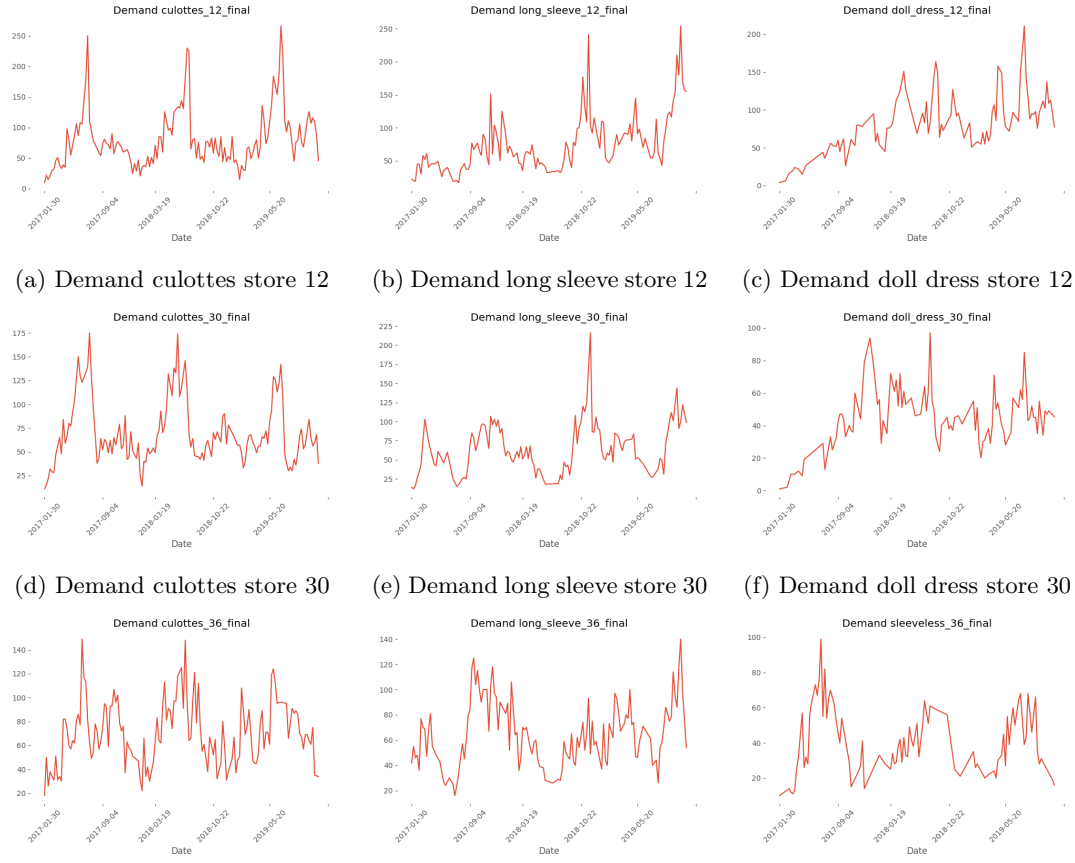
Using the same procedure for all 9 datasets, the final demand curves were created. They are shown in Figure 3.2. The demand for culottes exhibits strong seasonality, as demand increases in the summer. No significant growth trend is observed over time. Demand for culottes in store 36 however, shows significantly more variability than stores 12 and 30. Unless this variability can be

explained by the explanatory variables, it is highly likely that the prediction models for culottes in store 36 will perform worse compared to store 30 and 12.

Although the demand for long sleeves demonstrates seasonality with peak months during winter, it is less clearly defined than the seasonality in demand for culottes. While both in store 12 and 30 a slight growth trend can be observed, this characteristic is not visible in store 36.

In contrast to the demand for long sleeves and culottes, the demand for doll dresses shows a strong growth trend over time and a less clearly defined seasonality. Demand for doll dresses in stores 12 and 30 shows similar patterns, although the growth trend in store 12 continues more evenly over time.

Finally, the demand for sleeveless items in store 36 depicts no growth trends, but a strong seasonal component that drives up demand during summer months.



(a) Demand culottes store 12 (b) Demand long sleeve store 12 (c) Demand doll dress store 12

(d) Demand culottes store 30 (e) Demand long sleeve store 30 (f) Demand doll dress store 30

(g) Demand culottes store 36 (h) Demand long sleeve store 36 (i) Demand sleeveless store 36

Figure 3.2: Demand curves for all datasets

Using the resulted demand, features *lag_1* and *lag_2* were made by taking the demand from one and two weeks before respectively. Furthermore, the feature *rolling_sum_12w* was obtained by

taking the rolling sum of the demand over the last 12 weeks, giving a smooth proxy for the trend of the demand in the last 12 weeks.

Another significant driver for demand is the number of items that are available in the store. Using the fact that each individual item has a unique release date, the number of introductions per week can be calculated. The number of introduction for the 3 best selling items in store 12 is shown in Figure 3.3.

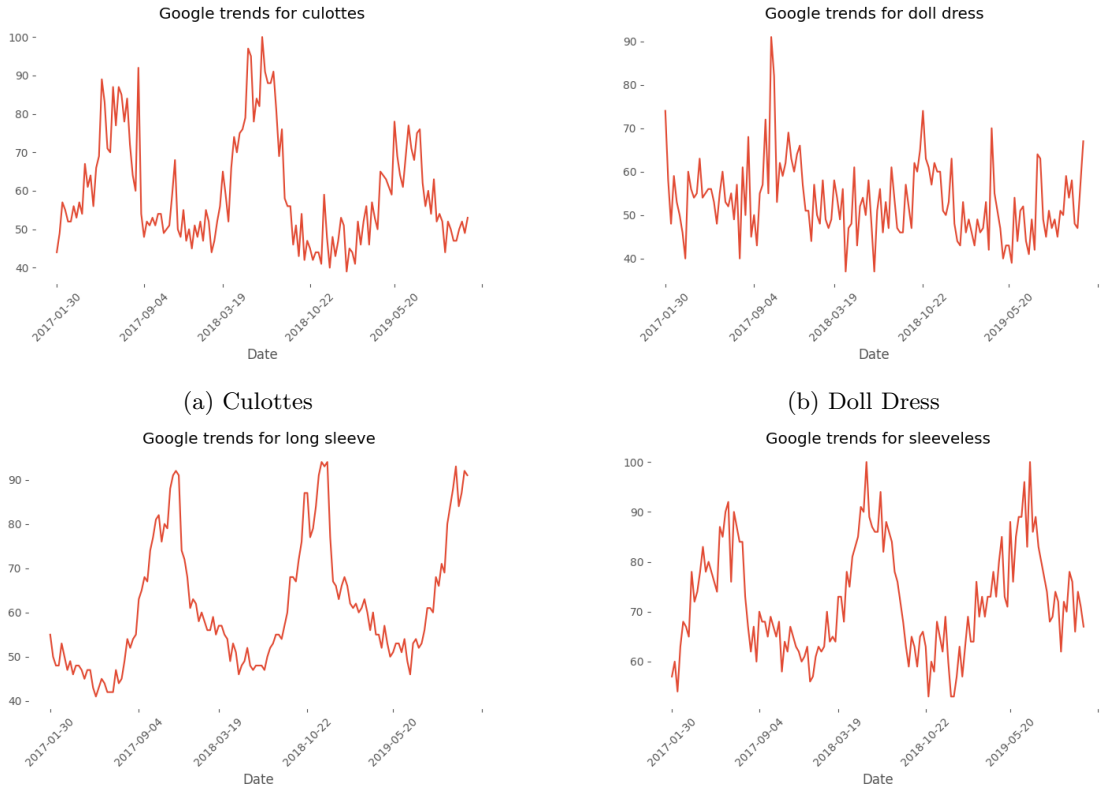
Figure 3.3: Number of item introductions in store 12

The plot shows that the number of introductions is highly seasonal: culottes and doll dresses are mostly introduced right before winter, while long sleeves are more pertained to winter. Similar patterns are observed in stores 30 and 36.

Finally, using the temporal properties of the release_date column, the features *year*, *month* and *week_of_year* were obtained.

3.3.2 Google trends data

Since the Google trends data is already in a weekly format, the features were selected only by selecting the columns referring to the category under scrutiny. In figure 3.4, we observe the Google trends data for the four different categories.



(a) Culottes

(b) Doll Dress

(c) Long Sleeve

(d) Sleeveless

Figure 3.4: Google trends data

We observe a significant seasonal effect of the popularity of google searches for culottes, long sleeves and sleeveless items, where popularity of sleeveless and culottes peaks during summer and long sleeves in winter. Interestingly, popularity of doll dresses barely shows seasonality. These seasonal effects, or lack thereof, were similarly noted in the demand of said items (See figure 3.2).

3.3.3 Weather data

The raw weather data contains daily observations for 12 different weather indicators. For reference, these are shown in table 3.2. Since these observations are on a daily basis, an effective approach for resampling the data into weekly intervals had to be devised to preserve the information hidden in the data. Given the inherent properties of the features, each feature was aggregated

from daily to weekly as follows:

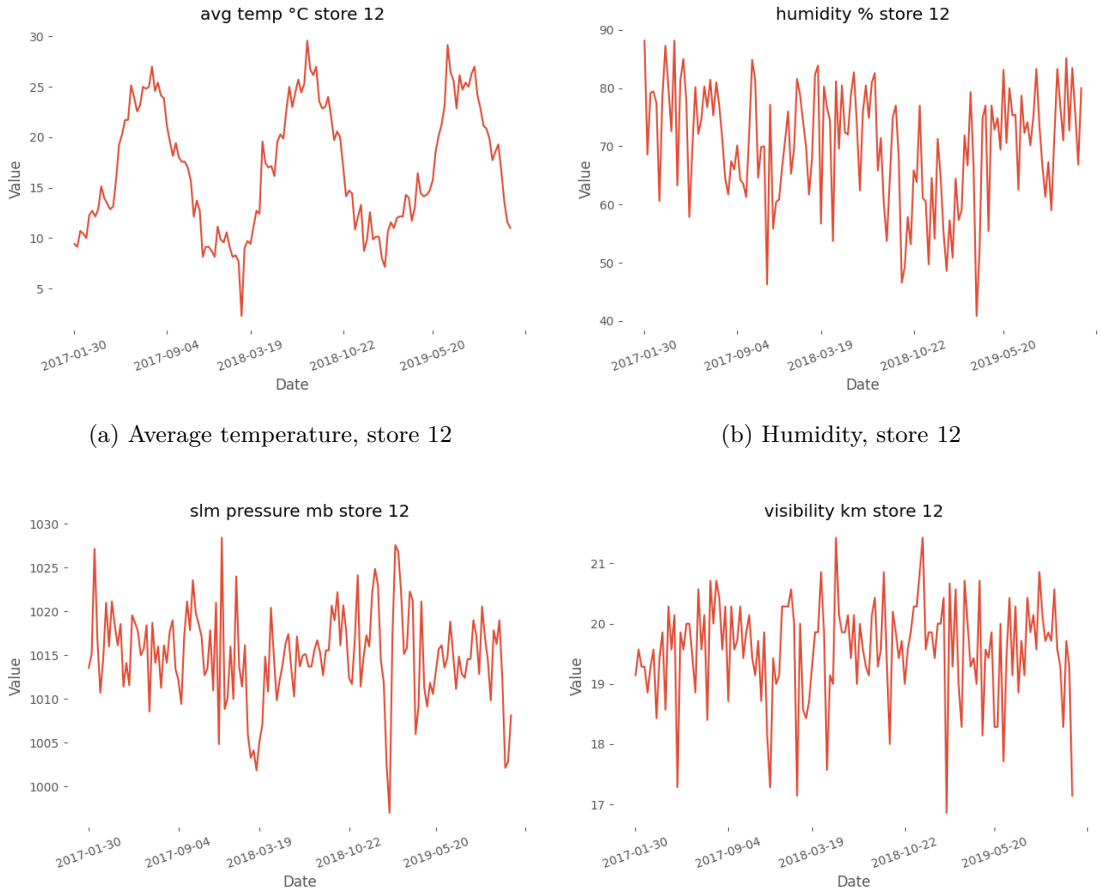
Mean: *avg_temp, dew_point, humidity, visibility, avg_wind_kmh, gust_kmh, slm_pressure_mb, avg_pressure_mb*

Max: *max_temp, max_wind*

Min: *min_temp*

Sum: *rain_mm*

For illustration purposes, the interpolated data for the average temperature, humidity and pressure observations are shown graphically in Figure 3.5.



(a) Average temperature, store 12

(b) Humidity, store 12

(c) Air pressure, store 12

(d) Visibility, store 12

Figure 3.5: Weather indicator data after aggregation

3.3.4 Economic indicators

In total, five economic indicators were obtained from the FRED database. A more detailed explanation for each variable is given below. The economic indicators were measured monthly and bi-yearly. Since we work with weekly data, linear interpolation was used to transform these indicators to a weekly frequency, ensuring a more granular analysis of the economic trends over time.

Harmonized Consumer Price Index: The Harmonized Consumer Price Index (HICP) is an inflation measure used in the European Union[12]. It measures the fluctuations in price of a predetermined basket of household goods, making for easy comparison of price changes between EU member states and fluctuations over time. Generally, HICP is negatively correlated with demand, since higher prices put pressure on the willingness of consumers to buy. For this specific instance however, an argument can be made for a positive correlation given that clothing is a primary need and we are analyzing relatively cheap fast fashion. Increased overall prices may nudge consumers that would otherwise buy expensive clothes to cheaper alternatives.

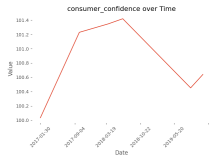
Gross Domestic Product: The Gross Domestic Product (GDP) is a measure of a countries' total economic output, measured in euros [11]. Higher GDP indicates economic growth, and is often an indicator that consumers have more disposable income. Consumers with more disposable income are more inclined to buy more expensive clothing. Therefore, a negative correlation with GDP and demand is assumed.

Household debt to GDP: Household debt to GDP is the percentage of the total debt held by households as a percentage of the countries' GDP[19]. Higher household debt to GDP would indicate that households are more leveraged and are probably more inclined to look for more affordable options for their basic needs. Therefore, a positive correlation between household debt to GDP and demand is hypothesized.

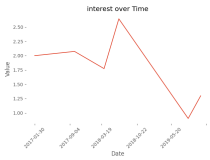
Interest rate: The interest rate is a macroeconomic indicator that measures the yearly percentual return on the safest possible investment: government bonds[20]. Interest rates are set by central banks and are used as a lever to increase or decrease spending. Generally, a higher interest rate will reduce spending, as the alternative (saving) will be more profitable. Consequently, a negative correlation between interest and demand is posed.

Consumer Confidence: Consumer confidence is a measure of the degree of confidence consumers have in the general economic state and their personal financial situation [10]. The indicator is normalized around 100, meaning that values higher than 100 indicate greater confidence, whereas values lower than 100 depict a pessimistic outlook. A high consumer confidence index indicates a positive outlook for consumers on their financial situation in the future. This may result in an increase in discretionary spending. Therefore, a positive correlation between consumer confidence and fast fashion demand is theorized.

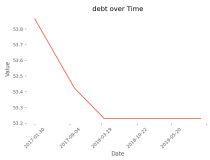
The resulting time series after linear interpolation are shown in Figure 3.6. Although research suggests economic indicators can have important predictive value, the low granularity of raw data results in steep charts with little variability. Although it is unlikely that these features will explain weekly variability in demand, they may be related to the longer term trend changes of demand.



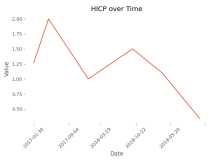
(a) Consumer Confidence



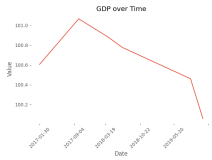
(b) Interest



(c) Debt



(d) HICP



(e) GDP

Figure 3.6: Economic indicators after linear interpolation

3.3.5 Feature selection

A common problem that can occur during the training of a model on a highly dimensional dataset, is the problem of multicollinearity. Alin (2010) describes the problem as follows: "Multicollinearity refers to the linear relationship among two or more variables, which also means lack of orthogonality among them" [2]. When two or more variables are highly correlated, a regression model may encounter difficulties in identifying the explanatory power of these variables, resulting in decreased model performance. Shreshta (2020) describes three ways to detect multicollinearity in a dataset [34]:

- Correlation Coefficient analysis
- Variance Influence Factor analysis
- Eigenvalues analysis

As it is the most common and easy to interpret, the choice was made to conduct a correlation coefficient analysis. Generally, there are two different methods for analyzing correlation between two variables X and Y: Pearson's correlation coefficient, and Spearman's Rank coefficient. Pearson's correlation assesses the strength and direction of the linear relationship between two variables, as described in equation 3.3.1

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{3.3.1}$$

where X_i and Y_i are individual data points, and $\bar{X}$ and $\bar{Y}$ are the means of X and Y, respectively. Spearman's Rank coefficient on the other hand, describes the strength and direction of the rank, or ordering of two variables X and Y. Its formula is given by equation 3.3.2

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2 - 1)} \tag{3.3.2}$$

where d_i is the difference between the ranks of corresponding data points X_i and Y_i, and n is the number of data points.

Both coefficients range from -1 to 1, where -1 indicates a perfect negative correlation, and 1 a perfect positive correlation. While Pearson's method is most widely used, it assumes normality. A common test to check whether data is distributed normally, is the Shapiro-Wilk test. Its formula is given by equation 3.3.3

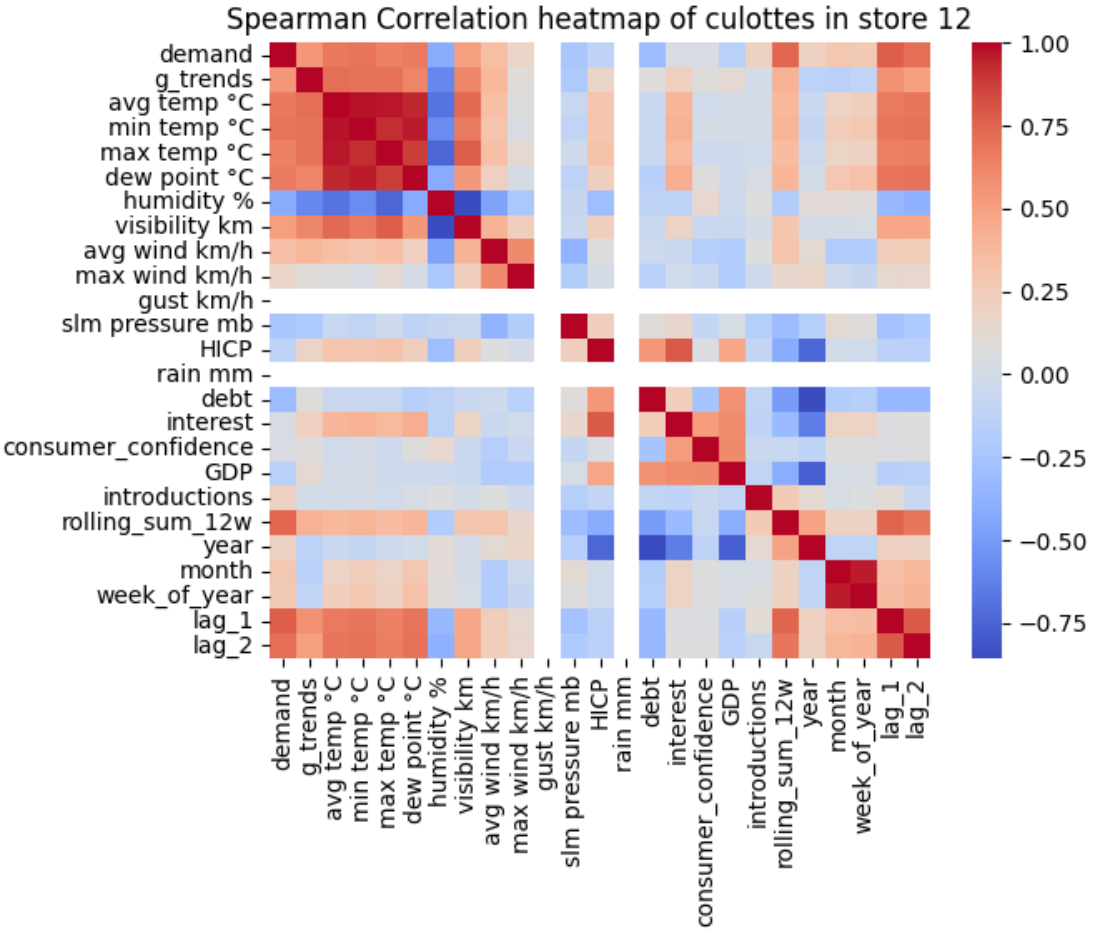


Figure 3.7: Spearman's Rank correlation for culottes in store 12

$$W = \frac{(\sum_{i=1}^{n} a_i x_{(i)})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3.3.3}$$

where $x_{(i)}$ are the ordered sample values, $\bar{x}$ is the sample mean, and a_i are constants derived from the covariance matrix of the ordered statistics under the null hypothesis of normality. When the P-value of a statistic is below 0.05, the null hypothesis of normality is rejected. Performing the Shapiro-Wilk test on all the features showed that only the *lag_1* and *lag_2* features possessed normality with p-values of 0.1652 and 0.3520 respectively, meaning Spearman's Rank coefficient is the most suitable way to perform a correlation analysis to prevent multicollinearity. For Spearman's Rank coefficient, the only assumption is that the data is ordinal. A widely used rule of thumb, is that a coefficient above 0.9 or below -0.9 indicates a high likelihood of collinearity within the data set [34]. Figure 3.7 shows a heat map of all correlations between the features of the dataset from store 12, pertaining culottes.

As can be observed, the gust and rain data is missing for store 12. These features are consequentially dropped. The four temperature variables *min temp*, *max_temp avg_temp* and *humidity* show high correlation (>0.9). Therefore, only average temperature was kept as a feature. Next, high positive correlation between *week_of_year* and *month* can be observed. However, since these date time features are cyclical and therefore not ordinal, this observation can be ignored and the features kept. Finally, a strong negative correlation between year and debt is observed. However, this correlation does not surpass the threshold, and thus both features are kept in the final dataset. A similar analysis was done on the remaining eight datasets, where all missing features, and features with correlation above 0.9 and below -0.9 were dropped. Coincidentally, the same features were dropped for each dataset, resulting in a similarly shaped data frame for each of the nine datasets.

3.3.6 Final data

In order to come to a final data frame that is ready to be fed to the models, all aforementioned features with weekly granularity were joined on the demand data using the *date* column as join key. For purposes of increased comparison accuracy and consistency, the choice was made to retain the same length of the data frame. Specifically, each of the nine data frames contain 147 rows, corresponding to weekly observations from 2017-01-30 to 2019-11-18. After feature selection using correlation analysis, each of the nine datasets contain the same 19 columns which correspond to the following features: *g_trends, avg temp °C, humidity %, visibility km, avg wind km/h, max wind km/h, slm pressure mb, HICP, debt, interest, consumer_confidence, GDP, introductions, rolling_sum_12w, year, month, week_of_year, lag_1, lag_2.*

Methodology

4.1 Algorithms

4.1.1 eXtreme Gradient Boosting

XGBoost is a gradient boosting algorithm developed in 2016 by Tianqi Chen and Carlos Guestrin (2016). XGBoost has emerged as a prominent and influential algorithm in ML, praised for its exceptional performance and versatility. It delivers state-of-the art results in multiple tasks, including regression, clustering and classification. Furthermore, regularization techniques such as L1 and L2 regularization are incorporated to reduce over fitting [9]. Although XGBoost is often the go-to algorithm for various ML tasks, it comes with various drawbacks. First of all, XGBoost is relatively sensitive to hyper-parameter tuning, compared to, for example, the RF model. While this increases flexibility, it may take more effort to optimize the model [22].

The name of XGboost draws on a procedure called boosting. Boosting is an ensemble learning method that combines various weak learners to create a stronger one. A weak learner, in the case of regression problems using XGB, is a regression tree. A regression tree uses a type of supervised algorithm that *"can be used to discover features and extract patterns in large databases that are important for discrimination and predictive modeling"* [27]. In Figure 4.1, an example can be found of how a regression tree would work in order to predict whether a certain person would be interested in video games. Scores are given to the output, based on the answers of the questions posed in the regression tree. These questions are learned by the regression tree algorithm while training, and are as discriminatory as possible, meaning that they would have the highest predictive power based on the training set.

On its own, a single regression tree is not powerful enough to make accurate predictions. Boosting is the process whereby multiple regression trees are formed, and their scores added to get a value that is more accurate than the output of a single regression tree (see Figure 4.2) [9].

More specifically, the predicted value is given by equation 4.1.1.

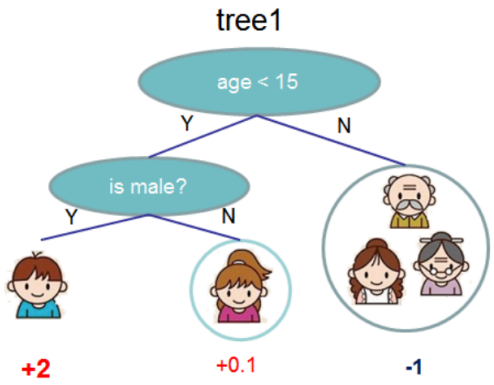


Figure 4.1: A regression tree used to predict whether a person would like video games. Figure from Chen et al. (2016) [9]

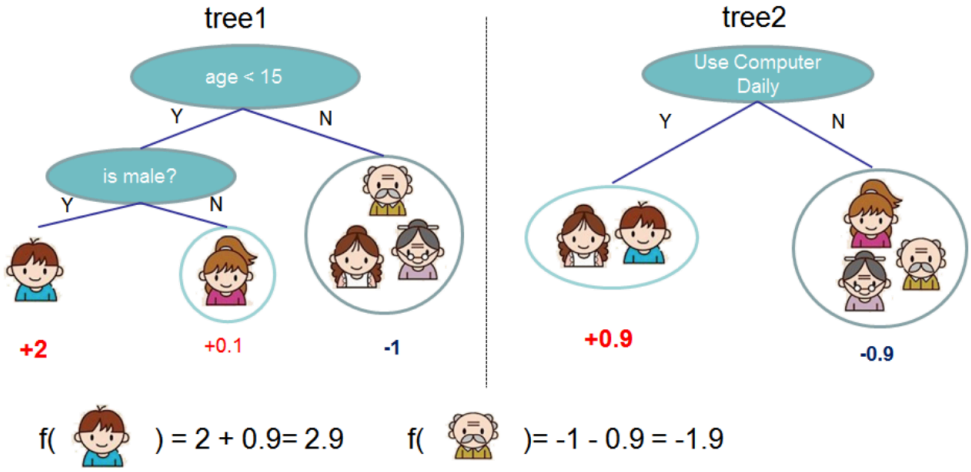


Figure 4.2: Boosting: combining multiple regression trees to obtain a more accurate prediction. Figure from [9]

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \quad f_k \in \mathcal{F} \tag{4.1.1}$$

Where K is the number of additive functions used, and F is the space of regression trees [9]. Using the training data, a first model is formed. Then, using gradient descent, a new model is formed for which the reduction in total error is maximized. This process is repeated until no improvement is possible, a maximum number of iterations is reached, or the decrease in loss for a new iteration is below a certain threshold. Specifically, the function that is minimized by the algorithm is given by equation 4.1.2.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y_i}^{(t-1)} + f_t\left(\mathbf{x}_i\right)\right) + \Omega\left(f_t\right) \tag{4.1.2}$$

Where l is the loss function that constitutes the error between the actual value y_i at instance i and the predicted value $\hat{y_i^{t-1}}$ at instance i and iteration $t-1$ in the optimization process. f_t is the tree function corresponding with with the $t-th$ tree.

4.1.2 Random Forest

In the literature review, numerous studies have been featured that use a RF algorithm to make a forecast. Cui et al. (2017) found that the RF model performed best when predicting sales using social media information [13]. Loureiro et al. (2018) also found RF to be the best performing model in their research on improving forecast accuracy or items that have little historical data [24].

The name Random Forest was first coined by Leo Breiman (2001) [4]. RF can be used for both classification and regression tasks. Contrary to XGBoost, RF is an ensemble model that uses bagging as opposed to boosting to improve its performance. The main idea of bagging is model averaging (see Figure 4.3). Bagging models train multiple base models in parallel instead of sequentially. In the case of this forecasting task, the algorithm does by creating multiple regression trees. Using the training data, these models are trained individually using bootstrapped data and are typically unconstrained, unless specified otherwise. Specifically, each tree is improved so that the squared loss given by equation 4.1.3 is minimized.

$$\min_{\hat{f}} \sum_{i=1}^{N} \left(y_i - \hat{f}\left(x_i\right)\right)^2 \tag{4.1.3}$$

where y_i is the actual value of the data point i, $\hat{f}(x_i)$ is the value obtained by using the tree $\hat{f}$ on the ith data point x. The function $\hat{f}$ is adapted in order to minimize the sum of squares.

After these separate trees are trained, predictions are made using the average answer of all trained trees [16]. Breiman (2001) demonstrated that Bagging may reduce an estimate's variance when compared to an estimator that uses only the original sample, which offers a means to increase a forecast's robustness [4].

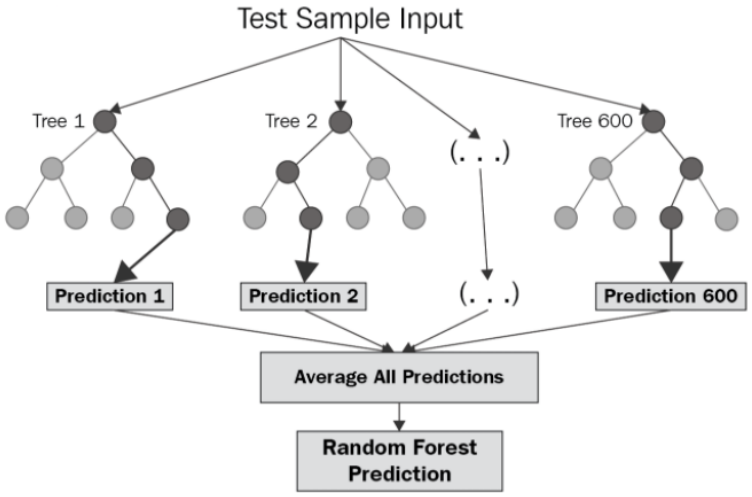


Figure 4.3: RF regressor using multiple decision trees to make a best guess

4.2 Performance metrics

In order to quantify the model's performance, a suitable metric should be used. Since we are dealing with a regression problem, we want to take a measure that calculates how close the predicted value was to the benchmark value. There are multiple performance metrics for this type of problem, each with its own advantages and drawbacks. If demand is high for a certain period, large mispredictions can be quite costly in terms of missed sales. Therefore, in addition to an absolute measure of deviation, a measure was chosen that penalizes larger errors more heavily. Finally, since we are dealing with multiple datasets that have different demand scales, comparison in absolute deviations will show difficult. Consequently, percentage variants of the aforementioned measures are introduced as well.

4.2.1 Mean Absolute Error (MAE)

The MAE is one of the most basic performance measures in regression problems. As the name suggests, the measure averages out all absolute errors between the prediction and the truth values. The MAE is easy to interpret and treats all errors equally, meaning larger error are not disproportionally penalized.

The equation for the MAE is given by:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.2.1}$$

Where: n is the number of data points, y_i is the actual value at the ith data point and $\hat{y}_i$ is the predicted value at the ith data point

4.2.2 Mean Absolute Percentage Error (MAPE)

The MAPE is a commonly used performance metric in forecasting and regression tasks. Similar to the MAE, it quantifies the average magnitude of errors between predicted values and actual values. However, the MAPE shows the final result as a percentage value, making for more intuitive comparison between regression problems that have a different order of magnitude. Because of its percentual nature, it does not weigh errors in peak seasons higher than errors in the low season, as the errors are measured as a percent of the total. However, a drawback is that when the absolute values are close to zero, the MAPE may give disproportionally high error values.
The formula for the MAPE is given by:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{4.2.2}$$

Where n is the number of data points, y_i is the actual value at the ith data point and $\hat{y}_i$ is the predicted value at the ith data point.

4.2.3 Root Mean Squared Error (RMSE)

The RMSE is similar to the aforementioned MAE. The only difference being that instead of taking the absolute errors between prediction and truth, the errors are squared, summed and averaged, after which the square root is taken to regain the original scale. Taking the square of the error values will result higher sensitivity to outliers, meaning larger errors having more impact on the overall RMSE value.
The formula for the RMSE is given by

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.2.3}$$

Where n is the number of data points, y_i is the actual value at the ith data point and $\hat{y}_i$ is the predicted value at the ith data point.

4.2.4 Root Mean Squared Percentage Error (RMSPE)

Another commonly used performance indicator in regression and forecasting applications is the RMSPE. The RMSPE is the percentual variant of the aforementioned RMSE, thereby being independent of scale of the target variable. It lends it self well for comparison among multiple datasets where larger errors are disproportionally penalized.
The formula for the RMSPE is given by:

$$RMSPE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2} \times 100\% \tag{4.2.4}$$

Where n is the number of data points, y_i is the actual value at the ith data point and $\hat{y}_i$ is the predicted value at the ith data point.
Whereas the MAPE lacks sensitivity, the RMSPE gives more weight to larger errors. This can be beneficial in problems where larger errors are indeed relatively more problematic to the final implementation.

4.3 Hyper parameter tuning

Hyper parameter tuning is a critical step in optimizing the performance of any ML model. While XGBoost and RF are inherently robust and flexible, their effectiveness relies heavily on setting the right hyper parameters. Among others, common methods to find optimal hyper parameter are manually through trial and error, grid search, randomized search, and Bayesian optimization. Grid search was chosen for both the XGBoost and RF model, because of optimal the trade-off between training time, effectiveness and ease of use [5]. In a grid search, a selection of hyper parameters is made, for which a number of possible values are manually predetermined. Next, an exhaustive search of all possible combinations is performed using a predetermined objective function, after which the optimal parameters are returned. For both XGBoost and RF, the default squared error metric is used to optimize the hyper parameters in combination with temporal cross-validation (see Section 4.4).

4.3.1 XGBoost

The XGBoostregressor class used in this research, has over 30 tunable hyper parameters. It is common procedure to take a subset of these parameters to tune, to prevent exceedingly long training times. Based on the book by Bartz et al., a subset of hyper parameters and possible

values was chosen [5]. The search space can be found in Table 4.1.

Parameter	Description	Possible Values
`n_estimators`	Number of boosting steps	50, 100, 200
`colsample_bytree`	Number of features that is chosen for the splits of a tree	0.6, 0.8, 1.0
`gamma`	Number of splits of a tree by assuming a minimal improvement for each split	0, 0.1, 0.2
`max_depth`	Maximum depth of a leaf in the decision trees	3, 5, 7
`min_child_weight`	Restriction of the number of splits of each tree	1, 3, 5

Table 4.1: Hyper parameters of XGBoost, their descriptions, and possible values for grid search

Using the GridSearchCV function from the sklearn package, optimal hyper parameters for each model were found and are displayed in Table 4.2.

Model	n_ estimators	colsample_bytree	gamma	max_ depth	min_ child_ weight	RMSE
culottes store 36	50	0.6	0	5	1	10.98
sleeveless store 36	50	0.8	0.2	3	1	8.15
long sleeve store 36	200	0.6	0.1	5	5	24.97
doll dress store 30	50	0.6	0.1	7	1	7.34
culottes store 30	100	1.0	0	5	5	12.85
long sleeve store 30	50	0.6	0.2	5	5	23.82
long sleeve store 12	50	0.6	0.1	3	1	41.03
doll dress store 12	200	0.8	0	5	1	17.37
culottes store 12	50	1.0	0	5	3	16.53

Table 4.2: Best parameters and RMSE scores for different models, XGBoost

Tuning the hyper parameters for all 9 XGBoost models took 01:20:11, and resulted in a reduction of the RMSE of 3.8%, compared to using the default parameters.

4.3.2 Random Forest

The tuning of the hyper parameters of the RF models follows a similar procedure. Following Bartz et al., the subset of tunable hyper parameters and their values are shown in Table 4.3 [5].

Parameter	Description	Possible Values
`n_estimators`	Number of trees in the forest	50, 100, 200, 500
`max_depth`	Maximum depth of each tree	None, 10, 20, 30, 40, 50
`min_samples_split`	Minimum number of samples required to split an internal node	2, 5, 10
`min_samples_leaf`	Minimum number of samples required to be at a leaf node	1, 2, 4
`max_features`	Number of features to consider when looking for the best split	'auto', 'sqrt', 'log2'

Table 4.3: Hyper parameters of RF Regressor, their descriptions, and possible values for grid search

After hyper parameter tuning using grid search, the optimal values were found and shown in Table 4.4.

Model	n_ estimators	max_ depth	max_ features	min_ samples_ leaf	min_ samples_ split	RMSE
culottes store 36	50	20	sqrt	1	5	11.4833
sleeveless store 36	100	50	log2	1	2	10.9219
long sleeve store 36	50	None	log2	1	2	12.8776
doll dress store 30	200	None	sqrt	1	2	9.2371
culottes store 30	50	None	sqrt	2	10	7.6206
long sleeve store 30	50	50	log2	1	2	11.6125
long sleeve store 12	50	None	log2	1	2	28.5560
doll dress store 12	50	50	log2	4	5	17.6858
culottes store 12	50	None	sqrt	1	10	9.1684

Table 4.4: Best parameters and RMSE Scores for different models, RF

Tuning the hyper parameters for all 9 RF models took 06:09:12, and resulted in an average reductions of the RMSE of 4.1%, compared to using the default parameters.

4.4 k-fold temporal cross-validation

A commonly used tactic to increase the performance of a predictive model, is to make use of cross-validation. Cross-validation means that instead of having a single predetermined train and test partition, the model is trained multiple times, whereby the train and test data differ each time. Following this procedure, the model becomes more generalizable. When performing regular cross-validation, the data is randomly split in train and test data. After the model has trained on train data and validated using the test data, a new random sample of train and test data is used.

However, since we are dealing with temporal data, regular cross-validation cannot be used. Taking random data to be test data, could mean that future values are used to predict past observations. This would make no sense, and will likely lead to over fitting. This problem is solved by putting restraints on the random slicing of the data, by making sure the temporal nature of the data is conserved. Figure 4.4 gives a visual representation of this process. Specifically, in the first iteration, the first 43 weeks are used as training data. Based on these observations, a prediction is made for the next two weeks. In the next iteration, these two weeks are added to the train data, resulting in a training set of 45 weeks. This process is repeated until the data runs out, resulting in exactly 104 weeks, or 2 years of predictions.

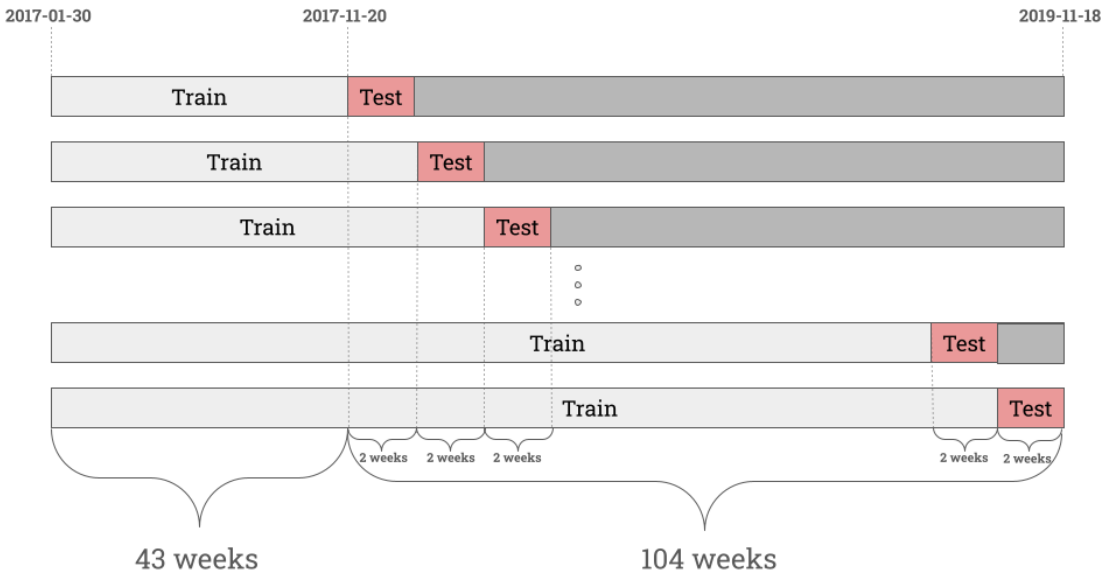


Figure 4.4: Visual representation of k-fold cross-validation. (k=52)

4.5 Cost analysis

In order to answer the research question, the model predictions should be translated to a cost-benefit analysis. To do so, a replenishment strategy has been formulated that will depict a fictional stock level for each week. Based on this weekly fictional stock level and cost assumptions, a conclusive decision on the benefit of using ML for forecasting can be made. The assumptions about the costs are as follows:

- Every store has a capacity to store items equal to the the average demand over the entire time period. In case the replenishment strategy based on the model's predictions exceeds this threshold, a fictional overstocking cost of 1 Euro per item, per week will be induced.
- Negative fictional stock levels are defined as lost sales: actual demand was higher than predicted. This will incur an opportunity cost of 10 Euros per item.
- Developing and implementing a stock replenishment system based on ML algorithms, is estimated to cost 7000 Euros: 2 months salary of a junior data scientist.

Together with the aforementioned cost assumptions, a restocking strategy based on the model's predictions and actual demand is formulated. The strategy that describes the stock level S in week i is given in equation 4.5.1.

$$S_i = \begin{cases} P_0 & \text{if } i = 0 \\ P_{i,2} - D_i & \text{if } i\%2 = 0 \text{ and } S_{i-1} < 0 \\ P_{i,2} - S_{i-1} - D_i & \text{if } i\%2 = 0 \text{ and } S_{i-1} \geq 0 \\ S_{i-1} - D_i & \text{if } i\%2 = 1 \end{cases} \tag{4.5.1}$$

Where:

S_i : Stock value at week i

P_i : Predicted demand for week i

D_i : Actual demand for week i

$P_{i,2}$: Predicted 2-week demand for week i (i.e., $P_i + P_{i+1}$)

In equation 4.5.1, we observe 4 different ways the stock level S in week i is calculated:

- At week 0, the stock level is assumed to be the predicted demand for the next week.
- In even weeks where last week's theoretical stock is negative (i.e., potential sales were missed), stock is replenished with the predicted 2 week demand ($P_{i,2}$). The actual demand for that week (D_i) is subtracted.
- In even weeks where last week's stock (S_{i-1}) was positive (i.e. we have excess stock), the replenishment amount ($P_{i,2}$) is reduced by last week's stock, after which demand for that week (D_i) is also subtracted.
- In odd weeks, the stock is equal to last week's stock minus actual demand in that week.

Using this replenishment strategy, a theoretical stock level for each week can be calculated, where negative stock indicates lost sales, and positive stock above the aforementioned threshold will induce storage costs.

Results

5.1 XGBoost

In Table 5.1, the results of the XGBoost model predictions are shown. While the MAE is the most intuitive to evaluate a model's performance, different data scales make it a challenging metric for comparison. Therefore, percentage metrics MAPE and RMSPE were introduced. Recall that MAPE penalizes errors linearly, whereas RMSPE assigns a higher relative penalty to larger deviations from the actual values.

Model	**MAE**	**MAPE (%)**	**RMSE**	**RMSPE (%)**
culottes store 36	15.06	0.25	19.44	0.33
sleeveless store 36	7.53	0.22	9.43	0.30
long sleeve store 36	15.45	0.27	19.69	0.35
doll dress store 30	10.42	0.22	14.10	0.30
culottes store 30	14.01	0.24	18.83	0.37
long sleeve store 30	14.22	0.22	22.66	0.33
long sleeve store 12	24.20	0.28	35.89	0.38
doll dress store 12	18.32	**0.20**	24.90	**0.27**
culottes store 12	22.86	0.30	32.63	0.44

Table 5.1: Performance metrics, XGBoost

We observe that for the doll dresses in store 12, the XGBoost model performs best with a MAPE of 0.20 and RMSPE of 0.27. On the long sleeve data of store 12, the model performed worst with a MAPE of 0.28 and RMSPE of 0.38.

In Figure 5.1, a visual representation of the actual demand and the predicted values is shown. We observe that the model catches the variability in demand well, but predictions lag actual demand. Furthermore, demand peaks are more likely to be under- than overpredicted.

In Figure 5.2, the actual and predicted demand of the worst performing model, culottes in store 12 is depicted. While the demand peaks in the summer of 2018 and 2019 are predicted by the model, the peak in 2018 is predicted too early, and the the peak in 2019 too late. These outliers with strong variability are likely the reason for the high MAPE and RMSPE scores.

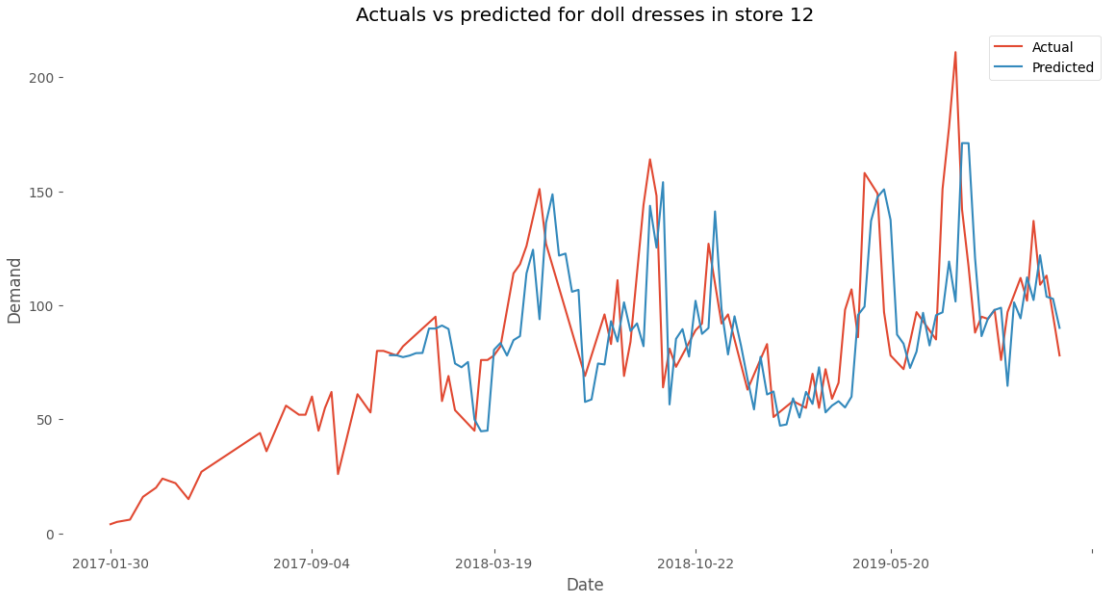


Figure 5.1: Actual vs. predicted values for doll dresses in store 12, XGBoost

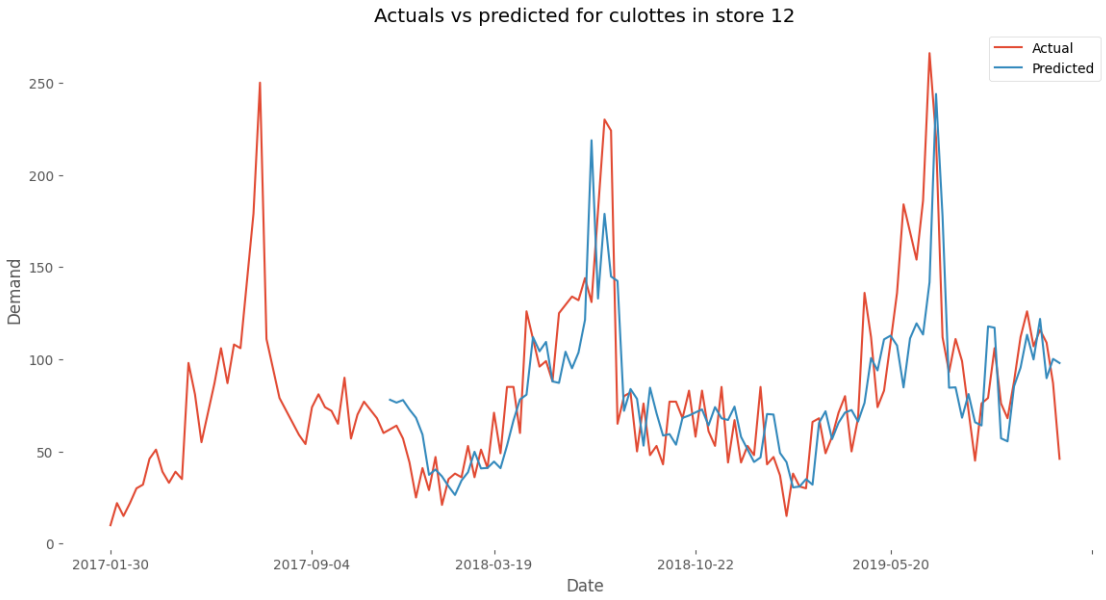
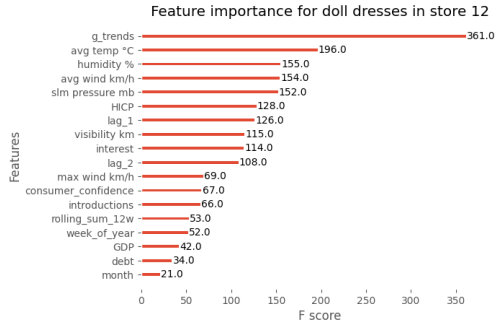


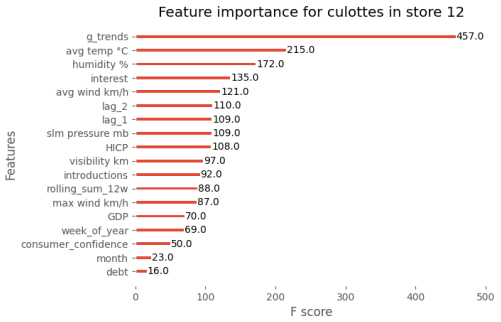
Figure 5.2: Actual vs. predicted values for culottes in store 12, XGBoost

Feature importance: Next, feature importance values for the models' independent variables were calculated. This is represented by the F-score. The F score tells us how many times a decision tree is split based on the value of a feature. The feature importance plot for doll dresses in store 12 is shown in Figure 5.3a. We observe *g_trends* to be the most effective predictor of demand, followed by *avg_temp* and *humidity*. According to the plot of feature importance, online popularity and weather are the most valuable factors in predicting demand. Economic indicators appear to have little predictive value. Especially *debt*, *GDP* and *consumer_confidence* are among the least contributing features.

For the model with the worst performance, the importance plot of features can be found in Figure 5.3b. Again, we observe *g_trends* to be the most predictive feature, followed by *avg_temp* and *humidity*. Notably, *interest* scores relatively high in this model, while other economic indicators are still lacking predictive power.



(a) Feature importance for doll dresses in store 12, XGBoost



(b) Feature importance for culottes store 12, XGBoost

Figure 5.3: Feature importance comparison, XGBoost

5.2 Random Forest

The numerical results for the RF models are shown in Table 5.2. Similarly to XGBoost, the model performs best on the doll dresses in store 12 with a MAPE of 0.20 and RMSPE 0.26, while performing worst on the culottes in store 12, with a MAPE of 0.27 and RMSPE of 0.41. The comparatively high RMSPE indicates that when mis-predictions are made, they are relatively large.

Model	**MAE**	**MAPE (%)**	**RMSE**	**RMSPE (%)**
culottes store 36	14.15	0.23	18.88	0.32
sleeveless store 36	6.59	0.20	8.82	0.30
long sleeve store 36	13.31	0.23	17.39	0.32
doll dress store 30	10.09	0.21	13.99	0.28
culottes store 30	13.91	0.25	19.19	0.40
long sleeve store 30	13.09	0.21	18.71	0.29
long sleeve store 12	21.38	0.25	30.61	0.32
doll dress store 12	18.73	**0.20**	24.40	**0.26**
culottes store 12	19.08	0.27	27.04	0.41

Table 5.2: Performance metrics, RF

In Figure 5.4, the actual vs predicted values for the doll dresses in store 12 are shown. We observe that the model is able to catch spikes in demand fairly accurately but is often one or two weeks too late with predicting demand. A likely reason for this can be that the model uses the *lag_1* feature as a strong predictor for demand.

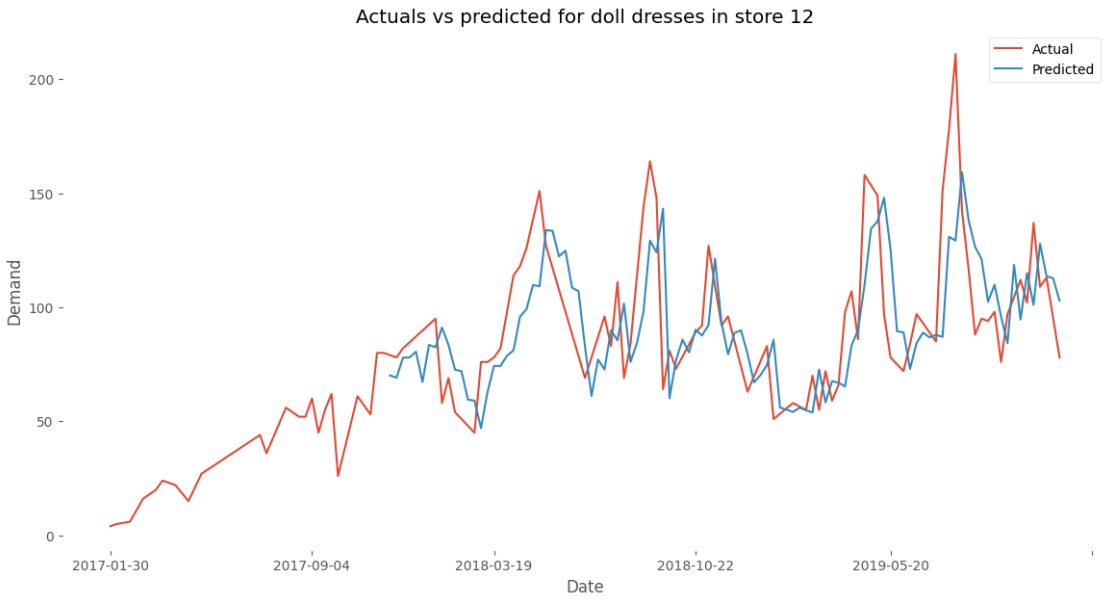


Figure 5.4: Actual vs. predicted values for doll dresses in store 12, RF

The actual vs predicted values for culottes in store 12 are shown in Figure 5.5. We observe sharp demand peaks in the summer of 2018 and 2019. Although the model anticipates these peaks, they are both under-predicted. While the variability of the demand is predicted accurately, we observe a recurring problem where sudden demand predictions lag actual demand.

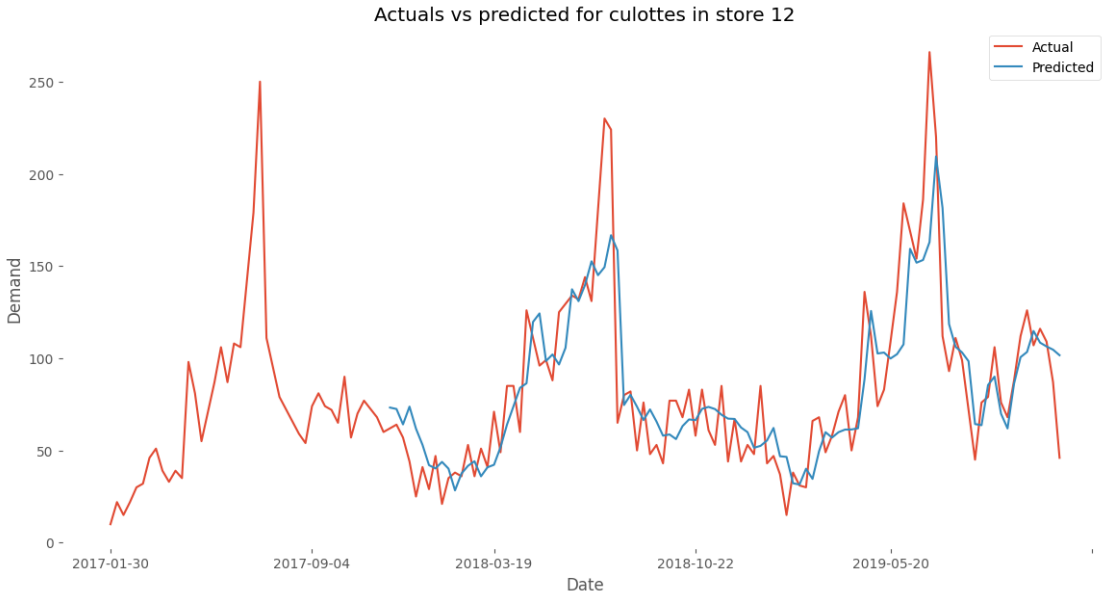


Figure 5.5: Actual vs. predicted values for culottes in store 12, RF

Feature importance: Feature importance for the RF models is computed in a different way compared to XGBoost, due to the nature of the algorithm. The feature importance values lie between 0 and 1 and represent the average reduction in impurity -for our regression problem, this means the MSE- that a feature brings across all trees in the forest.
Specifically, the reduction in impurity (ΔI) for a feature f at a particular node in a decision tree is given by:

$$\Delta I = I_{\text{parent}} - \left(\frac{N_{\text{left}}}{N_{\text{total}}} I_{\text{left}} + \frac{N_{\text{right}}}{N_{\text{total}}} I_{\text{right}} \right) \tag{5.2.1}$$

where:

- I denotes the impurity measure (e.g., mean squared error (MSE) for regression).
- N_{left} is the number of samples in the left child node.
- N_{right} is the number of samples in the right child node.
- N_{total} is the number of samples in the parent node.

For each feature f, the total reduction in impurity is summed across all nodes where f is used to make a split, over all trees in the forest. This reduction in impurity is given by equation 5.2.2.

$$\text{Importance}(f) = \sum_{t=1}^{T} \sum_{\text{nodes using } f \text{ in tree } t} \Delta I_t \tag{5.2.2}$$

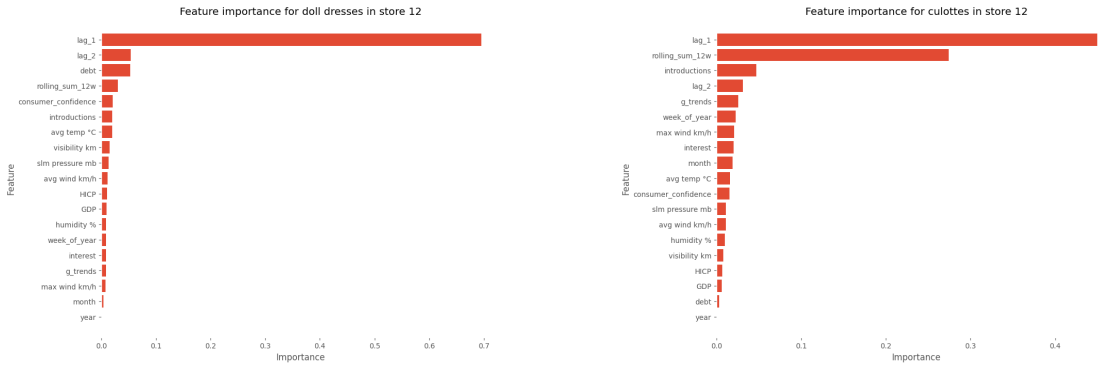
where T is the total number of trees in the forest.

The feature importances are then normalized so that they sum to 1:

$$\text{Normalized Importance}(f) = \frac{\text{Importance}(f)}{\sum_{\text{all features}} \text{Importance}(f)} \tag{5.2.3}$$

This normalized importance gives the relative importance of each feature, which can be interpreted as the percentage contribution of each feature to the model's predictions.

The importance of features for the best performing model is shown in Figure 5.6a. The *lag_1* feature possesses the most predictive power. This explains the observation found in Figure 5.4 where we saw that demand variability was predicted accurately but shifted to the future. Notably - while having low absolute predictive power - the *debt* feature has high relative predictive power compared to other features. While *g_trends* appeared to be the most powerful predictor in the XGBoost models, it scores low in the RF models. Although feature importances are calculated differently for XGboost and RF, the discrepancy is still noteworthy. In the worst-performing model, different patterns are observed (see Figure 5.6b). While *lag_1* is still the most valuable feature, the features *rolling_sum_12w* and *introductions* have considerable relative importance. The *g_trends* feature scores low in this model as well, empowering the hypothesis that discrepancies in feature importances between the XGboost models and RF models are likely due to different calculation procedures.



(a) Feature importance doll dress, store 12, RF

(b) Feature importance culottes, store 12, RF

Figure 5.6: Feature importance comparison, RF

5.3 Model comparison

In order to compare model performance over the same datasets, the performance measures for both models are juxtaposed in Table 5.3.

Model	**MAE (XGB)**	**MAE (RF)**	**RMSE (XGB)**	**RMSE (RF)**
culottes store 36	15.06	**14.15**	19.44	**18.88**
sleeveless store 36	7.53	**6.59**	9.43	**8.82**
long sleeve store 36	15.45	**13.31**	19.69	**17.39**
doll dress store 30	10.42	**10.09**	14.10	**13.99**
culottes store 30	14.01	**13.91**	**18.83**	19.19
long sleeve store 30	14.22	**13.09**	22.66	**18.71**
long sleeve store 12	24.20	**21.38**	35.89	**30.61**
doll dress store 12	**18.32**	18.73	24.90	**24.40**
culottes store 12	22.86	**19.08**	32.63	**27.04**

Table 5.3: Performance metrics comparison between XGBoost and RF

We observe that in terms of MAE, the RF model outperforms XGBoost on all datasets, except for the doll dresses in store 12. In terms of RMSE, the RF models outperform XGBoost on all data sets except for the culottes in store 30. Overall, we can conclude that the RF models are superior to the XGBoost models, and will hence be used to evaluate the added business value of using ML algorithms in forecasting fashion demand.

5.4 Cost analysis

Based on the replenishment strategy and cost assumptions mentioned in Section 4.5, theoretical stock values for each of the nine models were produced. A visual representation of these stock levels is show in Figure 5.7.

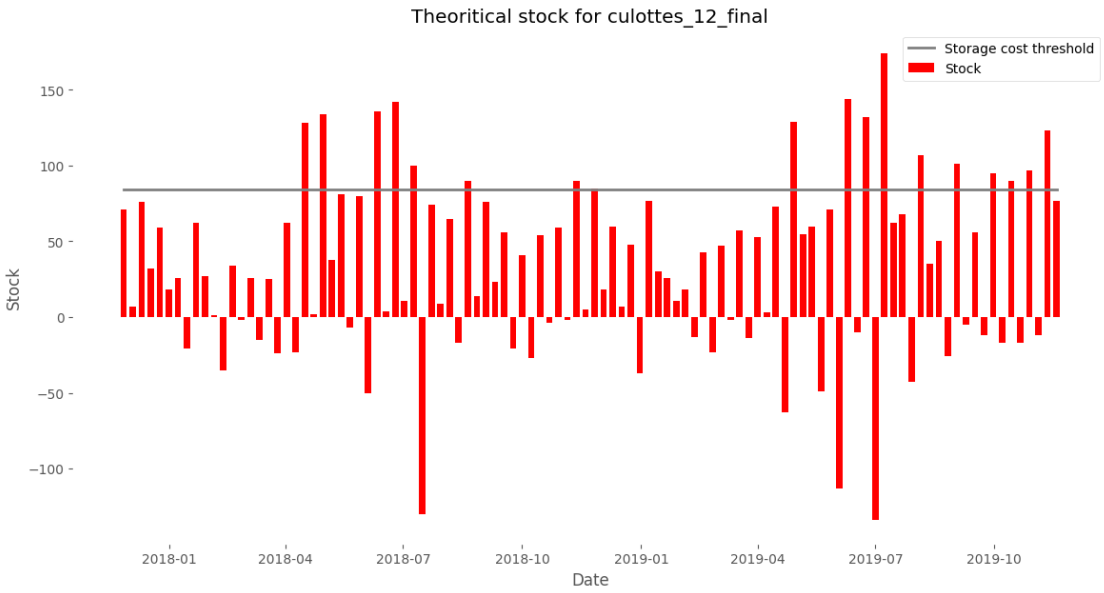


Figure 5.7: Theoretical stock values, culottes, store 12

We observe a high number of lost sales around the summer of 2018 and 2019. This is in line with the results of the prediction model in Figure 5.5, where we saw that the peaks of high demand were under predicted. The gray line represents the average demand over the entire time period and resembles the assumed threshold for storage costs: any stock above this threshold will induce storage costs. We observe multiple moments where stock is above the threshold level. These peaks are especially observed around the summer months, where demand variability is highest. Given the cost assumptions mentioned in Section 4.5 and the quantity of over- and understocks, the total cost for the nine models can be calculated according to equation 5.4.1.

$$C_{total} = N_{understocks} * C_{understocking} + N_{overstocks} * C_{overstocking} + C_{implementation} \quad (5.4.1)$$

In Table 5.4,the total number of over- and under-predictions for each model can be observed. In addition, percentage-based metrics have been added to account for differences in scale. In terms of under-predictions, the model on the culottes dataset in store 36 performs best with 5.94% of total demand being under-predicted. This is in line with the results from the forecasting models,

where the RF performed best on the culottes dataset in store 36 as well. The worst performance in terms of under predictions was measured on the long sleeve data in store 12, with 7.29% of total demand being missed.

In terms of over predictions, the best performance was measured on sleeveless data in store 36. The model makes the least over predictions at 12.15% of total demand. Percentage-wise, most over-predictions were made on the long sleeve data in store 12 at 13.60% of total demand. Notably, the percentage of over predictions for the culottes data in store 36 is relatively high at 13.17%, while this model scored best in terms of under predictions and forecasting.

Model	**Over Predictions**	**Under Predictions**	**Under Predictions (%)**	**Over Predictions (%)**
culottes store 36	960	433	**5.94%**	13.17%
sleeveless store 36	393	304	6.62%	**12.15%**
long sleeve store 36	900	419	6.52%	12.70%
doll dress store 30	623	250	6.13%	12.54%
culottes store 30	904	692	6.96%	12.55%
long sleeve store 30	810	533	7.11%	12.40%
long sleeve store 12	1630	704	7.29%	13.60%
doll dress store 12	1234	472	6.86%	13.44%
culottes store 12	991	617	6.89%	13.15%

Table 5.4: Total number of over- and under predictions per model

Filling in the formula with the total number of over- and under stocks over all nice models, we get the a total cost of 59,285 as defined by Equation 5.4.2.

$$C_{total} = 4,403 * 10 + 8,255 * 1 + 7000 = 59,285 \tag{5.4.2}$$

Finally, in Section 1.3, it was established that total costs may not exceed 5% of total revenue over the 2-year prediction period. Using the assumed average retail price of 10 Euro per item, the 5% threshold can easily be calculated using the total actual demand over said period. This calculation is shown in Equation 5.4.3.

$$Cost\_threshold = Total\_demand * Item\_price * Revenue\_threshold(\%) = 64,231 * 10 * 0.05 = 32,115.5 \tag{5.4.3}$$

Conclusion

In this section, the three sub-questions and research questions are answered on the basis of the findings of this research.

6.1 Sub-question 1

The first sub-question posed is: *Which ML algorithms perform best in forecasting sales?* In the literature review, multiple algorithms were highlighted and their performance discussed. ELMs, LSTMs, Prophet, XGboost, and RF are among the algorithms most commonly used for these types of forecasting problems. In this research, further inquiries were made about the effectiveness of XGBoost and RF. In terms of both MAE and RMSE, the RF model outperformed XGboost on eight out of nine datasets, concluding that RF is the superior algorithm in this specific case (see Table 5.3).

6.2 Sub-question 2

The second subqestion posed is: *What kind of external data can have a positive impact on the accuracy of clothing sales forecasts?* In the literature review, external data such as weather, promotions, pricing, holidays, and macroeconomic indicators were found to be effective in decreasing forecast errors. In this research, weather data, Google trends data, macroeconomic indicators, and derived sales features such as lag and rolling sum were added to test their effectiveness in increasing forecast accuracy. The extent to which features were effective, depended heavily on the choice of algorithm.

XGBoost For all nine datasets, the XGBoost model found the *g_trends* feature to be most important, i.e. most splits on the decision were made using that feature. *Humidity* and *avg temp* alternated third and second place for the 9 datasets. Macroeconomic indicators were often found

to have the lowest predictive value: especially debt and interest were often among the three lowest scoring variables.

Random Forest In eight out of nine datasets, the RF model deemed the *lag_1* feature to have the most predictive power. Only in the dataset for sleeveless items in store 36, average temperature scored highest. The *rolling_sum_12w* feature scored second place five out of nine times. Although the feature *g_trends* scored highest in all nine datasets with the XGBoost models, this feature scored second place only once with the RF models, while being of negligible importance in all 8 other datasets.

6.3 Sub-question 3

The third sub question posed is: *How well do ML forecasts approach real demand?* To answer this question, two percentage-based performance metrics, namely MAPE and RMSPE were introduced and used to evaluate and compare the performance of the models. The values are juxtaposed with the findings of Steinker et al. (2017), who have done extensive research in the benefit of adding weather variables to improve forecasting performance [36]. Since they used the MAPE as a performance measure, we can form an opinion about the found values for the performance measures in this research. Although their forecasts were limited to a daily time frame and extended no more than one week in advance, it is nevertheless possible to determine whether the findings of this research are comparable to those of similar studies. MAPE values found in the study of Steinker et al. ranged from 0.07 to 0.09, depending on the forecast window.

XGBoost For all nince XGBoost models, the RMSPE values ranged from 0.27 to 0.44, where the best performance was achieved on the demand of doll dresses in store 12, and the worst performance on the demand for culottes in store 12. The overall average of the RMSPE was 0.341. The MAPE ranged from 0.20 to 0.30, and the same datasets provided the best and worst performance, respectively. The overall average MAPE was 0.244. This means that overall, the prediction error is 24.4%. Given the relatively large discrepancy between the values found in this investigation and similar studies, we can conclude that for the XGBoost models, the forecasts using ML do not accurately approximate the actual demand.

RF In terms of MAPE and RMSPE, the RF models showed better performance than the XGBoost models. RMSPE values ranged from 0.26 to 0.41, with an overall average RMSPE of 0.322. Similarly to XGBoost, the best performance was achieved on the demand of doll dresses in store

12, whereas the worst performance was found on the demand of culottes in store 12. The MAPE ranged from 0.20 to 0.27, with an overall average MAPE of 0.228. Contrasting this to results found in the research of Steinker et al., we can conclude that for the RF models, forecasts using ML do not accurately approximate actual demand.

6.4 Research question

The main research question posed in this study is *Can ML algorithms in the field of fashion sales forecasting be of added business value to a fast-fashion retailer?*.

This research was conducted on publicly available sales data of the Italian fast-fashion retailer Nunalie. By selecting two algorithms commonly used in related literature (XGBoost and Random Forest), 9 forecasting models per algorithm were created by aggregating SKU-level sales data on category level for the three most selling items in the three most popular stores. By performing hyper-parameter tuning using grid search and k-fold cross-validation, the performance of the models was optimized as much as possible. Subsequently, a stock replenishment strategy based on the 2-week prediction of the best-performing models (RF) was established. Based on this replenishment strategy, a weekly theoretical stock level could be formed. Together with the formulated costs assumptions, a final cost calculation was performed to answer the research question. At the end of Section 5.4, the total costs of implementation, overstocking and lost sales due to over- and under-predictions was €59,285. 5% of the revenue over the 2 year prediction period was calculated to be €32,115.5. Since the costs exceed the preestablished threshold by a substantial margin, we can conclude that, based on the cost assumption made, ML algorithms in the field of fast fashion sales are of no added business value to a fast-fashion retailer.

Discussion

This section delves into the limitations of current research, offers explanations for the results obtained, and provides recommendations for future studies.

7.1 Costs and replenishment

Firstly, the drawn conclusion that ML forecasts are of no added business value, is heavily contingent on a variety of assumptions. The most predominant being the cost assumptions about storage, lost sales, implementation costs. Since actual information on these costs was not attainable, they were arbitrarily chosen. Thus, while a conclusion was drawn, the main aim of this study has been to provide a framework for assessing this question in a practical scenario, rather than definitively answering it for the company in question. In a situation where these costs are known, the drawn conclusion would be more applicable.

Secondly, costs are also related to the replenishment strategy. As shown in Equation 4.5.1, every odd week the stock is replenished by the expected demand for the next two weeks. Given the fact that storage costs per item are lower than opportunity costs due to lost sales, experimentation with a replenishment factor $a > 1$ can be done to increase the replenishment amount, and thereby reduce opportunity costs. Using the known cost function and an exhaustive search, an optimal value for a can be found. The proposed replenishment function is shown in Equation 7.1.1 (see Section 4.5 for variable references).

$$S_i = a * P_{i,2} - D_i \tag{7.1.1}$$

7.2 Data

Since the raw data contained information about more than 100 stores and more than 5,000 individual items, a subset of items and stores was taken for practical purposes. Moreover, since every individual item has a shelf life of only twelve weeks, an aggregation was performed on the category level to generate more historical data. While this aggregation is useful to better find longer-term patterns in demand, critical information might be lost. For example, a certain color of long sleeve might be trendy in a given year, while being out of fashion in the next year. It is imperative to be aware of this information loss when applying the framework provided in this research in a practical context.

Besides the information loss due to aggregation of the sales data, there are other notes to be made about the input data. While research of Sageart et al. (2018) found and increase of 18.8% in prediction accuracy when adding economic indicators to the model, feature importance plots in this research demonstrated that adding these features was of little to no impact on prediction accuracy [31]. This might be for two reasons: First, since the raw data was available in either monthly or bi-yearly frequency, linear interpolation was used to obtain weekly values. Intramonthly trends are therefore not depicted in the data. Especially when forecasting 2 weeks in advance, it is unlikely that these features contain predictive power. Investigation into higher frequency raw data, or more advanced interpolation methods is advised. Secondly, while these variables may contain long-term trend information that is related to the data, the plots in Figure 3.6 show little variability over the 2 year period. Economic indicators are hypothesized to be more valuable when predictions are made over a longer time.

7.3 Model performance

Concerning model performance, various improvements can be made in further research. Hyper parameter tuning, especially using grid search, is a computationally expensive procedure where the time complexity increases exponentially with every possible parameter value. Therefore, a subset of parameters and values was chosen based on related research. Given more time or computing power, a wider range of hyper parameters could be explored.

Another noteworthy observation, is the large discrepancy between feature importance for the XGboost and RF model. Whereas Google trends and weather data were of the most importance for the XGboost models, the RF models gained the most predictive power from generated sales features such as the lag and rolling sum. This might be explained due to the different nature

in which feature importance was calculated for the different algorithms, but is nevertheless an interesting finding that warrants further research. Since no manual benchmark forecasts were available, it was challenging to thoroughly evaluate the effectiveness of the XGBoost and RF models. Therefore, recommendations for further research include adding simpler models like exponential smoothing or SARIMA - which are often mentioned in relevant literature - to better evaluate to what extent advanced machine learning models and exogenous variables may contribute to forecast accuracy.

Since the data was aggregated on a weekly level and the input data frames were relatively small, no bottlenecks in terms of computation time were encountered. However, in a setting where more historic data or smaller time intervals are used, such limitations can be faced. To address these potential issues, it is recommended to use dimensionality reduction techniques such as L1 regularization or Principal Component Analysis (PCA). These methods ensure shorter computation times while maintaining model performance.

Next, for further research it is recommended to take price, promotion and social media data into consideration. Multiple studies have highlighted the importance of adding this information to the models [14], [37], [13]. Since these data were not available, they were omitted in this research. Finally, for further research it is recommended to examine a wider variety of algorithms. In the literature review, Prophet, LSTM's and ELM's were often mentioned and showed good performance.

Bibliography

[1] Ö. Ali et al. "SKU demand forecasting in the presence of promotions". In: *Expert Systems With Applications* 36.10 (Dec. 2009), pp. 12340–12348.

[2] A. Alin. "Multicollinearity". In: *Wiley interdisciplinary reviews. Computational statistics* 2.3 (May 2010), pp. 370–374.

[3] G. Allenby, L. Jen, and R. Leone. "Economic Trends and being Trendy: The influence of consumer confidence on retail fashion sales". In: *Journal of Business Economic Statistics* 14.1 (Jan. 1996), pp. 103–111.

[4] Leo B. "Random Forests". In: *Machine Learning* 45 (2001), pp. 5–32.

[5] E. Bartz et al. *Hyperparameter Tuning for Machine and Deep Learning with R*. Springer, Jan. 2023.

[6] S. Beheshti-Kashi et al. "A survey on retail sales forecasting and prediction in fashion markets". In: *Systems Science & Control Engineering* 3.1 (Jan. 2015), pp. 154–161.

[7] M. Carabantes. "Black-box artificial intelligence: an epistemological and critical analysis". In: *AI society* 35.2 (Apr. 2019), pp. 309–317.

[8] C. Catal et al. "Benchmarking of regression algorithms and time series analysis techniques for sales forecasting". In: *Balkan journal of electrical computer engineering* (Jan. 2019), pp. 20–26.

[9] T. Chen and C. Guestrin. "XGBoost: A Scalable Tree Boosting System". In: Mar. 2016.

[10] *Composite Leading Indicators: Composite Consumer Confidence Amplitude Adjusted for Italy*. Apr. 2024. URL: `https://fred.stlouisfed.org/series/CSCICP03ITM665S`.

[11] *Composite Leading Indicators: Reference Series (GDP) Normalized for Italy*. Apr. 2024. URL: `https://fred.stlouisfed.org/series/ITALORSGPNOSTSAM`.

[12] *Consumer Price Indices (CPIs, HICPs), COICOP 1999: Consumer Price Index: Total for Italy*. May 2024. URL: `https://fred.stlouisfed.org/series/ITACPALTT01CTGYM`.

[13] R. Cui et al. “The Operational Value of Social Media Information”. In: *Production and Operations Management* 27.10 (Apr. 2017), pp. 1749–1769.

[14] X. Dairu and Z. Shilong. “Machine Learning Model for Sales Forecasting by Using XGBoost”. In: *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE 2021)* (Jan. 2021).

[15] Y. Ensafi et al. “Time-series forecasting of seasonal items sales using machine learning – A comparative analysis”. In: *International journal of information management data insights* 2.1 (Apr. 2022), p. 100058.

[16] P Fuleky. *Macroeconomic Forecasting in the Era of Big Data*. Springer, Jan. 2020, pp. 389–431.

[17] Z.X. Guo, W.K. Wong, and M. Li. “A multivariate intelligent decision-making model for retail sales forecasting”. In: *Decision support systems* 55.1 (Apr. 2013), pp. 247–255.

[18] M. Gurnani et al. “Forecasting of sales by using fusion of machine learning techniques”. In: *2017 International Conference on Data Management, Analytics and Innovatio* (Feb. 2017).

[19] *Household Debt to GDP for Italy*. Dec. 2023. URL: `https://fred.stlouisfed.org/series/HDTGPDITQ163N`.

[20] *Interest Rates: Long-Term Government Bond Yields: 10-Year: Main (Including Benchmark) for Italy*. May 2024. URL: `https://fred.stlouisfed.org/series/IRLTLT01ITM156N`.

[21] S. Ji et al. “An Application of a Three-Stage XGBoost-Based Model to Sales Forecasting of a Cross-Border E-Commerce Enterprise”. In: *Mathematical Problems in Engineering* 2019 (Sept. 2019), pp. 1–15.

[22] M. Joharestani et al. “PM2.5 Prediction Based on Random Forest, XGBoost, and Deep Learning Using Multisource Remote Sensing Data”. In: *Atmosphere* 10.7 (July 2019), p. 373.

[23] A. López-Baucells, T. Choi, and Y. Yu. “Fashion retail forecasting by evolutionary neural networks”. In: *International Journal of Production Economics* 114.2 (Aug. 2008), pp. 615–630.

[24] A. L. Loureiro, V. Miguéis, and L. Da Silva. “Exploring the use of deep neural networks for sales forecasting in fashion retail”. In: *Decision Support Systems* 114 (Oct. 2018), pp. 81–93.

[25] J. Lv, S. Han, and J. Hu. “Clothing Sales Forecast Considering Weather Information: An Empirical Study in Brick-and-Mortar Stores by Machine-Learning”. In: *Journal of textile science and technology* 09.01 (Jan. 2023), pp. 1–19.

[26] S. Ma, R. Fildes, and T. Huang. “Demand forecasting with high dimensional data: The case of SKU retail sales forecasting with intra- and inter-category promotional information”. In: *European Journal of Operational Research* 249.1 (Feb. 2016), pp. 245–257.

[27] A. Myles et al. “An introduction to decision tree modeling”. In: *Journal of Chemometrics* 18.6 (June 2004), pp. 275–285.

[28] D. Nair and M. Saenz. *Pair People and AI for Better Product Demand Forecasting*. Jan. 2024. URL: https://sloanreview.mit.edu/article/pair-people-and-ai-for-better-product-demand-forecasting/.

[29] P. Pardalos and Ann E. Koehler. “Another look at measures of forecast accuracy”. In: *International Journal of Forecasting* 22.4 (Oct. 2006), pp. 679–688.

[30] N. Rose and L. Dolega. “It’s the Weather: Quantifying the Impact of Weather on Retail Sales”. In: *Applied Spatial Analysis and Policy* 15.1 (Aug. 2021), pp. 189–214.

[31] Y. Sagaert et al. “Tactical sales forecasting using a very large set of macroeconomic indicators”. In: *European journal of operational research* 264.2 (Jan. 2018), pp. 558–569.

[32] Ben Letham Sean J. Taylor. *Prophet: forecasting at scale*. Feb. 2017. URL: https://research.facebook.com/blog/2017/2/prophet-forecasting-at-scale/.

[33] A. Şen. “The US fashion industry: A supply chain review”. In: *International Journal of Production Economics* 114.2 (Aug. 2008), pp. 571–593.

[34] N. Shrestha. “Detecting multicollinearity in regression analysis”. In: *American journal of applied mathematics and statistics* 8.2 (June 2020), pp. 39–42.

[35] G. Skenderi et al. “The Multi-Modal Universe of Fast-Fashion: The Visuelle 2.0 Benchmark”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2022, pp. 2241–2246.

[36] S. Steinker, K. Hoberg, and U. Thonemann. “The Value of Weather Information for E-Commerce Operations”. In: *Production and Operations Management* 26.10 (Oct. 2017), pp. 1854–1874.

[37] Z. Sun et al. “Sales forecasting using extreme learning machine with applications in fashion retailing”. In: *Decision Support Systems* 46.1 (Dec. 2008), pp. 411–419.

[38] S. Thomassey. “Sales forecasts in clothing industry: The key success factor of the supply chain management”. In: *International Journal of Production Economics* 128.2 (Dec. 2010), pp. 470–483.

[39] Juan R. Trapero, N. Kourentzes, and R. Fildes. “On the identification of sales forecasting models in the presence of promotions”. In: *Journal of the Operational Research Society* 66.2 (Feb. 2015), pp. 299–307.

[40] N. Vairagade et al. “Demand Forecasting Using Random Forest and Artificial Neural Network for Supply Chain Management”. In: *Lecture notes in computer science* (Jan. 2019), pp. 328–339.

[41] M. Verleysen and D. François. *The Curse of Dimensionality in Data Mining and Time Series Prediction*. Springer Science+Business Media, Jan. 2005, pp. 758–770.

[42] W. C. Wong and Z. J. Guo. “A hybrid intelligent model for medium-term sales forecasting in fashion retail supply chains using extreme learning machine and harmony search algorithm”. In: *International Journal of Production Economics* 128.2 (Dec. 2010), pp. 614–624.

[43] Y. Xue. “An Overview of Overfitting and its Solutions”. In: *Journal of physics. Conference series* 1168 (Feb. 2019), p. 022022.

[44] L. Zhang et al. “Time series forecast of sales volume based on XGBoost”. In: *Journal of physics* 1873.1 (Apr. 2021), p. 012067.

[45] E. Zunic et al. “Application of Facebook’s Prophet algorithm for successful sales forecasting based on real-world data”. In: *International Journal of Computer Science and Information Technology* 12.2 (Apr. 2020), pp. 23–36.