Tuomas Niemelä

Demand forecasting in the retail environment

A comparative study of LightGBM, XGBoost, and MLP models

School of Technology and Innovations
Master of Science in Economics and Business Administration
Industrial Management

Vaasa 2025

UNIVERSITY OF VAASA
School of Technology and Innovations

Author:	Tuomas Niemelä		
Title of the thesis:	Demand forecasting in the retail environment: A comparative study of LightGBM, XGBoost, and MLP models		
Degree:	Master of Science in Economics and Business Administration		
Degree Programme:	Industrial Management		
Supervisor:	Petri Helo		
Year:	2025	**Pages:**	103

ABSTRACT:

Accurate demand forecasting is a critical operational factor in the retail environment, as organizational decision-making and management are increasingly dependent on it. Accurate forecasts enable strategic planning, inventory optimization, increased customer satisfaction, and reduction of surplus and waste. While advanced machine learning (ML) models are recognized for producing accurate forecasts, current literature often focuses on comparing algorithmic efficiency without sufficiently examining the contribution of external features to forecast accuracy.

This thesis aims to address this research gap by investigating how external variables, such as unemployment and inflation, influence the predictive accuracy of ML models and how feature selection affects their performance. The study conducts a comparative analysis of three algorithms: LightGBM, XGBoost, and Multilayer Perceptron (MLP). The models are tested and compared in relation to one another and benchmarked against a 52-week seasonal naïve forecast. The comparative analysis is based on comparing forecasts made with different feature sets, evaluating forecast accuracy using various error and performance metrics.

The empirical part of the research applies quantitative methods using simulated and anonymized time series data representing weekly sales figures from a U.S.-based retail chain operating in forty-five locations. The dataset covers approximately three years and includes seven original variables, consisting of macroeconomic, temporal, and store-specific features. Additional features were engineered to capture lagged and interaction effects within the data. The methodology involves data preprocessing, new feature engineering, a 65:35 train-test split, hyperparameter optimization, and evaluation using RMSE, MAE, MASE, and R^2 metrics. Permutation feature importance is used to assess the contribution of different features.

The findings indicate that all machine learning models significantly outperformed the seasonal naïve baseline, demonstrating their capability to produce more accurate forecasts. Gradient boosting models achieved the best overall performance, with LightGBM outperforming XGBoost with a slight margin, while the MLP model provided the weakest performance and highest computational cost. Answering the research questions, the results confirm that feature selection has a decisive effect on model performance. Lag features representing short-term temporal dependencies were found to dominate feature importance scores across all models. The optimal lag length was identified as one week, while macroeconomic variables such as unemployment and inflation showed limited significance in short-term forecasts. MLP was the only model for which holiday-related features showed notable importance.

KEYWORDS: Demand forecasting, machine learning, retail analytics, feature importance, LightGBM, XGBoost, MLP, time series analysis

VAASAN YLIOPISTO
School of Technology and Innovations
Tekijä: Tuomas Niemelä
Tutkielman nimi: Demand forecasting in the retail environment: A comparative study of LightGBM, XGBoost, and MLP models
Tutkinto: Kauppatieteiden maisteri
Koulutusohjelma: Tuotantotalouden maisteriohjelma
Opintosuunta: Tuotantotalous
Työn ohjaaja: Petri Helo
Valmistumisvuosi: 2025 **Sivumäärä:** 103

TIIVISTELMÄ:
Tarkka kysynnän ennustaminen on katsottu olevan kriittinen operatiivinen tekijä vähittäiskaupassa, mistä organisaation päätöksenteko ja johtaminen ovat yhä enemmän riippuvaisia. Tarkat ennusteet mahdollistavat strategisen suunnittelun, varastojen optimoinnin, asiakastyytyväisyyden parantamisen sekä ylijäämän ja hävikin vähentämisen. Vaikka kehittyneet koneoppimismallit tunnetaan tarkkojen ennusteiden tuottamisesta, nykyisessä kirjallisuudessa keskitytään usein algoritmien tehokkuuden vertailuun ilman, että ulkoisten tekijöiden vaikutusta ennusteiden tarkkuuteen tarkastellaan riittävästi.

Tämän tutkielman tarkoitus on vastata aiemman tutkimuksen puutteellisuuteen selvittämällä, kuinka ulkoiset muuttujat, kuten työttömyys ja inflaatio, vaikuttavat ML-mallien ennustustarkkuuteen ja kuinka ominaisuuksien valinta vaikuttaa niiden suorituskykyyn. Tutkimuksessa on toteutettu vertaileva analyysi kolmesta algoritmista, jotka ovat LightGBM, XGBoost ja MLP. Analyysi perustuu eri ominaisuusjoukoilla tehtyjen ennusteiden vertailuun ja ennusteiden tarkkuuden arviointiin käyttämällä erilaisia virhe- ja suorituskykymittareita. Työn metodologiaan sisältyy datan esikäsittely, uusien dataominaisuuksien luonti, tietokannan jakaminen harjoitus- ja testidataan, hyperparametrien optimointi, sekä virhe- ja suorituskykymittareiden validointi.

Tutkimuksen empiirisessä osassa sovelletaan kvantitatiivisia menetelmiä käyttäen simuloitua ja anonymisoitua aikasarjadataa, joka koostuu yhdysvaltalaisen vähittäiskauppaketjun viikoittaisista myyntiluvuista, kerättynä 45 eri toimipisteestä. Aineisto kattaa noin kolmen vuoden ajanjakson ja sisältää kahdeksan alkuperäistä muuttujaa, jotka koostuvat makrotaloudellisista, ajallisista ja myymäläkohtaisista ominaisuuksista. Muuttujien vaikutusta ennustetarkkuuteen mitataan permutaatiomenetelmällä.

Tulokset osoittavat, että koneoppimismallit suoriutuivat merkittävästi paremmin kuin kausittainen naiivi vertailuarvo, mikä osoittaa niiden kyvyn tuottaa tarkempia ennusteita kuin perinteiset ennustemallit. Gradient boosting -mallit saavuttivat parhaan kokonaistehokkuuden, joista LightGBM suoriutui hieman paremmin kuin XGBoost. MLP-malli puolestaan suoriutui heikoiten. Tulokset vahvistavat, että ominaisuuksien valinta vaikuttaa ratkaisevasti mallin suorituskykyyn. Lyhytaikaisia ajallisia riippuvuuksia edustavat viiveominaisuudet osoittautuivat tärkeimmiksi ominaisuuksiksi kaikissa malleissa. Optimaaliseksi viiveen pituudeksi on havaittu yksi viikko, kun taas makrotaloudelliset muuttujat, kuten työttömyys ja inflaatio, ovat osoittautuneet merkitykseltään rajallisiksi lyhyen aikavälin ennusteissa.

AVAINSANAT: Demand forecasting, machine learning, retail analytics, feature importance, LightGBM, XGBoost, MLP, time series analysis

Contents

Figures

Tables

Algorithms

Equations

1 Introduction

The structures of consumer-driven industries have reshaped over the past few decades as competition has intensified, and dynamics have increased due to globalization driven by free trade. The operational structures of global organizations have grown into complex entities, covering everything from the procurement of raw materials to the sales of the final product. As the operational structures of organizations expand, their administration and management become increasingly complex, as the volume and dimension of data affecting strategic planning increases. Based on current literature, organizational decision-making and management are increasingly dependent on accurate demand forecasts to sustain with growing competition. However, despite technological advances and availability of data, the complex nature and unpredictability of demand challenges consistent forecasting in both research and practice.

Accurate demand forecasting is critical operational factor in supporting strategic planning (Caniato et al., 2005; Lima et al., 2024; Mircetic et al., 2022). It is crucial for the planning of functional processes, such as financing, logistics, inventory management, marketing, and production of a profitable business (Lima et al., 2024; Mircetic et al., 2022, p. 2514). Forecasting demand can help optimize inventory, increase customer satisfaction, and reduce waste (Ganguly & Mukherjee, 2024, p. 884), making it an essential part of an effective organization management. A study on global retail market estimated that inefficiencies in inventory management alone cost retailers 1.106 billion U.S. dollars yearly (IHL, 2015, as cited in Disney et al., 2021). Inefficiencies in organizational structures can increase costs that could be minimized by systematic optimization. According to a report conducted by McKinsey & Company (2022) advanced forecasting and digital process optimization can significantly improve operational efficiency. The study states that companies were able to improve the accuracy of their demand forecasts from 60% to 90% by replacing manual forecasts with machine learning models. Furthermore, an intelligent procure-to-pay process reduced processing time from days to minutes and achieved 15-20 % lower costs. Consequently, Institute of

Business Forecasting and Planning (2018) reported that a 15 % increase in forecasting accuracy can yield a 3 % higher pre-tax improvement. Regardless of its size, a forecast error is significant in terms of a company's results. Even a one percent improvement in the forecast was able to improve the results of a company with a turnover of 50 million by 1.52 million during its fiscal year. Additionally, accurate demand forecasting can reduce the annual operating expenditure by 7% (Mitra et al., 2022, p. 3). Forecasts are therefore significant drivers in a changing competitive environment. Figure 1 provides a comprehensive overview displaying the volatility of the retail market in the euro area over the past 10 years. The figure displays two indices: trade volume and turnover excluding motor vehicles and motorcycles. The trade volume indicates inflation-adjusted real trade volume, and turnover measures the nominal retail turnover rate. Both indexes are monthly, seasonally and calendar adjusted data from the European Economic Area (EEA) and selected non-EU states.

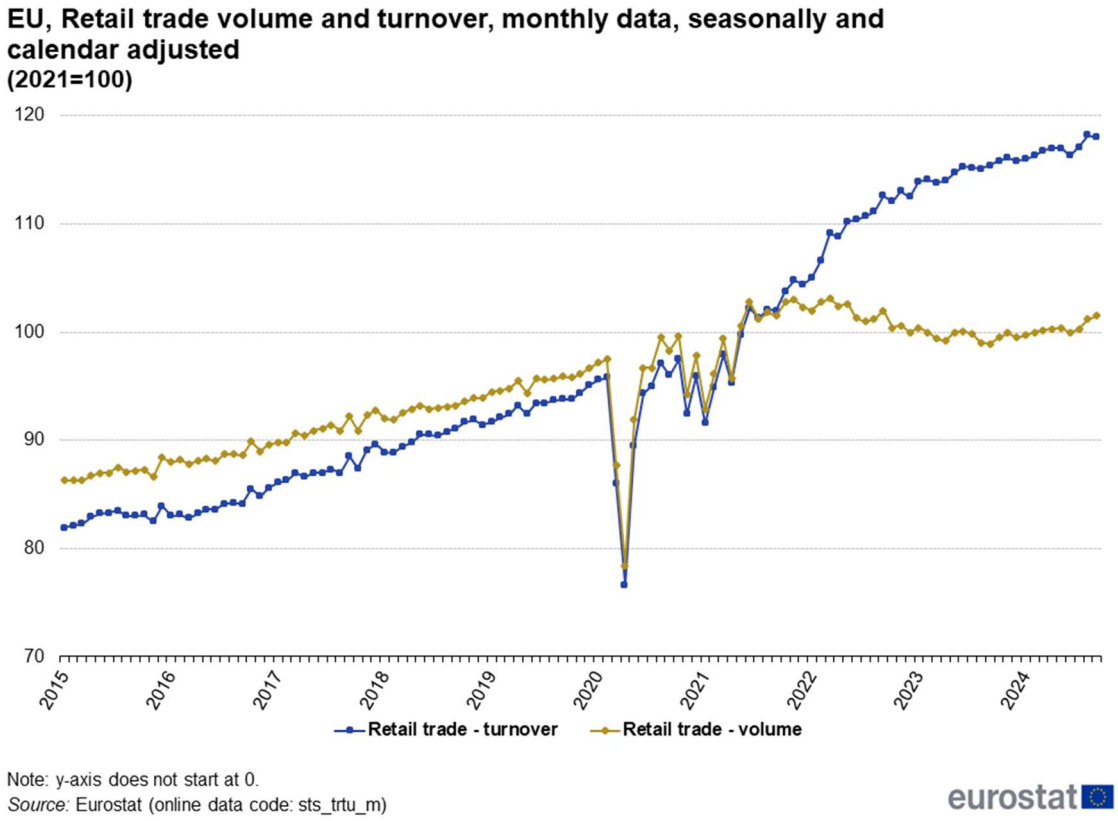


Figure 1. Retail trade volume and turnover (Eurostat, 2024).

The statistics illustrate how the macroeconomic effects triggered by the pandemic are reflected in consumer behavior. Strong volatility, starting from 2020, continues until 2022, after which the graph reveals the impact of inflation in the euro area. Nominal turnover would indicate that retail trade accelerated at the end of 2022, although in reality, according to inflation-adjusted retail sales volume, trade slowed down and declined in 2022-2023. The fluctuations of retail sales on a macroeconomic scale underscore the importance of effective forecasting. Consequently, volume and turnover indices provide an example of how, when making a forecast, it is necessary to be aware of which variables are used as inputs in the predictive analysis.

Due to the increased volume and dimensionality of data, traditional forecasting methods usually lack the requirements for efficient forecasting (Mediavilla et al., 2022, p. 1126), yet many forecasts are still conducted manually based on experience and intuition in the retail sector (Falatouri et al., 2022, p. 995). Current literature focuses on advanced forecasting methods, which are often based on machine learning (ML). These state-of-the-art algorithms can process large volumes of data and are efficient in finding causal connections between independent variables (Ganjare et al., 2023, p. 2237). Studies have shown that different ML methods can outperform traditional forecasts in specific prediction problems (Schmid et al., 2025, p. 2). Petropoulos and colleagues (2025) discuss the current state of retail demand forecasting. They depict the current situation as being twofold by providing a concrete example about an U.S. based retail company operating at over 10,000 different locations and managing approximately 200,000 unique stock keeping units (SKUs). Field tests and forecasting competitions have proven that advanced algorithms can be used to make accurate predictions on demand, but as the scale increases, computational constraints, the complexity of forecasts, and lack of practicality create clear limitations for effective forecasting (Petropoulos et al., 2025, p. 1564).

Forecasting demand can be effective for an individual store or product, but in modern, data-driven organizations, forecasting is not just a single process, but a multi-level

system that extends across different hierarchies of the business. Furthermore, the purpose of demand forecasting is not to estimate the sales of a single SKU, but rather to make an estimation of, for instance, regional purchasing power over a specific period. Such estimates can influence strategic decisions such as the location of a new business premises (Petropoulos et al., 2025, p. 1564). According to Yasir et al. (2024), the factors affecting demand are typically divided into internal and external. The internal factors are endogenous and organization-specific variables, and the external variables include, for instance, geographic location, temperature, seasons, and holidays (Falatouri et al., 2022, p. 995) as well as macroeconomic indicators, such as interest rates, trade volumes, prevailing employment rates, and exchange rates (Yasir et al., 2024, p. 2868). The external variables have proven to be valuable when conducting time series forecasts (Abolghasemi, Hurley, et al., 2020, p. 2; Falatouri et al., 2022, p. 995) but research on the significance of macroeconomic factors appears to be limited.

Despite the recognized value of forecasting in business operations, its implementation is relatively limited due to its complex nature. According to Schneider et al (2021), the difficulty of forecasting stems from the fact that measuring demand is not explicit. They state that the accuracy of an effective forecast is influenced by a plethora of potentially relevant variables, making it difficult to identify which factors truly improve forecasting accuracy. Additionally, traditional methods are usually too unsophisticated and advanced methods are still in the early stages of development, especially in complex decision-making situations (Schneider et al., 2021, p. 218). Abolghasemi and colleagues (2020) support the statement by concluding that when developing predictive models, it is important to find a balance between complexity and accuracy to maintain the efficiency and reliability of the model without unnecessary data usage.

This thesis focuses on the features used in demand forecasting, i.e., the external factors based on which the target (dependent) variable is predicted. According to previous research, forecasting is crucial in terms of operational efficiency. Advanced models, such as algorithms based on artificial intelligence, have also been found to produce accurate

predictions. However, there are relatively few studies focusing on the relationship between external factors used in predictions and the forecasting models. The study tests three AI-based models for making predictions and examines how the models control the features of the same time series dataset.

1.1 Research questions and purpose

Demand arises from the need for a specific good, product, or service. This need is influenced by various external factors, such as purchasing power, seasonality, and trends. Consequently, the final investment or purchase decision depends on this need, as well as, i.e., price, quality, economic situation, and substitutes. Thus, demand is influenced by numerous external variables, on the basis of which companies make important investment and operational decisions. When examining demand forecasting, the topic combines two highly complex areas: demand and forecasting. Currently, most research focuses on comparing the efficiency and predictive accuracy of different algorithms, and does not take external features into account when forecasting sales or demand (Deng et al., 2025, p. 156). Although studies focus on forecasting and specific exogenous factors related to demand, the emphasis is often solely on prediction error minimization without considering the contribution of the exogenous features.

Huber and Stuckenschmidt (2020) focused their research on analyzing forecast accuracy on specific calendar and holiday-related days in retail domain. They used various external features in their predictive analyses, such as store location and type, temporal characteristics (lag, rolling medians, etc.), and sales promotions and special days. The evaluation is based on comparing models in relation to the baseline forecasts and comparing the forecast error margins of the methods used. The study did not investigate the impact of external features used on forecast accuracy. However, they suggest in their proposal for further research that industry-related insights could be explored by studying the contributions of external features. Furthermore, Deng and others (2025) support the proposal by stating that analysis on consumer business domain is insufficient, thus

research on features of the retail market to optimize model structure is needed. They also conducted their research on retail by comparing a model consisting of LightGBM and Prophet to single prediction models such as LSTM, SVR, and ARIMA. Additionally, Falatouri compared machine learning models for an Austrian retail company and examined the effects from a supply chain management perspective. The results showed that profitability was increased by minimizing waste and increasing sales numbers. Nevertheless, he also notes that future studies could focus on examining the impact of external features, such as calendar events, weather, or availability of substitute products to provide domain-specific insights.

This thesis aims to address the research gap identified in previous research regarding retail demand forecasting by clarifying the contribution of external factors on the accuracy of forecasting made using machine learning models. The analyzed data consists of a U.S. -based retail chain. The data includes weekly sales figures from stores in 45 different locations, as well as data on contextual, macroeconomic, and temporal variables. This study aims to examine how machine learning models leverage external variables, such as unemployment, inflation, and fuel price data, in demand forecasting. This is examined by conducting predictions using three different machine learning algorithms. The algorithms are used to create different models by providing them with data with different sets of features, allowing the predictions obtained with different features to be compared. Additionally, the results of the best feature set are then used to produce a feature importance analysis using the permutation method. Motivated by this, previous research, and background of the study, the research questions are as follows:

RQ1: How do the algorithms evaluate external features in their predictions?
RQ2: How does feature selection affect the predictive accuracy?

The research questions are based on the purpose and background of the thesis, and they guide the research to fill the research gap identified within the field of study. The

empirical part of the thesis is done by conducting predictions with four different predictive models, one of which works as the baseline model for the other machine learning algorithms. Machine learning algorithms used include LightGBM, XGBoost, and Multilayer Perceptron (MLP), with the seasonal naïve forecast serving as the baseline.

1.2 Objectives and clear limitations

The objective of this thesis is to create a clear framework for how demand, its forecasting, and advanced predictive models are interconnected. Furthermore, the study aims to answer how the prediction models used in the study handle external variables in the creation of predictions. The thesis starts with a literature review, which examines the basic theories of the subject areas and previous research on the topic. During the literature review the key concepts and terminology are explained to create understanding of the fundamentals of forecasting, demand forecasting, as well as applications of artificial intelligence. Next, the thesis reviews previous research, which allows for a more detailed examination of topics related to demand forecasting.

During quantitative research, the topics studied in the literature review are examined in practice. Three different machine learning models are tested and compared with each other, as well as in relation to a seasonal naïve forecast. The aim is to examine the accuracy of the predictions using various errors and performance metrics and to produce results that are as objective as possible. These metrics allow the comparison of predictive accuracy of models with different feature sets. Consequently, it allows the generation of quantitative results on how external factors affect predictive accuracy. Last, the importance of features is measured by permutation feature importance, which depicts the contribution of a single feature by determining how much the model relies on such feature.

The clear limitations of the thesis are related to the characteristics of the data, the methodology of this study, and model-specific technicalities which may impact the

results. When conducting time series analysis, it would be desirable to have as much data available as possible, as is needed both in the training phase and in the testing phase of the model. The time series spans only three years, consisting of 6435 datapoints. Some algorithms, such as neural network architectures, require a lot of data to function properly. Furthermore, the data is not based on actual values as it is a simulated and anonymized dataset, therefore the relevance of the research results to real life situation decreases. Furthermore, the study lacks qualitative input that could support its objective and the generalizability of its results to real life situations. Technical limitations are related to tuning of model hyperparameters. Even if the study were repeated using the same dataset, data features, and algorithms, the results obtained could be different if the hyperparameters or their values are changed. To conclude, the thesis and its results are highly theoretical due to the mentioned limitations.

1.3 Structure of the paper

The structure of the thesis consists of five main chapters, which are introduction, literature review, methodology, results, and conclusions. The introduction chapter presents the background for this thesis, as well as the purpose, research questions, objectives, and clear limitations of this study. Second, the literature review presents the rationale for the topic, the main concepts, and the relevant terminology. After key concepts, it delves deeper into the research area at a more advanced level by examining previous research papers on the topic and the used algorithms. This is followed by the methodology of this paper, which explains the unit of analysis, process steps, data, and tools used to obtain the results. Its purpose is to provide the reader with guidelines on how this study was conducted. After the methodology, the results and how they were obtained are explained. The results section depicts the concrete quantitative results of the analysis and delve deeper into the results with visualization and comparisons and provides answers to the research questions. Finally, the conclusions section summarizes the findings of this research, generalizes the results beyond this thesis and its data, and possible suggestions for further research are provided.

2 Literature review

Forecasting demand is a crucial aspect of managing business. According to Punia et al. (2020) demand forecasting is done to solve two main problems of companies within manufacturing, retail or distribution business which are deciding the quantities for production or orders and allocation of resources. Furthermore, Sillanpää ja Liesiö (2018), in their study on demand forecasting in retail business, state that forecasts are required to effectively plan operations and incoming orders. They emphasize the need for information to apply demand forecasting in businesses and state that the data is most easily gathered from point-of-sales (POS) and stock keeping unit (SKU) sales data. However, creating forecasts solely on historical sales data can be problematic, since demand is influenced by numerous external variables. For example, forecast methods using POS-data can be unreliable in case of reschedules (Sillanpää and Liesiö, 2018, p. 4169). Because demand is a difficult variable to measure (Mitra et al., 2022, p. 3) a lot of research can be found on the topic: how forecasting methods are implemented and how they are created.

The purpose of this section is to review the fundamentals of forecasting, which will be used to continue discussion on demand forecasting to create a theoretical framework for the thesis. Next, machine learning models are examined, along with the principles of the models used in this thesis, and how they are used in demand forecasting. Finally, previous research related to the topic is examined.

2.1 Fundamentals of forecasting

Forecasting is about making predictions of the future. In their book *Forecasting Fundamentals* (2015) Nada Sanders explains that forecasting is all about predictions, whether it is the future weather, the outcome of tomorrow's match or who will win the election. They say that forecasting is the most important aspect of decision making and stress that inaccurate forecasting can lead business in a very unfortunate state and even

bankruptcy. Forecasting influences many managerial decisions such as the need for workers, how much inventory is enough and when to order more, what resources will be available, or how much production is appropriate for a given period (Sanders, 2015, pp. 4-5).

Armstrong (2001) proposes seven concrete principles to forecast with the intention to improve judgment by reducing bias and/or inconsistency of the forecast. The principles are displayed in table 1.

Table 1. Principles of forecasting according to Armstrong (2001, pp. 61–66).

Principle	**Effect on the forecast**
1. Using checklists	Increases consistency
2. Defining and delimiting precise criteria	Increases consistency and efficiency, minimizes bias
3. Comparison and evaluation of previous forecasts	Increases consistency and minimizes bias
4. Visualization of results for interpretation	Minimizes bias and decreases error
5. Utilizing patterns and trend lines	Increases consistency
6. Using multiple forecast methods	Increases robustness
7. Peer reviews	Minimizes bias

Armstrong states that two of the six, checklists (1) and utilization of trend lines (5) increase consistency of the forecast method. Usage of checklists emphasize systematic considerations of relevant variables and utilization of trend lines when making *judgmental* forecasts help visualize the data, thus providing possibility for pattern recognition and support more consistent decision-making. In contrast, (4) use of graphs for interpretation and (7) peer-reviewing the probability of success are for minimizing the bias within the forecast. Armstrong suggests that studying data in graphic rather than tabular form increases the forecasting accuracy and decreases error. Having peers

reviewing the probability of success decreases the amount of human error, which is usually shown as overconfident forecasts, thus increasing error of the forecast. Principles (2) and (3) help in both, decreasing bias and increasing consistency. Defining and delimitating precise criteria removes unnecessary variables from the forecast, making the forecast more efficient. Records from previous forecasts give forecasters a way to obtain cognitive feedback. However, Armstrong underlines that it is important to use the records appropriately to provide a reliable assessment (Armstrong, 2001, pp. 70-71).

Sanders (2015) proposes a different perspective on the forecasting principles with three main ideas. Whereas Armstrong's seven principles for forecasting are more traditional and concrete methods that should be considered when making predictions, Sanders' criteria are suitable for forecasting at a general level, with ML models for example. Sanders' principles are displayed in table 2.

Table 2. Forecasting principles according to Sanders (2015, pp. 18–19).

Forecasting principle	Rationale of the principle
1. Forecasts are rarely perfect	The goal of a good forecast is to **minimize error**, not to forecast perfectly
2. Forecasting clusters is more accurate than individual items	The overall variance can be minimized by diversification
3. Short-term forecasting is more accurate than long-term.	Shorter time horizons involve less uncertainty, making short-term forecasts more reliable

Sanders (2015) also proposes six-step process of forecasting, which starts with deciding what to forecast to identify the real problem. It follows with data cleaning, identifying data patterns, selecting models, generating the forecast, and measuring the forecast accuracy. Sanders highlights the importance of setting clear delimitations and focusing solely on the variables being forecasted. It starts with identifying the core issue to which the forecast is trying to find a solution. For instance, in a scenario where unexpected

demand causes a seller to run out of stock midway through the day, resulting in unfulfilled customer needs, the sales data will underestimate the actual demand for that day. Thus, the forecast for sales and demand would require different approaches, although at first, they seem to be measurable with the same parameters. To conclude, having a clear consensus on what the forecast is for is crucial in terms of the relevancy and reliability of the forecast results (Sanders, 2015, pp. 20–21). Determining the core issue provides the forecaster with a framework for data collection at the most detailed level possible.

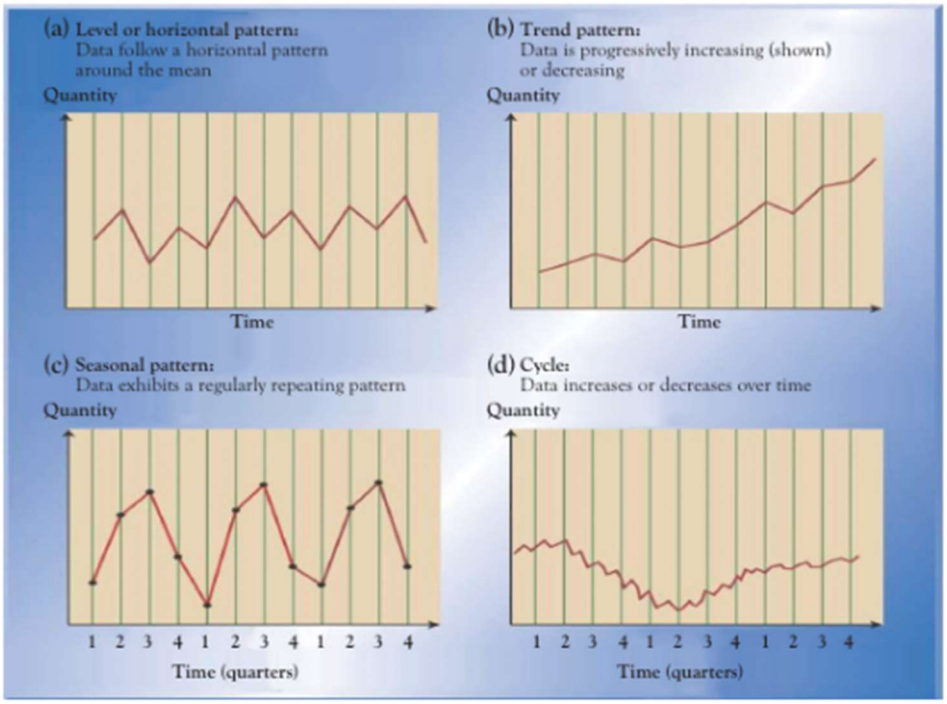


Figure 2. Common Data Patterns (Sanders, 2015, p. 23).

The patterns, level or horizontal, trend, seasonality, and cycles help the forecaster to choose the right forecasting model, which in turn is more likely to produce a more reliable forecast. Sanders (2015) explains that the clearer the trend is identified through data analysis, the more accurate the forecast is likely to be. This is because greater random variance increases the difficulty of producing reliable forecasts, as depicted in figure 3 below.

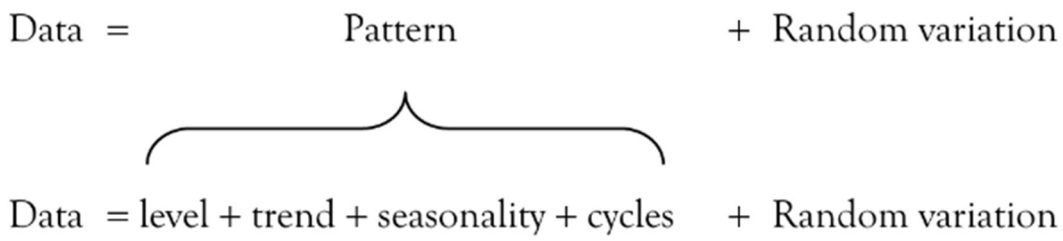


Figure 3. Components of data (Sanders, 2015, p. 24).

After cleaning and analyzing the data, the forecasting model can be selected. The selection might not be straightforward since there are plenty of different models for different types of datasets and patterns. Sanders highlights four factors that should be considered when choosing the model: forecast horizon, data patterns, the availability and quantity of data, and the required accuracy. After deciding which model to apply, the forecast can be generated, and the forecast results can be analyzed and interpreted. When the possible forecasting errors are identified and the model provides accurate results, the dataset should be updated as new relevant data is available (Sanders, 2015, pp. 25-26).

Demand forecasting can be implemented with either qualitative or quantitative methods. Qualitative methods, such as historical analogies, market research, questionnaires, and Delphi technique are used to predict market demand (Mitra et al., 2022, p. 58). Quantitative methods rely on numerical and measurable data, and most demand forecasting methods are evaluating causal relationships between independent and dependent variables through regression analysis. Additionally, other data-driven models utilized are time-series models such as exponential smoothing and moving average methods (Mitra et al., 2022, p. 58; Punia et al., 2020, pp. 2-3). This thesis focuses solely on quantitative demand forecasting.

2.2 Demand forecasting

In today's global economy organizations need to be efficient in terms of cost optimization, information flows, delivery, information transparency and development to be able to keep up with global competition (Abolghasemi, Beh, et al., 2020, p. 2; Feizabadi, 2022, p. 121; Mediavilla et al., 2022, pp. 1126–1127; Mitra et al., 2022, p. 2). Globalization has driven the trend of outsourcing increasing supply chain complexity and internationality, making lead-times of procurement longer (Feizabadi, 2022, pp. 121–122). When making purchases or planning production volumes, a company needs to have a plan on how much product on the shelf or in production is sufficient in quantities. Too much supply will be costly in terms of storage costs, surplus and labor costs. Insufficient supply, on the other hand, means lost revenue, decreased customer satisfaction and loyalty, loss of goodwill, and possible overstocking against future demand (Abolghasemi, Beh, et al., 2020, p. 1; Punia et al., 2020, pp. 1–2).

Demand forecasting can help address many of the issues by providing insight into decision making, as according to Abolghasemi et al (2020). However, demand is highly volatile and not an easy variable to predict, since it is affected by various exogenous and endogenous factors (Mitra et al., 2022, p. 3). Despite demand being highly dependent on exogenous factors, Falatouri (2022) states that many retailers still conduct demand forecasting manually, making the decision based on their individual biases. Usually this can lead to inaccuracies as the uncontrollable external factors, such as price volatility, market cannibalization, or consumer behavior impacting market demand. Efficient demand forecasting is conducted by strict processes and predictive analyses (Kilimci et al., 2019, pp. 1–2), utilizing advanced technologies and resources such as big data (Pereira & Frazzon, 2021, pp. 3–5).

Authors	Advantages of demand forecasting
Abolghasemi et al. (2020)	Help in addressing volatility over the entire demand series, mitigating issues in upstream supply chains and increasing cost-efficiency
Feizabadi (2022).	Improve efficiency across the processes of entire supply chain
Ho et al. (2025).	Optimize storage, increase customer satisfaction, and process efficiency Mitigates stockouts and overstocking
Huber & Stuckenschmidt (2020).	Increase in competitive advantage Minimize the amount of discarded goods and waste
Jackson et al. (2024).	Increase strategic decision-making efficiency and cost efficiency
Khan et al. (2020).	Help enterprises to formulate market strategies, increase inventory turn-over rates, customer satisfaction, and transparency. Reduce waste and overall costs
Kilimci et al. (2019).	Increased cost efficiency by minimizing excessive stocks and stockouts Increasing customer satisfaction
Lay et al. (2018).	Alleviates stockout and overstocking and increases customer satisfaction, which enables companies to gain sustainable competitive advantage

Table 3. Conclusion of advantages of demand forecasting.

Table 3 illustrates how different researchers depict the effectiveness of demand forecasting. The broad effects highlight the demand for accurate forecasts and advanced forecasting methods. Furthermore, the increasing availability of data further complicates the processes and slows down data processing if an organization lacks the necessary means to harness the data.

2.3 Artificial intelligence

Artificial intelligence (AI) is ubiquitous, regardless of the industry, as are all the buzzwords associated with it such as Machine Learning, Large Language Models, Deep Learning, and Big Data. Many researchers highlight how AI and its sub-areas work in different industries as drivers of efficiency (Dell'Acqua et al., 2023; Fosso Wamba et al., 2024; Jackson et al., 2024; Krakowski et al., 2023; Wasserbacher & Spindler, 2022).

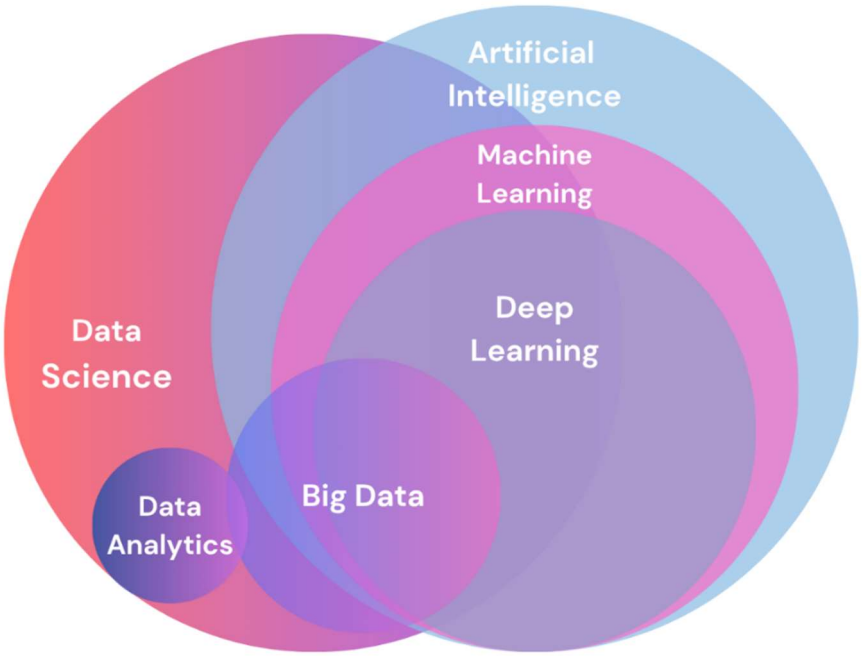


Figure 4. Venn diagram depicting the relationship between statistical concepts.

Figure 4 provides a conceptual illustration of how the subareas related to artificial intelligence are interconnected. It can be concluded that AI refers to a field of computer science focused on creating models performing tasks typically requiring human intelligence (Krakowski et al., 2023, p. 1426). artificial intelligence has gained popularity as Large Language Models (LLMs) became more common with the release of ChatGPT by OpenAI in 2022 (Jackson et al., 2024, p. 1). They quickly gained remarkable attention because of their generative capabilities (Dell'Acqua et al., 2023, p. 3; Jackson et al., 2024, p. 6120). LLMs are best known for their ability to provide their users with human-like answers, and creative and analytical capabilities (Dell'Acqua et al., 2023, p. 1) which can be utilized to complement or substitute human work.

2.3.1 AI as a driver of efficiency

The integration of AI into human work is seen as an opportunity to complement the efficiency of individuals (Dell'Acqua et al., 2023, p. 1) having impact on human cognition and problem-solving ability reducing the marginal cost of human thinking and reasoning similar to how internet lowered the cost of information sharing (Dell'Acqua et al., 2023, p. 18; Jackson et al., 2024, p. 6120). Applications of artificial intelligence enables autonomic learning of machines, which provides these machines the ability to co-operate in problem solving and decision making with humans (Krakowski et al., 2023, p. 1426). AI's ability to mimic the cognitive skills of humans is a unique capability in the field of technology according to Krakowski and others (2023). Humans' individual cognitive abilities have traditionally been difficult to duplicate, for which the supply has been scarce, thus AI's ability to provide cognitive skills provides a huge advantage since it can be utilized in decision making and managerial tasks (Krakowski et al., 2023, p. 1427). The added value of AI is not straightforward to calculate, but some researchers have examined possibilities to estimate it.

Efficiency can be increased by harnessing artificial intelligence, as according to Jackson (2024) AI tools can lead to a surge in productivity, enhance cognitive work efficiency,

support logistics and warehouse management, and even help in negotiating optimal contracts. From a demand forecasting perspective, implementation of AI methods can increase the accuracy of the forecast, which in turn increases supply chain resiliency (Mediavilla et al., 2022, p. 1130) improving order-picking performance, and accurately respond to upcoming demand spikes or uptrends (Ho et al., 2025, p. 2). They further examine the capabilities of artificial intelligence and identify five core characteristics that distinguish AI from traditional technologies, which collectively serve to define and explain the concept of AI. The core characteristics are Learning, Perception, Prediction, Interaction, Adaptation, and Reasoning (Jackson et al., 2024, p. 6123), from which learning, prediction, and reasoning are by far the most important features for Demand Forecasting.

Krakowski and others (2023) study the additional value of artificial intelligence from the perspective of resource-based view, which defines organizations' competitive advantage based on the availability and volume of resources. Traditionally, cognitive skills are considered difficult to duplicate, scarce in supply, heterogeneously distributed across individuals, and decisive in decision making and problem solving. Thus, from the perspective of resource-based view they are rendered as valuable organizational resources. However, the prediction of the potential added value of AI contradicts this, as AI's ability to learn and perform cognitive tasks affects the irreplaceability of cognitive skills and abilities, which have made them valuable resources. In addition, Krakowski and others note that generally technological resources, like AI, are subject to relatively few constraints on imitation and the marginal cost of reproducing them is almost negligible (Krakowski et al., 2023, p. 1426). On the contrary, when studying AI as a complementary to individuals' cognitive skills, it can further enhance cognitive resources thus adding value for the user. The unique capabilities of AI make it difficult to evaluate theoretically, which has led to a discussion about AI's potential as a substitute and complementary utility for cognitive workers.

2.3.2 Machine learning methods

Machine Learning is a subcategory of artificial intelligence (Ganjare et al., 2023, p. 2237; Mediavilla et al., 2022, p. 1126), and it is being harnessed in different industries to analyze masses of data. The industries utilizing ML, for instance, are search engines, finance, logistics, e-commerce, and inventory management, and a few examples of which tasks ML is used to provide are fraud detection, detecting spam emails, predictive analyses, optimizing inventory levels, and providing personalized feed in an e-commerce platform for the consumer (Ganjare et al., 2023, p. 2237). Barua et al. (2020) generalize that machine learning is a way to teach computers to learn new tasks naturally from experience resembling the way organisms acquire new knowledge. In ML the computer utilizes a computational method, an algorithm, for example, to learn directly from the dataset without a predetermined equation, unlike traditional statistical methods. Compared to conventional methods, ML is more efficient with faster, and more accurate analyses for large data sets, providing tools for better predictive data analyses while conventional statistical models provide relationships between variables based on predetermined models, such as regression (Ganjare et al., 2023, pp. 2236–2237; Rajula et al., 2020, pp. 1–2).

There are three general categories of machine learning, which differ in terms of the quality of the data and the methods used to teach the machine. They are called supervised, unsupervised, and reinforcement learning (Barua et al., 2020). They further divided supervised learning into two sub-categories: classic supervised learning and ensemble learning. In addition to the three main sub-categories, Wasserbacher and Spindler (2022) propose semi-supervised learning as the fourth category in machine learning. Semi-supervised methods include models utilizing small amounts of labeled data in addition to unlabeled data. Semi-supervised models aim to enhance supervised learning in environments where availability of labeled data is scarce (Wasserbacher & Spindler, 2022, p. 67). However, semi-supervised methods are not generally known in current literature.

In contrast with supervised learning, unsupervised learning models are not trained to create predictions based on pre-defined and labeled data. The task of unsupervised learning is to identify patterns and relationships within unlabeled data, without prior knowledge or predefined labels about what the data represents (Barua et al., 2020, pp. 2–3). The only known parameter of the unlabeled dataset is the joint distribution (Wasserbacher & Spindler, 2022, p. 67). The benefit of unsupervised learning is recognition of previously unseen insights in the data, according to Wasserbacher and Spindler (2022). They give customer segmentation as an example of a task unsupervised learning can provide based on consumers' demographic characteristics, socio-economic status, and behavior.

These kinds of hidden patterns and features are primarily recognized by clustering and principal component analysis (PCA) (Barua et al., 2020, p. 2). For instance, in a study conducted by Kılıç et al. (2025), two clustering algorithms were utilized to categorize the unlabeled data. They used K-means and Mean-Shift algorithms to cluster three datasets separately, from which, three distinct clusters (low, medium, and high) were identified by their performance metrics. By assigning cluster-based labels, the researchers were then able to further analyze the data by supervised learning algorithms to predict the future shipment performance. This hybrid approach enabled proactive operational management by allowing early identification of underperforming shipments (Kılıç Sarıgül et al., 2025, pp. 22–23). Thus, the unsupervised learning algorithms were used to organize an unspecified dataset so that it could be processed by supervised learning algorithms to model the relationship between input data and output values.

In reinforcement learning, a learning agent is learning from an environment it is operating in by trial and error (Barua et al., 2020, p. 2; Wasserbacher & Spindler, 2022, p. 67). Sutton and Barto (2015) state that the basic idea in reinforcement learning is to capture the most important aspects of a real problem by providing the learning agent feedback provided by the learning environment. The main reinforcement learning distinguishing aspects are the closed loop reward system, lack of direct instructions for

the learning agent, and where and how long the consequences of actions will affect the agent's performance. Because reinforcement learning is not utilizing pre-labeled examples and data, it might be often referred to as a subset of unsupervised learning algorithms (Sutton & Barto, 2015, p. 3).

Based on the reward system of the learning environment, after a failed attempt the agent decides the best action for it to take to succeed in the given task, to maximize the numerical reward (Barua et al., 2020, p. 3). Sutton and Barto (2015) presented an *Exploration-Exploitation dilemma,* which emphasize the dimensionality of reinforcement learning method. The learning agent must prefer past actions found to be effective producing reward—but it also must try new, unseen actions, to find the effective actions. Thus, the agent must *exploit* well proven actions at the same time *exploring* possible better actions to select in the future. Thus, unlike supervised learning, there are no instructions in reinforcement learning on how to proceed. The learning agent must decide how to approach the task purely based on gathered data via the reward system.

In supervised learning the machine is taught from a training set of labeled data, provided by a human annotator or domain expert (Sutton & Barto, 2015, p. 2). Supervised learning has two main steps, first training the model with predefined training dataset and then evaluating the model with separate, unseen dataset, which is referred to as test data. The predefined data is called labeled data and the purpose of using labeled training data is to teach the model to recognize the relationship between the input data and correct output values (Kılıç Sarıgül et al., 2025, p. 14), to create accurate predictions, which can inform decisions based on unseen data (Barua et al., 2020, p. 2; Sutton & Barto, 2015, p. 2; Wasserbacher & Spindler, 2022, p. 67). An example of where supervised learning can be utilized is predicting future sales based on known input variables, such as date, historical prices and sales, and availability of competitors' products (Wasserbacher & Spindler, 2022, p. 67).

Barua and others divide supervised learning into two, *Classic supervised learning* and *Ensemble learning.* Classic supervised learning involves training a single model using predefined labeled learning data. The most general algorithms Classic Supervised methods include are regression analyses, k-nearest neighbors (KNN), Artificial Neural Network (ANN), decision trees, and Support Vector Machine (SVM) (Barua et al., 2020, p. 2). Chaudhuri and others (2021) also state that decision tree, SVM, ANN, and random forest are among the most used analytical methods for forecasting. Interpretability, ease of use, and ability to handle categorical variables have popularized decision tree and random forest over ANN. On the other hand, ANN's capability to handle multidimensional datasets and efficiency in use of resources is superior compared to the easier-to-use random forest and decision tree (Chaudhuri et al., 2021, p. 3).

2.3.2.1 Decision tree, random forest, bagging, and boosting

Decision tree is a conceptually simple but effective and versatile (Barua et al., 2020, p. 2) non-parametric supervised learning algorithm. It is capable of both classification and regression tasks, and it forms a flowchart-like hierarchical structure consisting of a root node, internal nodes, branches, and leaf nodes. In a tree, each node splits the data into subsets based on feature values, and the branches represent the outcomes of these tests (Barua et al., 2020, p. 2). The straightforward implementation makes decision tree fast in performing a forecast (Barua et al., 2020, p. 2), and together with its interpretability it has become a widespread model for forecasting related applications (Chaudhuri et al., 2021, p. 3). Barua and others (2020) present a case study by Mohri and Haghshenas (2017), where a decision tree-based algorithm was used to determine when the use of shipping containers is optimal. The input variables included the price, weight, value, and distance of the shipment, for instance. The most important variables were item perishability, value of goods, distance, destination and point of departure. On the other hand, decision trees can be relatively unstable as they are sensitive to variance and noise in the data due to their tendency to overfit, according to Chaudhuri and others (2021). However, the sensitivity of decision tree can be addressed by composing multiple

decision trees as one ensemble learning model such as random forest (Huber & Stuckenschmidt, 2020, p. 1426). Ensemble methods improve accuracy and robustness while reducing variance (Barua et al., 2020, p. 2) compared to training a single decision tree.

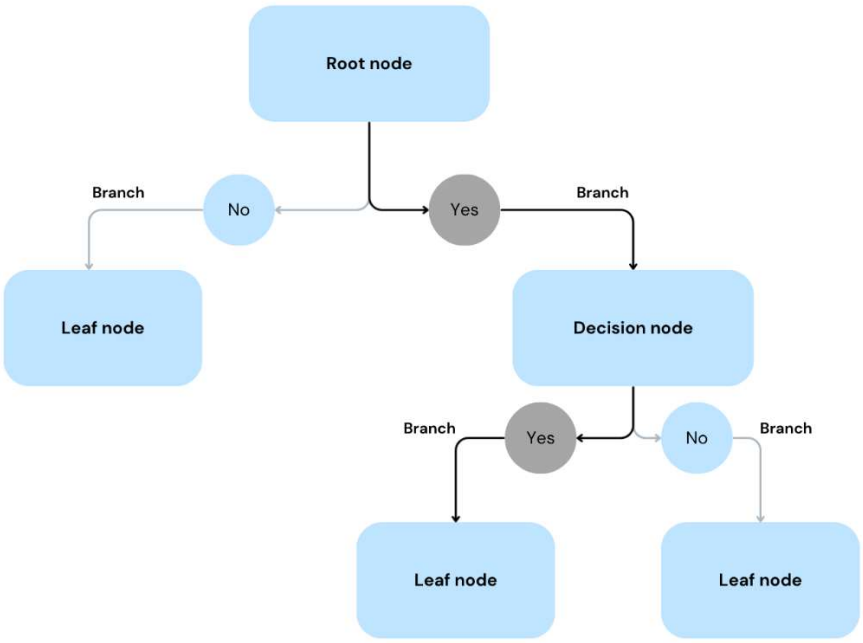


Figure 5. Decision tree architecture, adapted from Mohri et al (2017, p. 7).

Ensemble learning combines the predictions of several individual models to form a composite model. It uses the outputs of the individual models, referred to as *base learners* or *weak learners*, to produce its own predictions. Barua and others (2020) state that ensemble methods use a two-step process, first developing a population of base models from training data, and second, combining the base models to form the composite predictor (Barua et al., 2020, p. 2). Kilimci and others (2019) name the two steps of ensemble learning as “ensemble generation and ensemble integration”, and state that combining learning methods is done to boost the system performance (Kilimci et al., 2019, p. 2). Hastie and others (2009) also split ensemble learning into two tasks: creating a population of base learners and combining them to form the composition,

idea being to create a prediction model to combine the strengths of simple learning models (e.g. single decision tree) for a more efficient predictive model (e.g. random forest). Thus, the underlying concept of ensemble learning methods is to develop an enhanced form of supervised learning from base learners that enables more efficient predictive modeling compensating the weaknesses of individual models.

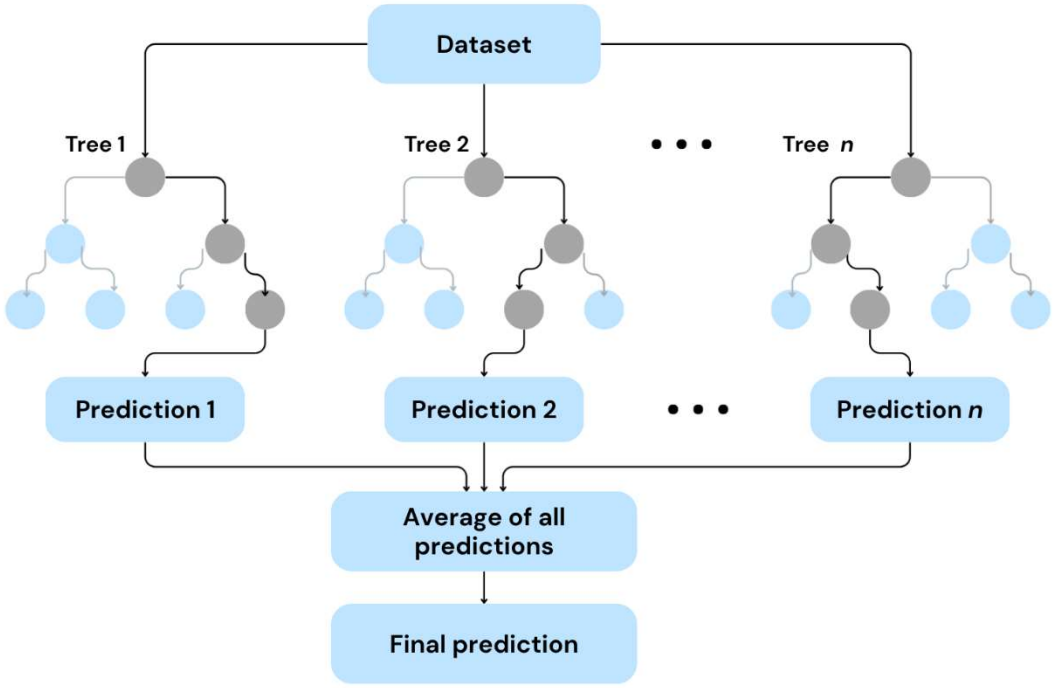


Figure 6. Random forest architecture, adapted from Jiang et al. (2016, p. 58).

Common ensemble techniques include bootstrap aggregating (bagging), boosting, and stacking, all of which employ a different strategy to achieve better performance by improving accuracy of predictions and robustness of the model minimizing bias and variance (Barua et al., 2020, p. 2). Boosting methods (e.g. LightGBM and XGBoost), for instance, train a series of weak learners and compiles the predictions of subsequent learners by the sum of trained simple models (Huber & Stuckenschmidt, 2020, p. 1426). For example, AdaBoost has been used in conjunction with SVMs to enhance predictive performance, as referenced by Ghareeb and others (2020, p. 1).

As Boosting focuses on reducing bias and variance by improving accuracy of the model and training learners subsequently, Bagging models combine homogenous weak learners, training them independently and in parallel with random subsets of training data, primarily prevents overfitting by reducing variance. random forest is a known Bagging method, which combines individual simple decision trees suitable for both, regression and classification tasks (Ghareeb et al., 2020, p. 1).

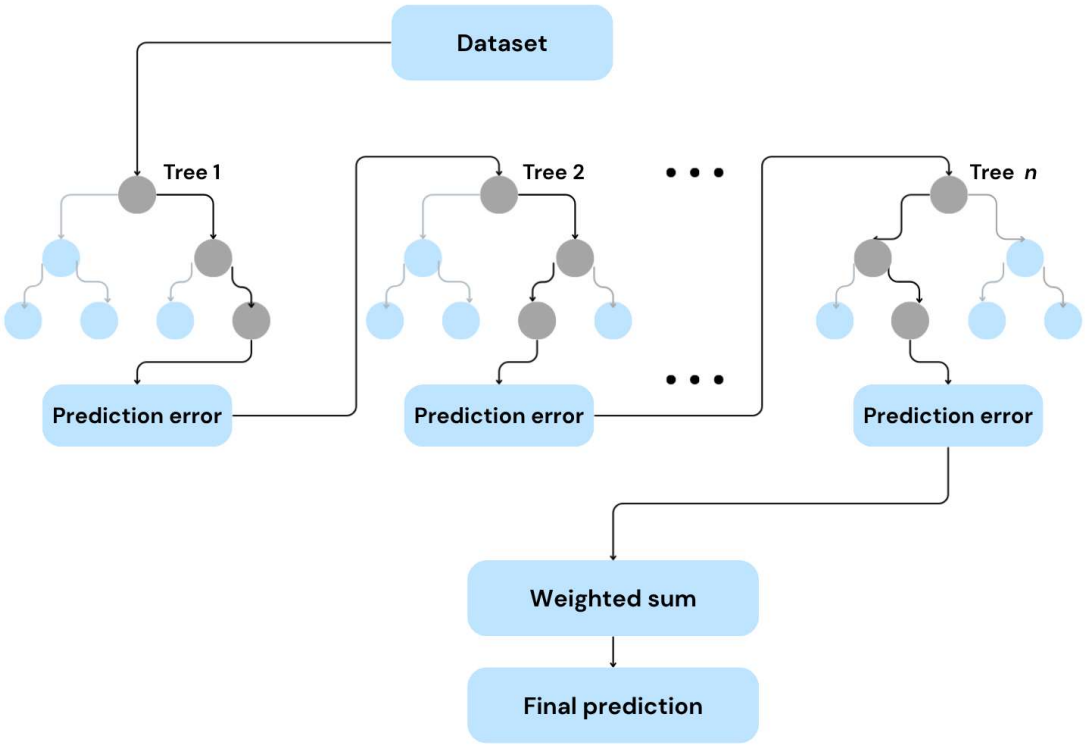


Figure 7. Gradient boosting architecture, adapted from Xu et al (Xu et al., 2023, p. 3).

Whilst bagging and boosting usually combines homogenous weak learners, stacking utilizes heterogeneous learners leveraging the strengths of different algorithms. The objective is to optimally incorporate the results of weak learners to improve the ability to make accurate predictions on new, unseen data (Ghareeb et al., 2020, p. 1). According to Ghareeb and others (2020) multistep predictions like Stacking are more sensitive to errors due to their complexity. However, the complexity of Stacking is also said to be viable making it more effective in forecasting complex datasets compared to other ensemble models.

2.3.2.2 Artificial Neural Networks

Artificial Neural Networks (ANN) consist of artificial neurons, connected to each other by arranged series of layers. The artificial neurons in ANNs are usually recalled as *units*, and a single ANN system can consist of dozens to millions of units, depending on how complex the neural network is (Barua et al., 2020, p. 2; Seyedan & Mafakheri, 2020, p. 12). The neurons are connected with *synapses*, which together construct the layers of neural networks. Neural networks usually include three layers: input layer, output layer, and hidden layer. ANNs with more than one, usually multiple hidden layers are called Deep Neural Networks (DNNs) (Punia et al., 2020, p. 4). The information processing of artificial neural networks resembles the way animals make decisions (outputs), based on a learned logical model of information processing (Barua et al., 2020, p. 2). Deep neural network models are used for complex problems, such as image recognition, and they form the core architecture of large language models (GPT-4, Llama 3, etc.). These models typically require a lot of computational resources to train and run. The common architecture of an artificial neural network is displayed in figure 8 below.

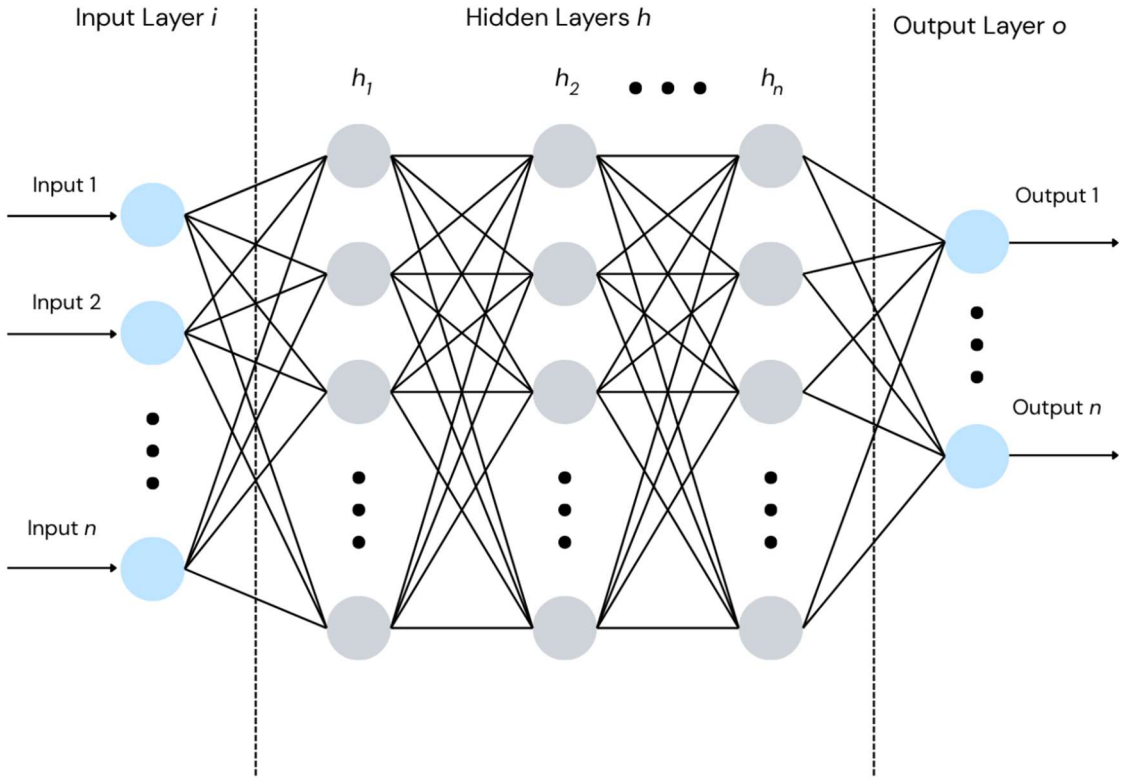


Figure 8. ANN Architecture, adapted from Bre et al. (2018, p. 1430).

This thesis uses a Multilayer Perceptron (MLP) regressor because of its relatively fast operation and training times, and because it is an effective model for analyzing time series data. Although different neural network-based models are based on the same type of architecture (Figure 8), there can be significant differences in their learning process according to Ramos et al (2023, p. 672). The relatively small amount of data used in the project might have been insufficient for highly complex neural network models, such as LSTM (long-short term memory) or RNN (recurrent neural network) models, which operate a different, more sophisticated memory concept compared to MLP (Ramos et al., 2023, pp. 672–673).

2.4 Creating forecasts with machine learning

The use of machine learning in forecasting has received a huge amount of attention in recent years. Forecasts conducted with machines are objective and mitigate human error (bias). Even if machine-based methods are applied by humans, with the right parameters and restrictions, most bias can be eliminated when making a forecast. Machine-based methods are particularly superior when forecasting demand, as demand itself is a multidimensional concept. First, there are different types of demand influenced by context and temporal features (Armstrong, 2001), and even if the form of demand is known, it is influenced by numerous macro- and microeconomic factors, not all of which are qualitative, i.e., they cannot be measured (Schneider et al., 2021, p. 218). Therefore, it is important to recognize demand forecasting as a complex domain and to consider it from different perspectives.

Falatouri et al. (2022) state that the objective of forecasting is to discover data patterns and provide accurate forecasts, for which machine learning is one of the most viable tools for processing data for accurate and transparent output. They propose that machine learning methods for demand forecasting can be divided into three categories which are time series analysis, regression-based methods, and supervised and unsupervised methods. Furthermore, they state that demand forecasting is done on long or short-term levels, short-term forecasts being six to twelve months and long-term forecasts for more than a year (Falatouri et al., 2022, p. 994).

2.4.1 Feature-based forecasts

Li and others (2023) researched the connection between intermittent demand and inventory management accuracy. Their research discusses the characteristics of demand, emphasizing that it is ubiquitous for demand to be intermittent, i.e. demand is often zero, which is rarely considered when studying demand. Predicting intermittent demand is particularly difficult due to uncertainty caused by stochasticity and timing of demand.

Previous literature has proposed methods to solve the problem of intermittent forecasting by dividing the time series into different intervals, but more recently the accuracy has been improved by machine learning methods, such as artificial neural networks. According to them, combining different forecasting methods have proved to provide efficient forecasts compared to individual methods, providing better or equal results. However, their own study concentrates on improving forecasting by engineering new features based on time series data. Li et al (2023) produced a forecasting model based on XGBoost, in which they utilize features derived from time series selected for intermittent demand. The feature-based model produced accurate predictions on variables having immediate impact on inventory managerial decisions (L. Li et al., 2023, p. 7568).

In Feizabadi's study (2022) on demand forecasting using autoregressive models and neural networks, demand forecasting is largely influenced by the characteristics (features) of the type of product and industry. They gave an example of metal products as so-called functional products. Functional products have less product variety, longer life cycles, lower profit margins and lower inventory risk. He notes that it is easier to predict demand for these products downstream of supply chain, closer to the consumer, than upstream, where demand is created mainly by organizational suppliers and buyers. However, they note that when moving downstream to upstream in the supply chain, updating demand forecasts is the single biggest cause of demand-supply mismatches and inefficiencies (Feizabadi, 2022, pp. 119–121), as separate parts of the supply chain update their own demand forecasts based on the purchase signal generated by the end customer. Updating demand data based solely on a signal from the end-customer, i.e. sales, is inaccurate on the scale of the whole supply chain. Therefore, more traditional methods, such as simple regression, are inadequate methods for forecasting demand. Machine learning can help organizations to better predict demand by dealing with complex dependencies even between causal factors with a non-linear relationship.

2.4.2 Promotions and seasonality

Various promotions are not uncommon in the domain of retailing. Promotions aim to increase sales during a specific seasonal period through various means, including price reductions, advertising campaigns, or free gifts for a purchase when specific conditions are met (e.g. minimum amount spent). Promotions are typically timed to coincide with seasonal holidays or events, such as Christmas, Thanksgiving, or Black Friday. Sales promotions tend to increase short-term sales and consumption, causing sudden fluctuations in demand patterns. Furthermore, usually after sales surges there is downward trend, which is explained by consumers' stockpiling (non-perishable) goods. Therefore, the consequences of promotions do not merely follow the traditional law of supply and demand, increasing demand as price decreases, but they also have more complex consequences due to consumer dynamics, which further undermines the complexity of demand forecasting. (Abolghasemi, Hurley, et al., 2020, p. 3)

According to Abolghasemi et al (2020), the level of demand can vary significantly during promotions, such as seasonal holiday weeks or campaigns, compared to non-promotional periods. They find that the variation in demand can increase by up to 6000% during different promotions in a high variance time series. They discuss how the behavior of a time series can be explained by an analysis and identification of its features. They analyze six features specific to time series which are: seasonality, stationarity, non-linearity, skewness, kurtosis, and spectral entropy. The first three depict whether the time series is dependent on time, what kinds of seasonal patterns are present, and if the time series is non-linear. The latter depicts the skewness of the trend patterns, i.e., how close the pattern is to normal distribution, kurtosis explains if the distribution is heavy-tailed or light-tailed, and spectral entropy is used to display the unpredictability of the data (2020, pp. 6–7). Their results conclude that uncertainty can be reduced, and volatility controlled by combining forecasts from various models. Some models in their study result in relatively accurate forecasts during non-promotional periods but drastically overfit during promotional periods with high variance. The occasional overfitting results in a low average forecast accuracy. The results also highlighted the

accuracy and efficiency of an artificial neural network (ANN) in time series forecasting. They describe the ANN results contradicting from literature, as it did not generalize well on the volatile time series and proposed that different architecture in the network, choosing relevant data selection (features) and adding data quantity could serve the ANN's performance.

In daily retail the calendric days usually stand for special promotional days rather than public holidays, which are often considered in time series methods. Huber and Stuckenschmidt (2020) conducted research on machine learning (ML) in forecasting of daily retail demand on specific calendric days. The study focused on a company with a large distribution network comprising over 100 individual retailers, each of which requires daily demand forecasts for its business operations. They compared a set of three machine learning models, including MLP as a feed-forward ANN, LSTM (long-short term memory) as a recurrent ANN, and LightGBM representing gradient-boosted regression trees (GBRTs). The models were evaluated by comparison of forecast errors, displayed with MAE and MASE. According to their research (2020, pp. 1435–1437), ML methods provided higher accuracy being more than 10% to 20% more accurate compared to time series models such as regularized linear regression model. Their conclusion was that ANNs were the strongest in forecasting daily demand, with LSTM as the top performer, followed by MLP and LightGBM being the worst of the comparison. However, the models were retained to fit the data, and no sophisticated hyperparameter optimization methods were used in their study. This can increase the probability of overfitting and lack generalization.

2.4.3 Uncertainty and difficulties in demand forecasting

The aim of demand forecasting is to create accurate forecasts for a specific future period to optimize supply as accurately as possible. Accurate forecasts help companies to optimize their operations by reducing excess costs, such as inventory costs or waste. Despite the benefits of accurate forecasts, forecasts aim to minimize error rather than

seek perfect prediction, as forecasts always involve inherent aleatoric uncertainty, which cannot be eliminated. Epistemic uncertainty, however, stems from a lack of knowledge and can potentially be reduced, which is why demand forecasts are produced. According to Hüllermeier and Waegeman (2021, p. 458) uncertainty can be roughly divided into two: *aleatoric* and *epistemic* uncertainty. Aleatoric (statistical) uncertainty is caused by inherent randomness. Aleatoric uncertainty therefore always involves a stochastic factor that cannot be eliminated by any statistical methods. Epistemic (systematic) uncertainty refers to uncertainty caused by ignorance or lack of knowledge. In other words, epistemic uncertainty can be eliminated, whereas aleatoric uncertainty always prevails when making predictions. In statistical fields, uncertainty is traditionally treated as a probabilistic concept, which often fails to explicitly distinguish the types of uncertainty.

Seyedan and Mafakheri (2020) examine supply chain demand forecasting from the perspective of big data analytics and forecasting. According to them, uncertainty is a key problem in supply chains and note that there is a common misconception associated with forecasting where variables such as cost, capacity, and demand are generally known parameters. In reality, these variables are subject to uncertainty due to external factors related to customer demand, deliveries, delivery times, and risks. Uncertainty created by demand plays a significant role, for which forecasting demand is the primary tool to mitigate uncertainty across the supply chain. In addition, demand uncertainty is a significant factor that affects, for instance, process scheduling, planning, and distribution. Forecasting demand is the primary means of reducing supply chain-wide uncertainty and minimizing disruptions caused by uncertainty, such as the bullwhip effect.

Feizabadi (2022) also addresses the risk as a key challenge in supply chains. According to the study, uncertainty mainly occurs in the form of demand uncertainty, which increases the imbalance between supply and demand. The fundamental argument of the study assumes that there are three key factors involved in forecasting demand: model uncertainty, parameter uncertainty, and data uncertainty. Traditional approaches to managing demand uncertainty include pull methods, i.e. make-to-stock (inventory

buffer) and make-to-order production methods. Additionally, advanced forecasting models, such as ANNs, are effective in predictive analysis for their ability to manage large volumes of varying data. Due to the complexity of uncertainty, prediction models that use a single algorithm are unable to address all sources of uncertainty simultaneously, which has led to the use of ensemble and hybrid prediction models are common in the forecast domain. However, when examining uncertainty from a machine learning perspective, Hüllemeier and Waegeman (2021) state that the distinction between the two might be unnecessary. For instance, in supervised learning, where the agent is a learning algorithm, and is forced to provide decisions or predictions, the distinction between the two is irrelevant. However, in some cases where a decision can be postponed or rejected altogether, the scenario might not always apply.

Inefficiencies in the supply chain caused by uncertainty can create cumulative negative effects that accumulate as they move up the supply chain structure. A primary example of this is the bullwhip effect and it occurs when different parties, such as individual companies (Sanders, 2015, p. 13), in the supply chain make their own, often unsuccessful, demand forecasts based on fluctuations in downstream demand. This means that small changes in consumer demand can cause much larger fluctuations in orders and stock levels upstream of the chain, amplifying the demand variance (Tai et al., 2022, p. 5). The phenomenon is exacerbated by poor information flow and inconsistent forecasting at different parts of the organization. For instance, inadequate information flow, long lead times, a distorted view of demand or inefficient inventory management can lead to recurring problems within the organization. The bullwhip effect causes over-stocking and poor customer service (Tai et al., 2022, p. 1) creating uncertainty which depletes operational efficiency (Disney et al., 2021, p. 5810).

Accurate demand forecasting plays a crucial role in reducing the probability of the bullwhip effect. According to Pereira et al. (2021), the cornerstone of creating an accurate forecast is choosing the right forecasting method and using machine learning algorithms to make the forecast, as machine learning provides better demand

predictability, which gives managers a better chance of identifying consumer needs. This in turn encourages managers to make more confident decisions, which in turn minimizes mismatches between supply and demand (Pereira & Frazzon, 2021, p. 11). According to Ganjare et al. (2023), the bullwhip effect can be prevented by careful inventory management, accurate order-up and replenishment strategies. To conclude, the literature suggests that transparent information flow across the supply chain, combined with highly accurate forecasting, is key to preventing the bullwhip effect.

A common conception in forecasting is that the longer the forecast period, the lower the forecast accuracy. Saoud, Kourentzes and Boylan (2025) discuss forecast uncertainty. They base their study on demand uncertainty and the bullwhip effect, emphasizing prevailing uncertainty as the primary driver of costs in the supply chain. However, they also highlight the impact of the forecast horizon length on uncertainty. Based on the study, forecast uncertainty increases as the forecast horizon lengthens. This is because errors accumulate over a longer period of time and the number of parameters affecting the variables to be forecast increases. Cerqueira et al (2025) examine the impact of the forecast horizon on the performance of forecast models. They discuss the robustness of forecast models in volatile environments where anomalies are present. They conclude that when selecting the model for predictive analysis, the best model is not necessarily the best choice for handling unexpected variation. Modern neural network models outperform traditional statistical models only in long-term forecasts. In the short term, there was no significant difference. This emphasizes that the length of the horizon is a critical factor when selecting a model.

3 Methodology and data

This section discusses the different stages of this thesis in chronological order. The objective of this section is to explain the unit of analysis, process steps, data, and tools used to obtain the results. The first chapters discuss how and where the data was acquired, the different stages of data pre-processing, and the statistical methods used in the pre-processing stage. Next, the algorithms selected for this study are presented and how they were utilized. Finally, the comparison of the models is discussed by explaining the selection of error and performance metrics and which data features were the most important for each model in the prediction process.

3.1 Data

The data for this project was sourced from a publicly available dataset from Kaggle.com. The dataset is in tabular format and consists of simulated and anonymized weekly sales data from a U.S. based retail chain. The dataset includes various variables depicting exogenous factors influencing demand in addition to the store-specific features and date columns for temporal information. In this thesis, following standard machine learning terminology, these external influencing factors, as well as temporal and store-specific factors used as inputs for the forecasting models, are referred to as features (Van Wyk, 2023, p. 7). This dataset forms the basis for this study and enables the examination of how these features impact demand forecasts generated by machine learning models.

In the first stage of data preprocessing, the nature of the data is analyzed by validating the data types of the dataset. The dataset consists of 6435 rows and 8 columns; thus, the dataset has a total of 55 480 observations. The data types for numerical features are *float* and *integer,* and *objects* for the date features (Appendix 1). The numerical features are Store, Weekly_Sales, Holiday_Flag, Temperature, CPI, Unemployment and Fuel_Price. The explanations for the variables can be found on table 4 below.

Table 4. Explanations for each variable of the dataset.

Variable	Explanation
Store	Indicates number of the store
Date	The week of sales
Weekly_Sales	Sales for the given store from one week
Holiday_Flag	Indicates whether the week is a special holiday week 1 = holiday week, 0 = non-holiday week
Temperature	Temperature (in °F) of the week during the week for the region of the store
Fuel_Price	Cost of Fuel in the region
CPI	Consumer Price Index
Unemployment	Prevailing regional unemployment rate

Holiday weeks mark the four most prominent holidays in the U.S. which are Super Bowl, Labor Day, Thanksgiving, and Christmas, as defined in the original Kaggle dataset (2021) documentation. An examination of the dataset reveals that the numerical variables *Date* and *Holiday_Flag* require conversion into categorical features to prevent the models from misinterpreting their numerical codes as ordered values. For example, this ensures that *Store 45* is not interpreted as of greater value than *Store 1*. Furthermore, as the *Date* column only refers to the week of sales, *Day, Week, Month,* and *Year*, columns were derived from the *Date* column to provide more accurate values for time series analysis.

3.1.1 Data acquisition and preparation

All data processing for this thesis was performed using Python programming language in Jupyter Notebook. Jupyter is an open-source web-based interactive computing environment used for data science (*The Jupyter Notebook — Jupyter Notebook 7.5.0b0 Documentation*, 2015). The data was analyzed using several libraries: pandas and NumPy for data handling, Matplotlib and Seaborn for visualization, and scikit-learn for machine learning methods. LightGBM and XGBoost were applied for gradient boosting models.

Data processing begins with importing the necessary libraries into the data processing environment, after which dataset file is also read and imported into Python with *pandas* library for further processing.

3.1.2 Data cleaning and preprocessing

After the initial overview of the dataset, the data pre-processing phase continued by checking the dataset for missing values, duplicated rows, and possible outliers. Pre-processing was done to ensure the quality of the raw data to prevent distorted analysis that would result from incorrect or biased data (Çetin & Yıldız, 2022, p. 300). Cleaning the dataset is an essential part of data analysis (Slater & Hasson, 2025, p. 723); thus, data analysis began with data cleaning by detecting missing or duplicated data, and identifying any anomalies in the data. In case there were any, the empty and duplicated rows were deleted from the dataset. Fortunately, the dataset chosen for this project did not include any empty or duplicated data, so no further processing was required at this point of the analysis.

After handling missing and duplicated values, the dataset was checked for possible outliers to ensure a robust data analysis. Outlier detection began with a visual inspection to help select the most efficient and objective method for identifying outliers (Alves et al., 2024a, p. 5). The numerical variables are assessed and analyzed with box plots for outlier visualization and histograms for displaying the skewness of the variable. In addition, outliers were calculated using interquartile range (IQR) method. All detected outliers were further validated to determine whether the deviations were caused by errors or represented natural variation.

3.2 Feature engineering

The next task in pre-processing of data is new feature engineering in which new features are derived from the existing dataset to create new insightful data into a form the machine learning algorithm can benefit from. Kampezidou and others (2024, p. 388) state that new features can be produced by generating, transforming, and combining existing features. The purpose of new feature engineering is to improve the efficiency of used machine learning models by minimizing generalization and training errors.

Some of the new features were created solely for exploratory data analysis. These features were dropped from the dataset before moving forward to the predictive analysis with the machine learning algorithms. This was done to prevent any data leakage during predictive analysis which can result in unreliable results. Data leakage refers to a situation where some of the test data is mixed with the training set, thus resulting in falsely great results on the test data, but decreasing the generalization of the model making the it useless in real world problems (Liu et al., 2022, p. 13).

Lag and Rolling features were engineered as they can increase the performance of a time series analysis especially when using tree-based algorithms (Kampezidou et al., 2024, p. 388). Since the data covers a relatively short period of three years, the Lag and Rolling features were created for time periods of one, four, and eight weeks. The four- and eight-week features were created to reflect time periods of approximately one and two months. Lag and Rolling features can improve the model's training efficiency and performance to predict seasonal patterns and trends (Tam et al., 2025, p. 23).

In addition to the temporal features, Interaction features were engineered to complement the temporary ones, capturing non-linear relationships and providing insight into how the original features of the dataset relate to one another. New features were engineered separately for both EDA and predictive analysis, because not all new engineered features needed for EDA could be used in the predictive analysis due to possible data leakage and biased parameter estimates.

3.3 Train-test split

When conducting predictive analysis on time series data, it is crucial to split the dataset into training and test sets before new feature engineering and training the algorithm. The data split was conducted into training and test sets by preserving the chronological order of the data. The ratio for the train-test split was 65 to 35, meaning 65 % of the data was used as the training set, and the remaining 35 % as the test data, which the models predict. The split was conducted with time series split, as a different split method could alter the time dependency, impacting the reliability of time series analysis.

3.4 Exploratory data analysis

Exploratory data analysis (EDA) was done by descriptive statistics to explore the patterns, possible trends, and overall structure of the dataset, and to visualize outlier detection and the train-test split. EDA started by summarizing the data with tables and bar charts. Histograms were also used to visualize the characteristics of the dataset and the skewness of numerical features. Correlation and correlation coefficients between individual features were visualized by a feature correlation heatmap. Outliers were visualized with boxplots, interquartile range (IQR), and tables, and possible trend patterns were visualized with a 12-week moving average of weekly sales. Holidays and non-holiday dates were visualized with simple pie chart and a histogram.

3.5 Machine Learning Models

The following predictive algorithms were used in this thesis: simple naïve, 52-week seasonal naïve, Multi Perceptron Neural Network (MLP), LightGBM, and XGBoost. Based on previous literature and research results, the gradient boosting models LightGBM and

XGBoost were selected as the primary comparison targets for this prediction. Naïve models were selected as baseline models to provide perspective when compared with more advanced algorithms to display the differences in predictive performances. Since it was assumed that the advanced machine learning models would outperform simple naïve predictions in the comparison, one more multivariate algorithm was selected for comparison to make the comparison more comprehensive. A neural network based algorithm, multilayer perceptron (MLP) was chosen due to its efficiency in processing time series data, as well as computational efficiency when compared to other neural network based algorithms, such as LSTM (Long-Short Term Memory) (Ramos et al., 2023).

The model training process started with pre-processing of the dataset. After the initial check for outliers, empty rows, and duplicates, the dataset was split into train and test sets. Once train-test split was done, new features were engineered to support the predictive performance of used models. The hyperparameter optimization was conducted using randomized search for gradient boosted models (LGBM & XGB) as well as for the Neural Network model (MLP). The models were evaluated and measured using four different metrics, which are mean absolute error (MAE), root mean square error (RMSE), mean absolute scaled error (MASE), and R-squared (R^2).

Root mean square error (Equation 1) measures the magnitude of prediction errors, with lower values indicating better accuracy (Kannadasan, 2025, p. 25).

Equation 1. Equation of root mean square error (RMSE).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - x_i)^2}{\mathrm{n}}}$$

where, n is the number of datapoints, y_i is the actual value for data point i, and x_i is the predicted value for data point i.

Mean absolute error measures the average absolute error (equation 2) between the predicted and actual values, indicating how close the prediction is to the reference point.

Equation 2. Equation of mean absolute error (MAE).

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{\mathrm{n}}$$

where, *n* is the number of datapoints, y_i is the actual value for data point *i* , and x_i is the predicted value for data point *i*.

Mean absolute scaled error (equation 3) indicates the effectiveness of the predictive model by comparing its Mean Absolute error (MAE) with the MAE of naïve forecast (Huber & Stuckenschmidt, 2020, p. 1430).

Equation 3. Equation of mean absolute scaled error (MASE).

$$MASE = \frac{MAE}{MAE_{naïve}}$$

where, *MAE* is the Mean Absolute error of the prediction and, MAE_{naive} is the actual MAE of Naïve forecast.

The coefficient of determination, also referred to as R-squared (R^2), quantifies the amount of variance in the dependent variable can be explained by the independent variables (Chicco et al., 2021, p. 2).

Equation 4. Equation of R-squared (R^2).

$$R^2 = \frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where, $\widehat{y_i}$ is the prediction of datapoint *i*, y_i represents the actual values on datapoint *i*, and $\bar{y}$ is the mean of all the observations (Scikit-Learn., 2025).

3.6 Feature Importance Analysis

Feature importance analysis is particularly important when the subject of analysis is highly dependable on exogenous factors. The feature importance method chosen for this project was permutation feature importance method, as it can be applied to evaluate any model, regardless of its operating principles. According to Yagmur and others (2024), it measures the significance of a feature by how much the model's performance metrics (e.g., MAE, RMSE, R^2) react when the values of a singular feature are randomly permuted, thus cutting its connection to the explanatory variable. In other words, permutation importance method can be used to determine the contribution of external variables on weekly sales.

The method works regardless of the model used, meaning it can be applied to both gradient boosting and neural network models. Permutation method was applied manually to test all models with different feature configurations, to display what features work best together and which features are repeated with the most contribution regardless of the configuration. The testing started with the baseline features of the dataset, after which new engineered features were gradually added into the feature configuration. This allowed the identification of the most important features, by which the final set of variables could be selected.

4 Results

In this section, the results from each section of the empirical part of the thesis are presented. The interpretation of results begins with exploratory data analysis, after which the predictive performance and feature importance metrics are evaluated. The models' results are first evaluated separately for each model, after which the results are analyzed relative to one another. The model performance section delves into the accuracy of the models, presenting the error and forecast metrics in tabular form. The feature importance and interpretation section analyzes how the models utilized external features in their forecasts and provides the results from feature importance analysis. Finally, the prediction accuracies of the models are compared with each other under comparative analysis where the prediction accuracy of each model is visualized with respect to actual sales.

4.1 Data analysis results

The data used in this thesis was collected from an open source. The dataset contains anonymized simulated weekly sales data from forty-five different stores from different regions, as well as exogenous factors implicating prevailing economic factors during the sales week. Before statistical data analysis, the usability of the dataset was ensured through data preprocessing and data validation methods. The steps are as follows: (1) preprocessing and validating the data, (2) outlier detection, (3) data cleaning, (4) new feature engineering, (5) normalizing of the data, and (6) balancing the dataset (Van Wyk, 2023, pp. 7–8).

4.1.1 Exploratory data analysis

Data analysis began with visualizing the dataset to explain and understand anomalies and the statistical nature of the data. Alves and others (2024b, p. 2) state that

exploratory data analysis is essential to gain deeper insight into the dataset as it helps identifying outliers and inliers, as well as deriving information on key variables and features of the data. In this thesis, the exploratory data analysis includes analysis of the time series to understand possible trends and patterns of the sales data. The exploratory data analysis began with a visual examination of the dataset. The nature of the numerical features of the data was examined based on the distribution of the variables. Based on this, it was found that most of the data was close to normally distributed, except for *Unemployment* (see appendix 7), which was heavily skewed. Furthermore, the correlation between the target variable, *Weekly_Sales* and continuous numerical features, such as *Temperature* and *Fuel_Price*, were examined using scatter plots. This preliminary analysis suggested that *Unemployment* data might contain outliers.

Figure 9 depicts the average weekly sales for the dataset, and the orange line represents the 12-week moving average, providing a fundamental perspective on the time series data. The graph clearly indicates that sales peaked before and during December, after which there is a sharp drop in sales right before January. This observation can be explained by the holiday seasons shown in the data, which occur in the last quarter of the year, including Labour Day, Thanksgiving, Christmas, and Super Bowl according to our dataset (2021).

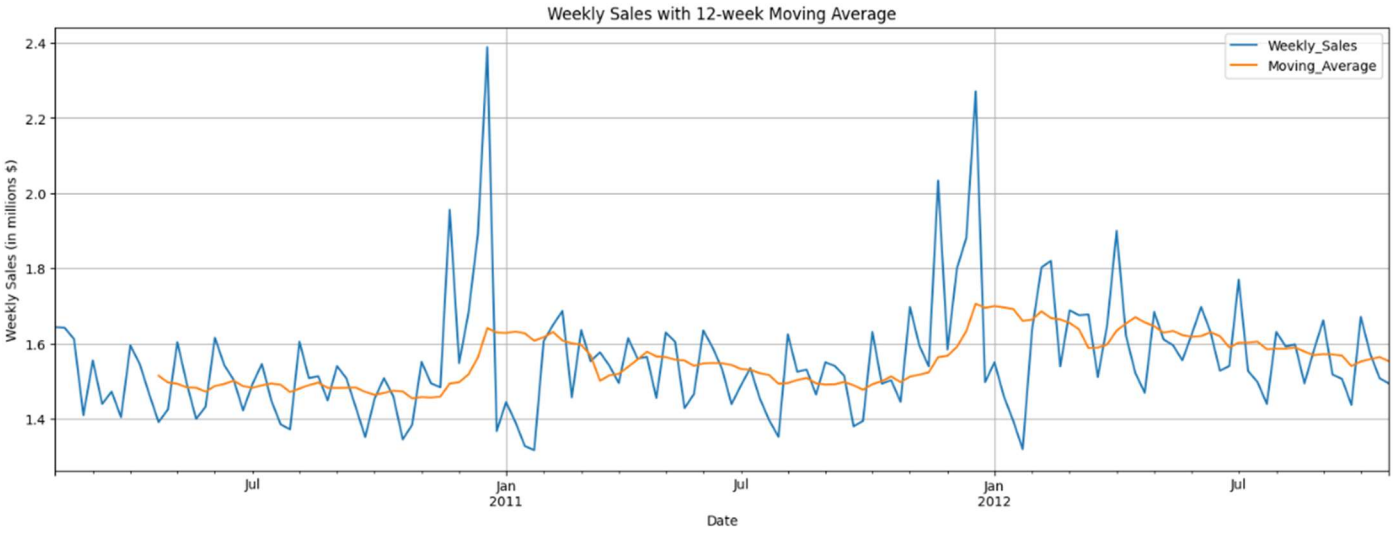


Figure 9. Weekly Sales, 12-Week Moving Average.

The time series analysis provides valuable insight when forecasting sales and demand. Neba and others (2024) state that identifying seasonal trends and patterns is important for forecasting accuracy, resource allocation and inventory management, promotional strategies, and financial planning (Neba et al., 2024, p. 11). As well as improving the above-mentioned, trends can increase consumer uncertainty challenging forecasting accuracy, creating risk for inefficiencies in inventory management (Ratra & Seth, 2025, p. 1024). Furthermore, Caniato et al (2005, pp. 39–40) studied demand variability and divided it into three main drivers, seasonal, promotional, and random (i.e. unpredictable) variability. They state that the variability of demand is mostly derived from the seasonal and promotional variability, further emphasizing the importance of understanding the patterns in the time series data to be forecasted.

Pairwise correlations between features were examined with Spearman Correlation, and the results are visualized by a Correlation Heatmap in figure 10. Correlation heatmap illustrates the correlation coefficients for each presented feature, indicating how strong the correlation between two variables is. The value for the correlation is between -1 and 1, negative values indicating a negative correlation, positive values indicating a positive correlation, and a value of zero means there is no correlation between the variables (J. Li et al., 2024, p. 152). The bar on the right side of the figure explains the color palette, giving a deeper color for a stronger correlation.

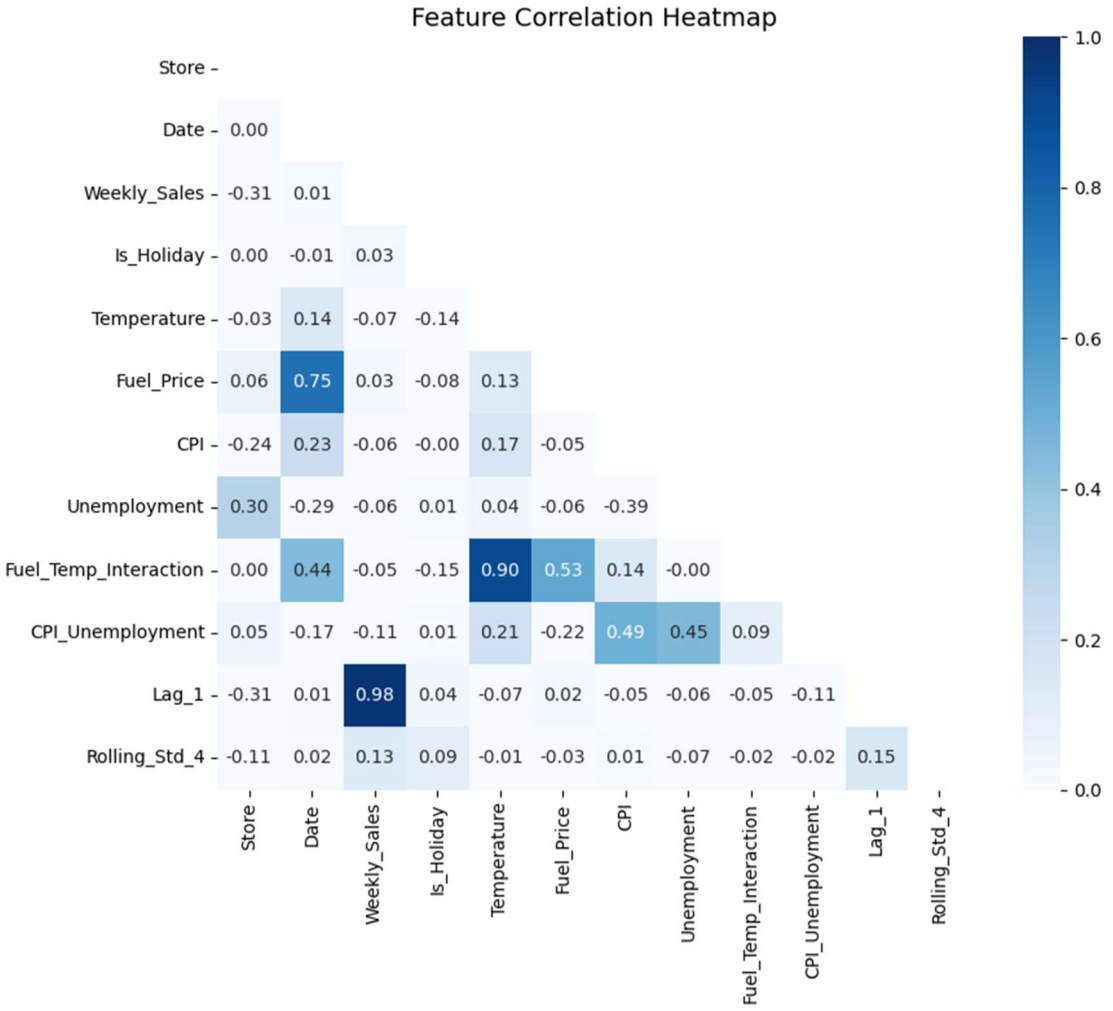


Figure 10. Feature Correlation Heatmap with Spearman's Correlation.

The correlation between the variables is, on average, weak according to the correlation matrix. The matrix shows that the correlation between weekly sales and features, such as *CPI, fuel price, temperature,* and *unemployment,* is between -0.07 and 0.03, i.e. almost neutral. Despite this, the models are expected to find dependencies in the data during the prediction process. It is important to note that the correlation between interaction features, such as *Fuel_Temp_Interaction* with *Fuel_Price* and *Temperature*, are not statistically significant, as *Fuel_Temp_Interaction* was derived from these two features.

4.1.2 Outlier detection

Outlier detection is done to detect unexpected, significantly deviated observations from the data (Reunanen et al., 2020, p. 287) to sustain the usability of the dataset. Multivariate datasets often have many outliers, for which, according to Alves and others (2024b), outlier detection is a principal task during data processing. They emphasize that detecting outliers is often essential, but the outlier detection method must be efficient and objective. It is also said that removing the outliers is potentially imperative, meaning the removal of the outliers is not always necessary (Alves et al., 2024b, p. 5).

The outlier detection method used in this thesis included outlier visualization with boxplots, after which interquartile range (IQR) was applied as a computational method. IQR and Boxplots both use the same computational formula to determine the upper and lower quartiles (Tukey, 1977, pp. 43–44). The interquartile range is defined by subtracting the first quartile from the third (Q3-Q1), and the inner *fences* are set at 1.5 times the IQR (Tukey, 1977, pp. 43–44). As outlier detection is dependent on the context, the flagged values were evaluated considering domain knowledge.

The IQR method was applied by creating a function (see Appendix 1) to calculate the interquartile range and detect outliers for each column passed to the function. The function was subsequently applied to all numerical features in the dataset. Finally, the output for the code displays all the values that fall below lower bound or above upper bound. Outliers detected with IQR are depicted in figure 11.

```
Weekly_Sales: 34 outliers
Temperature: 3 outliers
Fuel_Price: 0 outliers
CPI: 0 outliers
Unemployment: 481 outliers
```

Figure 11. Outlier detection with interquartile range.

Only two of five numerical columns included notable outliers. Before the possible outliers were removed from the data, the reasons for the deviation were checked to ensure that the observations were valid. The possible outliers in *Weekly_Sales* are displayed in figure 12 with a boxplot.

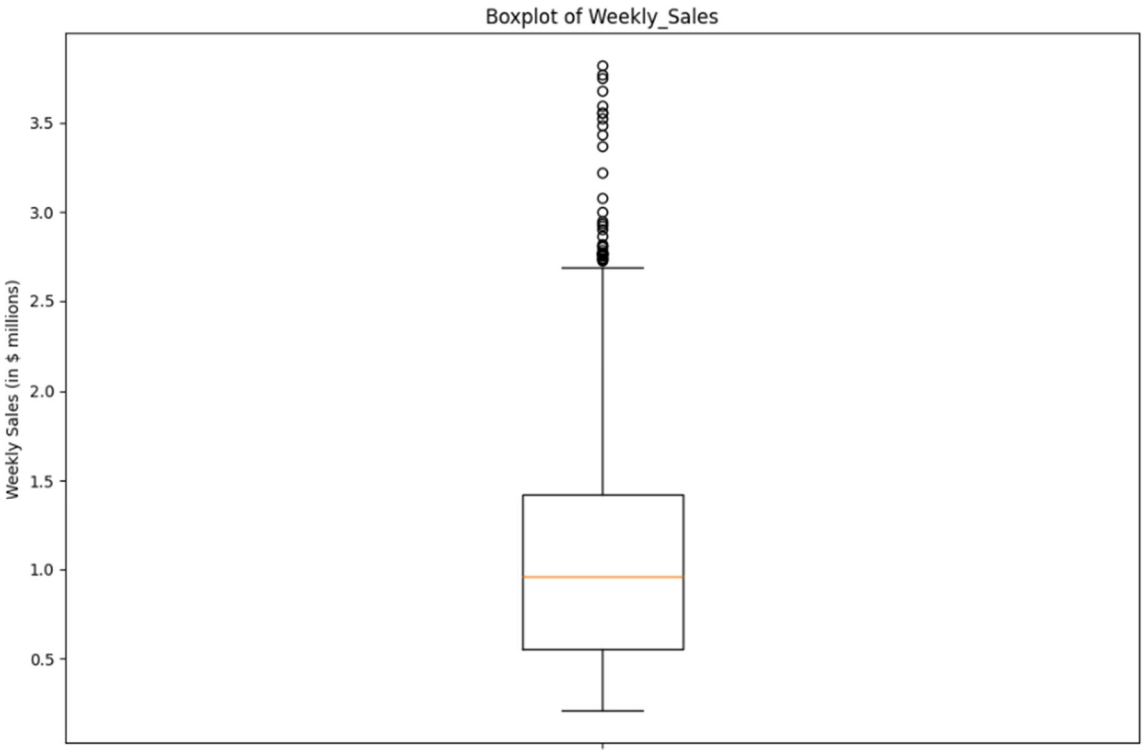


Figure 12. Boxplot of Weekly_Sales.

The dots represent the datapoints that fall outside the upper (or lower) bounds (*fences*) (Tukey, 1977, pp. 43–44). With *Weekly_Sales*, there were 34 datapoints falling outside the upper bound which was 2720371.49 million $ for Weekly Sales. However, the outlier count is relatively small considering the total of 6435 rows. By further examination it became clear that all 34 deviated values occurred in the last quarters of the year, which, based on previous analysis (Figure 2), were when the largest sales peaks occurred. Therefore these 34 outliers were preserved in the data to indicate the seasonal variation in the time series.

Unemployment, on the other hand, had 481 outliers detected by the IQR method. However, while the IQR is effective for identifying extreme values on quartiles (Alves et al., 2024b, p. 5), it is less suitable for non-Gaussian data (Jeong et al., 2017, p. 136) as the method assumes the data distribution is roughly symmetrical. As *Unemployment* is severely skewed, the calculated upper and lower boundaries with IQR method can overly classify skewed values as outliers. The skewness of the feature is visualized in figure 13 below.

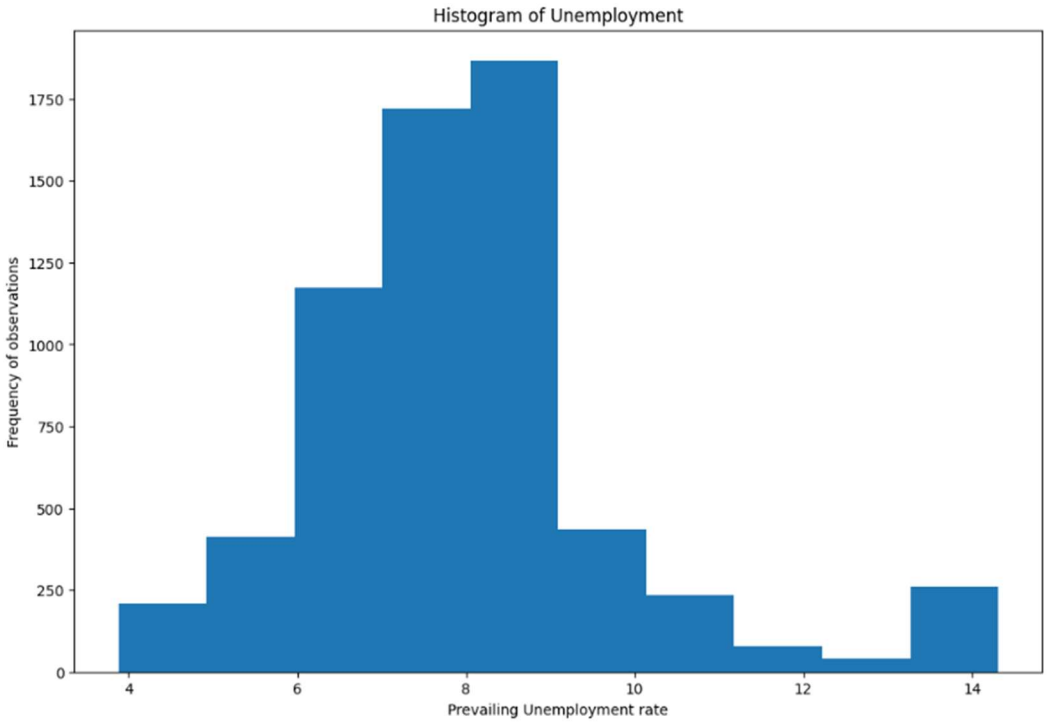


Figure 13. Histogram of Unemployment data.

From figure 13, the skewness of the *unemployment* feature is evident. To further validate the observation, skewness was calculated using the *skew()* function from the SciPy module in Python (Virtanen et al., 2020). The skewness value of the *Unemployment* column was 1.188. According to Trafimow and others (2019, p. 132) skewness of 1.188 means the data is severely right-skewed, confirming what was visually perceived. During the review period, the unemployment rate varies between 14.313 and 3.879, and the number of observations (frequency) is 43 at the maximum value and 4 observations at

the minimum value. However, as *Unemployment* depicts the prevailing unemployment rate of the region during the week of sales, the deviated results are not interpreted as outliers, but rather as real, rare economic situations influenced by underlying economic stress. To conclude, all the detected outliers were retained in the dataset as their deviation could be explained as normal deviations and potentially important threshold values for model learning, based on the context of the features.

4.1.3 Train-test split

Supervised machine learning requires splitting the data into separate training and test sets to ensure reliable predictive analysis and generalizability. Any data leakage would result in unreliable predictive results according to Van Wyk (2023, p. 8). He states that up to 80 % of the data can be used for the training set. As the time series used in this study is relatively short, and includes strong seasonal trends, using the maximum of 80 % for training would leave too little data for testing. Furthermore, with 80-20 split, the training data would include all the largest seasonal patterns, thus most of the variance would be left out from the test set. Ultimately, since the predictive model performance is measured based on predictions made using unseen data (Ensafi et al., 2022, p. 5), it is important that the train-test split is not too skewed. Otherwise, the results could be somewhat biased if the test set is much less variant than the training set, as seen in figure 14.

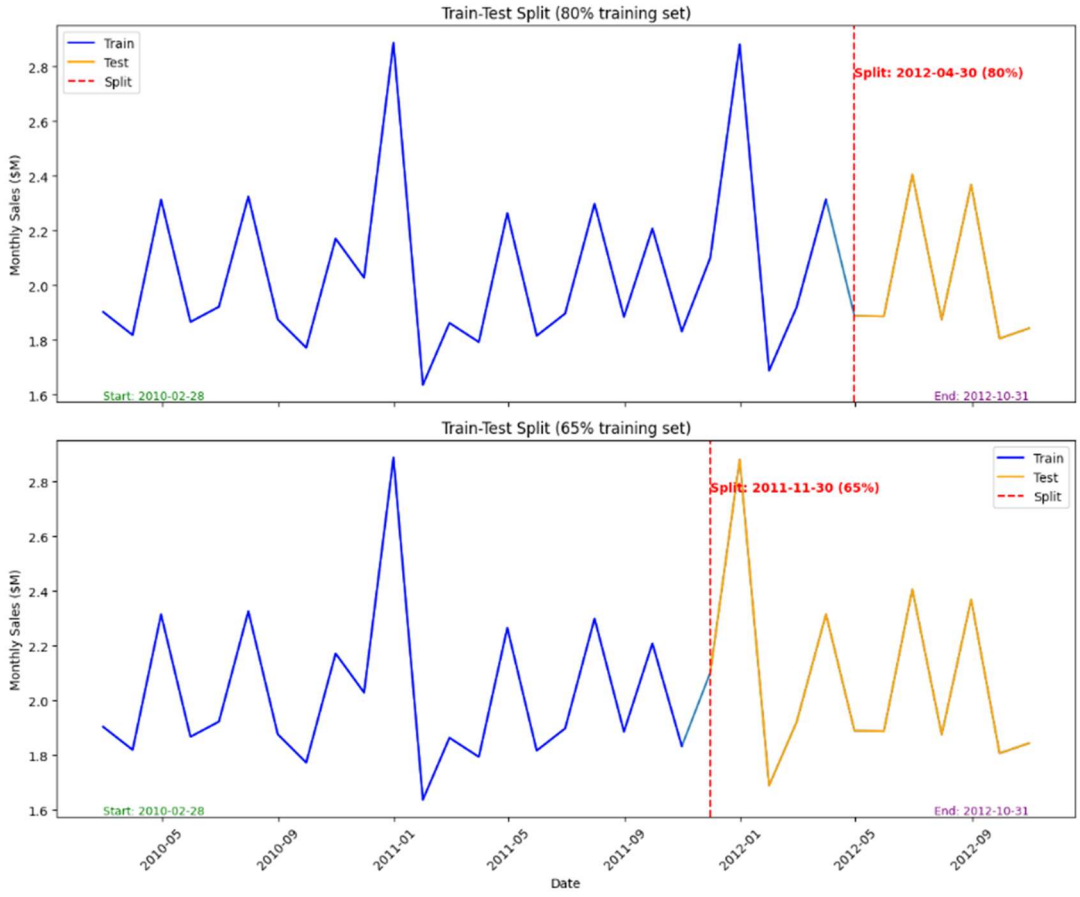


Figure 14. Visualization of train-test split with different ratios (80:20 vs. 65:35).

The line in the figure illustrates the average of the target variable (*Weekly_Sales*) during the review period. The blue line represents the training data, the orange line reresents the testing part, while the red dotted line indicates the point of the time series where the split occurred. The training set begins on February 28, 2010, and ends on April 30, 2012, while the test set contains data from the split date until October 31, 2012. As seen in figure 14, in the 65-35 split the largest fluctuations in the time series are distributed more evenly between the training and test sets, whereas an 80-20 split would provide a more biased environment for the model. Furthermore, for achieving accurate forecasts it is expedient that the training data includes seasonal patterns and trends allowing the predictive models to learn the most important features of the time series (Raghuvanshi, 2024, p. 2988). Thus, according to the forementioned, the 65-35 ratio provides a robust environment for the models to learn and predict on.

4.2 Hyperparameter optimization

Before conducting predictions with machine learning algorithms, the tuning parameters must be optimized to fit the specific dataset. The main objective in hyperparameter tuning is to find an appropriate set of parameters to optimize the model performance and to avoid any over or under fitting (Ganguly & Mukherjee, 2024, p. 886; Nematzadeh et al., 2022, p. 2). Nematzadeh and others (2022) state that adjusting parameters improves the efficiency of ML algorithms. They also emphasize that the selection of ML algorithms being data specific. However, the answer to which parameters should be used at what values cannot be found in current empirical research (Nematzadeh et al., 2022, p. 2) which highlights the challenges of hyperparameter tuning. To clarify, although the theory on how each parameter modifies the model's data processing, there is no clear answer as to when and on what scale certain parameters should be used. Parameter tuning is therefore case-specific, influenced by factors such as the amount and quality of data, the architecture of the model used, and computational limitations.

Mitigating overfitting is done to achieve a generalizable model. Brigato and Iocchi (2020) state that any model that is too complex, i.e. if the tuned parameters are not optimized, is prone to overfitting. They stress that complexity of the models used is heavily affected by the type and especially size of the dataset. However, their research implies a scenario where a complex algorithm with the correct parameters can work with small datasets, which is important in situations where only a limited amount of data is available.

Different algorithms have different tuning parameters, depending on the operating principle of the algorithm. For instance, since LightGBM and XGBoost are both based on decision trees, their tuning parameters have similarities, but are not completely identical. MLP, on the other hand, as a neural network has unique tuning parameters compared to gradient boosting models. According to Ganguly and Mukherjee (2024) minimizing RMSE score is one of the objectives of hyperparameter optimization. They state that for models

utilizing decision trees (e.g. lightgbm, xgboost), hyperparameters such as the number of estimators, maximum tree depth, and minimum samples for each node splitting are key parameters and crucial to tune for achieving a good fit to the data.

4.2.1 Hyperparameters of LightGBM and XGBoost

According to Van Wyk (2023) decision tree-based models are prone to overfitting due to overly complex and powerful tree structures. They propose that parameters managing the models' tree complexity and number of leaves is key to avoid overfitting. Ganguly and Mukherjee (2024) support this statement by examining the parameters of another decision tree-based model (Random Forest). The model complexity can be controlled by maximum tree depth (*max_depth*), minimum samples split (*min_samples_split*), number of estimators (*n_estimators*), and number of leaves (*num_leaves*) (Ganguly & Mukherjee, 2024, p. 886; *Parameters — LightGBM 4.6.0 Documentation*, 2025; Van Wyk, 2023, p. 21).

The hyperparameter search was conducted with randomized hyperparameter search. A grid of parameters was selected manually for randomized parameter search (see Appendix 1). The parameters in the parameter grid were selected with an emphasis on minimizing the possibility of model overfitting. After running the model with parameters set by randomized search, the value on *max_depth* was lowered from 4 to 1 to prevent overfitting. Additionally, decimal numbers were rounded up to the nearest decimal place. The optimized parameters were subsequently applied to the LightGBM model.

```
final_lgb = lgb.LGBMRegressor (learning_rate=0.06,
                               n_estimators=224,
                               num_leaves=44,
                               min_data_in_leaf=16,
                               max_depth=1,
                               reg_alpha=0.01,
                               reg_lambda=0.04,
                               random_state=42,
                               verbose=-1)
final.lgb.fit(X_train_full, y_train_full)
y_pred_test_lgb = final_lgb.predict(X_test)
```

Code 1. LightGBM with optimized parameters.

The same parameter optimization method was used for XGBoost to make the comparison of models as objective as possible. The parameter grid and results can be seen in Code 2.

```
final_xgb = xgb.XGBRegressor(n_estimators=372,
                             learning_rate=0.01,
                             max_depth=2,
                             subsample=0.9,
                             colsample_bytree=0.8,
                             reg_alpha=1.25,
                             reg_lambda=2.6,
                             min_child_weight=10,
                             gamma=0.4
                             tree_method="hist",
                             enable_categorical=True,
                             n_jobs=-1,
                             random_state=42,
                             verbosity=0)
final_xgb.fit(X_train_full, y_train_full)
y_pred_test_xgb = final_xgb.predict(X_test)
```

Code 2. XGBoost with optimized parameters.

The hyperparameters in XGBoost can be divided into two categories, the ones managing the learning process and those that control tree complexity. The parameters *learning_rate* and *n_estimators* control the quantity and contribution of trees in the boosting process (*XGBoost Parameters — XGBoost 3.0.5 Documentation*, 2022). In contrast, parameters *max_depth*, *subsample*, and *colsample_bytree* control the complexity of individual trees within the model and help prevent overfitting (Chen &

Guestrin, 2016, pp. 787–793; *XGBoost Parameters — XGBoost 3.0.5 Documentation*, 2022).

4.2.2 Hyperparameters of MLP

Like tree-based models, ANNs (e.g., MLP) are also highly efficient in managing large quantities of data and creating accurate predictions. Lainder and Wolfinger (2022) state in their paper about time-series forecasting that both, gradient boosted models and neural networks have taken over the field due to their efficiency and widespread availability. Both can learn causal relationships from nonlinear data exceptionally efficiently, which makes them prone to overfitting. However, the distinction between the types is that ANNs are known to work best with larger data volumes due to their architectural complexity (Brigato & Iocchi, 2020, p. 1; Ramos et al., 2023, p. 683). The parameter grid for MLP was selected using the same approach as for gradient boosting models, so that the results of the models would be as objective as possible and thus comparable despite the models' architectural differences.

The parameters can be divided into three categories based on their functions. The architecture (complexity) of the neural network is controlled by *hidden_layer_sizes* and *activation* parameters. *Solver, learning_rate*, and *max_iter* control the optimization of the model, e.g., how the model learns from the training data. The rest, *alpha* and *early_stopping* work as the regulators, helping to prevent overfitting (*MLPRegressor*, 2025). The randomized parameter search provided the following parameter values, except that the hidden layers were increased from two to three, and the number of neurons decreased from 40 and 20, to 25, 20, and 10 neurons.

```
mlp_model = MLPRegressor(
                activation='relu',
                hidden_layer_sizes=(25, 20, 10),
                max_iter = 200,
                learning_rate_init = 0.007,
                learning_rate = 'constant',
                alpha = 0.454,
                momentum = 0.64,
                tol = 0.001,
                validation_fraction=0.3,
                batch_size = 41,
                early_stopping = True,
                random_state = 42)
mlp_model.fit(X_train_scaled, y_train)
```

Code 3. MLP with optimized hyperparameters.

The neural network model is ready once the hyperparameters have been adjusted to the data being processed. The hyperparameters were not altered once they were tuned to fit the data. If the parameters were changed, for instance, for different feature sets, the results would be biased and not comparable.

4.3 Models' predictive performance

In this section of the thesis the results for all the models are examined and evaluated. The models in comparison are Light Gradient-Boosting Machine (LightGBM), Extreme Boosting Machine (XGBoost), and Multilayer Perceptron Network (MLP). Seasonal (52-week) Naïve forecast is used as the baseline for the comparison. The main evaluation of the models is done by comparing their forecast accuracy and margin of error in relation to the Naïve forecast and the actual sales results.

4.3.1 Baseline (seasonal naïve)

A baseline model is used to provide a benchmark forecast to which the advanced models can be compared (Micallef et al., 2025, pp. 11–13). The baseline model used in this thesis is seasonal Naïve forecast. The seasonal Naïve method assumes that the forecast for

each future period is equal to the observed value from the same period in the previous season (Micallef et al., 2025, p. 5). For instance, if the weekly sales in the first week of January last year were 100 units, a 52-week seasonal naïve forecast would assume 100 units will be sold during the first week of January this year. Adapted from Micallef and others (2025) the formula for seasonal naïve is displayed in Equation 5.

Equation 5. Equation of seasonal naïve forecast.

$$F_t = A_{t-h}$$

Where F_t is the forecast for time t, A_{t-h} is the actual observed value at time $t - h$, and h is the length of seasonality.

For the baseline model, only RMSE and MAE were calculated for evaluation. MAE is required for calculating MASE (Equation 3), which calculates the errors relative Naïve benchmark (Huber & Stuckenschmidt, 2020, p. 1430). R^2 is not calculated for the naïve forecast, as it just shifts past observations and is not fit with regression, thus making R-squared meaningless.

The seasonal naïve was set at 52 weeks to represent a year-long period. This ensures the inclusion of seasonal trends into the review period. However, the 52-week period is a relatively long forecast period for a time series that has only been collected from three years, which is why 26-week seasonal naïve was considered. The MAE for 52-week seasonal naïve score was 649,930.57 meaning that with the 52-week seasonal naïve the forecast would be typically off by approximately $ 650 000 per week. RMSE score was 815,004.55, meaning that the magnitude of error was even higher (approx. $ 815 000) than what simple MAE would suggest (Tam et al., 2025, p. 17), indicating large variability in the forecasted time series. When changing the length of the season from 52 to 26 weeks (*season_length=26)* the results were slightly worse. The MAE for 26-week seasonal naïve was 678,609.79 and RMSE was 841666.95 meaning that the full-year

seasonal naïve matched the data better, indicating that the seasonality in the dataset is closer to full than half-a-year.

4.3.2 Light gradient-boosting machine (LightGBM)

The Light Gradient-Boosting Machine was efficient in learning from the training set, as it works well with multivariate datasets and time series forecasting. Unlike the naïve forecast, LightGBM was evaluated using RMSE and MAE as well as R^2 and MASE. The robustness and performance of LightGBM is explained by first displaying the cross-validation scores and finally the actual predictions.

The cross-validation results indicate model's ability to learn and capture the variance of the data efficiently, as well as generalize well on unseen data. During cross-validation, the mean MAE score was $ 83806.63. This means that on the five different cross-validation folds the difference between the model's prediction and the real sales is approximately $ 84000 on average. The mean RMSE provided a value of $ 127889.68 indicating larger outliers, as RMSE penalizes errors more than MAE (Kannadasan, 2025, p. 25). This observation was expected, as the exploratory data analysis had already indicated that the time series reveals substantial variance.

The cross-validation results (Table 5) were substantially different across the folds, with Fold 2 standing out as an outlier. Its RMSE (317,734) is approximately 148 % higher than the cross-validation mean (128,213) and over 4.5 times worse than the best fold (Fold 4). This suggests that the second fold contains challenging samples as the results deviate from the others. When excluding fold 2, the average error (RMSE) decreases by approximately 38 %, highlighting fold 2 as an outlier.

Table 5. LightGBM cross-validation results.

LightGBM	RMSE	MAE	MASE	R-squared (R^2)
Fold 1	$ 82392.12	$ 59712.35	0.097	0.9748
Fold 2	$ 317743.44	$ 195773.78	0.322	0.7725
Fold 3	$ 108890.40	$ 75061.14	0.118	0.9580
Fold 4	$ 56748.91	$ 41443.07	0.067	0.9884
Fold 5	$ 75288.67	$ 47765.60	0.078	0.9803
Mean	$ 128212.71	$ 83951.19	0.136	0.9348

However, despite the variance in results between CV folds, the model seems to perform well with the data. The mean cross-validation results highlight the model's ability to capture the variance of the data better than those of the seasonal naïve prediction model. Furthermore, the comparison of the CV and test results demonstrates the robustness of the model. The final test results are displayed in table 6 below.

Table 6. LightGBM final test results.

LightGBM	RMSE	MAE	MASE	R-squared (R^2)
Test results	128353.54	77207.66	0.126	0.9504

At first glance, when comparing the test results with mean CV results, the results suggest overfitting or data leakage as the test results look substantially better than mean CV results. However, the forementioned deviating values in the CV results explain the gap between the test and CV results, and when excluding fold 2 the mean RMSE drops to $ 75,280.53 being much closer to the test results. This indicates the model generalizes well, but the heterogeneity between CV folds refers more to the difficulties of the model due to an outlier rather than to an overfitting model.

The MASE scores display the performance of the model with respect to the baseline model. The average MASE of 0.126 indicates that, on average, the model's mean absolute error was only ≈0.13 times of what it was with the naïve forecast.

The R^2 results indicate that the model has very strong predictive power, stating that LightGBM is efficient in capturing the variance of the data and can explain ≈95 % of the variance in the target variable. Despite the low average correlations between individual features (Figure 10), the model effectively explains the variance in the data, emphasizing the quality of the result.

4.3.3 Extreme gradient boosting (XGBoost)

Extreme Gradient Boosting (XGBoost) resulted in comparable results with LightGBM, in accordance with how the model managed the cross-validation folds. The CV results indicate XGBoost also had issues with the second fold, where the model achieved the worst results between the folds. XGBoost CV results are displayed in table 7 below. However, XGBoost results display a strong predictive performance and robustness when compared to validation and test results.

Table 7. XGBoost cross-validation results.

XGBoost	RMSE	MAE	MASE	R-squared (R^2)
Fold 1	80960.56	61541.16	0.099	0.98
Fold 2	318722.2	186644.2	0.307	0.77
Fold 3	95999.97	69112.72	0.109	0.97
Fold 4	69850.13	53275.91	0.086	0.98
Fold 5	88089.21	62003.79	0.101	0.97
Mean	130724.4	86515.55	0.1404	0.93

The values during the cross validation indicate strong predictive power of the model. Despite the high fluctuation during the second fold, the mean cross validation results display a strong understanding of the variance in the data. The results show a reasonably high accuracy in prediction of weekly sales. However, the mean RMSE (≈ 30,700)

indicates occasional bigger deviations when compared to mean MAE (≈86,500). When excluding the second fold, the average error of cross validation decreases by approximately 30-35 % for both RMSE (≈83,700) and MAE (≈61,500).

MASE indicates significantly better performance compared to the baseline model already during cross validation. The average MASE during cross validation shows that on average XGBoost resulted in 0.145 times absolute error (MAE) than the naïve forecast. Furthermore, on average during cross validation the model can explain ≈94 % of the variance of the target variable. Over 90% accuracy with R^2 is a sign of a strong model in terms of understanding and learning the variance in the data (Chicco et al., 2021, p. 7).

Table 8. XGBoost final test results.

XGBoost	RMSE	MAE	MASE	R-squared (R^2)
Test results	132975.74	82399.26	0.134	0.9467

The test results display impressive performance of the model highlighting its predictive accuracy. The high R-squared value underlines the model's effectiveness in capturing the variance in unseen data. The model's absolute error (MAE) is relatively low at about $ 82,000, compared to an average weekly sale of over one million. The impressive performance maintains when examining the RMSE, which is approximately $ 132,900 on the test set. The result establishes the forementioned variance in the data, but like MAE, it is a highly satisfactory result. As in the validation set, when predicting unseen data, XGBoost outperforms the naïve benchmark, with a mean absolute scaled error (MASE) of 0.134.

4.3.4 Multilayer Perceptron (MLP)

The Multilayer Perceptron (MLP) was the only model based on neural network architecture in this thesis. As a complex model, MLP was prone to overfit the dataset

used in the analysis. Additionally, the training efficiency of the model was negatively affected by the long training times, which further complicated the training of the model. However, after tuning the hyperparameters to prevent overfitting, the model displayed adequate efficiency and improved robustness.

Fold 1 was excluded as an outlier from the cross-validation examination due to insufficient training data. With only 637 observations during fold 1, MLP produced an R-squared value so low (R^2 ≈-0.66) it indicated a failed result. The rest of the cross-validation results are displayed in the table below.

Table 9. MLP cross-validation results.

MLP	RMSE	MAE	MASE	R-squared (R^2)
Fold 2	401848.88	294938.19	0.49	0.6362
Fold 3	257896.77	195549.35	0.31	0.7647
Fold 4	74370.51	57179.46	0.09	0.9801
Fold 5	102479.59	73128.41	0.12	0.9636
Average	209148.94	155198.85	0.25	0.8361

The cross-validation results are adequate, albeit volatile. The results indicate moderate difficulties in generalizing the data, which would indicate an incompatibility between the model and the dataset. The differences between the best and worst folds are significant. In fold 2 MLP was only 0.49 times more accurate than the naïve model, and MAE was ≈295,000, meaning that with this result the forecast on weekly sales would be $ 295 000 off on average. RMSE value further elaborates the magnitude of error during this fold, as it is ≈36% higher than MAE. On the other hand, as the training size of the fold increased, the results improved on fold 4 and fold 5, which indicates efficient predictive performance.

Table 10. MLP final test results.

MLP	RMSE	MAE	MASE	R-squared (R^2)
Test results	153996.36	97881.99	0.1548	0.9285

Multilayer Perceptron (MLP) performed well with the test data, producing commendable prediction results. The model can explain ≈93 % of the variance of the target feature indicating high learning efficiency. The Root Mean Square Deviation of ≈154,000 is respectable, when compared to the average value of over 1000,000 of weekly sales in the dataset. MAE being ≈30 % lower than RMSE refers to large prediction errors during the prediction on the test set. The model outperforms the baseline (naïve) model, which was to be expected. This observation is attributed by MASE of 0.1596, explaining that the absolute error of MLP was only ≈16 % of what it was with the seasonal naïve prediction method.

4.4 Feature importance and interpretation

Supervised machine learning models used for multivariate forecasting often benefit from assessing feature importance to select an optimal set of independent variables. The selection can significantly affect the model's performance and predictive accuracy, especially when forecasting a variable whose behavior is influenced by external factors (Javed & Akhtar, 2024, p. 2). In this thesis, the importance of individual features was conducted with the permutation feature importance method. Permutation importance evaluates the contribution of each feature by comparing the model's prediction error before and after randomly shuffling (permuting) the feature values (Yagmur et al., 2024, p. 2471). This section presents the results that aim to answer how the algorithms evaluate external features and how the feature selection affects predictive accuracy of each model.

The feature sets were evaluated, and the features with the most contribution for prediction accuracy were selected for the final set. The configurations were evaluated with baseline hyperparameters for each model, and the features were selected based on the results obtained by manually comparing the performance of the models with different sets of features. The selection began by testing the original features of the data, after which the new engineered features were gradually added into the mix. The performance was evaluated solely by the error metrics (MAE & RMSE) and the permutation importance scores (see Appendices 4-6).

Main summary based on the evaluation was that whenever 1-week lag feature was present in the feature set, it was the most influential according to permutation test. Another clear observation was that any of the interaction features (e.g. *Fuel_Temp_Interaction*) had close to no contribution to models' predictive performance, thus all the interaction features were left out from the feature set. Also, temporal features such as *day* and *year* were quickly discarded, as the data covers weekly sales and only covers three different years. The final feature set is displayed in table 11 below.

Table 11. Feature configuration.

Feature	Explanation	Source
Store	Indicates number of the store	Raw data
Month	The month of sales	Derived from 'Date' column
Holiday_Flag	Indicates whether the week is a special holiday week 1 = holiday week, 0 = non-holiday week	Raw data
Temperature	Temperature (in °F) of the week during the week for the region of the store.	Raw data
Fuel_Price	Cost of Fuel in the region	Raw data
CPI	Consumer Price Index	Raw data
Unemployment	Prevailing regional unemployment rate.	Raw data
Lag_1	Weekly sales from 1 week earlier.	Derived from 'Weekly_Sales' column
Lag_4	Weekly sales from 4 weeks earlier.	Derived from 'Weekly_Sales' column
Rolling_Mean_4	Four-week rolling average of weekly sales.	Derived from 'Weekly_Sales' column
Rolling_Std_4	Four-week rolling standard deviation of weekly sales.	Derived from 'Weekly_Sales' column

The lag and rolling features were engineered using *shift* function from Python's pandas-library. To conclude the selection of the feature configuration, *Date* and *Weekly_Sales* were excluded because *Date* column was further engineered to more specific temporal features (*Week* & *Month*) and *Weekly_Sales* is the target variable. Lag and rolling features were derived from *Weekly_Sales* after splitting the data into test and training sets to prevent any data leakage.

The permutation results highlight how all three models, regardless of differences in their operating principles, emphasize the features capturing temporal and correlative dependencies. The models were tested with different feature sets to examine dependencies between the models and the features, from the basis of which the best feature set was chosen. Based on the tests, despite differences between the models, certain features recurred among the most important ones from one test round to another. On the contrary, a few variables that were assumed to affect prediction accuracy appeared to be almost unnecessary for the models created for this project. As an example, LightGBM and the neural network based MLP had more similarities than LightGBM and XGBoost despite both being tree-based models. LightGBM and MLP shared four of the five most important features by permutation importance, albeit in a different order.

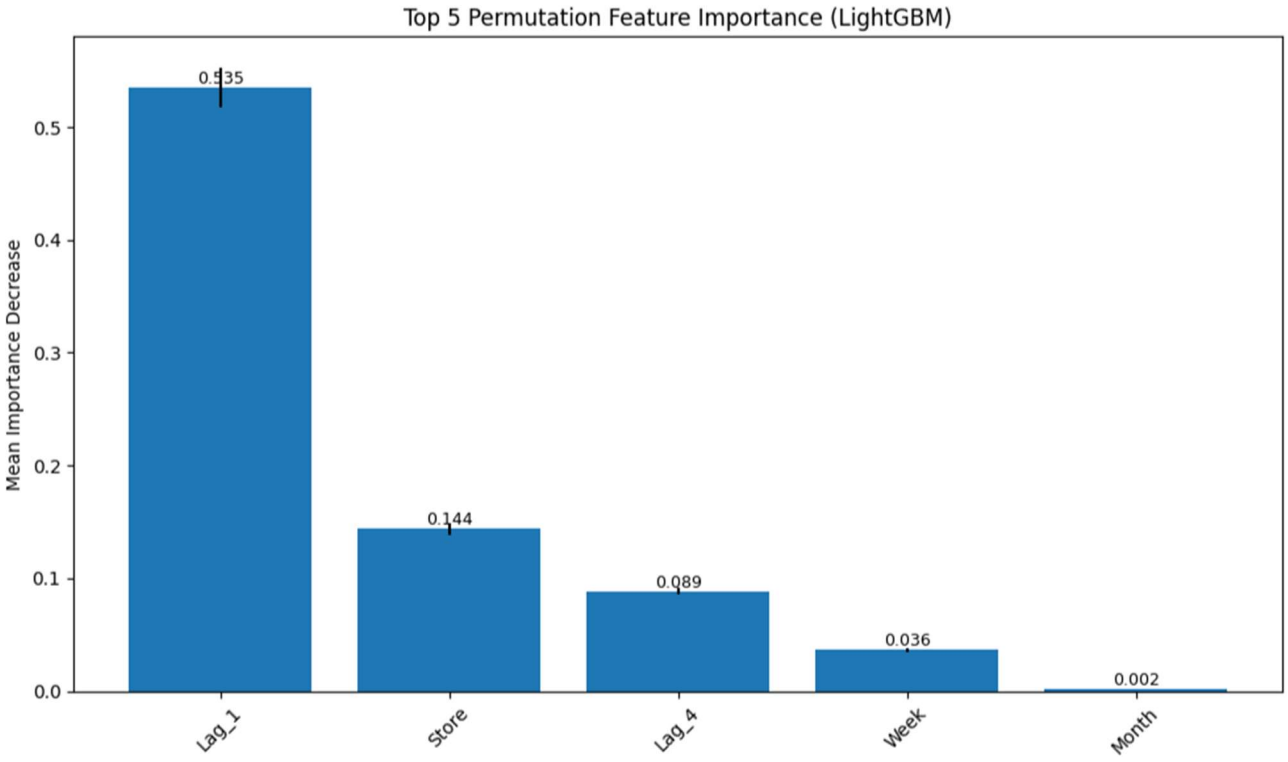


Figure 15. LightGBM permutation feature importance.

The permutation importance results display how lag and temporal features have the most contribution in LightGBM's prediction. The most important feature for LightGBM was the *1-week lag*, followed by *Store*, *4-week lag*, *week*, and *month*. Including the short-

term lag features in the feature set provides the best overall predictive performance. The remaining seven features outside the top five, assigned an average importance decrease score close to zero, indicating their contribution being negligible. For instance, rolling and interaction features offered only marginal benefit, indicating LightGBM's dependency on recent lagged signals for accurate forecasts (Appendix 4).

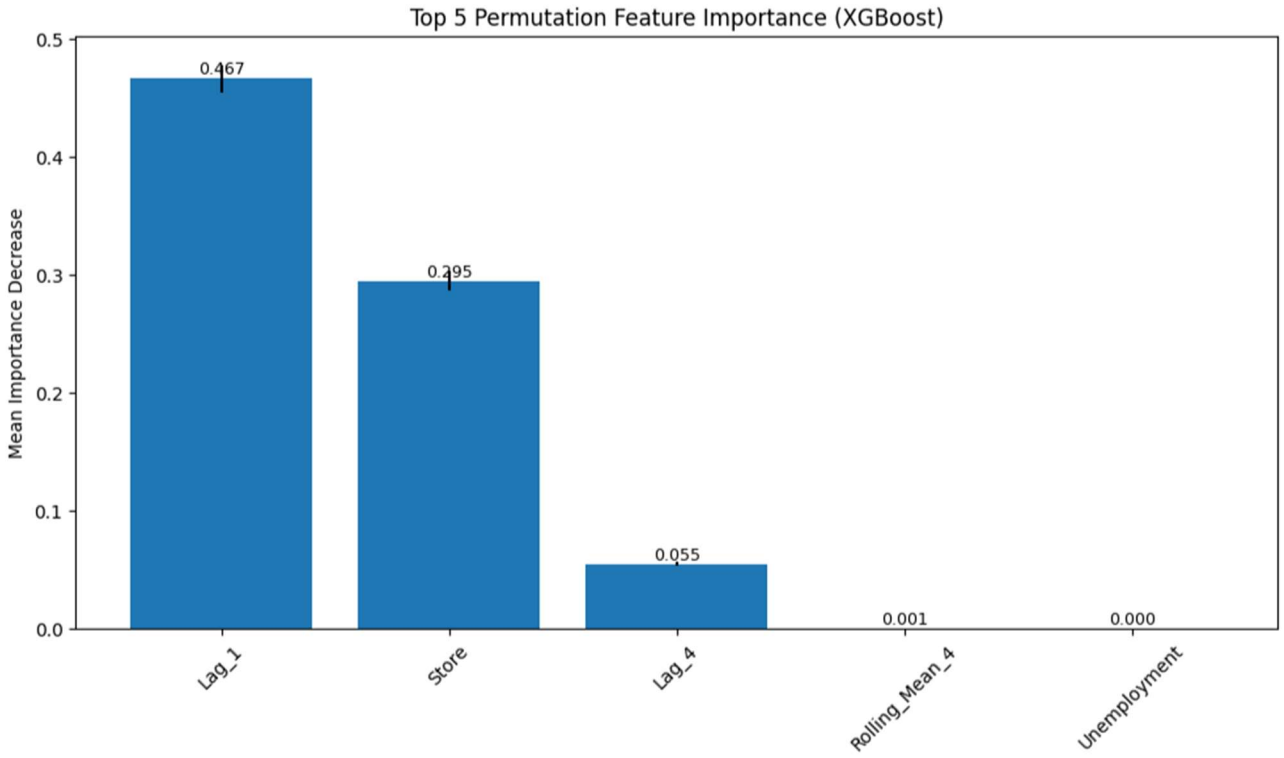


Figure 16. XGBoost permutation feature importance.

XGBoost was the model with the most variability in its feature importance scores. Unlike LightGBM and MLP, both temporal features (*Week & Month*) were excluded from the top 5 most influential features. Despite the differences, XGBoost and LightGBM had the top features on the list, *Lag_1* being the most influential, followed by *Store*, and *Lag_4*. XGBoost relies heavily on its top 3 features when conducting predictions, while the remaining features had contributions close to zero, according to the permutation importance scores. Similar to LightGBM feature importance scores, XGBoost is dependent on key temporal dependencies provided by one- and four-week lag features. Non-temporal features provided only marginal benefits to the predictive accuracy of the

model. Even the longer-term, eight-week lag feature provided only a limited value to XGBoost predictive accuracy (Appendix 5).

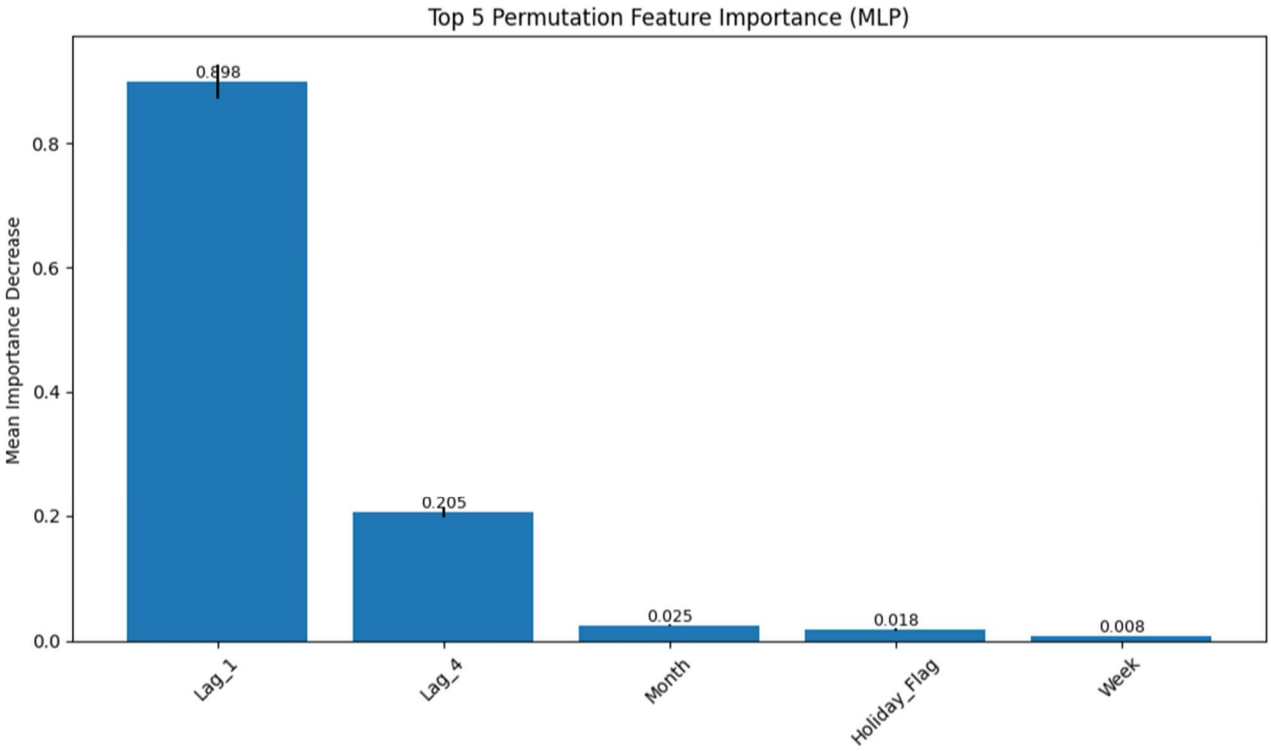


Figure 17. MLP permutation feature importance.

The MLP was highly sensitive to lag features. According to the permutation importance scores, 1-week lag had the most contribution also to MLP's predictive performance, despite MLP's structural difference compared to XGBoost and LightGBM. On the other hand, MLP was the only model for which the feature indicating seasonal holidays showed any contribution on forecast accuracy. MLP's forecast is heavily driven by the lag features it underfits the data. Other than lag features showed no significant contribution to the prediction accuracy when the connection to the target variable was cut by permutation. Thus, temporal inputs were critical for MLP predictive performance (Appendix 6).

The feature selection had a decisive impact on the predictive performance of all three models. In time series analysis, features capturing temporal dependencies and correlations within the data are crucial, as it helps to identify repeating trends and

patterns (Kannadasan, 2025, p. 21). The optimal lag length (e.g. 1-week, 4-week, or 8-week) should be evaluated and chosen depending on the strength of the correlation between the lagged feature and the target variable (Kannadasan, 2025, p. 21). In this project, the optimal lag length was 1 week, according to permutation feature importance scores. Indeed**,** the test results showed that lag features dominated the feature importance scores for each model. According to the results, none of the models used in this project base their forecasts on external factors. This may be due to the fact that the time series examined is subject to such strong seasonal trends and patterns, on which the models base their forecasts.

4.5 Comparative analysis

This section summarizes the final prediction results for each model, highlighting comparisons with the actual weekly sales data and across models. The prediction results are first presented on a graph to visualize the error for each model in comparison to the actual sales. The error and performance metrics (RMSE, MAE, MASE, and R^2) of the ML models are summarized in a single table to enable a clear and direct comparison of their performance.

4.5.1 Comparison of predictions and actual sales

The models were trained and tested using supervised learning on labelled data. Accordingly, the original dataset was initially split into training and test sets, with the training set being used to train the model, based on which the model then made its predictions for the test period. The test set was not shown for the data during the training phase to prevent data leakage, which would compromise the reliability of the prediction results. Compared to the baseline model, all models produced accurate forecasts. Based on mutual comparison, tree-based models were more effective than the ANN based MLP. The differences between the models are clearly visible in the graph below (Figure 18).

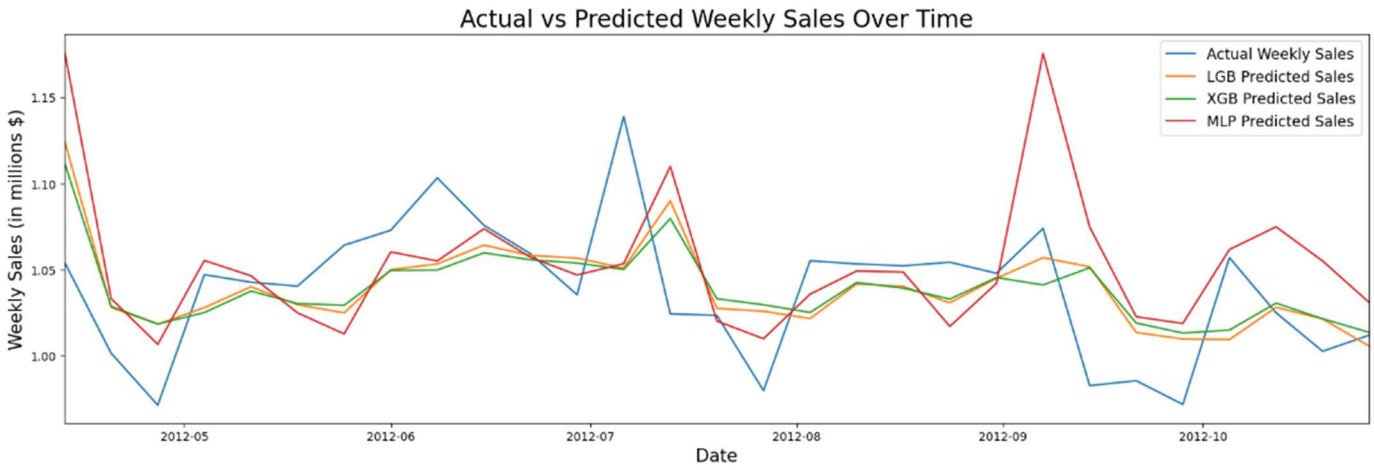


Figure 18. Actual weekly sales vs. predicted weekly sales.

The graph shows how the forecasts generated by tree-based models (green and orange lines) are aligned, showing minimal differences in prediction accuracy. On the contrary, the red line indicates how the MLP has the most variance in its prediction, especially during the 2012 forecast. Furthermore, evaluation of performance and error metrics are displayed in table 12 below. The error metrics used were Root mean square error (RMSE), Mean Average Error (MAE), Mean absolute scaled error (MASE), and R-squared (R^2) to measure how well the model can explain the variance in the target variable (Weekly_Sales).

Table 12. Comparison of the models' error (RMSE, MAE, MASE, R^2).

	RMSE	MAE	MASE	R-squared (R^2)
LightGBM	128353.54	77207.66	0.126	0.9504
XGBoost	132975.74	82399.26	0.134	0.9467
MLP	153996.36	97881.99	0.1596	0.9285
Naïve	815004.55	649930.57	-	-

The most robust values are marked with green, the second best with orange, the worst values on red, and the baseline values on grey for interpretation purposes. According to

the error and performance metrics, LightGBM was the best overall performer. The results argue that LightGBM had the lowest mean absolute error (MAE) in the forecast, it was able to explain the variation in the outcome most effectively (R^2 ≈95 %), and its error compared to the Naïve forecast was the smallest (MASE =0.126). However, although the RMSE result was lowest, the difference between RMSE and MAE was over 66 % indicating that errors in LightGBM's prediction were influenced by large, sudden variations (sales spikes) in the time series, suggesting its lower performance with sudden fluctuations.

XGBoost was the second-best performer according to the comparison, albeit by a minimal margin. The error values of XGBoost and the best performer LightGBM were almost identical for MASE and minimal in the mean average error (MAE) and the models' ability to explain the variance of the target variable. The difference between RMSE and MAE was lower than what LightGBM produced, indicating better ability to predict weekly sales even with sudden variations and outliers in the data.

The worst performer was MLP by a clear margin. On average its RMSE was 95 % ($ 74,700) higher, MAE was 86 % ($ 45,150) higher, and R-squared was ≈5.2 % lower compared to gradient boosting models' performance. The results indicate that the error in the weekly sales forecast is significantly greater compared to LightGBM and XGBoost. MLP's ability to explain the correlation of the target variable is also the weakest of the three models. In addition to the weakest error coefficients, training the neural network and searching for parameters was slow and computationally the most demanding.

4.5.2 Residual Analysis

Residual analysis is done to further evaluate the results of the prediction to support the error and performance metrics. It helps to determine the "goodness-of-fit" for the data, and bias and accuracy of the models' predictions by evaluating if the model is missing important features from the dataset (Verma, 2025, pp. 34–35). Thus, analyzing residuals

further helps to detect the best fitting model for the data type under analysis. The residual analysis was conducted by plotting graphs for the residual of each model and checking the normality of the residuals by Jarque-Bera (Jarque & Bera, 1980) test.

According to Verma (2025, pp. 36–37) the residuals of a well fitted model are randomly scattered on the plot excluding any trend or curve patterns. Residual plots help to visualize any heteroscedasticity as any systematic patterns would indicate that the model did not capture the variance in the data which might refer to poor fit of the model and the data (Verma, 2025, pp. 37–41). The residual plots for each model are displayed in figure 19.

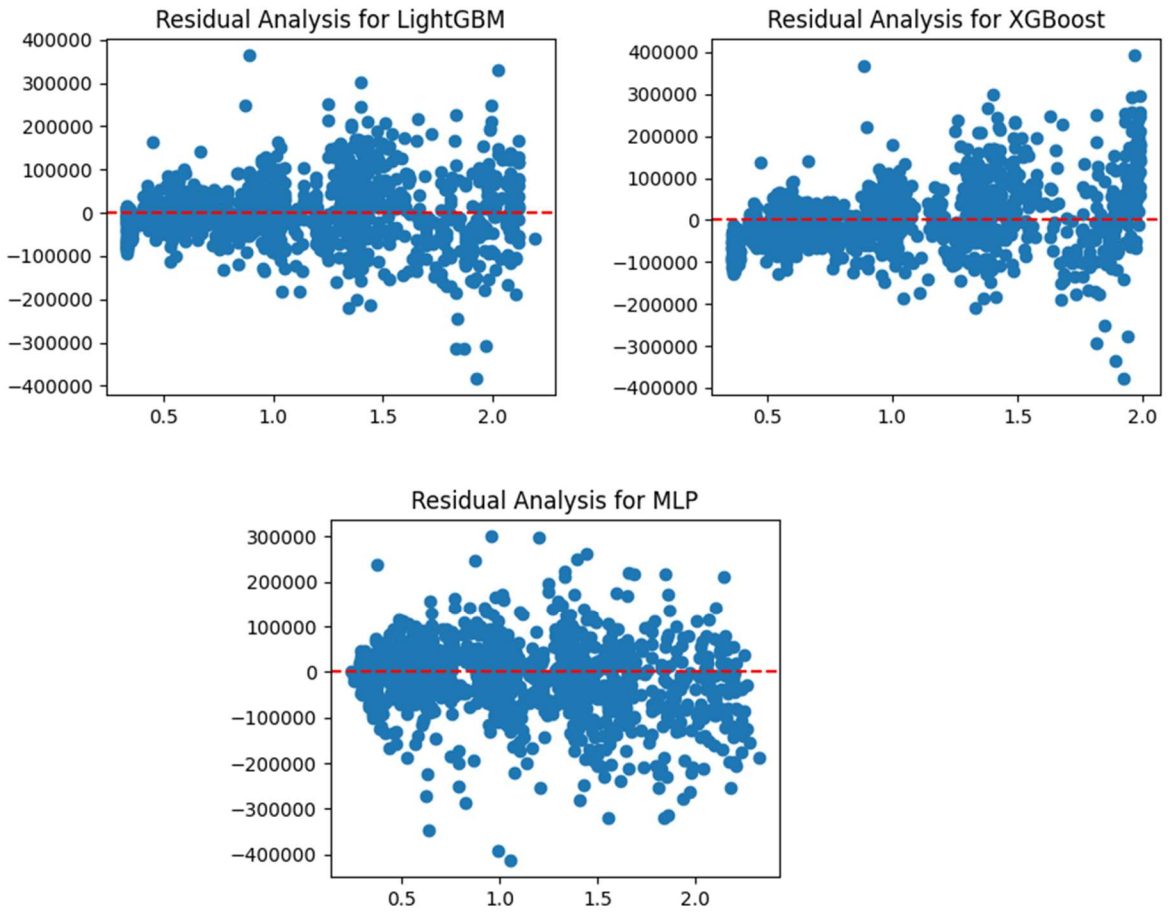


Figure 19. Residual Plots for each model.

The plots display how the residuals are randomly scattered for all the models, which validates the metrics used in model validation. This observation indicates that the models are not incorrectly specified for the data, thus the models are not missing significant features or interactions (Verma, 2025, p. 41). In addition to the residual plots, the normality of the residuals was tested to validate the robustness of the results.

To conclude the residual analysis, a normality test was performed with Jarque-Bera test. All the models' residuals scored a p-value < 0.05, which suggest non-normal residuals (Thadewald & Büning, 2007, p. 105; Verma, 2025, p. 40). The non-normality of the residuals confirmed the robustness of the error and performance metrics used in the evaluation of the ML models.

5 Conclusions

The aim of the study was to create a comparative study on three different machine learning models and to investigate how the models used in the project utilize different features of the dataset during the forecast process. The study was a time series analysis and based on anonymized and simulated sales data from a U.S-based retail company. The data consisted of weekly sales figures from 45 stores operating in different areas over a period of approximately three years. The target variable was weekly sales in US dollars. In addition to the target variable, the data included seven different variables (features) covering prevailing macroeconomic and temporal features. Furthermore, new features were engineered based on the original variables of the data, such as temporal, lag, and interaction features. The time series was divided into training and test sets in a ratio of 65:35, respecting the chronological order of the time series. The models used for the comparative analysis were LightGBM (Light Gradient-Boosting Machine), XGBoost (Extreme Gradient Boosting), and MLP (Multilayer Perceptron). The hyperparameters of all models were adjusted using randomized parameter search, and they predicted exactly the same time series data containing identical features.

5.1 Summary of comparative analysis

The comparative analysis was performed between LightGBM, XGBoost, and MLP, and a 52-week seasonal naïve forecast method was included to represent the baseline level of prediction. The use of a baseline prediction increases the objectivity of the assessment (Tam et al., 2025, p. 11) and provides a simple benchmark for the advanced ML models (Micallef et al., 2025, p. 4). All of the ML models were superior to the baseline model. The mean absolute prediction error results for all models were on average 86 % lower than the baseline prediction (see table 12). Gradient boosted models performed best according to error and performance metrics. LightGBM was more accurate by a slight margin compared to XGBoost meaning the difference between the models was not significant. LightGBM also presented a higher percentage difference between MAE and

RMSE, suggesting that XGBoost managed large fluctuations better in the time series. Although MLP, similarly to LightGBM and XGBoost, clearly outperformed the baseline model, it performed the weakest among the machine learning models by a clear margin. Its forecast errors were significantly larger than those of gradient boosted models, indicating a weaker forecast on weekly sales. The ability to explain the correlation of the target variable was also the weakest, resulting in ≈5.2 % lower R^2 value.

The neural network-based MLP caused the most difficulties in data processing, model training, and configuration phase. MLP had issues fitting the dataset used in the analysis, and its training and parameter search times were multiple times longer than those of GBDT-models. On the other hand, the issues probably were caused by scarcity of data, and the model could provide better performance in a different scenario where the volume of data is larger (Ramos et al., 2023, p. 683). For example, in a study by Chaudhuri et al (2021) a deep neural network model was the best performing model at predicting customer purchase behavior in online retail market, when compared to other conventional ML models. The dataset used in the study consisted of 429,013 unique online store visits, which means the volume is considerably larger than the 6,435 observations and 51,480 datapoints used in this thesis. To conclude, all ML models performed significantly better than the baseline model, breaking the null hypothesis of this thesis. The LightGBM model was the most efficient of the three machine learning models, with XGBoost coming in second by only a small margin. LightGBM and XGBoost seemed to capture the variance and generalize their predictions better, even though the volume of data was relatively small. Based on the results of error and performance metrics, as well as usability of the models, gradient boosted models were dominant in this research.

Several academic studies have been conducted based on the same dataset. For instance, Neba et al (2024) also conducted a comparative analysis to evaluate classical and advanced forecasting models based on the prediction accuracy using MAE and RMSE, as well as comparing their computational efficiency and interpretability. According to their

analysis, the forecast results were on average worse, with one exception. Four of the five models (ARIMA, SARIMA, Prophet, and Exponential Smoothing) produced average MAE of ≈$ 465.000 and RMSE of ≈$ 558.000, which are significantly higher than the results of this project. However, their best model (Gaussian Processes) produced the forecast with mean absolute error of only ≈$ 25.500 $ and root mean square error of ≈$ 34.000 outperforming the models used in this thesis. However, even if the fundamentals of the study were largely the same as in this thesis, there were some key differences during the data preprocessing. They used 80 % of the data for training and 20 % for testing and handled outliers using the Winsorization method, in which extreme values are capped with predetermined percentiles (e.g., the 95th and 5th percentiles) instead of removed. Also, hyperparameter tuning was minimal and no feature importance analysis was performed in their study. Therefore, due to differences in data processing, the research results are not directly comparable with each other. This highlights the significance of data processing in predictive analyses.

5.2 Main findings and recommendations for future research

The research questions were based on the background of the study and previous literature. Based on the literature review, most studies focus on the efficiency of a predictive model rather than the relationship between the model and the features of the data. This study sought to answer the question raised by the research gap identified: how do external factors (e.g. unemployment, inflation, fuel prices) affect the predictions of machine learning models, and how significant is the impact of the selected features on prediction accuracy? Essentially, the feature selection had a decisive impact on the predictive performance of all three models. In time series analysis, features capturing temporal dependencies and correlations within the data are crucial, as it helps to identify repeating trends and patterns (Kannadasan, 2025, p. 21). Indeed, the test results showed that lag features dominated the feature importance scores for each model despite their architectural differences. The optimal lag length (e.g. 1-week, 4-week, or 8-week) should be evaluated and chosen depending on the strength of the correlation between the

lagged feature and the target variable (Kannadasan, 2025, p. 21). In this project, the optimal lag length was 1 week, according to permutation feature importance scores.

In contrast, the feature depicting holiday weeks was among the most important only for MLP. This was contrary to expectations, as many studies (Abolghasemi, Hurley, et al., 2020; Ghareeb et al., 2020; Hirche et al., 2021) specifically studied the impact of seasons, holidays, and sales promotions features on the forecast effectiveness. Holidays and sales promotions generally increase momentary sales, causing increased data volatility, which was also evident in the time series used in this study (see figure 9). The feature indicating seasonal holidays could have been assumed to contribute to the forecasts across all models, but contrary to expectations, the holiday feature was only significant in prediction process of the MLP. Other temporal features such as rolling mean and rolling standard deviation had little to no contribution to the predictions when lag features were excluded. They provided only marginal gains for tree-based models and none for the MLP. Furthermore, the macroeconomic features were only significant when temporal features (except for week and month) were absent. Even then, MLP was the only model that leveraged macroeconomic data (see Appendix 6), CPI and unemployment in its forecast. Ultimately, all three models reduce the values of macroeconomic and contextual features when stronger temporal features are available. The substantial contribution of temporal variables to the forecasts suggests that macroeconomic variables do not significantly affect short-term forecasts in this context. Further research could aim to analyze a longer time series that is not so heavily weighted by seasonal trends to better assess the impact of external factors.

Building on the importance of feature selection, the models were tested with various feature configurations, starting with a feature set including only the original features of the data. Temporal features were the main drivers for prediction accuracy across all models. Adding lag features consistently reduced prediction errors, indicating the temporal dependency of each model; especially 1-week and 4-week lags had significant contributions whenever they were included in the feature set. This indicates that

temporal dependencies are essential in accurate forecasts. Including short-term lag features demonstrated the ability of both tree-based models (LightGBM & XGBoost) to leverage recent historical data in their forecasts. On average, the short-term lags improved RMSE approximately 10-15 % and increased R^2 above 0.94. MLP was the most sensitive to feature selection; it displayed poor performance without lag features, as its R-squared decreased significantly, to around 0.22–0.37, whereas it was ≈0.93 with the lag features included. In conclusion, new feature engineering is extremely important for generating accurate forecasts. In particular, including relevant lag features helps models capture temporal patterns, which is a prerequisite for accurate demand forecasts.

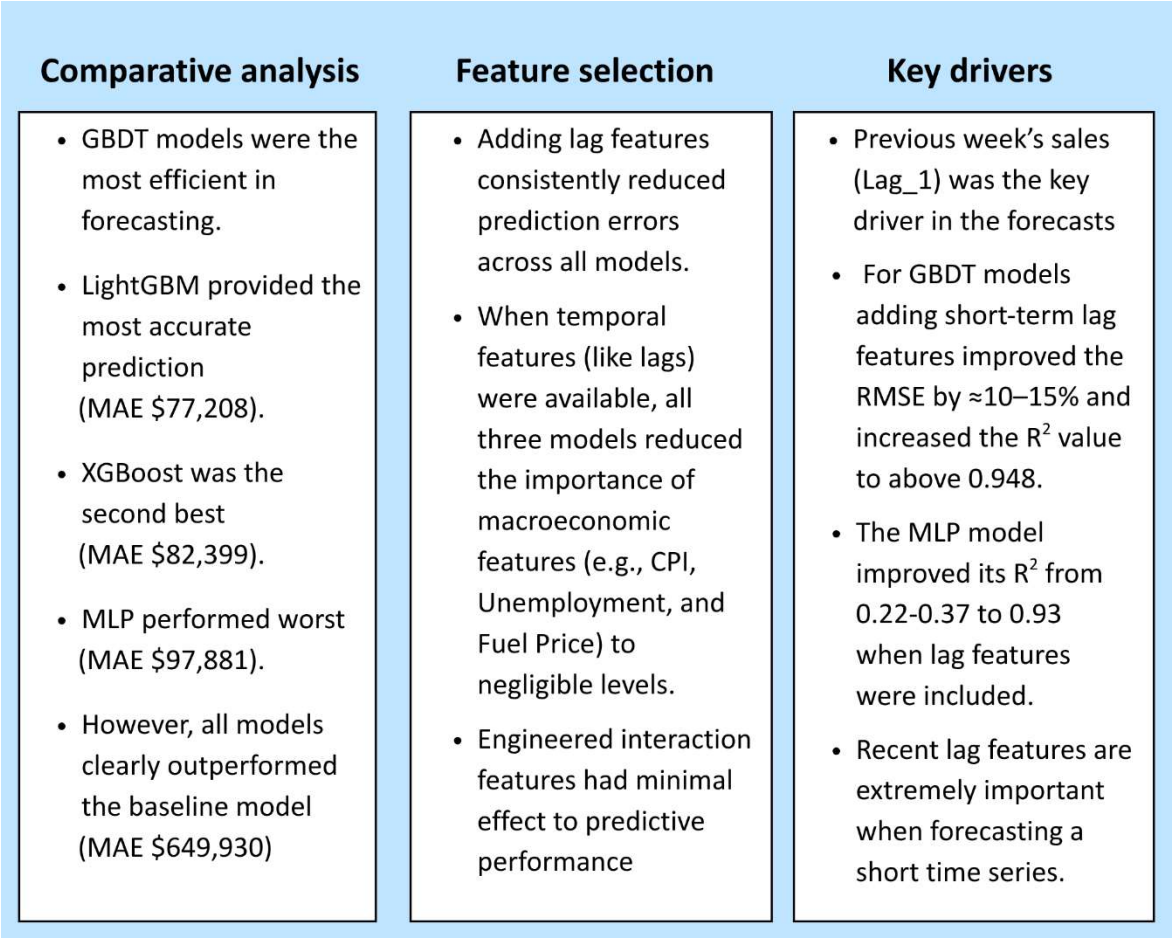


Figure 20. Summary of results.

In this study, the machine learning models created for demand forecasting did not base their forecasts on external factors. The data set imposed clear limitations on the research. For example, the store locations in the data were unknown, which prevented examining the relationship between macroeconomic and location-based features. Additionally, the time series data spanned over a period of only three years and it was heavily weighted toward seasonal patterns and trends. This may have contributed to the way the models utilized the data provided to them during the analysis. From the results of the study, we can conclude that the engineering of new features during the data preprocessing phase made a significant contribution to demand forecasts. The engineered features (e.g. lag features) proved to be the most influential in terms of accuracy during the permutation feature importance test. One of the objectives of this study was to examine the impact of external variables on demand forecasts generated by machine learning models using time series analysis. For further research on the topic, it would be essential to examine the subject using real-world figures and a longer review period. Particularly with regard to macroeconomic variables, it would be desirable for the time series to cover at least five years so that it includes multiple seasonal cycles and trends. Much older data may become irrelevant due to changes in environment and market dynamics. The topic could be studied through comprehensive figures referring to demand in the retail sector and prevailing macroeconomic factors, i.e., indices such as interest and inflation rates, consumer confidence, disposable income, and the retail sales index.

References

Abolghasemi, M., Beh, E., Tarr, G., & Gerlach, R. (2020). Demand forecasting in supply chain: The impact of demand volatility in the presence of promotion. *Computers & Industrial Engineering*, *142*, 106380. https://doi.org/10.1016/j.cie.2020.106380

Abolghasemi, M., Hurley, J., Eshragh, A., & Fahimnia, B. (2020). Demand forecasting in the presence of systematic events: Cases in capturing sales promotions. *International Journal of Production Economics*, *230*, 107892. https://doi.org/10.1016/j.ijpe.2020.107892

Alves, F., Souza, E. G. D., Sobjak, R., Bazzi, C. L., Hachisuca, A. M. M., & Mercante, E. (2024a). Data processing to remove outliers and inliers: A systematic literature study. *Revista Brasileira de Engenharia Agrícola e Ambiental*, *28*(9), e278672. https://doi.org/10.1590/1807-1929/agriambi.v28n9e278672

Alves, F., Souza, E. G. D., Sobjak, R., Bazzi, C. L., Hachisuca, A. M. M., & Mercante, E. (2024b). Data processing to remove outliers and inliers: A systematic literature study. *Revista Brasileira de Engenharia Agrícola e Ambiental*, *28*(9), e278672. https://doi.org/10.1590/1807-1929/agriambi.v28n9e278672

Armstrong, J. S. (Ed.). (2001). *Principles of Forecasting: A Handbook for Researchers and Practitioners* (1st ed. 2001). Imprint: Springer. https://doi.org/10.1007/978-0-306-47630-3

Barua, L., Zou, B., & Zhou, Y. (2020). Machine learning for international freight transportation management: A comprehensive review. *Research in Transportation Business & Management*, *34*, 100453. https://doi.org/10.1016/j.rtbm.2020.100453

Bre, F., Gimenez, J. M., & Fachinotti, V. D. (2018). Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, *158*, 1429–1441. https://doi.org/10.1016/j.enbuild.2017.11.045

Brigato, L., & Iocchi, L. (2020). *A Close Look at Deep Learning with Small Data* (No. arXiv:2003.12843). arXiv. https://doi.org/10.48550/arXiv.2003.12843

Caniato, F., Kalchschmidt *, M., Ronchi, S., Verganti, R., & Zotteri, G. (2005). Clustering customers to forecast demand. *Production Planning & Control*, *16*(1), 32–43. https://doi.org/10.1080/09537280512331325155

Cerqueira, V., Roque, L., & Soares, C. (2025). Modelradar: Aspect-based forecast evaluation. *Machine Learning*, *114*(10), 229. https://doi.org/10.1007/s10994-025-06877-z

Çetin, V., & Yıldız, O. (2022). A comprehensive review on data preprocessing techniques in data analysis. *Pamukkale University Journal of Engineering Sciences*, *28*(2), 299–312. https://doi.org/10.5505/pajes.2021.62687

Chaudhuri, N., Gupta, G., Vamsi, V., & Bose, I. (2021). On the platform but will they buy? Predicting customers' purchase behavior using deep learning. *Decision Support Systems*, *149*, 113622. https://doi.org/10.1016/j.dss.2021.113622

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, *7*, e623. https://doi.org/10.7717/peerj-cs.623

Dell'Acqua, F., McFowland, E., Mollick, E. R., Lifshitz-Assaf, H., Kellogg, K., Rajendran, S., Krayer, L., Candelon, F., & Lakhani, K. R. (2023). Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4573321

Deng, C., Zhang, J., Bai, S., Mo, Y., Liu, X., Gu, Y., & Li, N. (2025). *Retail Commodity Sales Prediction: A Prophet-LightGBM Combined Machine Learning Approach*. *65*, 150–157. https://doi.org/10.3233/ATDE250122

Disney, S. M., Ponte, B., & Wang, X. (2021). Exploring the nonlinear dynamics of the lost-sales order-up-to policy. *International Journal of Production Research*, *59*(19), 5809–5830. https://doi.org/10.1080/00207543.2020.1790687

Ensafi, Y., Amin, S. H., Zhang, G., & Shah, B. (2022). Time-series forecasting of seasonal items sales using machine learning – A comparative analysis. *International Journal of Information Management Data Insights*, *2*(1), 100058. https://doi.org/10.1016/j.jjimei.2022.100058

Eurostat. (2024). *Retail trade volume index overview. Statistics Explained.* https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Retail_trade_volume_index_overview

Falatouri, T., Darbanian, F., Brandtner, P., & Udokwu, C. (2022). Predictive Analytics for Demand Forecasting – A Comparison of SARIMA and LSTM in Retail SCM. *Procedia Computer Science*, *200*, 993–1003. https://doi.org/10.1016/j.procs.2022.01.298

Feizabadi, J. (2022). Machine learning demand forecasting and supply chain performance. *International Journal of Logistics Research and Applications*, *25*(2), 119–142. https://doi.org/10.1080/13675567.2020.1803246

Fosso Wamba, S., Guthrie, C., Queiroz, M. M., & Minner, S. (2024). ChatGPT and generative artificial intelligence: An exploratory study of key benefits and challenges in operations and supply chain management. *International Journal of Production Research*, *62*(16), 5676–5696. https://doi.org/10.1080/00207543.2023.2294116

Ganguly, P., & Mukherjee, I. (2024). Enhancing Retail Sales Forecasting with Optimized Machine Learning Models. *2024 4th International Conference on Sustainable Expert Systems (ICSES)*, 884–889. https://doi.org/10.1109/ICSES63445.2024.10762950

Ganjare, S. A., Satao, S. M., & Narwane, V. (2023). Systematic literature review of machine learning for manufacturing supply chain. *The TQM Journal*, *36*(8), 2236–2259. https://doi.org/10.1108/TQM-12-2022-0365

Ghareeb, A., Al-bayaty, H., Haseeb, Q., & Zeinalabideen, M. (2020). Ensemble learning models for short-term electricity demand forecasting. *2020 International Conference on Data Analytics for Business and Industry: Way Towards a*

Sustainable Economy (ICDABI), 1–5. https://doi.org/10.1109/ICDABI51230.2020.9325623

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition* (2nd ed.). Springer New York. http://ebookcentral.proquest.com/lib/tritonia-ebooks/detail.action?docID=6314834

Hirche, M., Haensch, J., & Lockshin, L. (2021). Comparing the day temperature and holiday effects on retail sales of alcoholic beverages – a time-series analysis. *International Journal of Wine Business Research*, *33*(3), 432–455. https://doi.org/10.1108/IJWBR-07-2020-0035

Ho, G. T. S., Tang, V., Tong, P. H., & Tam, M. M. F. (2025). Demand-driven storage allocation for optimizing order picking processes. *Expert Systems with Applications*, *272*, 126812. https://doi.org/10.1016/j.eswa.2025.126812

Huber, J., & Stuckenschmidt, H. (2020). Daily retail demand forecasting using machine learning with emphasis on calendric special days. *International Journal of Forecasting*, *36*(4), 1420–1438. https://doi.org/10.1016/j.ijforecast.2020.02.005

Hüllermeier, E., & Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *ResearchGate*, *110*, 457–506. https://doi.org/10.1007/s10994-021-05946-3

Institute of Business Forecasting and Planning. (2018, July 12). *How Much Does Forecasting Software Cost, & How Much Will It Save? – Demand Planning, S&OP/ IBP, Supply Planning, Business Forecasting Blog*. https://demand-planning.com/2018/07/12/how-much-does-forecasting-software-cost/

Jackson, I., Ivanov, D., Dolgui, A., & Namdar, J. (2024). Generative artificial intelligence in supply chain and operations management: A capability-based framework for analysis and implementation. *International Journal of Production Research*, *62*(17), 6120–6145. https://doi.org/10.1080/00207543.2024.2309309

Jarque, C. M., & Bera, A. K. (1980). Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, *6*(3), 255–259. https://doi.org/10.1016/0165-1765(80)90024-5

Javed, S., & Akhtar, R. (2024). Data driven Approaches for Demand Forecasting in Supply Chain for Business Decisions. *2024 5th International Conference on Advancements in Computational Sciences (ICACS)*, 1–7. https://doi.org/10.1109/ICACS60934.2024.10473234

Jeong, J., Park, E., Han, W. S., Kim, K., Choung, S., & Chung, I. M. (2017). Identifying outliers of non-Gaussian groundwater state data based on ensemble estimation for long-term trends. *Journal of Hydrology*, *548*, 135–144. https://doi.org/10.1016/j.jhydrol.2017.02.058

Jiang, X., Abdel-Aty, M., Hu, J., & Lee, J. (2016). Investigating macro-level hotzone identification and variable importance using big data: A random forest models approach. *Neurocomputing*, *181*, 53–63. https://doi.org/10.1016/j.neucom.2015.08.097

Kaggle.com. (2021). Walmart Store Sales Prediction - Regression Problem. https://www.kaggle.com/datasets/yasserh/walmart-dataset

Kampezidou, S. I., Tikayat Ray, A., Bhat, A. P., Pinon Fischer, O. J., & Mavris, D. N. (2024). Fundamental Components and Principles of Supervised Machine Learning Workflows with Numerical and Categorical Data. *Eng*, *5*(1), 384–416. https://doi.org/10.3390/eng5010021

Kannadasan, T. (2025). Deep Learning based Seasonality and Trend Detection in Sales Forecasting. *International Journal of Data Informatics and Intelligent Computing*, *4*(2), 16–29. https://doi.org/10.59461/ijdiic.v4i2.170

Khan, M. A., Saqib, S., Alyas, T., Ur Rehman, A., Saeed, Y., Zeb, A., Zareei, M., & Mohamed, E. M. (2020). Effective Demand Forecasting Model Using Business Intelligence Empowered With Machine Learning. *IEEE Access*, *8*, 116013–116023. IEEE Access. https://doi.org/10.1109/ACCESS.2020.3003790

Kilimci, Z. H., Akyuz, A. O., Uysal, M., Akyokus, S., Uysal, M. O., Atak Bulbul, B., & Ekmis, M. A. (2019). An Improved Demand Forecasting Model Using Deep Learning Approach and Proposed Decision Integration Strategy for Supply Chain. *Complexity*, *2019*(1), 9067367. https://doi.org/10.1155/2019/9067367

Kılıç Sarıgül, R., Erkayman, B., & Usanmaz, B. (2025). Increasing load factor in logistics and evaluating shipment performance with machine learning methods: A case from the automotive industry. *Scientific Reports*, *15*(1), 12434. https://doi.org/10.1038/s41598-025-94713-8

Krakowski, S., Luger, J., & Raisch, S. (2023). Artificial intelligence and the changing sources of competitive advantage. *Strategic Management Journal*, *44*(6), 1425–1452. https://doi.org/10.1002/smj.3387

Lainder, A. D., & Wolfinger, R. D. (2022). Forecasting with gradient boosted trees: Augmentation, tuning, and cross-validation strategies. *International Journal of Forecasting*, *38*(4), 1426–1433. https://doi.org/10.1016/j.ijforecast.2021.12.003

Lay, R. Y. K., Zhang, W., & Xu, W. (2018). Parallel Aspect-Oriented Sentiment Analysis for Sales Forecasting with Big Data. *Production and Operations Management*, *27*(10), 1775–1794. https://doi-org.proxy.uwasa.fi/10.1111/poms.12737

Li, J., Zhang, J., Sun, M., & Zhu, R. (2024). Vegetable Sales Forecasting Based on XGBOOST Algorithm and Random Forest. *2024 International Conference on Computers, Information Processing and Advanced Education (CIPAE)*, 150–155. https://doi.org/10.1109/CIPAE64326.2024.00032

Li, L., Kang, Y., Petropoulos, F., & Li, F. (2023). Feature-based intermittent demand forecast combinations: Accuracy and inventory implications. *International Journal of Production Research*, *61*(22), 7557–7572. https://doi.org/10.1080/00207543.2022.2153941

Lima, S., Gonçalves, A. M., & Costa, M. (2024). Predictive accuracy of time series models applied to economic data: The European countries retail trade. *Journal of Applied Statistics*, *51*(9), 1818–1841. https://doi.org/10.1080/02664763.2023.2238249

Liu, F., Chen, L., Zheng, Y., & Feng, Y. (2022). A Prediction Method with Data Leakage Suppression for Time Series. *Electronics*, *11*(22), 3701. https://doi.org/10.3390/electronics11223701

McKinsey & Company. (2022). *CPG Operations: How to win in a rapidly changing environment* (p. 120).

Mediavilla, M. A., Dietrich, F., & Palm, D. (2022). Review and analysis of artificial intelligence methods for demand forecasting in supply chain management. *Procedia CIRP*, *107*, 1126–1131. https://doi.org/10.1016/j.procir.2022.05.119

Micallef, A., Apap, M., Licari, J., Spiteri Staines, C., & Xiao, Z. (2025). A comparative framework for evaluating machine learning models in forecasting electricity demand for port microgrids. *Energy and AI*, *20*, 100494. https://doi.org/10.1016/j.egyai.2025.100494

Mircetic, D., Rostami-Tabar, B., Nikolicic, S., & Maslaric, M. (2022). Forecasting hierarchical time series in supply chains: An empirical investigation. *International Journal of Production Research*, *60*(8), 2514–2533. https://doi.org/10.1080/00207543.2021.1896817

Mitra, A., Arnav, J., Kishore, A., & Kumar, P. (2022). A Comparative Study of Demand Forecasting Models for a Multi-Channel Retail Company: A Novel Hybrid Machine Learning Approach. *Operations Research Forum*, *3*(58). https://doi.org/10.1007/s43069-022-00166-4

MLPRegressor. (2025). Scikit-Learn. https://scikit-learn/stable/modules/generated/sklearn.neural_network.MLPRegressor.html

Mohri, S. S., & Haghshenas, H. (2017). *Modeling the Container Selection for Freight Transportation: Case Study of Iran*.

Neba, C., Cyril, Shu, G., F., Nsuh, G., Amouda, P., A., Neba, A., F., Webnda, F., Ikpe, V., Adeyinka, O., & Sylla, N., A. (2024). View of A Comprehensive Study of Walmart Sales Predictions Using Time Series Analysis. *2024*, *20*(7), 9–30. https://doi.org/10.9734/arjom/2024/v20i7809

Nematzadeh, S., Kiani, F., Torkamanian-Afshar, M., & Aydin, N. (2022). Tuning hyperparameters of machine learning algorithms and deep neural networks using metaheuristics: A bioinformatics study on biomedical and biological cases. *Computational Biology and Chemistry*, *97*, 107619. https://doi.org/10.1016/j.compbiolchem.2021.107619

Parameters—LightGBM 4.6.0 documentation. (2025). https://lightgbm.readthedocs.io/en/stable/Parameters.html

Pereira, M. M., & Frazzon, E. M. (2021). A data-driven approach to adaptive synchronization of demand and supply in omni-channel retail supply chains. *International Journal of Information Management*, *57*, 102165. https://doi.org/10.1016/j.ijinfomgt.2020.102165

Petropoulos, F., Grushka-Cockayne, Y., Siemsen, E., & Spiliotis, E. (2025). Wielding Occam's razor: Fast and frugal retail forecasting. *Journal of the Operational Research Society*, *76*(8), 1564–1583. https://doi.org/10.1080/01605682.2024.2421339

Punia, S., Singh, S. P., & Madaan, J. K. (2020). From predictive to prescriptive analytics: A data-driven multi-item newsvendor model. *Decision Support Systems*, *136*, 113340. https://doi.org/10.1016/j.dss.2020.113340

Raghuvanshi, S. (2024). Assessing the Impact of Seasonal Decomposition on the Time Series Analysis Accuracy: A Comprehensive Study. *International Journal for Research in Applied Science and Engineering Technology*, *12*(5), 2988–2992. https://doi.org/10.22214/ijraset.2024.61811

Rajula, H. S. R., Verlato, G., Manchia, M., Antonucci, N., & Fanos, V. (2020). Comparison of Conventional Statistical Methods with Machine Learning in Medicine: Diagnosis, Drug Development, and Treatment. *Medicina*, *56*(9), 455. https://doi.org/10.3390/medicina56090455

Ramos, F. R., Pereira, M. T., Oliveira, M., & Rubio, L. (2023). The memory concept behind deep neural network models: An application in time series forecasting in the e-Commerce sector. *Decision Making: Applications in Management and Engineering*, *6*(2), 668–690. https://doi.org/10.31181/dmame622023695

Ratra, K. K., & Seth, D. K. (2025). AI-Driven Hybrid Edge-Cloud Architecture for Real-Time Big Data Analytics and Scalable Communication in Retail Supply Chains. *SoutheastCon 2025*, 1023–1029. https://doi.org/10.1109/SoutheastCon56624.2025.10971468

Reunanen, N., Räty, T., Jokinen, J. J., Hoyt, T., & Culler, D. (2020). Unsupervised online detection and prediction of outliers in streams of sensor data. *International*

Journal of Data Science and Analytics, *9*(3), 285–314. https://doi.org/10.1007/s41060-019-00191-3

Sanders, N. (2015). *Forecasting Fundamentals*. Business Expert Press. http://ebookcentral.proquest.com/lib/tritonia-ebooks/detail.action?docID=4742536

Saoud, P., Kourentzes, N., & Boylan, J. E. (2025). The importance of forecast uncertainty in understanding the Bullwhip effect. *International Journal of Production Research*, 1–22. https://doi.org/10.1080/00207543.2025.2527957

Schmid, L., Roidl, M., Kirchheim, A., & Pauly, M. (2025). Comparing Statistical and Machine Learning Methods for Time Series Forecasting in Data-Driven Logistics—A Simulation Study. *Entropy*, *27*(1), 25. https://doi.org/10.3390/e27010025

Schneider, J.-V., Alavi, S., Guba, J. H., Wieseke, J., & Schmitz, C. (2021). When do forecasts fail and when not? Contingencies affecting the accuracy of sales managers' forecast regarding the future business situation. *Journal of Personal Selling & Sales Management*, *41*(3), 218–232. https://doi.org/10.1080/08853134.2020.1859941

Scikit-Learn. (2025). *Model selection and evaluation*. 3.4. Metrics and Scoring. Quantifying the Quality of Predictions. https://scikit-learn.org/stable/modules/model_evaluation.html

Seyedan, M., & Mafakheri, F. (2020). Predictive big data analytics for supply chain demand forecasting: Methods, applications, and research opportunities. *Journal of Big Data*, *7*(1), 53. https://doi.org/10.1186/s40537-020-00329-2

Slater, P., & Hasson, F. (2025). Quantitative Data Quality Assurance, Analysis and Presentation. *Journal of Psychiatric and Mental Health Nursing*, *32*(3), 723–727. https://doi.org/10.1111/jpm.13143

Sutton, R. S., & Barto, A. G. (2015). *Reinforcement Learning: An Introduction*.

Tai, P. D., Buddhakulsomsiri, J., & Duc, T. T. H. (2022). Revisiting measurement of compound bullwhip with asymmetric reference price. *Computers & Industrial Engineering*, *172*, 108510. https://doi.org/10.1016/j.cie.2022.108510

Tam, B., C. ,. M., Tang, S.-K., & Cardoso, A. (2025). Multi-level lag scheme significantly improves training efficiency in deep learning: A case study in air quality alert service over sub-tropical area. *Journal of Big Data*, *12*(1), 3. https://doi.org/10.1186/s40537-024-01043-z

Thadewald, T., & Büning, H. (2007). Jarque–Bera Test and its Competitors for Testing Normality – A Power Comparison. *Journal of Applied Statistics*, *34*(1), 87–105. https://doi.org/10.1080/02664760600994539

The Jupyter Notebook—Jupyter Notebook 7.5.0b0 documentation. (2015). https://jupyter-notebook.readthedocs.io/en/latest/notebook.html

Trafimow, D., Wang, T., & Wang, C. (2019). From a Sampling Precision Perspective, Skewness Is a Friend and Not an Enemy! *Educational and Psychological Measurement*, *79*(1), 129–150. https://doi.org/10.1177/0013164418764801

Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley Publishing Company. https://www.consoleflare.com/blog/wp-content/uploads/2022/09/Exploratory-Data-Analysis-1977-John-Tukey.pdf

Van Wyk, A. (2023). *Machine Learning with LightGBM and Python: A practitioner's guide to developing production-ready machine learning systems*. Packt Publishing Ltd.

Verma, V. (2025). A Comprehensive Framework for Residual Analysis in Regression and Machine Learning. *Journal of Information Systems Engineering and Management*, *10*(31s), 34–46. https://doi.org/10.52783/jisem.v10i31s.4958

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … van Mulbregt, P. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

Wasserbacher, H., & Spindler, M. (2022). Machine learning for financial forecasting, planning and analysis: Recent developments and pitfalls. *Digital Finance*, *4*(1), 63–88. https://doi.org/10.1007/s42521-021-00046-2

XGBoost Parameters—XGBoost 3.0.5 documentation. (2022). https://xgboost.readthedocs.io/en/stable/parameter.html

Xu, N., Wang, Z., Dai, Y., Li, Q., Zhu, W., Wang, R., & Finkelman, R. B. (2023). Prediction of higher heating value of coal based on gradient boosting regression tree model. *International Journal of Coal Geology*, *274*, 104293. https://doi.org/10.1016/j.coal.2023.104293

Yagmur, N., Taskin, G., Musaoglu, N., & Erten, E. (2024). Forecasting surface movements based on PSI time series using machine learning algorithms. *International Journal of Remote Sensing*, *45*(7), 2462–2485. https://doi.org/10.1080/01431161.2024.2331977

Yasir, M., Ansari, Y., Latif, K., Maqsood, H., Habib, A., Moon, J., & Rho, S. (2024). Machine learning–assisted efficient demand forecasting using endogenous and exogenous indicators for the textile industry. *International Journal of Logistics Research and Applications*, *27*(12), 2867–2886. https://doi.org/10.1080/13675567.2022.2100334

Appendices

Appendix 1. Functions created in data analysis

Interquartile range function

```
def detect_outliers_iqr(data, column):
            Q1 = data[column].quantile(0.25)
            Q3 = data[column].quantile(0.75)
            IQR = Q3 - Q1
            lower_bound = Q1 - 1.5 * IQR
            upper_bound = Q3 + 1.5 * IQR
            outliers  =  data[(data[column]  <  lower_bound)|
            (data[column] > upper_bound)]
            return outliers
```

Function for 52-week seasonal naïve

```
def seasonal_naive_forecast(y_train, steps, season_length=52):
    y_train = np.asarray(y_train)

    if len(y_train) < season_length:
        raise ValueError("Training series is shorter than the
season length.")

    last_season = y_train[-season_length:]

    reps = int(np.ceil(steps / season_length))
    forecast = np.tile(last_season, reps)[:steps]

    return forecast
```

Appendix 2. Parameter grids

LightGBM parameter grid

```
param_grid_lgbm = {
            'n_estimators': randint(100, 300),
            'learning_rate': loguniform(0.01, 0.1),
            #Tree Complexity
            'max_depth': randint(3, 9),
            'num_leaves': randint(31, 128),
            'min_data_in_leaf': randint(5, 30),
            'reg_alpha': loguniform(1e-3, 0.5),
            'reg_lambda': loguniform(1e-3, 0.5)
}
```

MLP parameter grid

```
param_grid_mlp = {
    'hidden_layer_sizes': [(40,20), (60,30), (40,20,5)],
    'learning_rate_init': loguniform(1e-4, 0.1),
    'alpha': loguniform(1e-2, 1),
    'batch_size': randint(16, 128),
    'activation': ['relu', 'tanh'],
    'learning_rate': ['constant', 'adaptive'],
    'momentum': uniform(0.5, 0.5),
    'tol': [1e-3, 1e-2, 1e-1],
    'validation_fraction': [0.1, 0.2, 0.3]
}
```

Appendix 3. Summary of the dataset

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6435 entries, 0 to 6434
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Store         6435 non-null   category
 1   Date          6435 non-null   datetime64[ns]
 2   Weekly_Sales  6435 non-null   float64
 3   Holiday_Flag  6435 non-null   category
 4   Temperature   6435 non-null   float64
 5   Fuel_Price    6435 non-null   float64
 6   CPI           6435 non-null   float64
 7   Unemployment  6435 non-null   float64
 8   Year          6435 non-null   int32
 9   Month         6435 non-null   int32
 10  Week          6435 non-null   UInt32
 11  Day           6435 non-null   int32
dtypes: UInt32(1), category(2), datetime64[ns](1), float64(5), int32(3)
memory usage: 422.7 KB
```

Appendix 4. LightGBM Results with different feature sets

LightGBM with different feature sets

Default features

'Store', 'Month', 'Week', 'Temperature', 'Fuel_Price',
'CPI', 'Unemployment', 'Holiday_Flag'

```
LGB: Top 5 features:        LightGBM Final Test Results
Store, 1.689275             Final RMSE: 143647.50
Week, 0.063719              Final MAE: 90986.43
Month, 0.002515             Final MASE: 0.148
Holiday_Flag, 0.000000      Final R²: 0.9378
Fuel_Price, 0.000000
```

Set 1 (best performance)

'Store', 'Month', 'Week',
'Temperature', 'Fuel_Price', 'CPI', 'Unemployment', 'Holiday_Flag',
'Rolling_Mean_4', 'Rolling_Std_4', 'Lag_4', 'Lag_1'

```
LGB: Top 5 features:        LightGBM Final Test Results
Lag_1, 0.535235             Final RMSE: 128353.54
Store, 0.144100             Final MAE: 77207.66
Lag_4, 0.088750             Final MASE: 0.126
Week, 0.036418              Final R²: 0.9504
Month, 0.001551
```

Set 2 (no lags, only rolling/interactions)

'Store', 'Month', 'Week',
'Temperature', 'Fuel_Price', 'CPI', 'Unemployment', 'Holiday_Flag',
'Rolling_Mean_4', 'Rolling_Std_8', 'CPI_Unemployment'

```
LGB: Top 5 features:        LightGBM Final Test Results
Store, 1.682528             Final RMSE: 143960.09
Week, 0.062045              Final MAE: 91234.18
Month, 0.002564             Final MASE: 0.149
Rolling_Mean_4, 0.000301    Final R²: 0.9375
CPI_Unemployment, 0.000000
```

Set 3 (Lag_8 and interaction terms):

'Store', 'Month', 'Week',
'Temperature', 'Fuel_Price', 'CPI', 'Unemployment', 'Holiday_Flag',
'Rolling_Mean_4', 'Lag_8', 'Rolling_Std_4', 'Fuel_Temp_Interaction'

```
LGB: Top 5 features:        LightGBM Final Test Results
Store, 1.446081             Final RMSE: 142134.83
Week, 0.062865              Final MAE: 88164.75
Lag_8, 0.021263             Final MASE: 0.144
Month, 0.002760             Final R²: 0.9391
Rolling_Mean_4, 0.000281
```

Appendix 5. XGBoost results with different feature sets

XGBoost with different feature sets

Default features

'Store', 'Month', 'Week', 'Temperature', 'Fuel_Price',
'CPI', 'Unemployment', 'Holiday_Flag'

```
XGBoost: Top 5 features:  XGBoost Final Test Results
Store, 1.591297           Test RMSE: 149833.37
Week, 0.056147            Test MAE: 98910.02
Month, 0.002523           Test MASE: 0.161
Unemployment, 0.001764    Test R²: 0.9323
CPI, 0.000246
```

Set 1 (best performance)

'Store', 'Month', 'Week',
'Temperature', 'Fuel_Price', 'CPI', 'Unemployment', 'Holiday_Flag',
'Rolling_Mean_4', 'Rolling_Std_4', 'Lag_4', 'Lag_1'

```
XGBoost: Top 5 features:  XGBoost Final Test Results
Lag_1, 0.424236           Test RMSE: 132975.74
Store, 0.266259           Test MAE: 82399.26
Lag_4, 0.075494           Test MASE: 0.134
Week, 0.032911            Test R²: 0.9467
Month, 0.001087
```

Set 2 (no lags, only rolling/interactions)

'Store', 'Month', 'Week',
'Temperature', 'Fuel_Price', 'CPI', 'Unemployment', 'Holiday_Flag',
'Rolling_Mean_4', 'Rolling_Std_8', 'CPI_Unemployment'

```
XGBoost: Top 5 features:  XGBoost Final Test Results
Store, 1.537092           Test RMSE: 151973.28
Week, 0.045809            Test MAE: 99914.16
Rolling_Mean_4, 0.008828  Test MASE: 0.163
Month, 0.002179           Test R²: 0.9304
Unemployment, 0.000177
```

Set 3 (Lag_8 and interaction terms):

'Store', 'Month', 'Week',
'Temperature', 'Fuel_Price', 'CPI', 'Unemployment', 'Holiday_Flag',
'Rolling_Mean_4', 'Lag_8', 'Rolling_Std_4', 'Fuel_Temp_Interaction'

```
XGBoost: Top 5 features:  XGBoost Final Test Results
Store, 1.110899           Test RMSE: 146855.23
Lag_8, 0.078453           Test MAE: 93472.77
Week, 0.050649            Test MASE: 0.152
Rolling_Mean_4, 0.002142  Test R²: 0.9350
Month, 0.001826
```

Appendix 6. MLP results with different feature sets

MLP with different feature sets

Default features

'Store', 'Month', 'Week', 'Temperature', 'Fuel_Price',
'CPI', 'Unemployment', 'Holiday_Flag'

MLP: Top 5 features:	MLP Final Test Set Results
Store, 0.397338	Final RMSE: 508367.36
CPI, 0.245198	Final MAE: 421823.64
Month, 0.042519	Final MASE: 0.6879
Temperature, 0.028058	Final R^2: 0.2212
Unemployment, 0.027263	

Set 1 (best performance)

'Store', 'Month', 'Week',
'Temperature', 'Fuel_Price', 'CPI', 'Unemployment', 'Holiday_Flag',
'Rolling_Mean_4', 'Rolling_Std_4', 'Lag_4', 'Lag_1'

MLP: Top 5 features:	MLP Final Test Set Results
Lag_1, 0.897991	Final RMSE: 153477.02
Lag_4, 0.205306	Final MAE: 94950.60
Month, 0.024775	Final MASE: 0.1548
Holiday_Flag, 0.018207	Final R^2: 0.9290
Week, 0.007549	

Set 2 (no lags, only rolling/interactions)

'Store', 'Month', 'Week',
'Temperature', 'Fuel_Price', 'CPI', 'Unemployment', 'Holiday_Flag',
'Rolling_Mean_4', 'Rolling_Std_8', 'CPI_Unemployment'

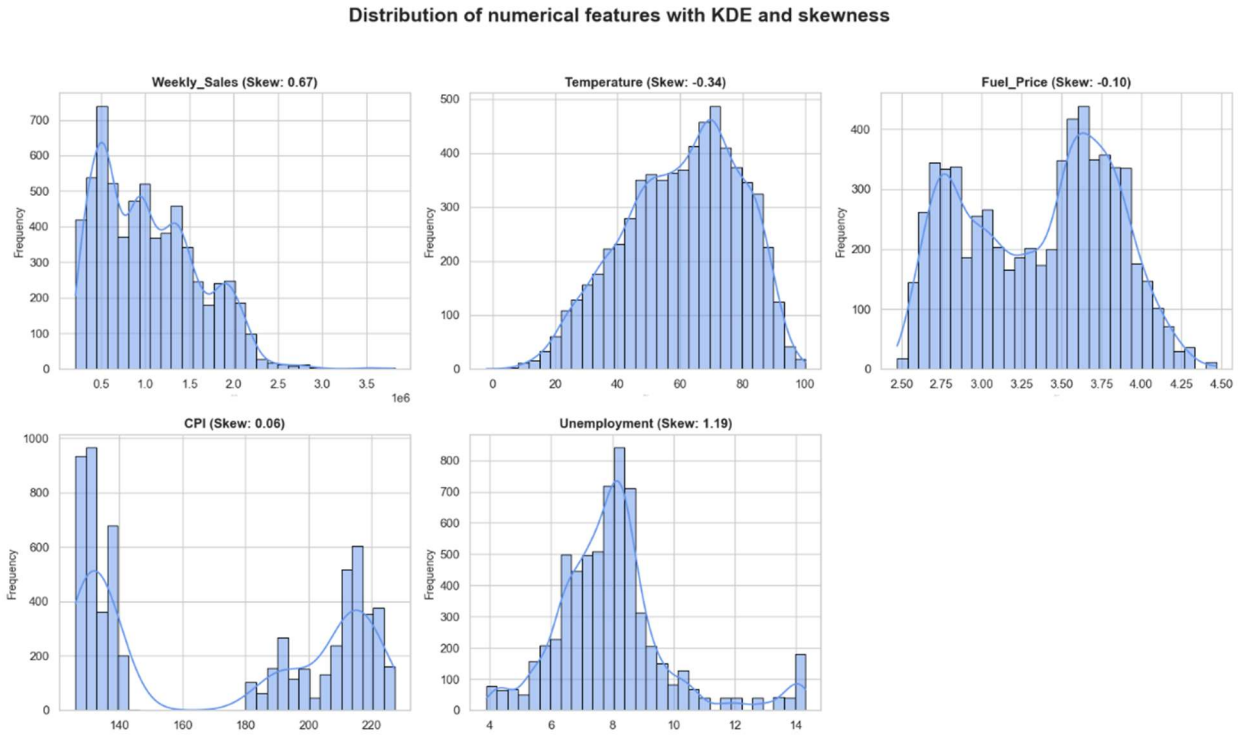
MLP: Top 5 features:	MLP Final Test Set Results
CPI, 0.354687	Final RMSE: 458064.56
Rolling_Mean_4, 0.312105	Final MAE: 371167.33
Store, 0.289075	Final MASE: 0.6053
CPI_Unemployment, 0.20244	Final R^2: 0.3677
Unemployment, 0.136773	

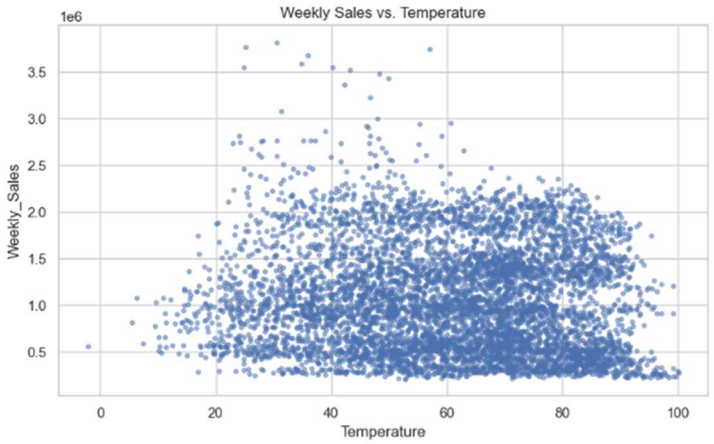
Set 3 (Lag_8 and interaction terms):

'Store', 'Month', 'Week',
'Temperature', 'Fuel_Price', 'CPI', 'Unemployment', 'Holiday_Flag',
'Rolling_Mean_4', 'Lag_8', 'Rolling_Std_4', 'Fuel_Temp_Interaction'

MLP: Top 5 features:	MLP Final Test Set Results
Lag_8, 1.534507	Final RMSE: 212144.91
Month, 0.058229	Final MAE: 142521.14
Temperature, 0.017256	Final MASE: 0.2324
Week, 0.012694	Final R^2: 0.8644
Fuel_Temp_Interaction, 0.012349	

Appendix 7. Exploratory data analysis






```
Correlation (Sales vs Temp): -0.0638100131794696
Correlation (Sales vs Fuel): 0.009463786314475135
```