MACHINE LEARNING APPROACH FOR RETAIL DEMAND FORECASTING: INTEGRATING FEATURE ENGINEERING WITH STACKED TREE-BASED ENSEMBLE AND DEEP LEARNING MODEL FOR IMPROVED ACCURACY

Thesis Submitted
In partial Fulfillment of the Requirements for the
Degree of

MASTER OF TECHNOLOGY
in
Industrial Engineering and Management
by

Shishir Acharya
(Roll No. 23/IEM/11)

Under the Supervision of
Dr. S.K. Garg
Professor, Department of Mechanical Engineering
Delhi Technological University

To the

Department of Mechanical Engineering

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi – 110042, India

June, 2025

DEPARTMENT OF MECHANICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

ACKNOWLEDGEMENT

I want to express my deepest gratitude to the Almighty for blessing me with the wisdom, resilience, and strength to see this research through to its completion.

I embrace the opportunity to express my deep sense of gratitude to my supervisor **Dr. S.K. Garg**, Professor, Department of Mechanical Engineering, Delhi Technological University, Delhi, for his constant guidance, valuable suggestions, and encouragement during this research. His encouragement, support, intellectual stimulation, perceptive guidance, immensely valuable ideas, and suggestions from the initial to the final level enabled me to develop an understanding of the subject. His scholarly suggestions, constant help and affectionate behavior have been a source of inspiration for me. I am extremely grateful to him for his continuous guidance.

I am cordially thankful to **Prof. Prateek Sharma**, Hon'ble Vice-Chancellor, Delhi Technological University, Delhi for providing this world class platform to conduct this research work. My acknowledgement will never be complete without the special mention of **Dr. B.B. Arora**, HOD Department of Mechanical Engineering, Delhi Technological University, Delhi.

I am thankful to Central library, Delhi Technological University for providing me the access to world class research literature that helped me in this research work.

I am thankful to all the authors and publishers whose research work has helped me in this project.

Shishir Acharya

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE'S DECLARATION

I, **SHISHIR ACHARYA**, hereby certify that the work which is being presented in the thesis entitled "**MACHINE LEARNING APPROACH FOR RETAIL DEMAND FORECASTING: INTEGRATING FEATURE ENGINEERING WITH STACKED TREE-BASED ENSEMBLE AND DEEP LEARNING MODEL FOR IMPROVED ACCURACY"** in partial fulfillment of the requirements for the award of the Degree of Master of Technology, submitted in the Department of Mechanical Engineering, Delhi Technological University is an authentic record of my own work carried out during the period of January 2025 to June 2025 under the supervision of Dr. S.K. Garg, Professor, Department of Mechanical Engineering, Delhi Technological University, Delhi.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Candidate's Signature

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE BY THE SUPERVISOR

Certified that **Shishir Acharya** (23/IEM/11) has carried their research work presented in this thesis entitled **"MACHINE LEARNING APPROACH FOR RETAIL DEMAND FORECASTING: INTEGRATING FEATURE ENGINEERING WITH STACKED TREE-BASED ENSEMBLE AND DEEP LEARNING MODEL FOR IMPROVED ACCURACY"** for the award of **Master of Technology** from Department of Mechanical Engineering, Delhi Technological University, Delhi, under my supervision. The thesis embodies the results of original work, and studies are carried out by the student himself, and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Signature

Dr. S.K. Garg

Professor

Department of Mechanical Engineering

Delhi Technological University, Delhi

Date:

Machine Learning Approach For Retail Demand Forecasting: Integrating Feature Engineering With Stacked Tree-Based Ensemble And Deep Learning Model For Improved Accuracy

Shishir Acharya

ABSTRACT

Retail demand forecasting plays a critical role in supply chain management by enabling businesses to predict future sales, manage inventory efficiently, and enhance production planning. With the advancement of machine learning, particularly tree-based ensemble methods and deep learning techniques, traditional forecasting systems have evolved to better handle the complex and non-linear patterns present in retail data. This study evaluates the forecasting performance of a stacked ensemble comprising tree-based models—Random Forest, XGBoost, LightGBM, and CatBoost—using Gradient Boost as the meta-learner, in comparison to an artificial neural network (ANN), a widely used deep learning model. The analysis is conducted on a five-year dataset covering multiple stores and products, using comprehensive feature engineering methods such as lag variables, rolling windows, month-over-month sales growth, and interaction terms to uncover significant temporal and cross-sectional patterns. Forecasts are generated for a three-month horizon to aid inventory control and production planning. An ANOVA test indicated that approximately 71% of the sales variance could be explained by engineered features, validating its effectiveness. The stacked ensemble model significantly outperformed the ANN, achieving a maximum R^2 value of 0.994 compared to 0.924 from the ANN. Moreover, the ensemble approach surpassed the performance of individual models, with the best-performing tree-based model incorporated into the stack. Overall, the study highlights that when supported by effective feature engineering, tree-based stacking ensembles offer superior accuracy in capturing non-linear relationships in retail demand forecasting, and statistical methods can be used to make decision for feature engineering to improve the forecast.

Keywords: demand forecasting, supply chain management, ensemble model, deep learning, ANN

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	**Full Form**
ANOVA	Analysis of Variance
ANN	Artificial Neural Network
LSTM	Long Short-Term Memory
RFR	Random Forest Regressor
XGBR	XGBoost Regressor
LGBM	Light Gradient Boosting Machine
CNN	Convolutional Neural Network
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error

CHAPTER 1

INTRODUCTION

1.1. General Introduction

Today in the competitive retail environment, customer demand forecasting is more than a numbers game; instead, it is an imperative strategic necessity. To anticipate what, when, and where of customer buys, retailers must forecast so that their supply chain efforts can respond accordingly. Every process, from procurement and manufacturing to assigning and replenishing inventory, depends heavily on accurate forecasts. Even a one-percentage-point enhancement in forecast accuracy can deliver significant value to companies—liberating tens of millions of working capital, increasing stock turnover, and enhancing both supplier cooperation and downstream customer satisfaction. These related effects make retail demand forecasting an integral part of supply chain efficiency and aggregate company profitability.

Despite its importance, retail forecasting is still a challenging task. Retail sales data, particularly in time series format, are inherently volatile. Volatility can be caused by many unforeseen events, such as promotional activities, consumer sentiment fluctuations fueled by social media, meteorological extremes, pay-day effects, and even worldwide crises like pandemics. Such fluctuations make linearity and constant variance assumptions underlying common statistical models such as exponential smoothing, ARIMA (Auto-Regressive Integrated Moving Average), and simple multiple regression outdated. Such models therefore have the propensity to fail to capture the sophisticated behavior typical of today's retail environments. Under forecasting results often results in stock shortages and customer dissatisfaction, whereas over forecasting results in excess stock, tied-up funds, and costly markdowns. Due to this reason, there is necessary to develop powerful forecasting techniques which can handle stable and seasonal patterns in the data and handle non-linear relationships and volatility(Zhang & Qi, 2005)(Kuvulmaz et al., 2005).

To solve the complex challenges faced with forecasting, Machine Leaning has been introduced and used to solve forecasting problems over the past decade. Machine models can solve complex, non-linear, high-dimensional relationships in large data. In several industries such as retail(Huber & Stuckenschmidt, 2020)(Güven et al., 2021), logistics(Salais-Fierro & Martínez, 2022)(Kantasa-ard et al., 2021), healthcare(Jain et al., 2024), energy(Emami Javanmard & Ghaderi, 2023)(Ahmad & Chen, 2018), machine learning models such as LSTM, Random Forest(RF),etc. are used. Due to the unique characteristic each machine learning model has, they can uncover complex correlations in the data which cannot be found by statistical models and help in improving the accuracy of the forecast.

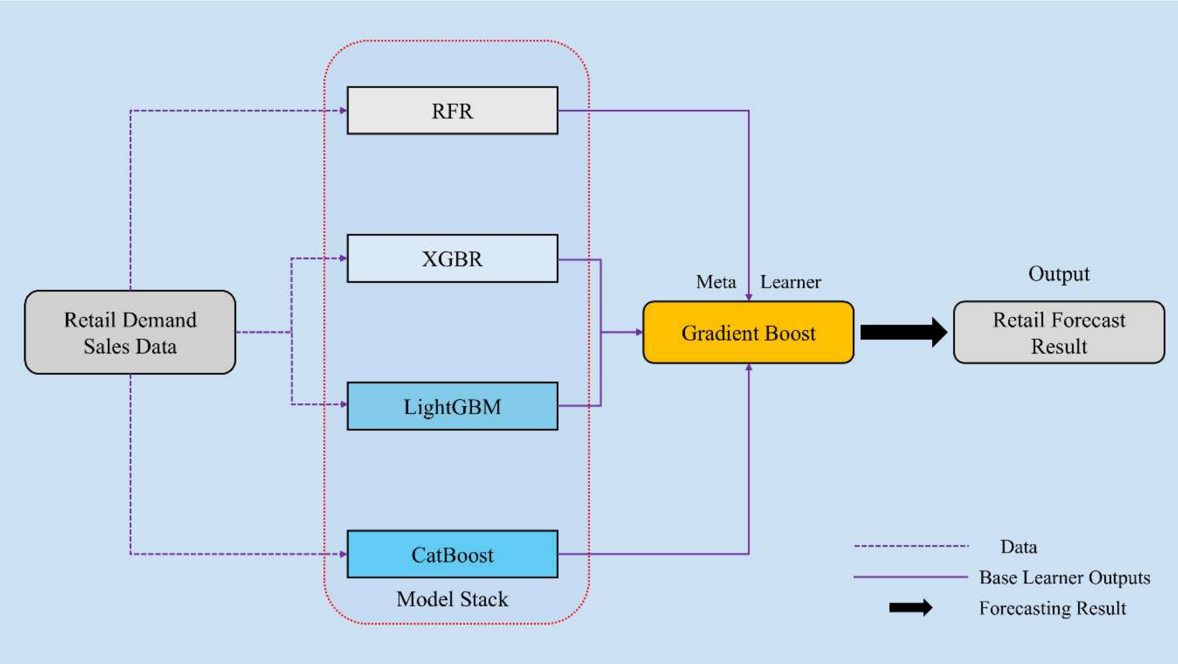


Figure 1.1 : Basic Stacking Process for Retail Demand Forecast

Ensemble learning has emerged as a powerful model to formulate robust forecasting by combining predictions from multiple models. Ensemble learning uses several machine learning models and combines the strengths of each model for improving the accuracy of the forecast. Different ensemble methods such as bagging, boosting, stacking are used which minimizes variance, reduces bias and combines several models into one respectively. The underlying philosophy of ensemble methods is that while no algorithm works best for all patterns of data or retail segments, a well-coordinated team of distinct models can improve overall accuracy by a large margin. Particularly in retail environments where demand shows sudden spikes, seasonal patterns, and intermittent zeros, ensemble methods have beaten their individual counterparts .

However, despite the presence of such technological leaps, one inherent fact remains: feature engineering forms a necessary foundation for true forecasting. While deep learning models are built to learn pertinent representations autonomously, true-world utility is greatly promoted through intentional preprocessing efforts. Domain-specific features—i.e., lagged sales, rolling averages, holiday flags, weather codes, and promotion flags—enable machine learning models to interpret raw data in terms that closely replicate real-world business contexts. This is particularly relevant to tree-based models and neural networks, as such features enable quicker convergence, avoid overfitting, and yield more meaningful generalizations. Methods like ANOVA-based feature selection are beneficial in identifying which combination of products and stores needs targeted interventions, thereby optimizing the intelligence and resource efficiency of the models. This research uses stacking of tree-based machine learning models and compares its performance with ANN, which is a deep learning model.

1.2 Research Gap Identification

1. Under exploration of stacking of tree-based models: While the existing research compares several tree-based models with different machine learning algorithms, the combination of ensemble models is not explored.
2. Focus on single product or single store: Many studies focus on forecasting as a whole leaving question for the robustness of the model across store-item pairs.
3. Limited integration of statistical validation into model design and feature engineering: Most existing research uses statistical models to evaluate the model performance but is not used earlier in the process for making feature engineering decisions.

1.3 Objectives of the Study

Based on the research gaps in existing research articles, the objectives of the research are given below:

1. To develop a robust forecasting model with comparison the performance of a stacked tree-based ensemble model (RF+XGBR+LGBM+CatBoost-GBR) with ANN.
2. Perform statistical analysis (ANOVA) for explaining store-item effect and variation in sales which is only used for forecasting not for basis for feature engineering.
3. Using feature engineering for making robust forecasts across all store-item combinations.

1.4 Methodology

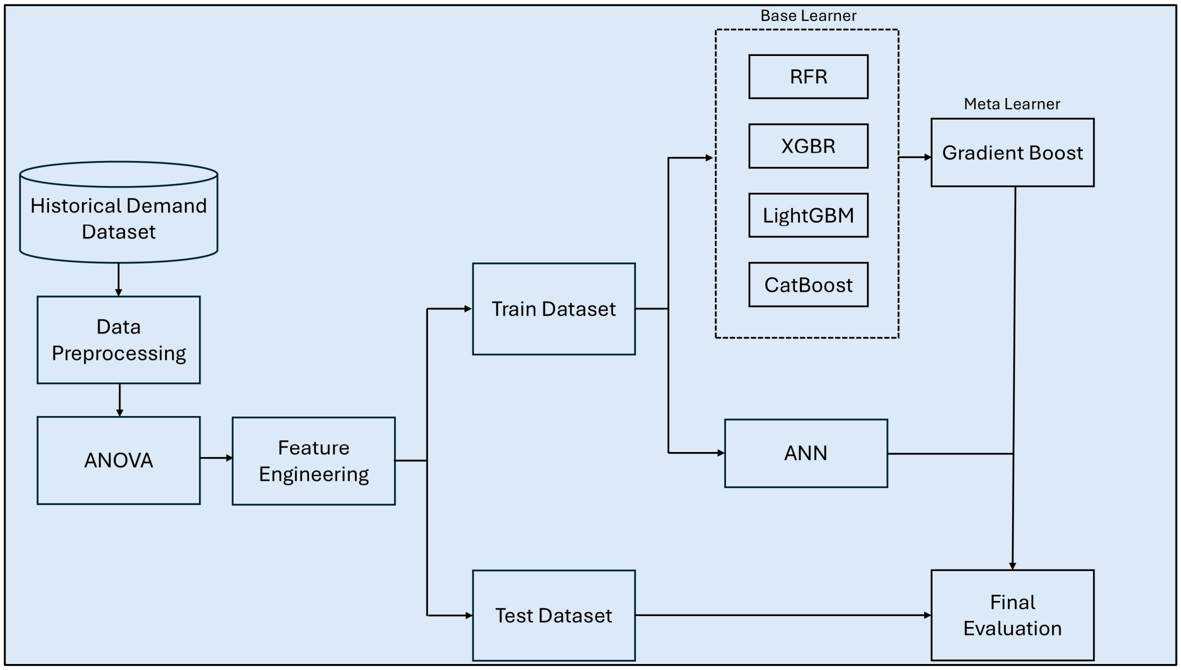


Figure 1.2 : Flowchart of Research for Retail Demand Forecasting

This research includes various steps to conduct the machine learning study for retail demand forecasting. Generally, two different models are formulated, stacking and ANN and they are evaluated based on key performance metrics. The first step includes collecting the historical demand dataset for retail demand forecast which is taken from Kaggle competition and the dataset is allowed to use for research purpose. The data preprocessing is carried out like checking for null values and converting the date into proper date format. After that ANOVA is conducted using SPSS and the result is interpreted. Based on the result, features such as lag, rolling, month-over-month sales, cumulative sales and interaction features are calculated for feature engineering. The next step involves training the dataset and later testing it for the last 3 months. Two models are trained in feature engineering, one stacked tree-based ensemble and another ANN. Stacking ensemble consists of RFR, XGBR, LightGBM and CatBoost as base learner and Gradient Boost as meta learner. The last step involves evaluating the performance of the model based on commonly used key performance metrics such as RMSE, MAE, MAPE and R^2.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Demand forecasting methodologies has evolved a lot with statistical innovation and computational advancement for decades with development of machine learning for improving the forecast accuracy. During early ages, demand forecasting in the retail sector depended upon qualitative methods such as expert judgment, market surveys, and simple historical averaging. The qualitative methods have very less precision for effective inventory management and have increased the inventory cost of the companies.

During the 1960s and 1970s more sophisticated statistical techniques were introduced called exponential smoothing methods founded by Brown, Holt, and Winters. The concept of weighted average was introduced with it which gives greater importance to recent observations and uses older data for forecasting.

Simple Exponential Smoothing (SES) used level components which generally uses average over time, Holt's Linear Trend method used trend components which uses fluctuations, and the Holt-Winters method used seasonal components(Lu, 2014). The temporal patterns in the retail demand data were captured due to the introduction of these methods.

The evolution of Box-Jenkins methodology during the 1970s was another milestone when Autoregressive Integrated Moving Average (ARIMA) models were introduced. ARIMA gave a general framework for modeling time series data by mixing autoregressive parts, differencing to induce stationarity, and moving average components. ARIMA allowed for more advanced treatment of temporal dependencies and remained a standard for time series forecasting for decades. (Thomassey, 2010) points out that good sales forecasting became a key success factor in retail supply chain management in the 1980s and 1990s as retailers were under pressure to maximize stock levels and minimize costs. During these time, casual variables and external factors were included for forecasting and to handle these multivariate methods such as Vector Autoregression (VAR) and regression-based models was introduced (Punia et al., 2020).

The use of machine learning methods for demand forecasting was begin during the late 1990s and early 2000s. Slowly, there was development in computing power as well as evolution in machine learning algorithms like k-Nearest Neighbors (kNN), Support Vector Machines (SVMs), and neural network structures which has an ability to handle non-linearity and volatility in the retail data effectively. (L. Liu et al., 2024). The introduced machine learning models were able to analyze complex patterns in the data and solve the problems which was not detected by statistical techniques.

Due to the inability of linear models to determine seasonal trends and behavior in time series data, the machine learning models were introduced to cope up with the existing problems(Zhang & Qi, 2005). The research shows that if the machine learning models, i.e. neural networks, are designed correctly then it could perform better with datasets having non-linear trends which opens door for further improvement in the existing models, maybe with the introduction of grid search techniques. Several researchers are going on to introduce new machine learning models having more forecast accuracy than the existing one which can make robust forecasts for rapidly changing retail markets. Several machine learning models such as deep learning, and hybrid techniques have been introduced which have improved forecasting accuracy as well as helped retail to optimize its inventory.

The evolution of forecasting methodologies shows the need for more accurate and adaptable approaches to predict consumer demand in increasingly complex and dynamic retail environments. Also, due to the advancement there is collaboration of advanced machine learning, deep learning, and hybrid approaches where the models compete to push the boundaries of forecasting accuracy.

2.2 Bibliometric Analysis

For reviewing and analyzing the scientific literature, bibliometric analysis is the popular method which gives insights about the works carried out by the researchers (Merigó & Yang, 2017). We have used bibliometrix package of R library which is designed by (Aria & Cuccurullo, 2017).

The study utilizes research from SCOPUS database of "dtulibrary.remotexs.in" which is library provided by DTU and contains collection of the SCOPUS indexed researches. The database is filtered for a span of 10 years from 2015 to 2025. To ensure that the bibliometric analysis aligns with the study, the database is searched for relevant keywords. The library is searched with "AND" keywords so that all the relevant papers are included in the study. The databased is searched for the following keywords: "feature engineering" AND "time series forecasting", "demand forecasting" AND "supply chain management" AND "machine learning", "demand forecasting" AND "supply chain management" AND "machine learning" "ensemble" and "machine learning" AND "time series forecasting" AND "retail". 298 documents were found from the SCOPUS database combined with the above keywords where 15 duplicates files were found.

The keywords along with associated papers from 2015-2025 are as follows:

Table 2.1: Paper count for bibliometric analysis with keywords

Keywords	Paper Count
"feature engineering" AND "time series forecasting"	109
"demand forecasting" AND "supply chain management" AND "machine learning"	165
"demand forecasting" AND "supply chain management" AND "machine learning" "ensemble"	8
"machine learning" AND "time series forecasting" AND "retail"	31

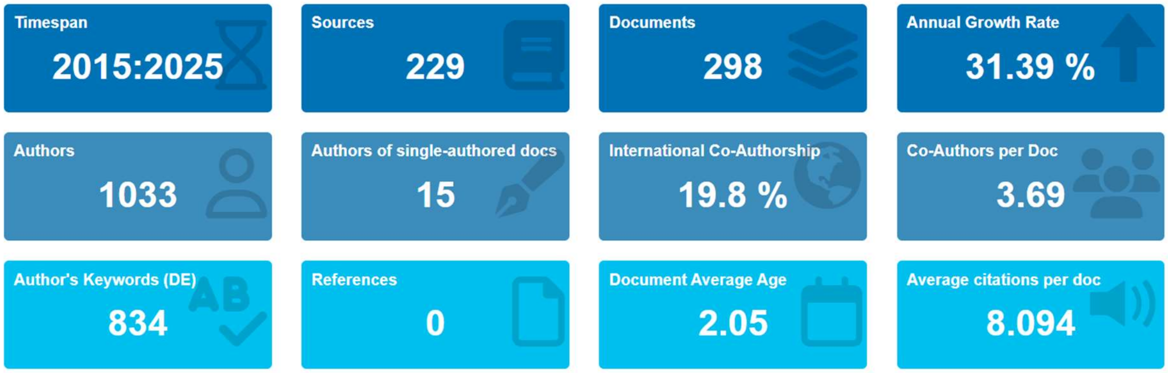
Table 2.2 shows the description of the articles used for bibliometric study along with its results. The average citation per doc is 8.094 with an annual growth rate of 31.39% paper production which is impressive.

Table 2.2 : Information of bibliometric study for Retail Demand Forecasting

Description	Results
Main Information About Data	
Timespan	2015:2025
Sources (Journals, Books, etc)	229
Documents	298
Annual Growth Rate %	31.39
Document Average Age	2.05
Average citations per doc	8.094
References	0
Document Contents	
Keywords Plus (ID)	1637
Author's Keywords (DE)	834
Authors	
Authors	1033
Authors of single-authored docs	15
Authors Collaboration	
Single-authored docs	23
Co-Authors per Doc	3.69
International co-authorships %	19.8
Document Types	
article	99
article conference paper	1
book	1
book chapter	29
conference paper	148

conference paper book	1
conference review	7
editorial	2
review	10

Figure 2.1 shows the summary of bibliometric study for Retail Demand Forecasting where there are a total of 1033 authors in 298 documents and international Co-Authorship of 19.8%.



Timespan
2015:2025

Sources
229

Documents
298

Annual Growth Rate
31.39 %

Authors
1033

Authors of single-authored docs
15

International Co-Authorship
19.8 %

Co-Authors per Doc
3.69

Author's Keywords (DE)
834

References
0

Document Average Age
2.05

Average citations per doc
8.094

Figure 2.1 : Summary of bibliometric study for Retail Demand Forecasting

Figure 2.2 shows the annual production of paper per year which is increasing yearly with an increase of 109 from 49 in 2024.

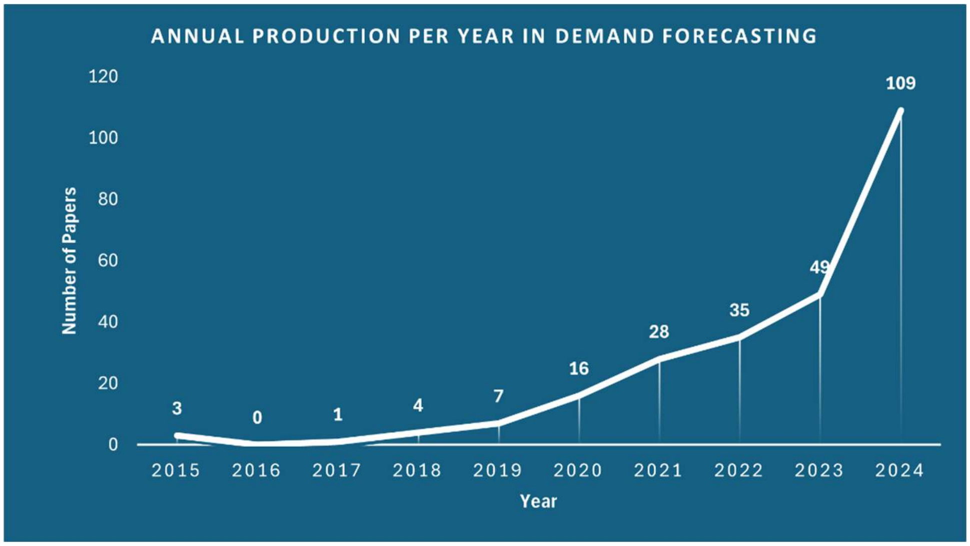


Figure 2.2: Annual production per year in demand forecasting

Figure 2.3 shows the most global cited documents in 10 years, and the plot shows top 10 papers in the bibliometric articles list.

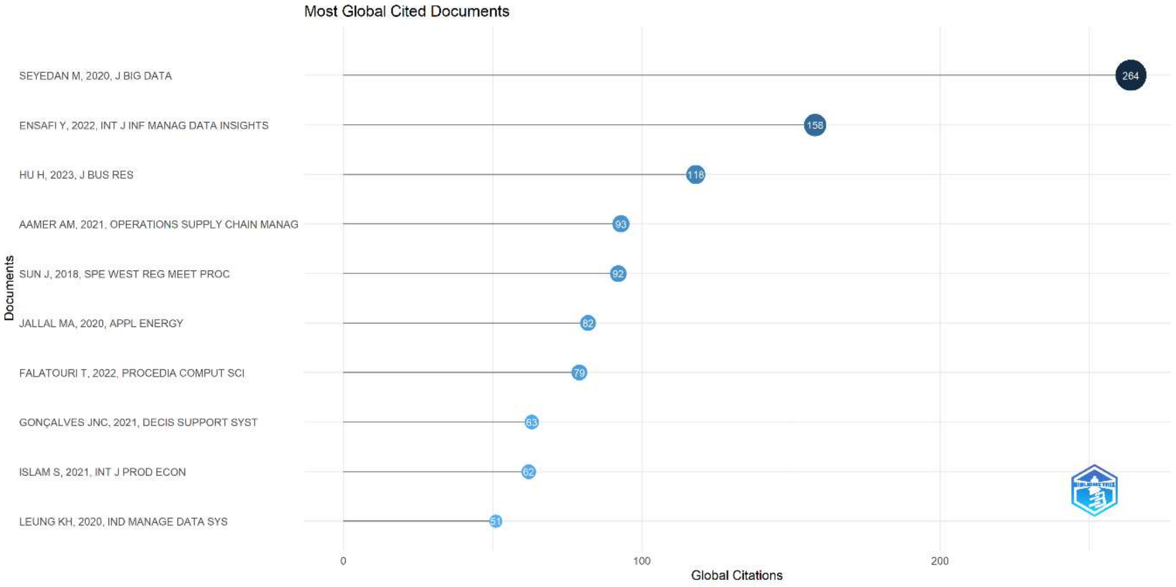


Figure 2.3: Most Global Cited Documents

Figure 2.4 shows the most occurring words in the keyword for the bibliometric study papers with “forecasting” occurring in 109 papers followed by “supply chain management” in 92 papers and Figure 2.5 shows WordCloud plot for it.

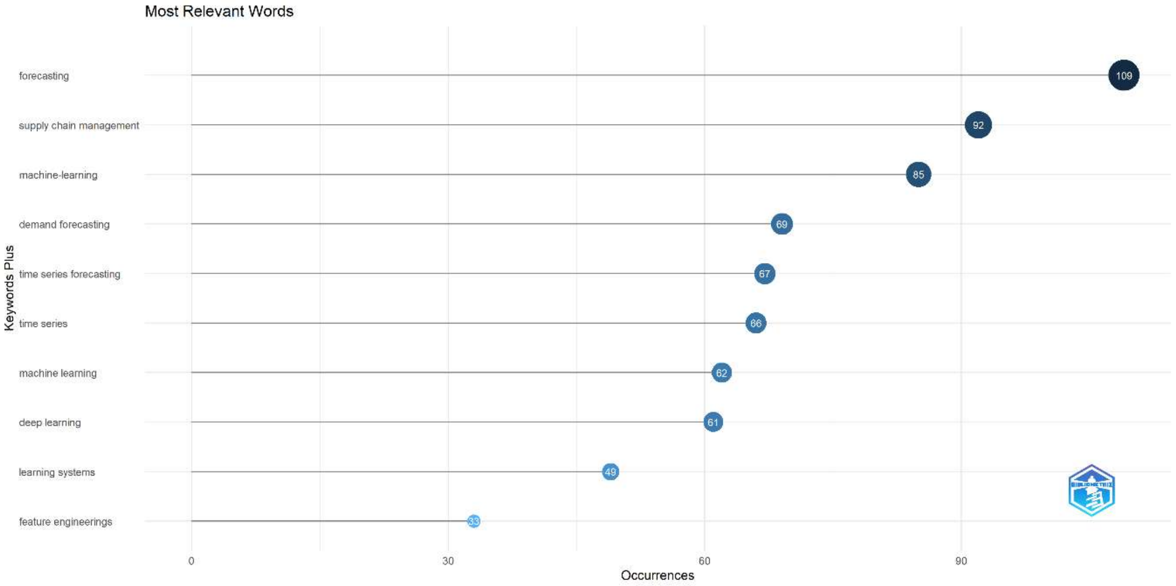


Figure 2.4: Plot for Most Relevant keywords

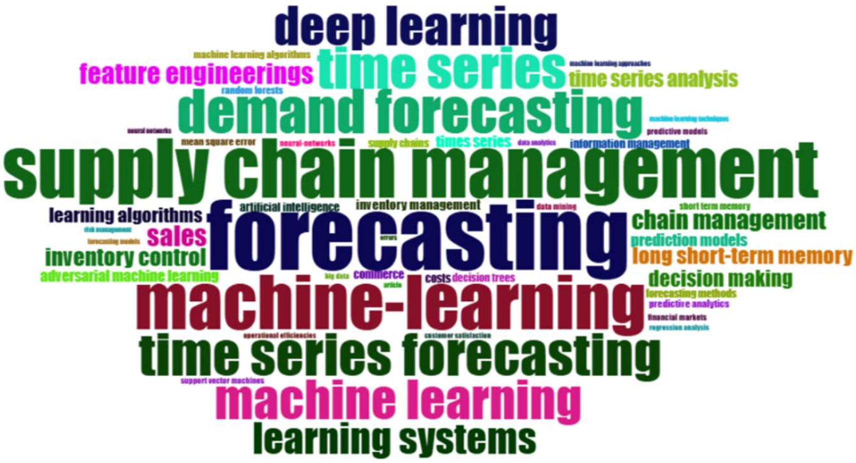


Figure 2.5: WordCloud for Most Relevant keywords

Figure 2.6 shows the Co-occurrence network for the keywords which is modelled using VOS viewer with main keywords such as supply chain management, machine learning, forecasting, demand forecasting, time series forecasting, time series and machine-learning.

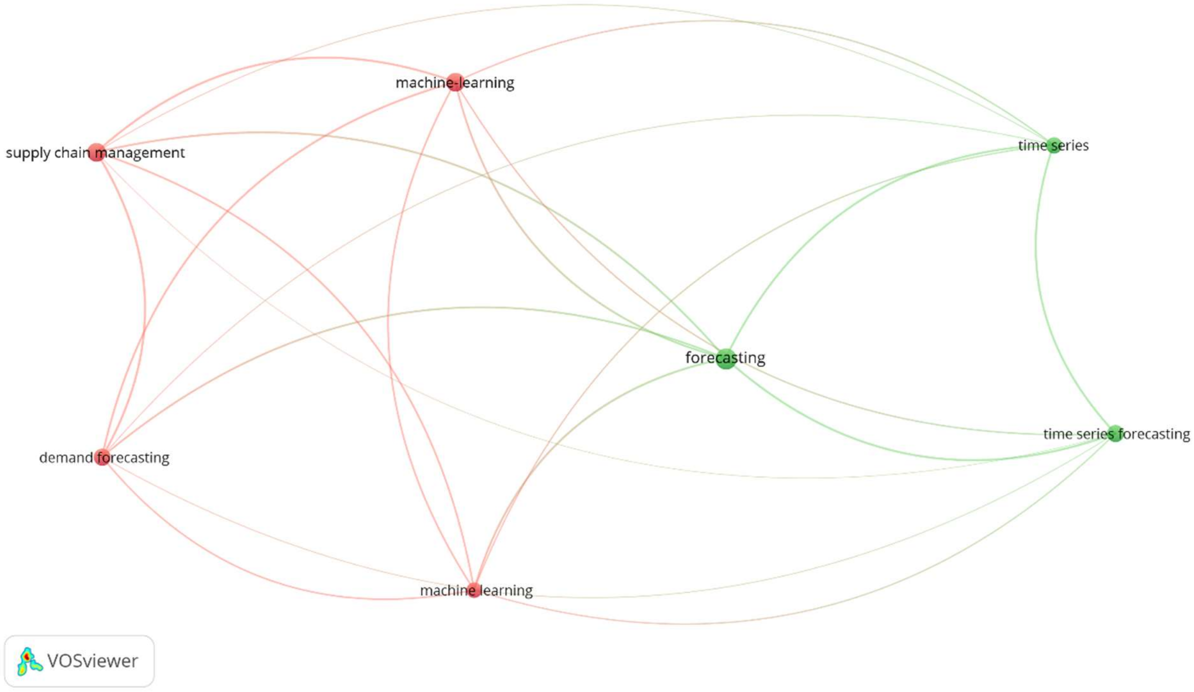


Figure 2.6: Co-occurrence Network for main keywords using VOS viewer

2.3 Statistical Methods for Demand Forecasting

In retail applications, traditional statistical methods have been widely used for a long time, particularly with time series techniques to achieve forecasting results. ARIMA (AutoRegressive Integrated Moving Average) models have dominated among traditional methods due to their remarkable longevity and utility. According to Falatouri et al (Falatouri et al., 2022) , SARIMA (Seasonal ARIMA) has been widely used for demand forecasting because of its maturity and interpretability, and excels in capturing both seasonal and trend components in retail data.

SARIMA models have been applied for specific retail contexts which has shown promising results. (Falatouri et al., 2022) used a SARIMA model for demand forecasting of computer parts which performed better compared to simple smoothing methods. The research shows the importance of incorporating seasonal components in forecasting models for products with recurring demand patterns, which is a common characteristic in retail environments. Exponential smoothing has been also used as a statistical forecasting method. Exponential smoothing along with Simple, Holt's, and Holt-Winters methods, applies exponentially decreasing weights to historical observations(Gahirwal, 2013). Due to the interpretability , computational efficiency, and reasonable accuracy of the model, it has been widely used for adoption of retail demand forecasting (Chowdhury et al., 2025). Due to the intuitive nature of exponential smoothing, it has been preferred in retail forecasting. It is mainly used for products with stable demand patterns. Similarly, regression is also a popular model used in retail demand forecasting. It computes the relation between dependent and independent variables particularly between sales and various predictor variables, such as price, promotions, and seasonality. Also, multiple regression models are popular types of regression which incorporate external factors that influence demand. It allows retailers to compute the impact of marketing initiatives, pricing strategies, and competitive actions.

Although statistical methods have historical importance and continued utility, there are plenty of limitations of these modes in contemporary retail environments. (Lu, 2014) describes statistical models often struggle to capture complex, non-linear relationships in the data of modern retail patterns. Also, (Zhang & Qi, 2005) explains that the traditional statistical methods tends to underperform in retail time series data when there is combined seasonal and trend patterns in the data. Statistical methods lack adaptability in structural changes of demand patterns. These methods generally assume stationary processes and sometime needs explicit differencing to achieve stationarity due to which it is less responsive to evolving market dynamics (Kuvulmaz et al., 2005). It becomes the major problem in the fast-moving retail item category in which there is rapid change in customer preferences and market conditions. There is development in statistical models by integration of external variables such as ARIMAX (ARIMA with exogenous variables) which used traditional ARIMA along with incorporation of external factors such as promotional activities, pricing changes, and competitor actions(Punia et al., 2020)(Feizabadi, 2020). Because of this incorporation, there is a significant improvement in the forecast accuracy of the retail sector where external factors significantly influence demand patterns. Although the statistical models improve accuracy, it often face challenges to capture complex interactions between multiple external factors and its non-linear effects on demand (Feizabadi, 2020).

While statistical methods have been partially superseded by more advanced machine learning approaches in many applications, they continue to serve as important benchmarks and components in hybrid forecasting systems. When evaluating the performance of more sophisticated algorithms, researchers have preferred to use statistical methods as baseline comparisons(Punia & Shankar, 2022).

2.4 Machine Learning Approaches for Forecast Enhancement

To improve the accuracy of the forecast, researchers have developed machine learning models which can predict complex demand patterns with greater accuracy. In retail Decision Trees, Random Forests (RF) are popular machine learning models. Apart from tree-based models, boosting algorithms are also used in the retail sector.

Random Forest is type of ensemble machine learning model that combines multiple decision tree and uses average of all decision trees for forecasting and due to its ability to prevent overfitting, it is preferred in retail forecasting.(Yazici & Domínguez-Gutiérrez, 2025). Random Forest is preferred as it has ability to handle non-linear relationships in the data due to which it can to reduce overfitting where there is volatility (Priyadarshi et al., 2019). Also, when multiple factors impact the demand in the forecasting. Using Random Forest can give better accuracy. Gradient Boosting is also another type of supervised machine learning algorithm which is preferred for retail demand forecasting as is suitable for forecasting of individual items at individual stores. For tree based-forecasting, when speed and accuracy along with the scalability is required, XGBoost is used by many researchers.(T. Chen & Guestrin, 2016). If there is any type of complex seasonal pattern and promotional effects in the dataset then researchers prefer to use XGBoost in retail demand forecasting rather than statistical foresting methods (Haselbeck et al., 2022). Similarly, LightGBM and CatBoost, which is a boosting ensemble machine learning algorithm is used for retail demand forecasting along with other models. If there is holidays and changing consumer behavior to be considered while forecasting which can greatly affect the accuracy then LightGBM is used (Huber & Stuckenschmidt, 2020). Also, for categorical variables, CatBoost is used as it does not need encoding to handle the model and can automatically integrate within the model (Nasseri et al., 2023). Similarly, Support Vector Regression (SVR) is also used for forecasting in retail sector and it has incorporated variable selection and is widely used in computer demand forecasting (Lu, 2014). SVR can convert non-linear problems into linear to solve the complex patterns in the data by using the kernel functions, so it is widely used.

Another machine learning technique used for retail forecasting is K-Nearest Neighbors (KNN). KNN was used for intermittent demand forecasting in retail apparel by (Güven et al., 2021), who discovered that it worked effectively in situations involving sparse data, which are typical in the fashion retail industry. Retailers without a lot of data science experience can now use the method because of its ease of use and straightforward approach to prediction based on comparable past trends. Retail demand forecasting has shown promise when clustering and predictive modeling are combined. (I. F. Chen & Lu, 2021) ceated a method for multichannel fashion shops that integrated machine learning techniques with k-means clustering. Their method obtained higher accuracy than applying machine learning algorithms to the entire dataset by first segmenting products based on demand patterns before applying predictive models. This tactic is a prime example of how preprocessing methods can improve machine learning models' performance in retail forecasting applications.

Feature engineering is widely used in machine learning for retail applications which helps to improve forecast accuracy. The relevant features such as lag, rolling, interaction features must be extracted from the data to achieve robust forecast (Mejía & Aguilar, 2024). By capturing temporal dynamics and contextual elements in the retail environment, the accuracy of the forecast can be significantly improved with machine learning models. In many cases of retail forecasting, machine learning models have performed exceptionally well compared to traditional statistical methods. Ensemble machine learning models performs better and achieve high forecast accuracy compared to ARIMA-based methods in retail (Joseph et al., 2022). A research conducted by (Güven et al., 2021) signifies that the accuracy of Random Forest and KNN is way better compared to exponential smoothing methods in apparel demand forecasting.

Although machine learning models are widely utilized in retail demand forecasting, it has numerous drawbacks. There is a problem of interpretation of the model when the complex ensemble approaches are used and there is no information on the effect of features in the prediction. Also, a large volume of data is required for machine learning models, and it cannot perform well in case of new products. Deep learning models are also evolving machine learning models which are widely used for forecasting purposes in many sectors. Deep learning models have hidden layers due to which they can learn from complex patterns of the data. There is also no requirement for explicit feature engineering in the deep learning models due to this. As feature engineering needs time to model the dataset, deep learning models are introduced to solve the problem due to which the models can learn all the patterns in the data itself.

Long Short-Term Memory (LSTM) is a popular deep learning model which is recently preferred for retail demand forecasting. It can remember the pattern of sales in the time series dataset for long time due to which it can make accurate forecasts. (Punia et al., 2020) also conducted research on deep learning which showed that LSTM performs better than traditional forecasting methods like ARIMA and SARIMA which he forecasted for multichannel retail. This research shows that LSTM can capture complex seasonal patterns in the data for long time periods in the retail sector. Another type of deep learning is Artificial Neural Network (ANN) and it is widely considered for retail demand forecasting. Research has also shown that ANN can forecast properly in multi-channel retail sector when proper tuning is done in the dataset and related input features are provided (Mitra et al., 2022). The architecture used in ANN helps it to learn complex non-linear relationships in the retail data and gives better results even when multiple factors are affecting the demand. Recently transformers are evolving deep learning model in the retail sector. Along with it, hybrid and ensemble machine learning models are also preferred.

Ensemble models combine the strength of all the machine learning models used for ensemble forecasting due to which it has better accuracy compared to individual models. As the models can combine strengths of the machine learning models used, it can determine complex patterns in the data and help in forecasting. It has been used for short-term prediction of agribusiness time series with the utilization of boosting, bagging and stacking which is the type of ensemble models (G. T. Ribeiro et al., 2019). (Punia & Shankar, 2022) integrated the components of classical time series analysis for demand forecasting with deep learning-based decision support system for predictive analytics. Similarly, (Joseph et al., 2022) forecasted for store item demand forecasting using a hybrid model of integrated CNN and bidirectional LSTM, and showed the improvement in forecasting performance from architectural improvements between two deep learning models.

While ensembles generally improve accuracy in forecasts, they also simultaneously introduce added complexity and computational requirements. (Seyedan et al., 2022) addressed the problem by developing a cluster-based demand forecasting framework that utilizes Bayesian model averaging as a robust ensemble learning approach. Their method successfully balanced the improvements in accuracy against resource constraints posed by computational capabilities, offering a workable solution for retailers with resource constraints.

2.5 Feature Engineering in Forecasting

Feature engineering has emerged as a critical component of successful forecasting systems, particularly in retail contexts where demand patterns are influenced by numerous interconnected factors. The process of feature engineering involves transforming raw data into meaningful representations that enhance a model's ability to capture underlying patterns and relationships. In retail demand forecasting, effective feature engineering can dramatically improve prediction accuracy by incorporating domain knowledge and extracting relevant temporal patterns.

Temporal features represent a fundamental category of engineered variables for retail forecasting. Lag features, created by shifting historical sales data at various intervals (e.g., 1 day, 7 days, 30 days, and 365 days), enable models to capture recurring patterns and autoregressive relationships in demand data. (Huber & Stuckenschmidt, 2020) demonstrated that lag features capturing weekly, monthly, and yearly patterns significantly improved forecast accuracy for retail sales, particularly for products with strong seasonal components. Similarly, (Ma & Fildes, 2021) found that incorporating lag variables at multiple time scales enhanced the model's ability to detect both short-term and long-term dependencies in retail sales data.

Rolling window statistics provide another valuable set of features for retail forecasting. These include moving averages, standard deviations, minimums, and maximums calculated over various time windows (e.g., 7 days, 30 days, 365 days). (Mejía & Aguilar, 2024) showed that rolling statistics helped identify longer-term trends while smoothing short-term anomalies, reducing the model's tendency to overfit to noise in the data. Furthermore, these features provide crucial context for recent sales performance, enabling more accurate predictions during periods of volatility or changing trends. Growth rate features, such as month-over-month or year-over-year sales growth, capture dynamic aspects of demand that absolute values might miss. These features are particularly valuable for detecting acceleration or deceleration in sales trends, which can signal emerging patterns in consumer behavior. (Nasseri et al., 2023) incorporated growth rate features in their comparison of tree-based ensembles and deep learning approaches for retail demand prediction, finding that these features contributed significantly to forecast accuracy, especially during periods of changing market conditions.

Interaction features represent combinations of different variables that together may provide additional predictive power. Store-month interactions, for example, can capture store-specific seasonal patterns that might be lost in more general representations. Similarly, weekday-sales interactions help models understand how demand varies across different days of the week, which is particularly important for retail operations planning. These interaction terms allow models to capture non-additive relationships between variables, enhancing their ability to represent complex retail dynamics. Calendar-based features have proven especially valuable for retail forecasting. Huber and Stuckenschmidt emphasized that calendar-based special days significantly impact retail demand, requiring specialized features to capture these anomalies effectively(Huber & Stuckenschmidt, 2020). Their research demonstrated that encoding holidays, promotional periods, and special events as explicit features allowed models to anticipate the distinct demand patterns associated with these calendar events. This approach has been widely adopted in retail forecasting systems to account for the substantial impact of holidays and promotional calendars on consumer behavior.

Categorical encoding techniques transform nominal variables like store identifiers, product categories, or day-of-week indicators into numerical representations suitable for machine learning algorithms. While simple one-hot encoding is common, more sophisticated approaches such as target encoding or feature hashing may be employed for high-cardinality variables. (Tong et al., 2023) demonstrated that appropriate encoding of categorical variables related to store characteristics and product attributes enhanced the performance of spatial-temporal CNN models for ultra-short-term load forecasting. External data integration represents an advanced form of feature engineering that incorporates variables beyond the immediate sales history. (Haque et al., 2023) highlighted the value of enriching retail forecasting models with macroeconomic indicators such as Consumer Price Index (CPI), Index of Consumer Sentiment (ICS), and unemployment rates. Similarly, (Saha et al., 2022), explored the integration of weather data, social media signals, and competitive intelligence into retail demand forecasting models. These external variables provide crucial context about the broader environment in which retail sales occur, allowing models to account for factors that might otherwise appear as unexplained variance in demand patterns.

Statistical validation of engineered features represents an important but often overlooked aspect of the feature engineering process. The use of techniques such as Analysis of Variance (ANOVA) to evaluate the explanatory power of potential features can guide more effective feature selection. Statistical analysis can reveal which store and item characteristics significantly impact sales variability, informing the creation of more relevant features. This approach ensures that feature engineering efforts are directed toward variables with genuine predictive power, rather than introducing noise into the model. Feature selection and dimensionality reduction techniques help manage the proliferation of features that often result from extensive feature engineering. Techniques such as Principal Component Analysis (PCA), recursive feature elimination, and regularization methods help identify the most informative features while reducing model complexity. (Lu, 2014) incorporated variable selection methods with support vector regression for computer product demand forecasting, achieving improved accuracy with a more parsimonious model. This balance between feature richness and model simplicity is crucial for creating forecasting systems that generalize well to new data.

Automation of feature engineering processes has gained attention as the complexity of retail forecasting applications increases. Automated feature engineering tools can generate and test thousands of potential features, identifying combinations and transformations that human analysts might overlook. These approaches are particularly valuable for large-scale retail forecasting systems that must generate predictions for thousands of products across hundreds of locations, where manual feature engineering becomes prohibitively time-consuming.

2.6 Applications in Retail and Related Sectors

The application of advanced forecasting methodologies extends across diverse sectors, with retail serving as a primary domain for innovation and implementation. Within retail itself, forecasting applications span multiple subcategories, each presenting unique challenges and opportunities for predictive analytics(Haque et al., 2023).

Fashion retail represents a particularly challenging forecasting domain due to short product lifecycles, high seasonality, and the significant impact of trends and consumer preferences. (Thomassey, 2010) highlighted that sales forecasting in the clothing industry serves as a key success factor for supply chain management, enabling better inventory planning and reducing the risk of stockouts or excess inventory. (I. F. Chen & Lu, 2021) developed an approach specifically for multichannel fashion retailers that integrated clustering with machine learning algorithms, achieving superior accuracy by accounting for the distinct demand patterns across different product categories and sales channels. Grocery and fresh food retail presents another specialized application area where forecasting accuracy directly impacts profitability and sustainability. The perishable nature of products adds urgency to forecast accuracy, as errors can result in significant waste or lost sales opportunities. (Priyadarshi et al., 2019) analyzed sales forecasting for selected vegetables at the retail stage, demonstrating how performance varied across different forecasting approaches. Their findings underscored the importance of product-specific forecasting strategies that account for the unique characteristics of fresh food items, including seasonality, shelf life, and sensitivity to external factors such as weather conditions.

Electronics and technology retail features complex demand patterns influenced by product lifecycles, technological innovation, and competitive dynamics. (Lu, 2014) developed a sales forecasting approach for computer products based on variable selection and support vector regression, addressing the challenges of predicting demand for items with relatively high unit values and irregular purchase frequencies. This application demonstrated how specialized forecasting approaches could be tailored to the characteristics of technology products, where traditional time-series methods often proved inadequate. Home appliance retail represents another significant application area with distinct forecasting challenges. (Duan & Dong, 2024) constructed an ensemble learning model specifically for home appliance demand forecasting, addressing the complex interplay between seasonality, promotional effects, and replacement cycles that characterize this product category. Their approach combined multiple forecasting models to capture different aspects of demand generation for appliances, achieving superior accuracy compared to single-model approaches.

Beyond specific retail categories, forecasting applications have extended to supply chain operations that support retail activities. (Aburto & Weber, 2007) developed an improved supply chain management approach based on hybrid demand forecasts, demonstrating how accurate predictions could enhance inventory planning, transportation scheduling, and warehouse operations. Their research highlighted the cascading benefits of forecast accuracy throughout the retail supply chain, from manufacturers to distribution centers to individual stores. Cross-industry applications have provided valuable insights into retail forecasting. Energy sector forecasting, particularly short-term load forecasting, has developed sophisticated methodologies that have been subsequently adapted for retail applications. (Manandhar et al., 2024) evaluated new forecasting metrics using Prophet, Random Forest, and LSTM models for load forecasting, generating insights regarding model selection and evaluation that transferred effectively to retail contexts. Similarly, (Emami Javanmard & Ghaderi, 2023) developed an optimization model based on machine learning algorithms for energy demand forecasting across seven sectors, introducing methodological innovations later adopted in retail forecasting applications.

Steel industry demand forecasting has also contributed valuable methodological insights. (Raju et al., 2022) demonstrated the effectiveness of ensemble learning approaches for demand forecasting in steel industries, combining multiple base learners with meta-models to achieve superior accuracy. The principles of ensemble construction and model integration developed in this industrial context have informed similar approaches in retail forecasting, highlighting the cross-pollination of methodologies across different forecasting domains. Agricultural forecasting represents another related field with significant parallels to retail demand prediction. (M. H. D. M. Ribeiro & dos Santos Coelho, 2020) explored ensemble approaches based on bagging, boosting, and stacking for short-term prediction in agribusiness time series, developing methodologies subsequently adapted for retail applications. The seasonal nature of agricultural production and its sensitivity to external factors such as weather conditions present forecasting challenges like those encountered in retail, facilitating methodological transfer between these domains.

The integration of retail forecasting with broader business intelligence systems represents an emerging application area. Demand forecasting has become increasingly connected to other business functions, including marketing, store operations, and financial planning**.** This integration enables retailers to leverage forecast insights beyond inventory management, informing decisions regarding staffing levels, promotional planning, and capital investment. The development of interactive dashboards and visualization tools has facilitated this broader application of forecast data, making predictive insights accessible to stakeholders across retail organizations. Mobile and e-commerce platforms present unique forecasting challenges and opportunities. The availability of rich customer interaction data, combined with rapid demand fluctuations and the absence of physical inventory constraints, has driven innovation in forecasting methodologies for online retail. (Joseph et al., 2022) developed a hybrid deep learning framework with CNN and bi-directional LSTM specifically designed for this context, demonstrating how architectural innovations could enhance forecast accuracy for digital retail channels. The integration of clickstream data, search trends, and social media signals has further enriched forecasting models for online retail, enabling more responsive and accurate demand predictions.

2.7 Previous Works

Below table shows some of the works carried out by the researchers between 2018-2024. The research works are related to demand forecasting with different machine learning approaches with its key findings and includes whether they have used ensemble learning or not. The most common metrics used in Table 2.3 research is RMSE, but it is not sufficient to determine the model performance as it only shows the deviation of forecasted value with the actual value. To validate performance, R^2 is commonly preferred as it shows how close the forecasted value is with actual value having range from 0 to 1 where 1 is perfect fit. The summary of various works is given below:

Table 2.3 : Previous research carried out in the field of Demand Forecasting

Publication	Year	Sector	Approach	Key Findings	Ense-mble
(L. Wang et al., 2018)	2018	Food Price Volatility	EEMD	Factors such as trade policies, financial factors, events causes food price fluctuations	No
(Priyadarshi et al., 2019)	2019	Retail Demand Forecasting	LSTM, SVR, RFR, GBR, XGBoost, ARIMA	Forecasted from Tuesday to Sunday daily on Tomato, Potato and Onion with minimum MAPE of 0.18 with LSTM for Potato	Yes
(Wanchoo, 2019)	2019	Retail Demand Forecasting	DNN, GBM	Forecasted univariate sales achieving RMSE as low as 0.023	Yes

(M. H. D. M. Ribeiro & dos Santos Coelho, 2020)	2020	Agricultural Commodities Price Forecasting	RF, GBM, XGB, MLP, KNN, SVR	Achieved MAPE between 11.04%–40.36%	Yes
(Huber & Stuckenschmidt, 2020)	2020	Retail Demand Forecasting	LGBM, LSTM, MLP, REG	Achieved minimum MAE of 0.965 with LSTM-REG	Yes
(Feizabadi, 2020)	2020	Demand Forecasting	ARIMAX, NN	Achieved forecast overall accuracy of about 89.4% for NN	No
(Güven et al., 2021)	2021	Retail Demand Forecasting	RF, KNN	Achieved minimum RMSE of 119.31 with RF	Yes
(I. F. Chen & Lu, 2021)	2021	Fashion Retail Forecasting	KM-ELM, KM-SVR	MAPE up to 0.13% achieved with KM-ELM	Yes
(Giri & Chen, 2022)	2022	Retail Demand Forecasting	SVM, RF, NN, NB	Achieved AUC of 0.716 with NN	Yes
(Punia & Shankar, 2022)	2022	Demand Forecasting	RF, LSTM, ARIMA-NN, LSTM-RF	Used PCA which improved forecast having RMSE 0.4638 and without PCA 0.4782 for LSTM-RF	Yes
(Joseph et al., 2022)	2022	Store Item Demand Forecasting	SGD, LR, KNN, RF, SVR, XGB,	Achieved R^2 of 0.847 with CNN-BiLSTM	Yes

			CNN-BiLSTM		
(Majed, 2022)	2022	Short-Term Load Forecasting	DNN, ANN, DT	Achieved R^2 of 0.985 with DNN	No
(Raju et al., 2022)	2022	Steel Demand Forecasting	ELM+GBR+XGBR-SVR, ELM+GBR+XGBR-LASSO	Got R^2 of 0.977 for both the models with hyperparameter tuning	Yes
(Saha et al., 2022)	2022	Retail Demand Forecasting	LGBM, LSTM	LGBM performed better than LSTM with RMSE of 4436 whereas it was 5024 for LSTM	Yes
(Nasseri et al., 2023)	2023	Retail Demand Forecasting	ETR, RFR, XGBR, GBR, LSTM	Achieved R^2 of 0.60 with ETR	Yes
(Ramos et al., 2023)	2023	Energy Consumption Forecasting	RNN, LSTM, GRU, XGBR	Forecasted for 1min, 15min and 1hr with XGBR having MAPE of 0.0313 for 15min	Yes
(Duan & Dong, 2024)	2024	Home Appliance Demand Forecasting	LSTM-RF-XGBoost	Achieved R^2 of 0.9116 with combination	Yes
(Sukolkit et al., 2024)	2024	Steel Demand Forecasting	RNN-LSTM, CNN-LSTM, GRU-LSTM	MAPE of 1.2 was achieved with RNN-LSTM	Yes

(Manandhar et al., 2024)	2024	Load Forecasting	Prophet, LSTM, RF	LSTM performed better with minimum MAPE of 1.59%	Yes
(J. Wang et al., 2024)	2024	Retail Demand Forecasting	ARIMA, LSTM, GDBT, XGBoost, ST-GBT	ST-GBT performed better than other models with best RMSE of 1.231	Yes

2.8 Current Challenges and Future Directions

Despite significant advancements in retail demand forecasting methodologies, several challenges persist that limit the effectiveness of current approaches and highlight opportunities for future research. The complex nature of retail demand forecasting is still an issue, and several studies are being conducted to formulate robust forecast.

Data quality and availability remain fundamental challenges for retail forecasting systems. Many retailers struggle with inconsistent data collection practices, missing values, and varying granularity across different data sources. (Saha et al., 2022) noted that data imbalance across retail channels and the lack of cross-channel data integration posed significant challenges for deep learning frameworks in multinational retail environments. Future research directions include developing more robust preprocessing techniques and imputation methods specifically designed for retail time series data, as well as creating frameworks for seamless integration of data across physical and digital retail channels. Scalability presents another significant challenge, particularly for retailers with thousands of products across hundreds of locations. The computational resources required for training sophisticated models at this scale can be prohibitive, limiting the practical application of advanced methodologies. (Nasseri et al., 2023) identified computational costs as a key limitation in their comparison of tree-based ensembles and deep learning approaches for retail demand prediction. Future developments will likely focus on more efficient algorithmic implementations, distributed computing approaches, and automated model selection techniques that balance accuracy against computational requirements.

The integration of diverse datasets beyond traditional sales history represents both a challenge and an opportunity. (Haque et al., 2023) highlighted the value of incorporating macroeconomic indicators into retail forecasting models but noted the difficulties in establishing reliable relationships between these external factors and product-specific demand patterns. Future research will explore more sophisticated approaches for feature selection and fusion when working with heterogeneous data sources, potentially leveraging techniques from transfer learning and multi-task learning to enhance model performance with external variables.

The explainability and interpretability of forecasting models have gained increasing importance as retailers seek to build trust in automated prediction systems and extract actionable insights from model outputs. The "black box" nature of many advanced forecasting approaches, particularly deep learning models, limits their adoption in business contexts where transparency is valued. Research directions in this area include developing more interpretable architectures, employing post-hoc explanation techniques such as SHAP (SHapley Additive exPlanations) values, and creating visualization tools that communicate forecast rationale to business stakeholders in accessible formats. Dynamic market conditions and evolving consumer preferences challenge the stability of forecasting models. Traditional approaches often assume relatively static relationships between predictors and demand, an assumption increasingly challenged by rapidly changing retail environments. (J. Wang et al., 2024) addressed this issue by developing spatial-temporal gradient boosting methods that could adapt to shifting geographical and temporal patterns in retail demand. Future methodologies will likely incorporate more sophisticated approaches for concept drift detection and adaptation, enabling models to maintain accuracy despite changing underlying relationships.

The balance between forecast accuracy and operational feasibility represents another ongoing challenge. While highly complex models may achieve superior mathematical accuracy, they may prove difficult to implement and maintain in practical retail contexts. (Manandhar et al., 2024) emphasized the importance of selecting appropriate forecast horizons and granularity levels that align with actual business decision processes. Future research will increasingly consider the operational context of forecasting systems, developing methodologies that optimize for decision quality rather than purely statistical metrics. The handling of special events, promotions, and anomalous periods continues to challenge retail forecasting systems. (Huber & Stuckenschmidt, 2020) demonstrated that calendar-based special days significantly impact retail demand patterns, requiring specialized modeling approaches. Future developments will likely include more sophisticated methods for detecting and incorporating the effects of promotional activities, holidays, and other exceptional circumstances, potentially leveraging techniques from anomaly detection and causal inference to isolate and model these effects more accurately.

Multi-level forecasting represents an emerging challenge as retailers seek to generate consistent predictions across different hierarchical levels (e.g., individual products, categories, departments, and total store sales). (Alon et al., 2001) highlighted the importance of maintaining coherence between forecasts at different aggregation levels for effective retail planning. Future research directions include developing more sophisticated hierarchical forecasting frameworks that maintain consistency while optimizing accuracy at each level, potentially incorporating advances in reconciliation techniques and probabilistic graphical models.

The integration of forecasting with optimization and decision-making systems represents a promising frontier for retail applications. Beyond generating accurate predictions, retailers increasingly seek to translate forecast insights into optimal decisions regarding inventory levels, pricing strategies, and promotional planning. Research in this direction will explore closed-loop systems that combine prediction with prescription, potentially incorporating techniques from reinforcement learning and stochastic optimization to maximize business outcomes rather than merely forecast accuracy. Automated machine learning (AutoML) for retail forecasting represents another emerging direction that addresses the challenge of model selection and hyperparameter tuning at scale. As the number of potential models and configurations grows, manual optimization becomes increasingly impractical. Future developments will likely include specialized AutoML frameworks for time series forecasting in retail contexts, enabling more efficient discovery of optimal model architectures and configurations across thousands of product-location combinations.

Cross-learning between products and locations offers untapped potential for improving forecast accuracy, particularly for new products or stores with limited historical data. Techniques from transfer learning and meta-learning could enable retailers to leverage patterns identified in data-rich contexts to enhance predictions in data-sparse situations. (Ma & Fildes, 2021) explored this direction through meta-learning approaches for retail sales forecasting, demonstrating how knowledge transfer between different forecasting tasks could improve overall system performance.

CHAPTER 3

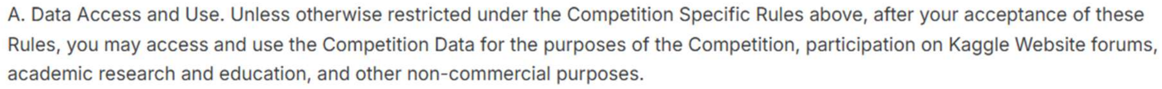
METHODOLOGY

3.1 Introduction

The main aim of this research is to compare the stacked ensemble tree-based machine learning model with deep learning model ANN along with validation that ensemble model performs better than the individual models. The research includes several steps such as data collection from Kaggle competition, preprocessing it, conducting statistical analysis and incorporate feature engineering, developing the machine learning models and evaluating the performance of it. The more detailed explanation of methodology used in this research is given below.

3.2 Data Collection and Preprocessing

This research uses time series data from Kaggle competition "Store Item Demand Forecasting Challenge" having daily demand from 2013 to 2017. The main aim of this competition data is to develop a robust forecasting model which can have forecast value the same as the actual value of the sales. The general rule of the competition is considered so that the data can be used for research purposes which is shown in Figure 3.1.



A. Data Access and Use. Unless otherwise restricted under the Competition Specific Rules above, after your acceptance of these Rules, you may access and use the Competition Data for the purposes of the Competition, participation on Kaggle Website forums, academic research and education, and other non-commercial purposes.

Figure 3.1 : Data accessibility for academic research

Only four parameters are recorded in the forecasting challenge dataset which are given below:

- date - Date of the sale data. There are no holiday effects or store closures.
- store - Store ID
- item - Item ID
- sales - Number of items sold at a particular store on a particular date.

To manage computational complexity while still ensuring valid comparisons, the research focuses on a subset of the full dataset-specifically two stores and five product categories. Due to this it helps to balance computational efficiency and sufficient stores and items are considered for forecasting purposes. The research contains different steps in the data preprocessing which are as follows:

a) Data Importation: The dataset is imported into the Python environment using the Pandas library, which facilitates subsequent manipulation and analysis.
b) Null Value Identification: The dataset is checked for any missing value which can impact the performance of the model if present and will decrease forecast accuracy.
c) Date Formatting: The "date" column is converted to date-time format which is required for forecasting as it helps to extract required information from the data like day of week, month, day, month, month, year, etc.

The preprocessing process used in this research aligns with the most research conducted in retail forecasting. In the demand pattern, the nature of demand may recur at regular interval so proper conversion of date to date-time format is necessary. Also, proper handling of null values prevent any error occurring in them model training and helps to improve its accuracy by enabling them to learn proper pattern from the data.

3.3 Statistical Analysis Using ANOVA

A distinguishing feature of the methodology is the integration of statistical analysis prior to model development. This approach allows for empirical validation of underlying assumptions about factors affecting sales performance, which subsequently informs feature engineering decisions.

3.3.1 Two-Way ANOVA Implementation

The research employs a Two-Way Analysis of Variance (ANOVA) using SPSS to determine the impact of stores and items on sales patterns. This statistical technique evaluates how two categorical independent variables (store and item) affect a continuous dependent variable (sales), both individually and in interaction with each other. It is done because the sales may vary from store to store between the items. Suppose the item which is selling better in one store may not have same sales in another store.

ANOVA is carried out in the following steps in SPSS:

1. Importing the dataset into the SPSS environment
2. Go to Analyze>General Linear Model>Univariate
3. Configuring the Univariate Analysis of Variance (UNIANOVA) with the following:
4. Sales as the dependent variable
    a. Store and item as fixed factors
    b. Store × item as the interaction term
    c. Significance level set at 0.05
    d. Setting up Univariate options
5. Run the model

The two-way ANOVA is appropriate when analyzing the effects of two independent categorical variables (such as gender and college major) – both alone and in combination with each other – on a continuous dependent variable (such as an exam score). In this research context, the approach allows for systematic assessment of how store and item variables independently and jointly influence sales patterns. If the p-value is less than this cutoff value then it is considered statistically significant at a significance level of 0.05.

3.3.2 Levene's Test and Post Hoc Analysis

The methodology incorporates Levene's Test of Equality of Error Variances to assess whether the assumption of equal variances across groups is met. This is critical for validating the robustness of ANOVA results. Levene's Test is used to determine whether two or more groups have equal variances. It is widely used because many statistical tests use the assumption that groups have equal variances. Additionally, Post Hoc testing is employed to identify specific differences between groups when the ANOVA indicates significant overall effects. Post Hoc testing will help to determine what type of data structure is available and helps to formulate the model of machine learning according to that.

The results of the ANOVA analysis directly inform subsequent methodological decisions, particularly in feature engineering. With approximately 71.3% of sales variability explained by the model, the statistical findings justify the development of store-specific and item-specific features to capture the differential patterns identified.

3.4 Feature Engineering for Time Series Forecasting

The process of feature engineering is a very sophisticated part of the research design. Rather than employing pure raw sales figures, the process employs the understanding of retail demand patterns to construct predictive features with temporal dependencies and cross-sectional relationships.

3.4.1 Lag Features Implementation

Lag features are constructed by shifting current sales history through several ranges (1 day, 7 days, 30 days, and 365 days) so that the models can learn from past sales trends. This follows common conventions in the time-series forecasts, where the past is a good predictor of the future.

Lagged features is a feature engineering method used to extract temporal dependencies and patterns in time series data. It is built by taking the value of a variable at an earlier time period and using it as a feature in the model of the current time. The use of differing lag intervals—daily, weekly, monthly, and yearly—is methodologically sound, as it captures different cyclical patterns in retailed sales.

Lag features are extremely important to time series analysis, exploring their importance, creation process, and practical application towards prediction. The technique leverages this concept to enable models to use past values to predict future outcomes.

3.4.2 Rolling Window Features

The approach is more than just lagging methods in the sense that it embeds rolling window statistics which best capture trends and variation in different temporal dimensions. Specifically, the approach employed here computes:

1. 7-day rolling mean and standard deviation (weekly patterns)

2. 30-day rolling mean and standard deviation (monthly trends)

3. 365-day rolling mean and standard deviation (annual trends)

This multi-window method solves the most important retail forecasting problem: separating short-term anomalies from significant longer-term trends. By estimating central tendency (mean) and dispersion (standard deviation) at various time scales, the model captures the level and volatility of sales patterns.

Cleaning of time series can be a multi-step procedure involving missing value handling, trend removal, removal of seasonality, stationarity testing, data normalization, outlier removal, and data smoothing. The rolling window method addresses some of these issues by assisting in trend detection and minimizing the effect of outliers.

3.4.3 Cumulative Sales Features

The approach combines cumulative sum characteristics to offer insight into the general trend of sales across time horizons. The approach values that absolute sales size contains valuable information beyond seasonality and short-term movement.

This approach enables models to learn long-term growth or decline trends that may be missed by features that learn short-term or cyclical trends.

3.4.4 Month-over-Month Sales Growth

One very novel component of the feature engineering process is computing month-over-month sales increases. This feature measures the change rate in sales, not absolute values, which can be particularly significant to spot increases or declines in patterns of demand.

Month-over-Month Growth (M/M) is a way of expressing how fast a specific measure is changing from month to month, and it is expressed as a percentage of the original value. The calculation is a two-step procedure: First, we do the division of the figure in the current month by the figure in the previous month. Then, the value obtained in the first step is decreased by one.

This approach provides valuable information about sales momentum that complements absolute level features.

3.4.5 Interaction Features Development

The methodology recognizes that relationships between different factors can be as important as the factors themselves. Consequently, it incorporates several interaction features:

1. Store-Month Interaction: Captures store-specific seasonal patterns by combining store identifiers with monthly information
2. Weekday-Sales Interaction: Identifies how sales patterns vary across different days of the week

The development of these interaction features is methodologically justified by the ANOVA results, which indicated significant interaction effects between stores and items. Implementation involves:

3.5 Model Training Methodology

The model training methodology represents a systematic approach to comparing different machine learning techniques for retail demand forecasting. The research specifically contrasts two approaches: a stacked ensemble of tree-based models and an artificial neural network (ANN).

3.5.1 Stacked Ensemble Architecture

The stacked ensemble approach employed here is a combined model structure with the predictions produced by different groups of machine learning models being combined to improve prediction accuracy. This stacking ensemble approach contrasts with the traditional approaches of bagging or boosting, where different types of base learners are combined rather than repeating the same type of model. The basic notion is to capitalize on the strengths of each model and offset their weaknesses by finding the best way of combining their outputs using a meta-model.

In this research, the stacked models uses feature engineering such as lag, rolling, store and month interaction for modelling purposes. In the machine learning models temporal variables are incorporated in the dataset like day, month, day of the week, etc. along with engineered features such as sales_lag_7, sales_roll_mean_30, and store_month_interaction. With the use of these features in the dataset, the model can determine the seasonality, trends and store-item variations due to which the model is able to predict the sales with great accuracy.

In these stacked ensemble models, four tree-based regression models are used which are Random Forest Regressor, XGBoost Regressor, LightGBM, and CatBoost as base learners. Each base learners have unique features, each having ability to handle non-linear relationships, missing values and incorporate categorical variables in it which is combined in stacking to get better forecasting results. Random Forest uses the bagging method for variance reduction, whereas XGBoost and LightGBM use gradient boosting for bias reduction as well as learning efficiency optimization. Similarly, CatBoost is better at handling categorical data in the model because of which there is no requirement of converting categorical data into numerical like one hot encoding. All four types of learners are fed the same training data and make individual predictions for sale, and each utilize its ability to forecast differently. The result of each base learner is used for training the Gradient Boosting Regressor which is used as meta learner in the stacking model. It combines the predictions of all four base learners and making predictions to provide final forecasted sales as output. Gradient Boosting is an excellent option for the meta-model because of its ability to learn residual patterns that the previous models have been unable to pick up and improve forecasts by focusing on more difficult cases to predict. With passthrough=True, the stacking process also feeds the original feature inputs and predictions from the base models to the meta-model, augmenting its decision-making capability.

The model is trained on sales history up to September 2017, and predictions are generated from a given test set based on the chosen feature columns. The performance of the ensemble is measured against four standard metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R-squared (R^2). The metrics give a comprehensive view of the accuracy, error responsiveness, and overall fitness of the model. Lastly, actual vs. predicted sales are plotted and saved as graphs for each item-store pair to enable interpretability and monitoring of the performance of the model over time.

3.5.2 Artificial Neural Network Implementation

In contrast to the tree-based ensemble, the methodology implements an ANN as a representative deep learning approach. The network architecture and training parameters are carefully specified:

The Network Architecture of ANN model is given below:

1. Three hidden layers with 128, 64, and 32 neurons respectively
2. Hyperbolic tangent (tanh) activation function
3. L2 regularization with coefficient 0.001
4. Dropout rate of 0.2 between layers
5. Training Configuration:
    a. Maximum 500 epochs
    b. Early stopping with patience of 3 epochs
6. Feature-engineered dataset as input

Dropout is a technique for addressing this problem. The key idea is to randomly drop units (along with their connections) from the neural network during training. This prevents units from co-adapting too much."

The design decisions within the neural network structure are based on established methodological best practices with the intent to enhance performance and avoid overfitting. The use of a diminishing sequence of neurons within subsequent layers—beginning with 128 in the initial hidden layer and decreasing to 64, then to 32—is an established strategy referred to as progressive dimensionality reduction. This strategy assists the network to step-by-step extract intricate features and condenses them into more abstract forms, with the ultimate layers concentrating on the most precise patterns. Using the tanh activation function in the ANN model brings non-linearity, which helps it to learn complicated relationships between inputs and outputs. In contrast of functions such as ReLU, tanh sends inputs to a normalized interval between -1 and 1, which assists with gradient flow and model stability, particularly in deeper models.

To avoid the model becoming overly complicated and memorizing the training data, L2 regularization is used. The method discourages big weights by appending a penalty term to the loss function and keeping the weight value small while making the model less complicated. Apart from regularization, dropout is also introduced as a regularizer. At training time, dropout is used to randomly silence some of the neurons in every layer, causing the network to learn more redundant and stronger representations which perform better on unseen data. The randomness used in the neural networks helps to reduce overfitting.

Lastly, the model uses early stopping, which tracks the performance on a validation set while training. If the model's performance ceases to improve over a given number of iterations, training is automatically stopped. This prevents the model from fitting noise in the training data if it continues to be trained, and thus its performance on new data will ultimately be harmed. In combination, dropout and early stopping act as protection against overtraining and ensure that the network learns significant patterns and not noise or unimportant input features, resulting in a more generalizable and consistent model. The mathematical equation for a single neuron in ANN is expressed in Equation (1) and (2).

$$y = \frac{1}{1+ e^{-z}} \qquad (3.1)$$

$$z = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b \qquad (3.2)$$

Where,

Inputs = $x_1, x_2, ..., x_n$

Weights = $w_1, w_2, ..., w_n$

Bias = b

3.6 Model Evaluation Methodology

To determine the performance of the machine learning models in forecasting, mainly four key metrics are preferred by researchers which can provide view of overall performance of the model which are Root Mean Square Error (RMSE), Mean Square Error (MAE), Mean Absolute Percentage Error (MAPE) and Coefficient of Determination (R^2).

3.6.1 Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \qquad (3.3)$$

The Root Mean Square Error (RMSE) calculates the square root of the average squared difference between actual and anticipated values. In statistical forecasting, root mean square error is calculated by taking the average of the square of the difference between actual and forecast values and taking its square root. It is mainly used in financial forecasting or high-risk inventory planning to see the deviation of the forecasted value with actual one.

3.6.2 Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \bar{y}_i| \qquad (3.4)$$

It is another key metric which is used in forecasting. Mean Absolute Error (MAE) is the average of absolute difference of actual and the forecasted result values. It is mostly used in regression as it is easy to interpret and helps us to know how close the forecasted value are to the actual. As it is absolute, it is always positive due to which does not we cannot know whether the forecast result is overestimated or underestimated.

3.6.3 Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n} \left|\frac{y_i - \bar{y}_i}{y_i}\right| \qquad (3.5)$$

Mean Absolute Percentage Error (MAPE) is the metrics used to determine the performance of forecasting model which calculates the average percentage of absolute predictions from actual values. In simple terms, it can be considered as an accuracy of the forecasting model with 100% denoting as the perfect forecasting result in which all actual and forecasted values are the same. It is easy to understand and also shows the percentage deviation from the actual value but if it is very close to 0 then overfitting of data can occur.

3.6.4 Coefficient of Determination (R²)

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\bar{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \qquad (3.6)$$

The Coefficient of Determination (R²) is used to measure the variance in the output which is forecasting result due to independent variable. It essentially compares the model's performance to a benchmark model, which forecasts the average of the observed values. A high R² value implies a strong fit, as it explains a significant part of the variability in the result variable. R² should be read properly, especially in non-linear or complex models, as it may not always provide a full understanding of the result.

The choice of the metrics used for demand forecasting in this research is based on the commonly used key metrics by many researchers. Using this metrics, it is easy to see the actual performance of the model used in the research

CHAPTER 4

RESULTS AND DISCUSSIONS

This chapter discusses the result of the research conducted on retail demand forecasting. Initially, ANOVA is performed using SPSS to determine the correlation between the store and the items on sale and study its effect. The result of ANOVA provides insight into how sales vary by store and item and explains the variability in sales. Features such as lagging, rolling, cumulative sum, month-over month sales, and weekday sales interaction are used to improve the model's performance. The stacking of RF+XGBR+LGBM+CatBoost-GBR is compared with the deep learning model ANN based on the model evaluation metrics. Also, the two best performing models of the stacking is compared with ensemble model itself to validate the result conducted by any researchers that ensemble model performs better than individual models. For that the performance is validated based on key performance metrics, mainly R^2.

4.1 ANOVA Result

This research uses Univariate Analysis of two-way Analysis of Variance (ANOVA) to find the relation between store and item, evaluating the sales performance in each store-item combination. The analysis helps to understand the underlying data structure before applying feature engineering in machine learning models. According to the result of ANOVA, the selected model explains approximately 71.3% of the total variability in sales, indicating a strong relationship between the independent variables (store and item) and the dependent variable (sales). The results support for the employment of feature engineering in the model. The descriptive statistics from the ANOVA analysis indicated a mean sales value of 34.52

with a standard deviation of 20.901 across the sample size of 18,260 observations, offering important context about the central tendency and dispersion of the retail sales data being modeled.

Table 4.1 : ANOVA result Summary

Source	**Type III Sum of Squares**	**df**	**Mean Square**	**F**	**Partial Eta Squared**
Corrected Model	5685129.29[a]	9	631681.033	5030.521	0.713
Intercept	21753565.42	1	21753565.42	173238.949	0.905
Store	643418.711	1	643418.711	5123.996	0.219
Item	4895191.149	4	1223797.787	9745.963	0.681
Store * Item	146519.433	4	36629.858	291.709	0.060

Levene's test for equality of error variances yielded a highly significant result (p < 0.000), conclusively demonstrating that the variances across different stores and items were not equal. This heterogeneity of variance is a common characteristic in retail data due to the inherent differences in product popularity, pricing strategies, and store-specific factors affecting sales patterns. The ANOVA results are summarized in Table 2, which presents the Type III Sum of Squares calculation – a method particularly appropriate for unbalanced models with no empty cells. The Type III methodology adjusts each effect in the model for all other "appropriate" effects, making it suitable for analyzing complex retail datasets where multiple factors interact simultaneously. The ANOVA results demonstrated strong statistical significance for both main effects: store (F = 5123.996, p < 0.001) and items (F = 9745.963, p < 0.001), confirming that both factors significantly impact sales variability. The Partial Eta Squared values, which measure effect size, revealed that the item factor explained 68.1% of the variance in sales (after accounting for other variables), while the store factor explained 21.9%. These substantial effect sizes underscore the importance of both store-specific and product-specific characteristics in retail demand patterns. The Partial Eta Squared values surpass Cohen's traditional threshold of 0.14 for large effects, indicating exceptionally strong relationships between these factors and sales outcomes.

Moreover, the interaction effect between store and item was also statistically significant (F = 291.709, $p < 0.001$), with a Partial Eta Squared value of 0.060. This interaction effect, while smaller than the main effects, still represents a medium effect size according to Cohen's guidelines and demonstrates that the impact of item selection on sales varies depending on the specific store, and vice versa. This interaction justified the use of store-item combination features in the subsequent machine learning models. The significant interaction term indicates that the effect of one independent variable (store or item) on sales depends on the level of the other independent variable, suggesting a complex relationship that simple additive models might fail to capture adequately. The ANOVA findings provided crucial guidance for the feature engineering process, highlighting the need to account for both store-specific and item-specific characteristics, as well as their interactions, in the forecasting models. As per findings, feature engineering was implemented in the model such as lag features, rolling windows, cumulative sales metrics, and interaction terms so that it can capture complex patterns in the data to make forecasting. The statistical analysis of ANOVA helps complicated machine learning modes to capture and predict the complex pattern in the retail sales data.

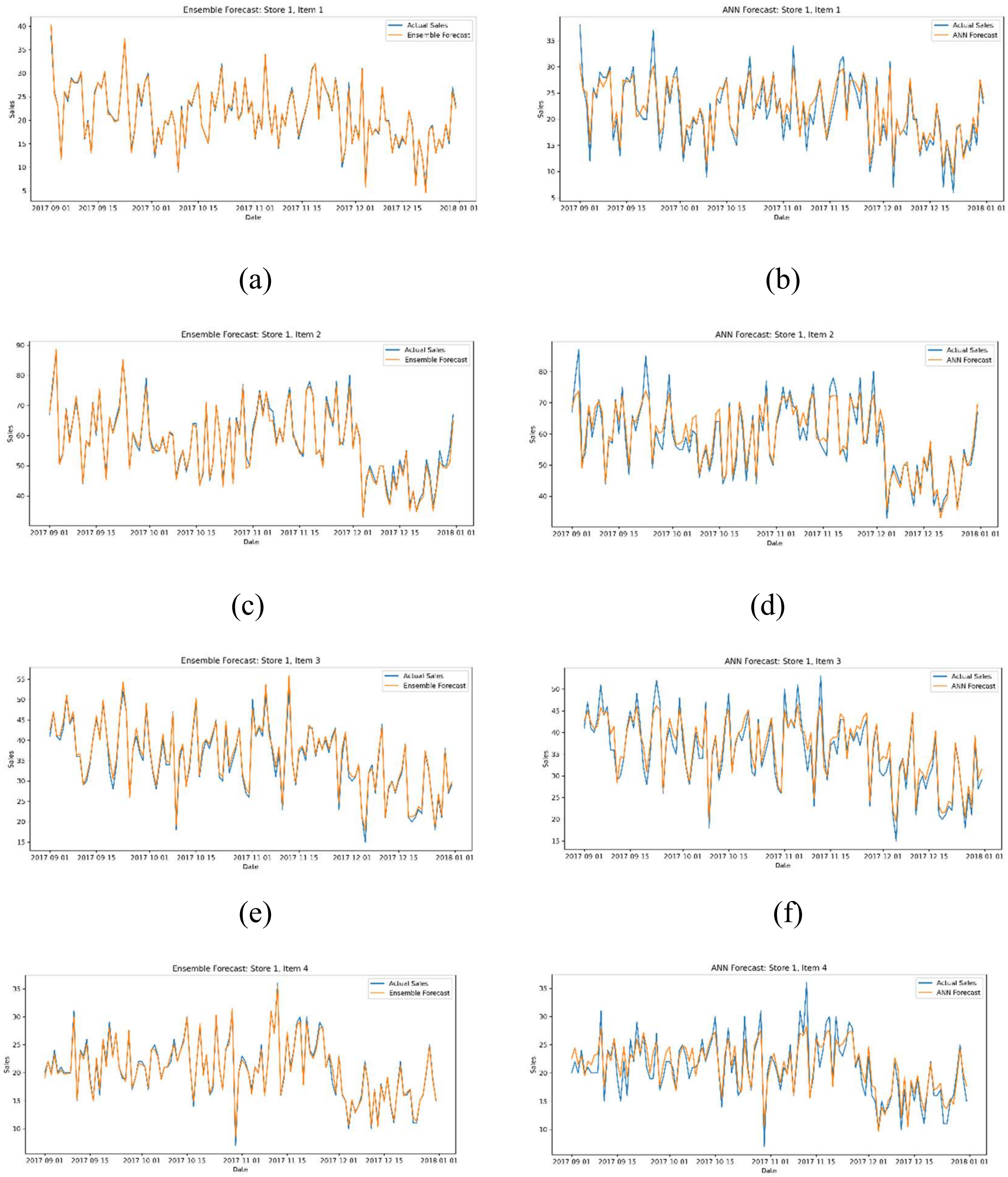
4.2 Comparison between Stacking and ANN

The results of comparison between the stacking ensemble and the ANN model reveals the performance between the store-item combinations in the dataset based on the key performance metrics. The performance of stacking ensemble of tree-based models is way better compared to the ANN model, and also the performance gap varies across different store-item combinations but remains substantial in all combinations. The result shows the strong superiority of the stacking of tree-based approach for retail demand forecasting. Table 3 indicates the performance of stacking ensemble across all store-item combinations which were able to achieve R^2 ranging from 0.982 to 0.994. This result signifies that the stacking model of this research could explain variance between 98.2% and 99.4% in the sales data and shows the exceptional predictive accuracy. But the ANN model performance was very less in comparison with stacking model having values of R² between 0.769 to 0.924. There is very much fluctuation in the performance of ANN model across the combinations which showed poor performance in some cases such as in Store 2 and Item 1 combination, R^2 was only 0.769.

Table 4.2 : Comparison between stacking model and ANN

Store	Item	Model	RMSE	MAE	MAPE	R^2
1	1	Stacking	0.483	0.351	2.012	0.994
		ANN	1.997	1.504	8.442	0.896
	2	Stacking	1.382	1.022	1.783	0.985
		ANN	3.152	2.447	4.123	0.923
	3	Stacking	1.108	0.881	2.662	0.982
		ANN	2.428	1.964	5.848	0.911
	4	Stacking	0.560	0.410	2.184	0.988
		ANN	2.013	1.585	8.242	0.851
	5	Stacking	0.611	0.410	2.609	0.988
		ANN	2.469	1.668	10.027	0.798
2	1	Stacking	1.055	0.673	2.684	0.984
		ANN	3.963	3.043	11.027	0.769
	2	Stacking	1.823	1.424	1.934	0.989
		ANN	5.674	3.149	3.566	0.891
	3	Stacking	1.327	0.967	2.015	0.986
		ANN	3.648	2.585	4.876	0.893
	4	Stacking	0.709	0.509	1.667	0.991
		ANN	2.130	1.410	4.348	0.915
	5	Stacking	0.620	0.475	1.904	0.991
		ANN	1.780	1.103	4.371	0.924

The root mean square error (RMSE) values further underscore the performance gap between the two approaches. The stacking ensemble achieved RMSE values between 0.483 and 1.823 across different store-item combinations, while the ANN model's RMSE values ranged from 1.780 to 5.674. Similarly, for mean absolute error (MAE), the stacking ensemble values ranged from 0.351 to 1.424, compared to 1.103 to 3.149 for the ANN model. These error metrics consistently show that the stacking ensemble's predictions were much closer to the actual sales values than the ANN predictions. The mean absolute percentage error (MAPE) values provide perhaps the most interpretable comparison of forecast accuracy. The stacking ensemble achieved MAPE values between 1.667% and 2.684% across all store-item combinations, indicating that on average, its predictions deviated from actual sales by less than 3%. In stark contrast, the ANN model's MAPE values ranged from 3.566% to 11.027%, with several combinations showing MAPE values above 8%. This means that in some cases, the ANN predictions were off by more than 11% on average, making them significantly less reliable for inventory planning and supply chain management decisions.



Ensemble Forecast: Store 1, Item 1

ANN Forecast: Store 1, Item 1

Ensemble Forecast: Store 1, Item 2

ANN Forecast: Store 1, Item 2

Ensemble Forecast: Store 1, Item 3

ANN Forecast: Store 1, Item 3

Ensemble Forecast: Store 1, Item 4

ANN Forecast: Store 1, Item 4

(a)

(b)

(c)

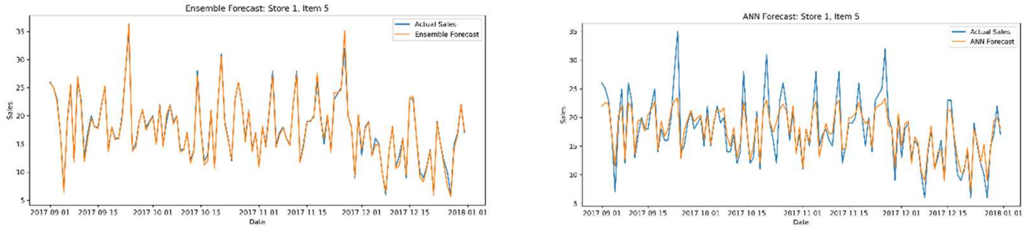
(d)

(e)

(f)

(g)

(h)

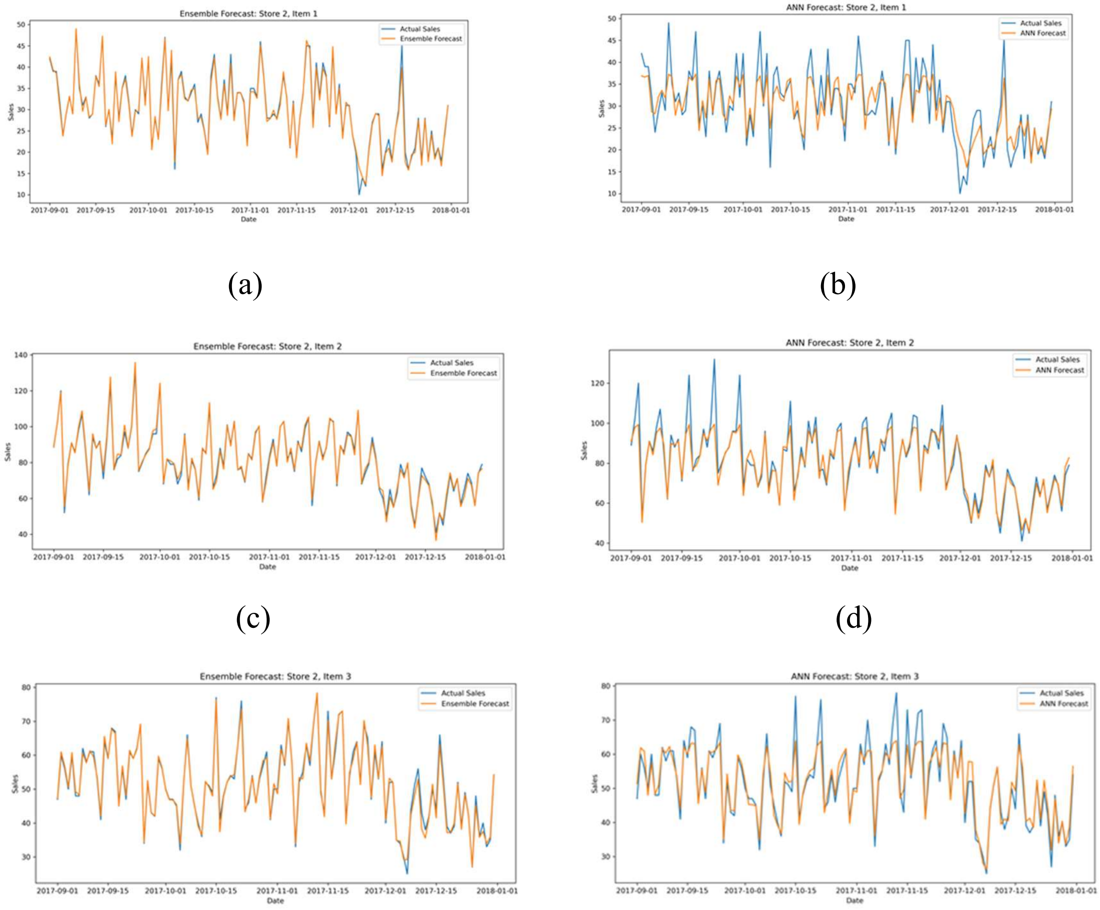


(i) (j)

Figure 4.1: Daily forecast over 3 months: (a) Stacking result of tree-based model forecast for store 1 and item 1. (b) ANN forecast result for store 1 and item 1. (c) Stacking result of tree-based model forecast for store 1 and item 2. (d) ANN forecast result for store 1 and item 2 (e) Stacking result of tree-based model forecast for store 1 and item 3. (f) ANN forecast result for store 1 and item 3. (g) Stacking result of tree-based model forecast for store 1 and item 4. (h) ANN forecast result for store 1 and item 4. (i) Stacking result of tree-based model forecast for store 1 and item 5. (j) ANN forecast result for store 1 and item 5.

A particularly notable aspect of the stacking ensemble's performance is its consistency across different store-item combinations with varying levels of sales volatility. The researchers noted that the standard deviation in the sales data varied from 5.6 to 20.3 between store-item combinations, indicating substantial differences in sales volatility. Despite these variations, the stacking ensemble maintained high performance across all combinations, demonstrating its robustness to data variability. The ANN model, however, showed greater sensitivity to these variations, with its performance deteriorating more notably for store-item combinations with higher volatility. The visual comparisons presented in Figures 2 and 3 further illustrate the performance difference between the two approaches. The line charts comparing forecast versus actual sales show that the stacking ensemble's predictions closely follow the actual sales trend with minimal deviation, while the ANN forecasts exhibit larger discrepancies and greater variance from actual sales, particularly during peaks and troughs. This visual evidence supports the quantitative metrics in demonstrating the stacking ensemble's superior ability to track sales patterns accurately.

This research highlights the stacking ensemble's superior performance due to several factors. First, the tree-based models that form the base learners in the ensemble are particularly good at capturing non-linear relationships and complex interactions between features, which are common in retail sales data. Second, the stacking models are good at combining strengths of different base models and using meta-learner to learn and combine their predictions. Finally, the feature engineering helped to identify the hidden patterns from them and helped to enhance the accuracy of the model. The weaker performance of the ANN model was also due to many factors. Neural networks are more sensitive when it comes to hyperparameter choices and initialization conditions due to which the models fail to perform well without extensive tuning. Also, the deep learning models need larger dataset so it can learn complex patterns effectively. When dealing with high-dimensional feature spaces, the volume of data should be huge. Neural networks have "black box" nature which makes it difficult to diagnose and address specific weaknesses in their predictions.



(a)

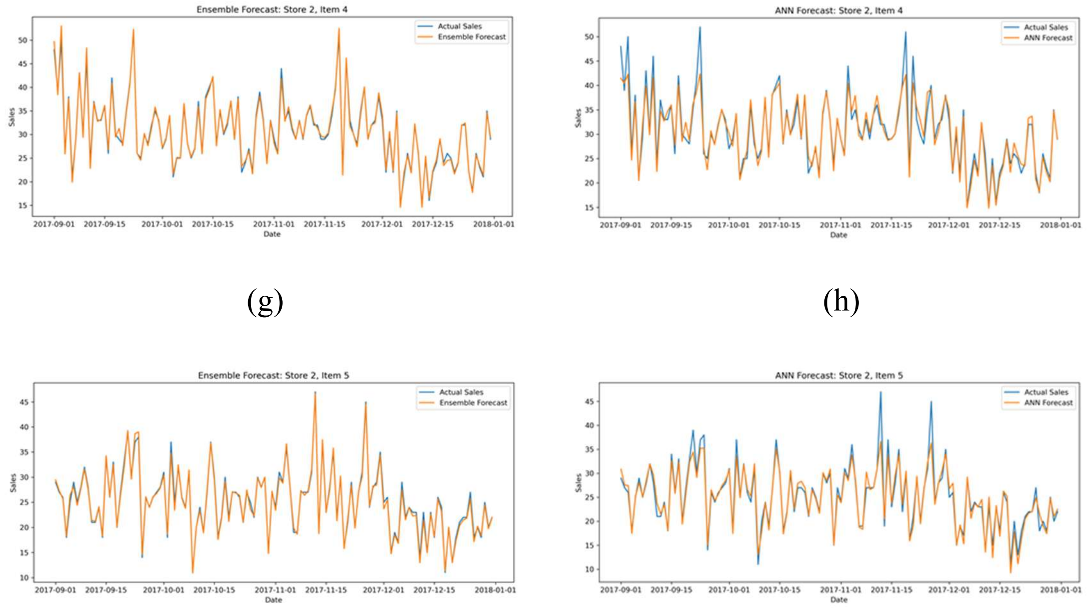
(b)

(c)

(d)

(e)

(f)



(g) (h)

(i) (j)

Figure 4.2: Daily forecast over 3 months: (a) Stacking result of tree-based model forecast for store 2 and item 1. (b) ANN forecast result for store 2 and item 1. (c) Stacking result of tree-based model forecast for store 2 and item 2. (d) ANN forecast result for store 2 and item 2 (e) Stacking result of tree-based model forecast for store 2 and item 3. (f) ANN forecast result for store 2 and item 3. (g) Stacking result of tree-based model forecast for store 2 and item 4. (h) ANN forecast result for store 2 and item 4. (i) Stacking result of tree-based model forecast for store 2 and item 5. (j) ANN forecast result for store 2 and item 5.

The comparison of performance between the stacking ensemble and the ANN model in this research shows the advantages of tree-based ensemble methods in retail demand forecasting. The stacking ensemble shows consistent performance across multiple evaluation metrics and store-item combinations surpassing ANN and demonstrates its practical value for retail inventory management and supply chain optimization applications.

4.3 Comparison between CatBoost and XGBR

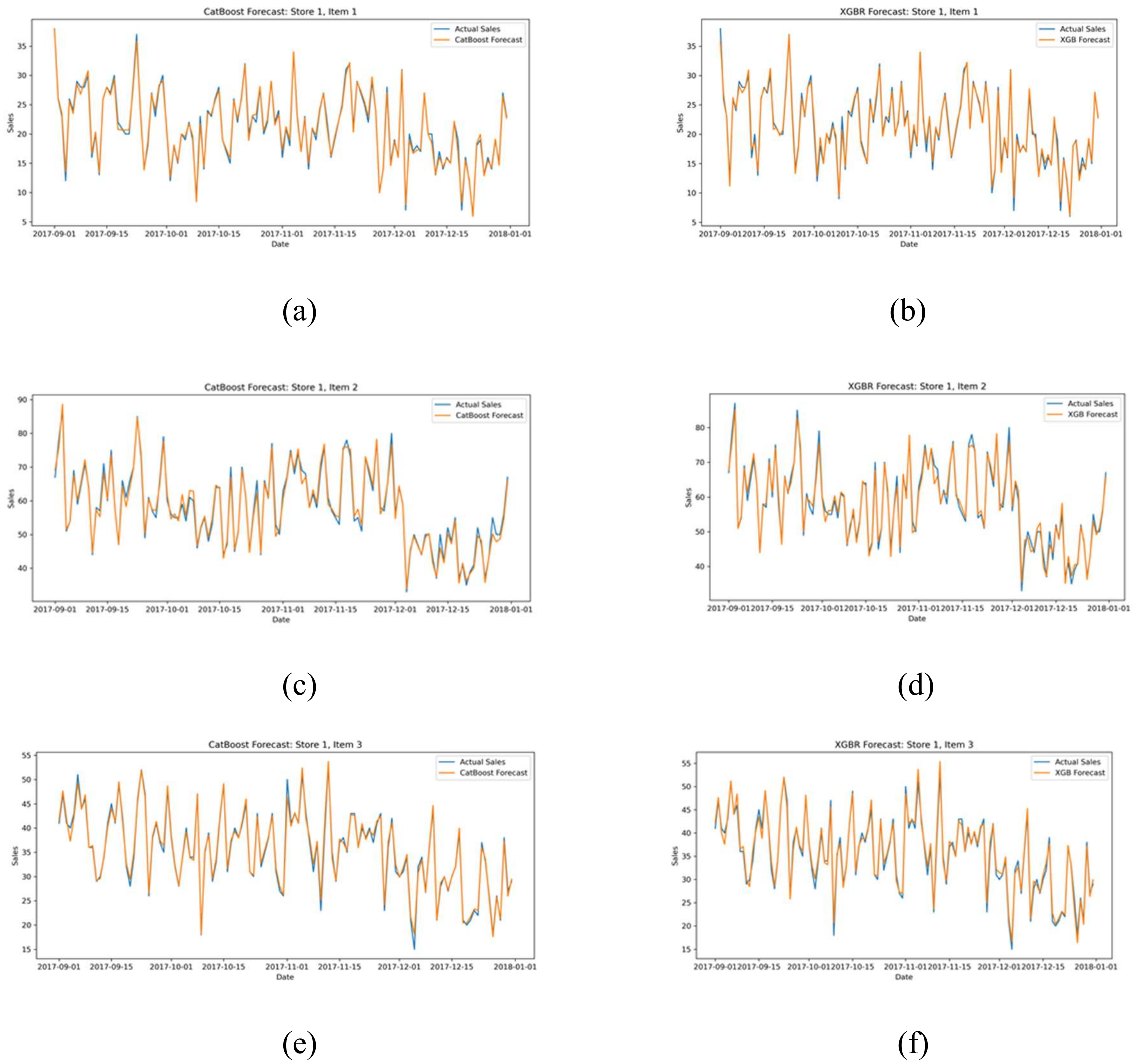
Apart from comparison of the stacking ensemble and ANN model, this research also evaluates CatBoost and XGBoost Regression (XGBR), which are two key tree-based algorithms used in the stacking process. This evaluation helps to understand the contribution of contributions of these individual models to the stacking ensemble's performance and supports the research that ensemble model performance is superior compared to individual models. This research also shows which individual models have superior performance for retail demand forecasting. The choice of comparison of CatBoost and XGBR is considered based on the superior performance of the models compared to other base learners.

Table 4.3 : Comparison between CatBoost and XGBR

Store	Item	Model	RMSE	MAE	MAPE	R^2
1	1	CatBoost	0.629	0.488	2.621	0.990
		XGBR	0.758	0.582	3.252	0.985
	2	CatBoost	1.560	1.245	2.157	0.981
		XGBR	1.658	1.330	2.373	0.979
	3	CatBoost	0.873	0.660	2.010	0.989
		XGBR	1.117	0.905	2.780	0.981
	4	CatBoost	0.598	0.442	2.324	0.987
		XGBR	0.722	0.524	2.822	0.981
	5	CatBoost	0.410	0.316	2.002	0.994
		XGBR	0.591	0.438	2.825	0.988
2	1	CatBoost	0.985	0.617	2.569	0.986
		XGBR	1.201	0.761	3.162	0.979
	2	CatBoost	2.255	1.737	2.290	0.983
		XGBR	2.597	1.998	2.745	0.977
	3	CatBoost	1.450	1.179	2.410	0.983
		XGBR	1.567	1.108	2.249	0.980
	4	CatBoost	0.700	0.523	1.694	0.991
		XGBR	0.937	0.697	2.304	0.984
	5	CatBoost	0.677	0.501	1.969	0.989
		XGBR	0.808	0.631	2.568	0.984

In this research, the optimal hyperparameters are identified for both CatBoost and XGBR by using grid search optimization, which ensures a fair comparison between the two algorithms. The hyperparameter tuning is essential for maximizing the performance of machine learning models which helps to obtain reliable comparative results. With the use of grid search technique, both the models performed consistently with R^2 above 0.97 across all store-item combinations. The high-level performance of both models makes it suitable for retail demand forecasting which were considered in the application before stacking. Table 4 shows that

CatBoost performs better than XGBR across all evaluation metrics for both stores and all items. For Store 1, CatBoost achieved R² values ranging from 0.981 to 0.994, while XGBR's R² values ranged from 0.979 to 0.988. Similarly, for Store 2, CatBoost's R² values ranged from 0.983 to 0.991, compared to 0.977 to 0.984 for XGBR. While the performance difference is relatively small in terms of R² values, CatBoost consistently achieved lower error metrics (RMSE, MAE, and MAPE) across all store-item combinations, indicating more accurate predictions.



(a)

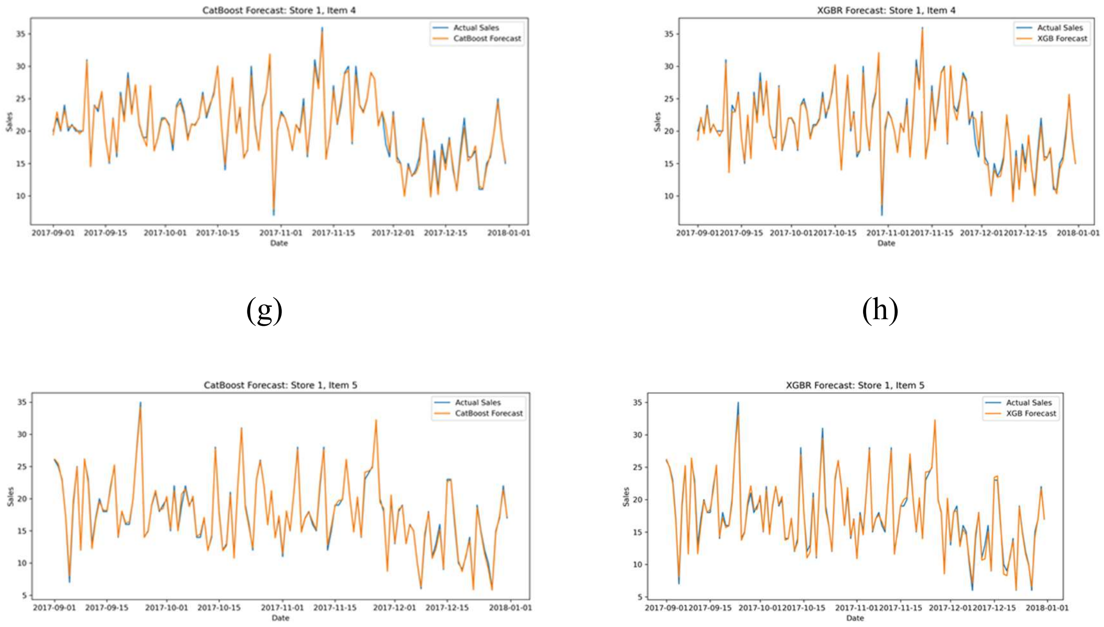
(b)

(c)

(d)

(e)

(f)



CatBoost Forecast: Store 1, Item 4

XGBR Forecast: Store 1, Item 4

CatBoost Forecast: Store 1, Item 5

XGBR Forecast: Store 1, Item 5

(g)

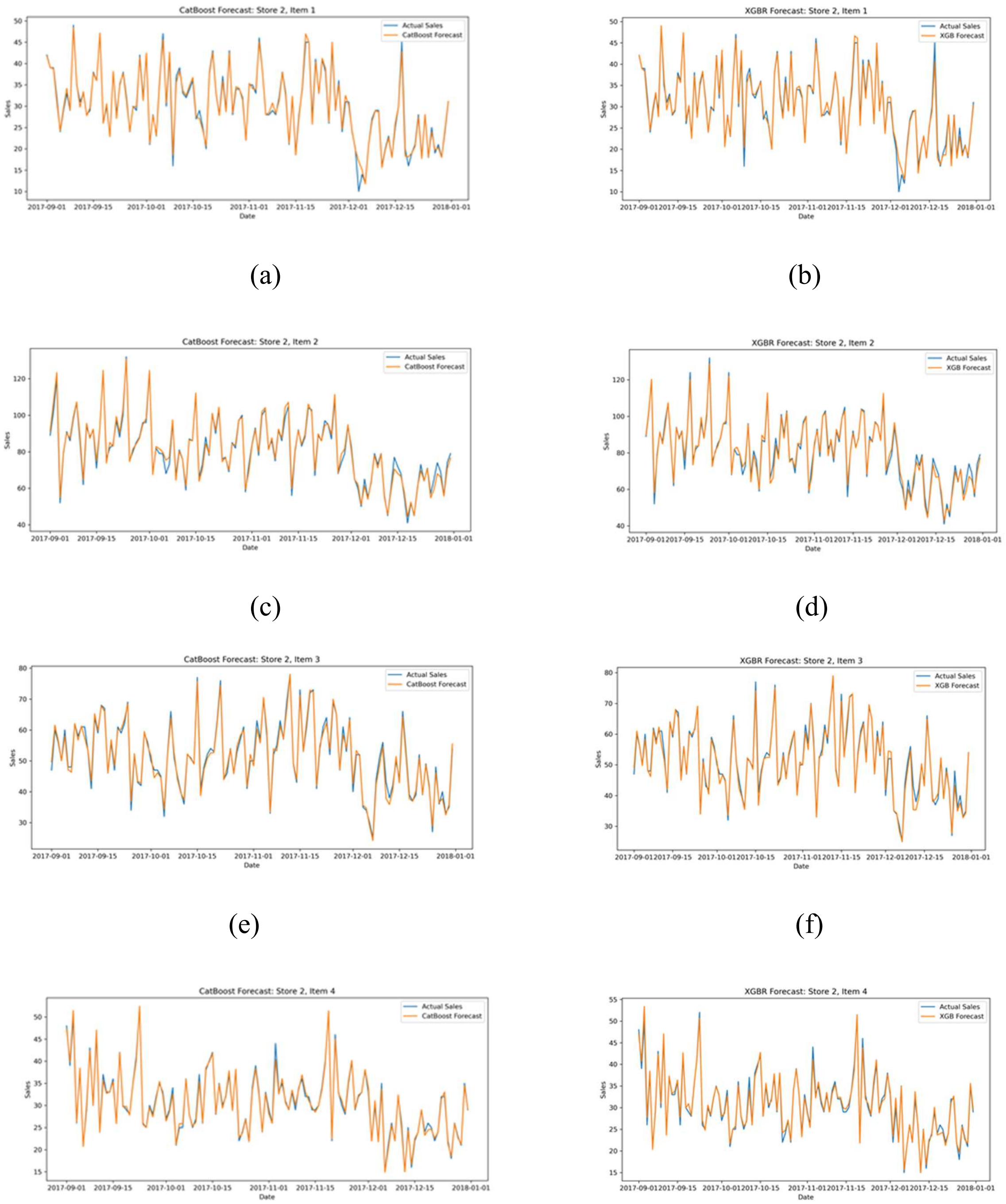
(h)

(i)

(j)

Figure 4.3: Daily forecast over 3 months: (a) CatBoost forecast result for store 1 and item 1. (b) XGBR forecast result for store 1 and item 1. (c) CatBoost forecast result for store 1 and item 2. (d) XGBR forecast result for store 1 and item 2. (e) CatBoost forecast result for store 1 and item 3. (f) XGBR forecast result for store 1 and item 3. (g) CatBoost forecast result for store 1 and item 4. (h) XGBR forecast result for store 1 and item 4. (i) CatBoost forecast result for store 1 and item 5. (j) XGBR forecast result for store 1 and item 5.

The RMSE values for CatBoost ranged from 0.410 to 2.255 across all store-item combinations, compared to 0.591 to 2.597 for XGBR. For MAE, CatBoost values ranged from 0.316 to 1.737, while XGBR ranged from 0.438 to 1.998. The MAPE values, which represent the percentage deviation from actual sales, ranged from 1.694% to 2.621% for CatBoost and from 2.249% to 3.252% for XGBR. These consistent differences across all metrics and store-item combinations demonstrate CatBoost's superior performance compared to XGBR in this retail forecasting context. The visual comparisons presented in Figures 4 and 5 further illustrate the performance difference between CatBoost and XGBR. The line charts show that CatBoost's predictions generally align more closely with actual sales patterns than XGBR's predictions, with smaller deviations during both normal periods and unusual fluctuations. This visual evidence corroborates the quantitative metrics in demonstrating CatBoost's advantage over XGBR for retail demand forecasting.



(a)

(b)

(c)

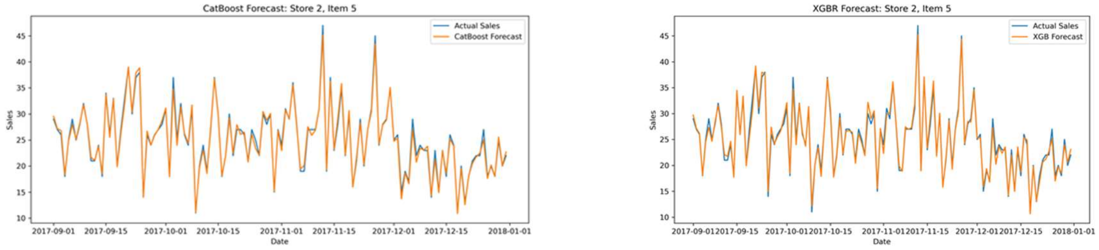
(d)

(e)

(f)

(g)

(h)



(i) (j)

Figure 4.4: Daily forecast over 3 months: (a) CatBoost forecast result for store 2 and item 1. (b) XGBR forecast result for store 2 and item 1. (c) CatBoost forecast result for store 2 and item 2. (d) XGBR forecast result for store 2 and item 2. (e) CatBoost forecast result for store 2 and item 3. (f) XGBR forecast result for store 2 and item 3. (g) CatBoost forecast result for store 2 and item 4. (h) XGBR forecast result for store 2 and item 4. (i) CatBoost forecast result for store 2 and item 5. (j) XGBR forecast result for store 2 and item 5.

Several factors may contribute to CatBoost's superior performance compared to XGBR. CatBoost employs an ordered boosting technique that reduces prediction shift caused by target leakage, which can be particularly beneficial for time-series data like retail sales. It also uses a novel approach for handling categorical features, which might have better captured the categorical aspects of the data (such as store and item identifiers). Additionally, CatBoost implements symmetric trees and a combination of oblivious and non-oblivious trees, which can improve its ability to model complex relationships in the data. Although the performance of CatBoost was superior compared to XGBR but it was worse compared to full stacking ensemble. The importance of stacking of tree-based ensemble is highlighted by this study which increases the performance of the model than any individual model by combining the predictions from multiple models. This stacking approach uses strengths from each machine learning models and makes more accurate predictions across diverse data patterns and store-item combinations.

The detailed comparison between CatBoost and XGBR helps us to make decision for which algorithms should be selected for retail demand forecasting. The study shows the advantage of combining multiple tree-based algorithms in a stacking ensemble rather than relying on a single algorithm. The findings of this research fits perfectly with the finding of several studies that often outperform individual models for complex prediction tasks.

CHAPTER 5

CONCLUSION, FUTURE SCOPE AND SOCIAL IMPACT

5.1 Conclusion

This research combines the domain of operation and production management with machine learning. In this interdisciplinary research era, this research focuses on integrating demand forecasting problem with machine learning to increase the forecasting accuracy which will help to minimize the inventory cost. This research study indicates that tree-based stacking ensemble performs better than ANN which is a deep learning model. The stacking model which consists of RF, XGBoost, LightGBM, and CatBoost as base learner and Gradient Boosting as the meta-learner aligns with the existing research than ensemble models performs better than individual models (Eglite & Birzniece, 2022)(Mejía & Aguilar, 2024). Ensemble model achieved maximum R^2 of 0.994 whereas it was only 0.924 for ANN. Each model of base learners combine its unique abilities and help to achieve improved forecasting accuracy. Gradient Boost is preferred as meta learners instead of other tree-based model due to its unique ability to learn properly from residual patterns in the data. Similarly, the two best performing models of stacking CatBoost and XGBR was evaluated where the performance of CatBoost was better than XGBR and was almost in line with the performance of ensemble. Grid search technique used for the model training of CatBoost and XGBR helped to achieve forecasting result with optimal performance of the model.

The ANOVA result showed 71.3% variability in sales which is explained by the model and suggested for proper feature engineering in the dataset so that the hidden pattern for forecasting can be determined. Also, the result showed sales varies by store and items. The relevant features such as rolling, lag, month-over-month growth were incorporated in the model to help the model to capture the patterns in the data.

5.2 Future Scope

The future scope of this study are as follows:

1. Error adaptation for stacking : The future study can be conducted to formulate a model that can adjust the weight of each model used in stacking to increase the accuracy of forecasting.
2. Forecasting among multiple domains : The same model can be used in various domains such as energy, logistics, and healthcare to validate the effectiveness of the model.
3. Comparison with broader deep learning models : ANN is baseline models so future study can compare the stacking with more advanced models such as Transformer, CNN-LSTM.
4. Forecasting for larger chain : While this research only forecasts for two stores and five items, future research can conduct study on multinational chain consisting of thousands of categories.
5. Effect on inventory cost : Future research can conduct study on the percentage of inventory cost saving due to the ensemble forecast.

5.3 Social Impact

The social impact of this study are as follows:

1. Improvement in retail efficiency: With this forecast model, the problem of overstocking and understocking in the retail sector can be minimized, which can reduce the wastages of materials.
2. Increase in customer satisfaction: This forecast will help to prevent stockout even during peak demand time and will help to improve customer satisfaction.
3. Green supply chain: Proper demand forecast will minimize the need for unnecessary frequent transportation which will help to reduce carbon emission as well as energy consumption in the supply chain.

REFERENCES

Aburto, L., & Weber, R. (2007). Improved supply chain management based on hybrid demand forecasts. *Applied Soft Computing Journal*, *7*(1), 136–144. https://doi.org/10.1016/j.asoc.2005.06.001

Ahmad, T., & Chen, H. (2018). Utility companies strategy for short-term energy demand forecasting using machine learning based models. *Sustainable Cities and Society*, *39*(March), 401–417. https://doi.org/10.1016/j.scs.2018.03.002

Alon, I., Qi, M., & Sadowski, R. J. (2001). Forecasting aggregate retail sales: A comparison of artifcial neural networks and traditional methods. *Journal of Retailing and Consumer Services*, *8*(3), 147–156.

Aria, M., & Cuccurullo, C. (2017). bibliometrix: An R-tool for comprehensive science mapping analysis. *Journal of Informetrics*, *11*(4), 959–975. https://doi.org/10.1016/j.joi.2017.08.007

Chen, I. F., & Lu, C. J. (2021). Demand forecasting for multichannel fashion retailers by integrating clustering and machine learning algorithms. *Processes*, *9*(9), 1–14. https://doi.org/10.3390/pr9091578

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-Augu*, 785–794. https://doi.org/10.1145/2939672.2939785

Chowdhury, A. R., Paul, R., & Rozony, F. Z. (2025). *A SYSTEMATIC REVIEW OF DEMAND FORECASTING MODELS FOR RETAIL E-COMMERCE ENHANCING ACCURACY IN INVENTORY AND DELIVERY PLANNING*. *6*(1), 1–27. https://doi.org/10.63125/mbbfw637

Duan, G., & Dong, J. (2024). Construction of Ensemble Learning Model for Home Appliance Demand Forecasting. *Applied Sciences (Switzerland)*, *14*(17). https://doi.org/10.3390/app14177658

Eglite, L., & Birzniece, I. (2022). Retail Sales Forecasting Using Deep Learning: Systematic Literature Review. *Complex Systems Informatics and Modeling Quarterly*, *2022*(30), 53–62. https://doi.org/10.7250/csimq.2022-30.03

Emami Javanmard, M., & Ghaderi, S. F. (2023). Energy demand forecasting in seven sectors by an optimization model based on machine learning algorithms. *Sustainable Cities and Society*, *95*(April), 104623. https://doi.org/10.1016/j.scs.2023.104623

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*(2019), 993–1003. https://doi.org/10.1016/j.procs.2022.01.298

Feizabadi, J. (2020). 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

Gahirwal, M. (2013). *Inter Time Series Sales Forecasting*. http://arxiv.org/abs/1303.0117

Giri, C., & Chen, Y. (2022). Deep Learning for Demand Forecasting in the Fashion and Apparel Retail Industry. *Forecasting*, *4*(2), 565–581. https://doi.org/10.3390/forecast4020031

Güven, İ., Uygun, Ö., & Şimşir, F. (2021). Machine learning algorithms with intermittent demand forecasting: An application in retail apparel with plenty of predictors. *Tekstil ve Konfeksiyon*, *31*(2), 99–110. https://doi.org/10.32710/tekstilvekonfeksiyon.809867

Haque, M. S., Amin, M. S., Miah, J., Cao, D. M., Sayed, M. A., & Ahmed, S. (2023). Retail Demand Forecasting: A Comparative Study for Multivariate Time Series. *Journal of Mathematics and Statistics Studies*, *4*(4), 40–46. https://doi.org/10.32996/jmss.2023.4.4.4

Haselbeck, F., Killinger, J., Menrad, K., Hannus, T., & Grimm, D. G. (2022). Machine Learning Outperforms Classical Forecasting on Horticultural Sales Predictions. *Machine Learning with Applications*, *7*(December 2021), 100239. https://doi.org/10.1016/j.mlwa.2021.100239

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

Jain, S., Agrawal, S., Mohapatra, E., & Srinivasan, K. (2024). A novel ensemble ARIMA-LSTM approach for evaluating COVID-19 cases and future outbreak preparedness. *Health Care Science*. https://doi.org/10.1002/hcs2.123

Joseph, R. V., Mohanty, A., Tyagi, S., Mishra, S., Satapathy, S. K., & Mohanty, S. N. (2022). A hybrid deep learning framework with CNN and Bi-directional LSTM for store item demand forecasting. *Computers and Electrical Engineering*, *103*(August), 108358. https://doi.org/10.1016/j.compeleceng.2022.108358

Kantasa-ard, A., Nouiri, M., Bekrar, A., Ait el cadi, A., & Sallez, Y. (2021). Machine learning for demand forecasting in the physical internet: a case study of agricultural products in Thailand.

International Journal of Production Research, *59*(24), 7491–7515. https://doi.org/10.1080/00207543.2020.1844332

Kuvulmaz, J., Usanmaz, S., & Engin, S. N. (2005). Time-series forecasting by means of linear and nonlinear models. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3789 LNAI*, 504–513. https://doi.org/10.1007/11579427_51

Lu, C. J. (2014). Sales forecasting of computer products based on variable selection scheme and support vector regression. *Neurocomputing*, *128*, 491–499. https://doi.org/10.1016/j.neucom.2013.08.012

Ma, S., & Fildes, R. (2021). Retail sales forecasting with meta-learning. *European Journal of Operational Research*, *288*(1), 111–128. https://doi.org/10.1016/j.ejor.2020.05.038

Majed, A. A. (2022). *Machine Learning Approach for Short-Term Load Forecasting Using Deep Neural Network*. 1–23.

Manandhar, P., Rafiq, H., Rodriguez-Ubinas, E., & Palpanas, T. (2024). New Forecasting Metrics Evaluated in Prophet, Random Forest, and Long Short-Term Memory Models for Load Forecasting. *Energies*, *17*(23), 1–30. https://doi.org/10.3390/en17236131

Mejía, S., & Aguilar, J. (2024). A demand forecasting system of product categories defined by their time series using a hybrid approach of ensemble learning with feature engineering. *Computing*, 3945–3965. https://doi.org/10.1007/s00607-024-01320-y

Merigó, J. M., & Yang, J. B. (2017). A bibliometric analysis of operations research and management science. *Omega*, *73*, 37–48. https://doi.org/10.1016/j.omega.2016.12.004

Mitra, A., Jain, A., 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*(4), 1–22. https://doi.org/10.1007/s43069-022-00166-4

Nasseri, M., Falatouri, T., Brandtner, P., & Darbanian, F. (2023). Applying Machine Learning in Retail Demand Prediction—A Comparison of Tree-Based Ensembles and Long Short-Term Memory-Based Deep Learning. *Applied Sciences (Switzerland)*, *13*(19). https://doi.org/10.3390/app131911112

Priyadarshi, R., Panigrahi, A., Routroy, S., & Garg, G. K. (2019). Demand forecasting at retail stage for selected vegetables: a performance analysis. *Journal of Modelling in Management*, *14*(4), 1042–1063. https://doi.org/10.1108/JM2-11-2018-0192

Punia, S., Nikolopoulos, K., Singh, S. P., Madaan, J. K., & Litsiou, K. (2020). Deep learning with long short-term memory networks and random forests for demand forecasting in multi-channel retail. *International Journal of Production Research*, *58*(16), 4964–4979. https://doi.org/10.1080/00207543.2020.1735666

Punia, S., & Shankar, S. (2022). Predictive analytics for demand forecasting: A deep learning-based decision support system. *Knowledge-Based Systems*, *258*, 109956. https://doi.org/10.1016/j.knosys.2022.109956

Raju, S. M. T. U., Sarker, A., Das, A., Islam, M. M., Al-Rakhami, M. S., Al-Amri, A. M., Mohiuddin, T., & Albogamy, F. R. (2022). An Approach for Demand Forecasting in Steel Industries Using Ensemble Learning. *Complexity*, *2022*. https://doi.org/10.1155/2022/9928836

Ramos, P. V. B., Villela, S. M., Silva, W. N., & Dias, B. H. (2023). Residential energy consumption forecasting using deep learning models. *Applied Energy*, *350*(July), 121705. https://doi.org/10.1016/j.apenergy.2023.121705

Ribeiro, G. T., Mariani, V. C., & Coelho, L. dos S. (2019). Enhanced ensemble structures using wavelet neural networks applied to short-term load forecasting. *Engineering Applications of Artificial Intelligence*, *82*(August 2017), 272–281. https://doi.org/10.1016/j.engappai.2019.03.012

Ribeiro, M. H. D. M., & dos Santos Coelho, L. (2020). Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series. *Applied Soft Computing Journal*, *86*, 105837. https://doi.org/10.1016/j.asoc.2019.105837

Saha, P., Gudheniya, N., Mitra, R., Das, D., Narayana, S., & Tiwari, M. K. (2022). Demand Forecasting of a Multinational Retail Company using Deep Learning Frameworks. *IFAC-PapersOnLine*, *55*(10), 395–399. https://doi.org/10.1016/j.ifacol.2022.09.425

Salais-Fierro, T. E., & Martínez, J. A. S. (2022). Demand Forecasting for Freight Transport Applying Machine Learning into the Logistic Distribution. *Mobile Networks and Applications*, *27*(5), 2172–2181. https://doi.org/10.1007/s11036-021-01854-x

Seyedan, M., Mafakheri, F., & Wang, C. (2022). Cluster-based demand forecasting using Bayesian model averaging: An ensemble learning approach. *Decision Analytics Journal*, *3*(February), 100033. https://doi.org/10.1016/j.dajour.2022.100033

Sukolkit, N., Arunyanart, S., & Apichottanakul, A. (2024). An open innovative inventory management based demand forecasting approach for the steel industry. *Journal of Open Innovation: Technology, Market, and Complexity*, *10*(4), 100407. https://doi.org/10.1016/j.joitmc.2024.100407

Thomassey, S. (2010). Sales forecasts in clothing industry: The key success factor of the supply chain management. *International Journal of Production Economics*, *128*(2), 470–483. https://doi.org/10.1016/j.ijpe.2010.07.018

Tong, C., Zhang, L., Li, H., & Ding, Y. (2023). Attention-based temporal–spatial convolutional network for ultra-short-term load forecasting. *Electric Power Systems Research*, *220*(June 2022), 109329. https://doi.org/10.1016/j.epsr.2023.109329

Wanchoo, K. (2019). 5Hwdlo ' Hpdqg) Ruhfdvwlqj D & Rpsdulvrq Ehwzhhq ' Hhs 1Hxudo 1Hwzrun Dqg * Udglhqw % Rrvwlqj 0Hwkrg. *2019 5th International Conference for Convergence in Technology (I2CT)*, 5–9.

Wang, J., Chong, W. K., Lin, J., & Hedenstierna, C. P. T. (2024). Retail Demand Forecasting Using Spatial-Temporal Gradient Boosting Methods. *Journal of Computer Information Systems*, *64*(5), 652–664. https://doi.org/10.1080/08874417.2023.2240753

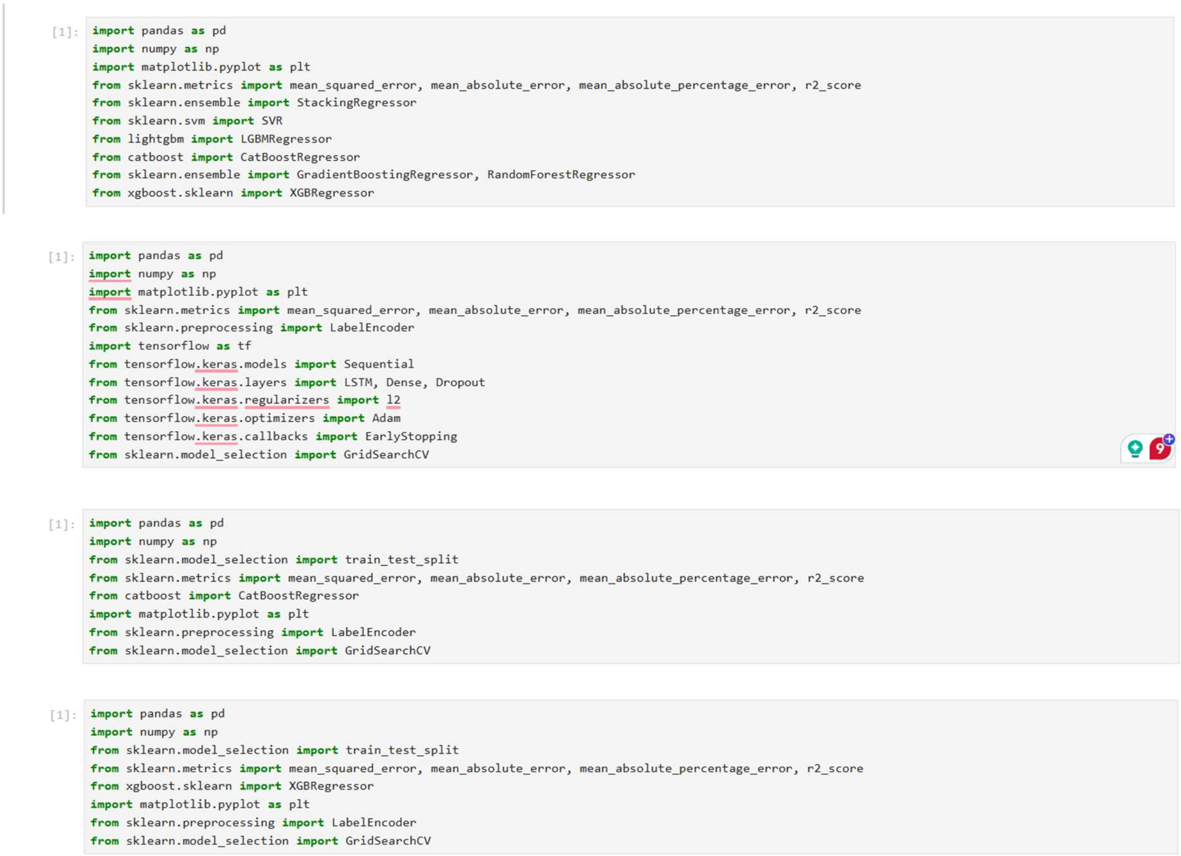
Wang, L., Duan, W., Qu, D., & Wang, S. (2018). What matters for global food price volatility? *Empirical Economics*, *54*(4), 1549–1572. https://doi.org/10.1007/s00181-017-1311-9

Yazici, C., & Domínguez-Gutiérrez, F. J. (2025). Machine learning techniques for estimating high–temperature mechanical behavior of high strength steels. *Results in Engineering*, *25*(December 2024), 104242. https://doi.org/10.1016/j.rineng.2025.104242

Zhang, G. P., & Qi, M. (2005). Neural network forecasting for seasonal and trend time series. *European Journal of Operational Research*, *160*(2), 501–514. https://doi.org/10.1016/j.ejor.2003.08.037

APPENDICES

Figure 1 : Importing the necessary libraries for ensemble machine learning , ANN, CatBoost and XGBR



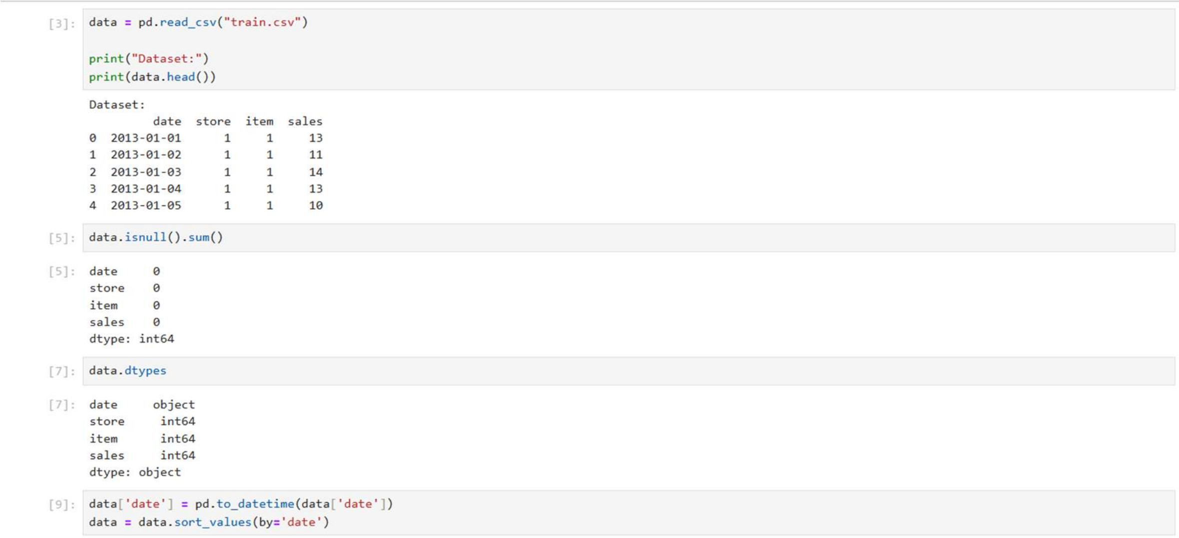
```
[1]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     from sklearn.metrics import mean_squared_error, mean_absolute_error, mean_absolute_percentage_error, r2_score
     from sklearn.ensemble import StackingRegressor
     from sklearn.svm import SVR
     from lightgbm import LGBMRegressor
     from catboost import CatBoostRegressor
     from sklearn.ensemble import GradientBoostingRegressor, RandomForestRegressor
     from xgboost.sklearn import XGBRegressor
```

```
[1]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     from sklearn.metrics import mean_squared_error, mean_absolute_error, mean_absolute_percentage_error, r2_score
     from sklearn.preprocessing import LabelEncoder
     import tensorflow as tf
     from tensorflow.keras.models import Sequential
     from tensorflow.keras.layers import LSTM, Dense, Dropout
     from tensorflow.keras.regularizers import l2
     from tensorflow.keras.optimizers import Adam
     from tensorflow.keras.callbacks import EarlyStopping
     from sklearn.model_selection import GridSearchCV
```

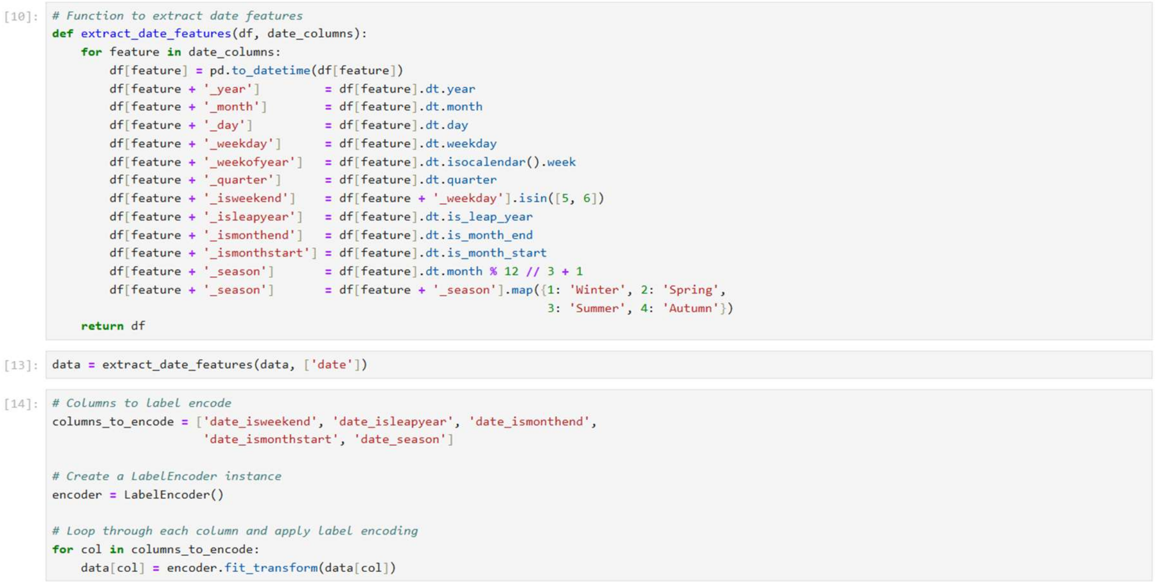
```
[1]: import pandas as pd
     import numpy as np
     from sklearn.model_selection import train_test_split
     from sklearn.metrics import mean_squared_error, mean_absolute_error, mean_absolute_percentage_error, r2_score
     from catboost import CatBoostRegressor
     import matplotlib.pyplot as plt
     from sklearn.preprocessing import LabelEncoder
     from sklearn.model_selection import GridSearchCV
```

```
[1]: import pandas as pd
     import numpy as np
     from sklearn.model_selection import train_test_split
     from sklearn.metrics import mean_squared_error, mean_absolute_error, mean_absolute_percentage_error, r2_score
     from xgboost.sklearn import XGBRegressor
     import matplotlib.pyplot as plt
     from sklearn.preprocessing import LabelEncoder
     from sklearn.model_selection import GridSearchCV
```

Figure 2 : Data Preprocessing



```
[3]: data = pd.read_csv("train.csv")

     print("Dataset:")
     print(data.head())

     Dataset:
              date  store  item  sales
     0  2013-01-01      1     1     13
     1  2013-01-02      1     1     11
     2  2013-01-03      1     1     14
     3  2013-01-04      1     1     13
     4  2013-01-05      1     1     10

[5]: data.isnull().sum()

[5]: date     0
     store    0
     item     0
     sales    0
     dtype: int64

[7]: data.dtypes

[7]: date      object
     store      int64
     item       int64
     sales      int64
     dtype: object

[9]: data['date'] = pd.to_datetime(data['date'])
     data = data.sort_values(by='date')
```

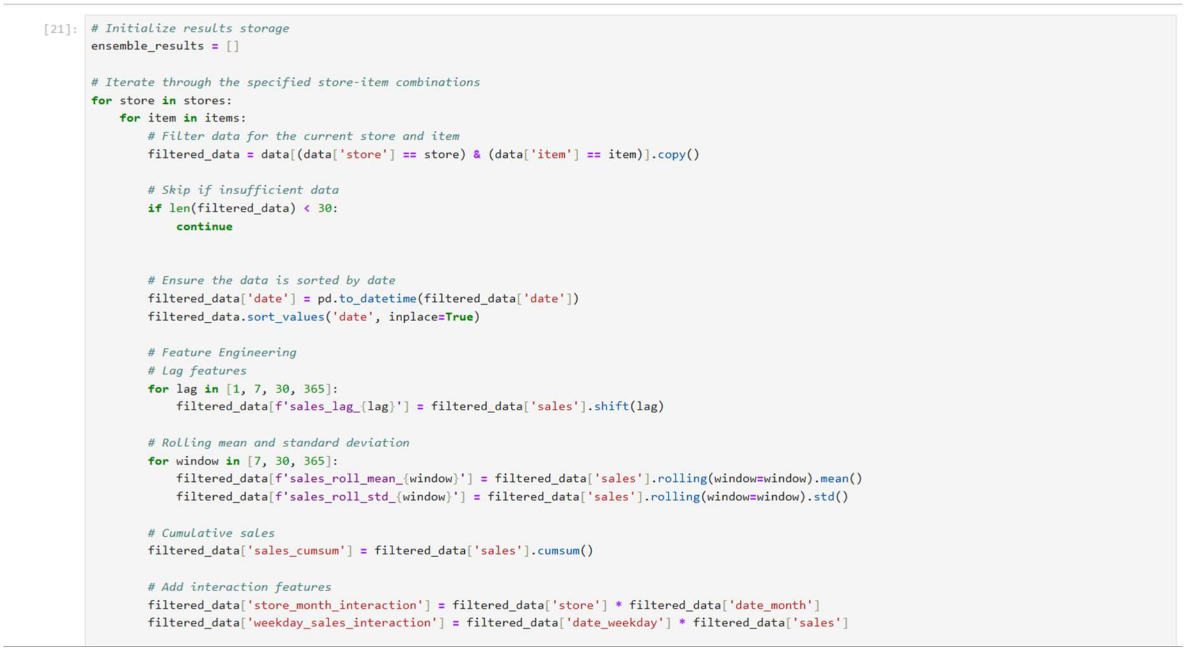


```
[10]: # Function to extract date features
      def extract_date_features(df, date_columns):
          for feature in date_columns:
              df[feature] = pd.to_datetime(df[feature])
              df[feature + '_year']         = df[feature].dt.year
              df[feature + '_month']        = df[feature].dt.month
              df[feature + '_day']          = df[feature].dt.day
              df[feature + '_weekday']      = df[feature].dt.weekday
              df[feature + '_weekofyear']   = df[feature].dt.isocalendar().week
              df[feature + '_quarter']      = df[feature].dt.quarter
              df[feature + '_isweekend']    = df[feature + '_weekday'].isin([5, 6])
              df[feature + '_isleapyear']   = df[feature].dt.is_leap_year
              df[feature + '_ismonthend']   = df[feature].dt.is_month_end
              df[feature + '_ismonthstart'] = df[feature].dt.is_month_start
              df[feature + '_season']       = df[feature].dt.month % 12 // 3 + 1
              df[feature + '_season']       = df[feature + '_season'].map({1: 'Winter', 2: 'Spring',
                                                                          3: 'Summer', 4: 'Autumn'})

          return df
```

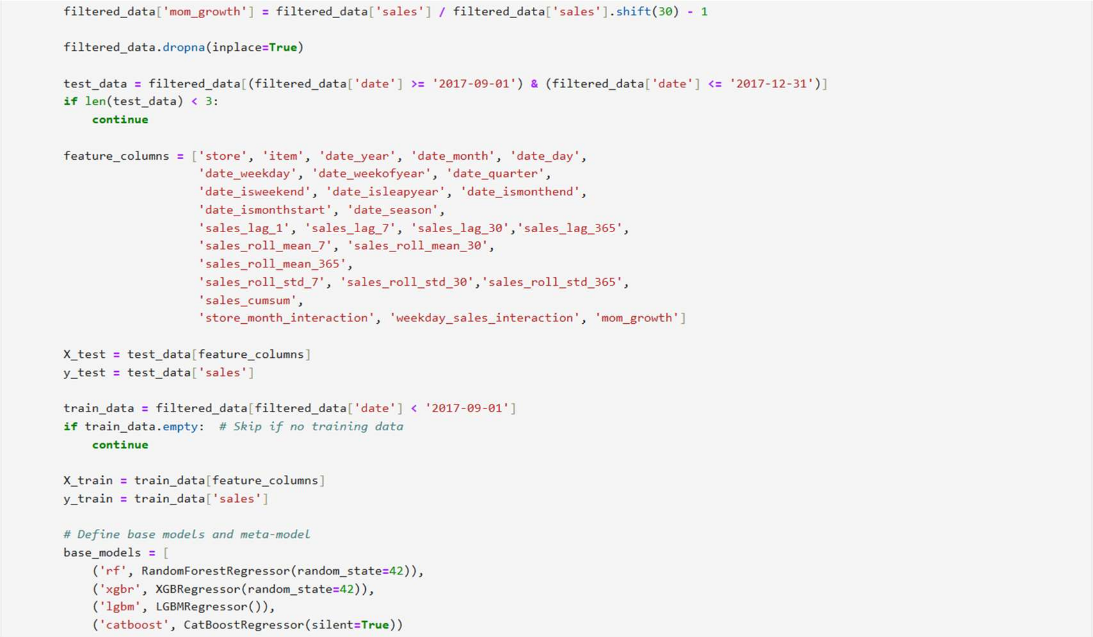
```
[13]: data = extract_date_features(data, ['date'])
```

```
[14]: # Columns to label encode
      columns_to_encode = ['date_isweekend', 'date_isleapyear', 'date_ismonthend',
                           'date_ismonthstart', 'date_season']

      # Create a LabelEncoder instance
      encoder = LabelEncoder()

      # Loop through each column and apply label encoding
      for col in columns_to_encode:
          data[col] = encoder.fit_transform(data[col])
```

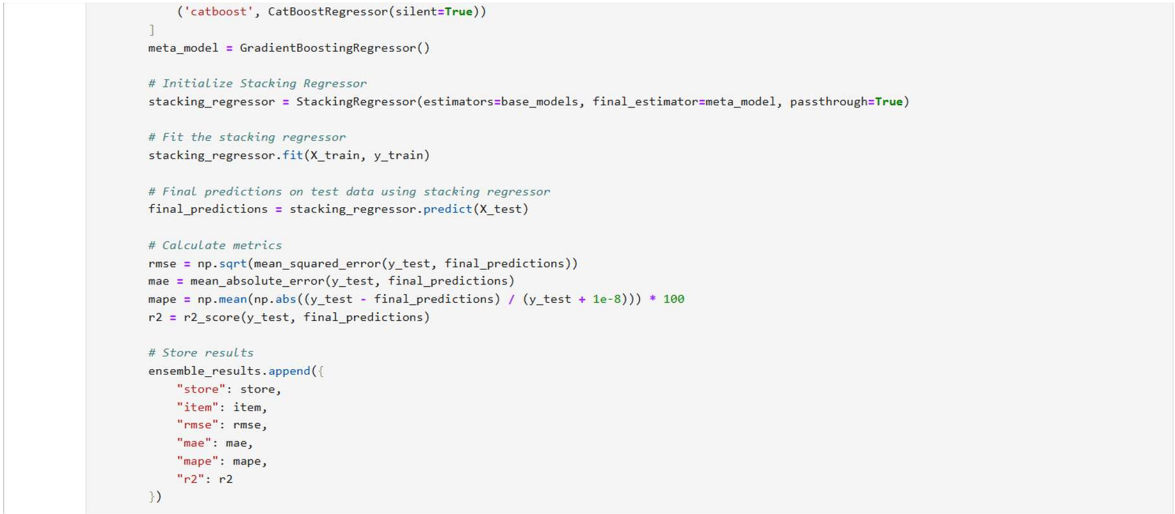
Figure 3 : Model Training for ensemble machine learning with integrated feature engineering



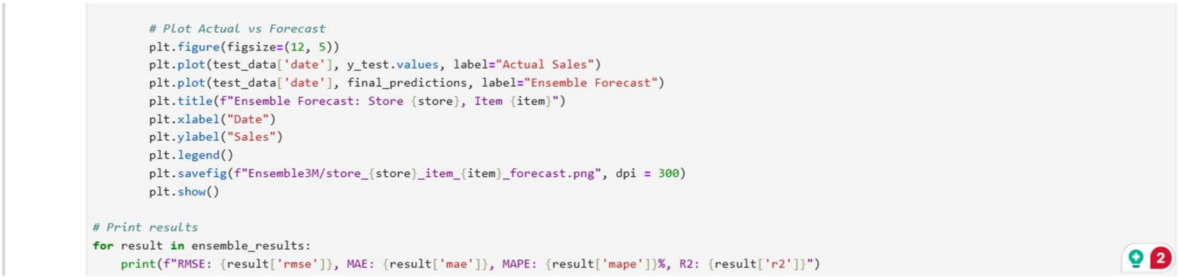
```
[21]: # Initialize results storage
      ensemble_results = []

      # Iterate through the specified store-item combinations
      for store in stores:
          for item in items:
              # Filter data for the current store and item
              filtered_data = data[(data['store'] == store) & (data['item'] == item)].copy()

              # Skip if insufficient data
              if len(filtered_data) < 30:
                  continue


              # Ensure the data is sorted by date
              filtered_data['date'] = pd.to_datetime(filtered_data['date'])
              filtered_data.sort_values('date', inplace=True)

              # Feature Engineering
              # Lag features
              for lag in [1, 7, 30, 365]:
                  filtered_data[f'sales_lag_{lag}'] = filtered_data['sales'].shift(lag)

              # Rolling mean and standard deviation
              for window in [7, 30, 365]:
                  filtered_data[f'sales_roll_mean_{window}'] = filtered_data['sales'].rolling(window=window).mean()
                  filtered_data[f'sales_roll_std_{window}'] = filtered_data['sales'].rolling(window=window).std()

              # Cumulative sales
              filtered_data['sales_cumsum'] = filtered_data['sales'].cumsum()

              # Add interaction features
              filtered_data['store_month_interaction'] = filtered_data['store'] * filtered_data['date_month']
              filtered_data['weekday_sales_interaction'] = filtered_data['date_weekday'] * filtered_data['sales']
```



```
        filtered_data['mom_growth'] = filtered_data['sales'] / filtered_data['sales'].shift(30) - 1

        filtered_data.dropna(inplace=True)

        test_data = filtered_data[(filtered_data['date'] >= '2017-09-01') & (filtered_data['date'] <= '2017-12-31')]
        if len(test_data) < 3:
            continue

        feature_columns = ['store', 'item', 'date_year', 'date_month', 'date_day',
                           'date_weekday', 'date_weekofyear', 'date_quarter',
                           'date_isweekend', 'date_isleapyear', 'date_ismonthend',
                           'date_ismonthstart', 'date_season',
                           'sales_lag_1', 'sales_lag_7', 'sales_lag_30','sales_lag_365',
                           'sales_roll_mean_7', 'sales_roll_mean_30',
                           'sales_roll_mean_365',
                           'sales_roll_std_7', 'sales_roll_std_30','sales_roll_std_365',
                           'sales_cumsum',
                           'store_month_interaction', 'weekday_sales_interaction', 'mom_growth']

        X_test = test_data[feature_columns]
        y_test = test_data['sales']

        train_data = filtered_data[filtered_data['date'] < '2017-09-01']
        if train_data.empty:  # Skip if no training data
            continue

        X_train = train_data[feature_columns]
        y_train = train_data['sales']

        # Define base models and meta-model
        base_models = [
            ('rf', RandomForestRegressor(random_state=42)),
            ('xgbr', XGBRegressor(random_state=42)),
            ('lgbm', LGBMRegressor()),
            ('catboost', CatBoostRegressor(silent=True))
```

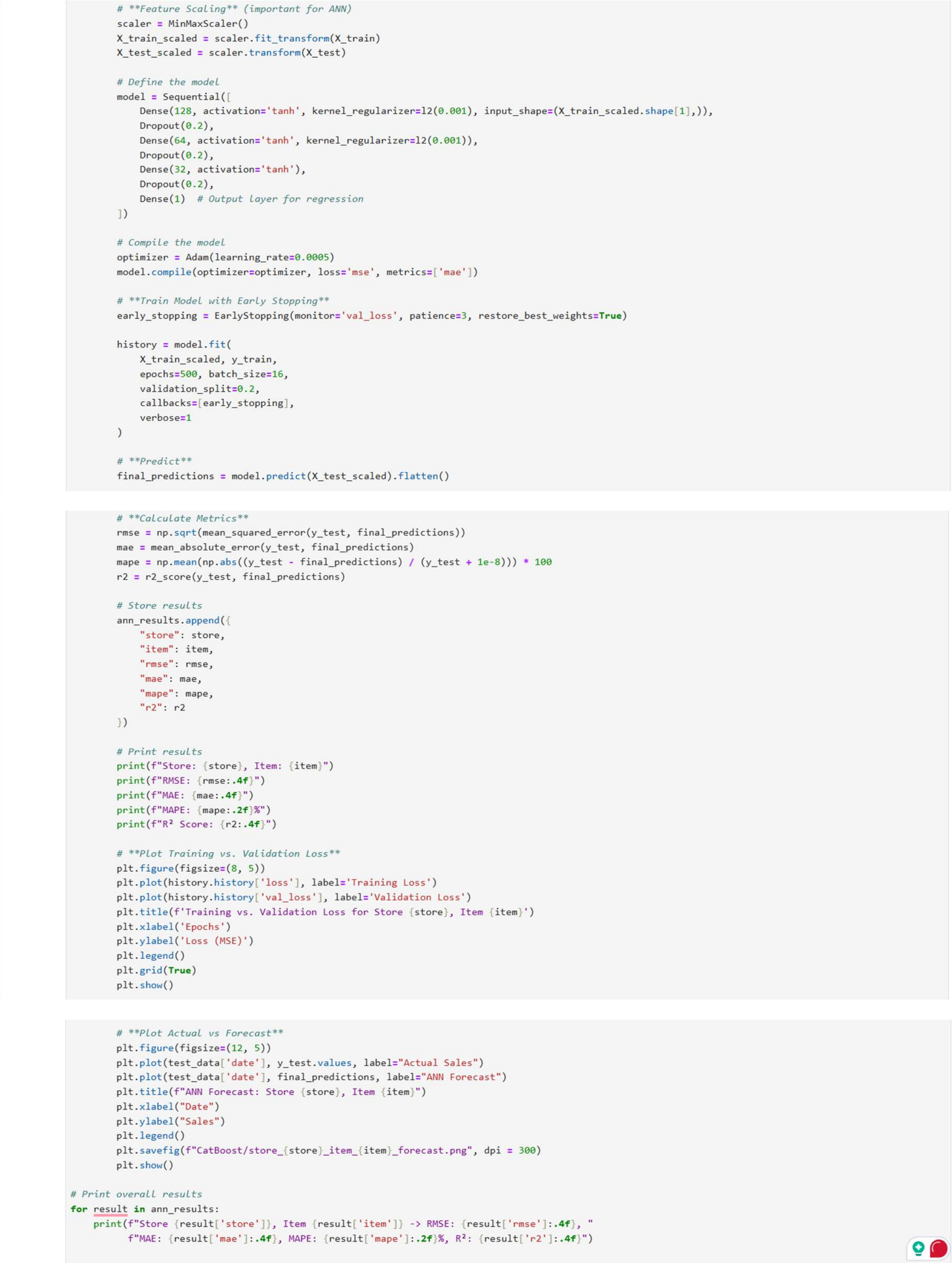


```
            ('catboost', CatBoostRegressor(silent=True))
        ]
        meta_model = GradientBoostingRegressor()

        # Initialize Stacking Regressor
        stacking_regressor = StackingRegressor(estimators=base_models, final_estimator=meta_model, passthrough=True)

        # Fit the stacking regressor
        stacking_regressor.fit(X_train, y_train)

        # Final predictions on test data using stacking regressor
        final_predictions = stacking_regressor.predict(X_test)

        # Calculate metrics
        rmse = np.sqrt(mean_squared_error(y_test, final_predictions))
        mae = mean_absolute_error(y_test, final_predictions)
        mape = np.mean(np.abs((y_test - final_predictions) / (y_test + 1e-8))) * 100
        r2 = r2_score(y_test, final_predictions)

        # Store results
        ensemble_results.append({
            "store": store,
            "item": item,
            "rmse": rmse,
            "mae": mae,
            "mape": mape,
            "r2": r2
        })
```



```
        # Plot Actual vs Forecast
        plt.figure(figsize=(12, 5))
        plt.plot(test_data['date'], y_test.values, label="Actual Sales")
        plt.plot(test_data['date'], final_predictions, label="Ensemble Forecast")
        plt.title(f"Ensemble Forecast: Store {store}, Item {item}")
        plt.xlabel("Date")
        plt.ylabel("Sales")
        plt.legend()
        plt.savefig(f"Ensemble3M/store_{store}_item_{item}_forecast.png", dpi = 300)
        plt.show()

# Print results
for result in ensemble_results:
    print(f"RMSE: {result['rmse']}, MAE: {result['mae']}, MAPE: {result['mape']}%, R2: {result['r2']}")
```

Figure 4 : ANN Forecast Modelling

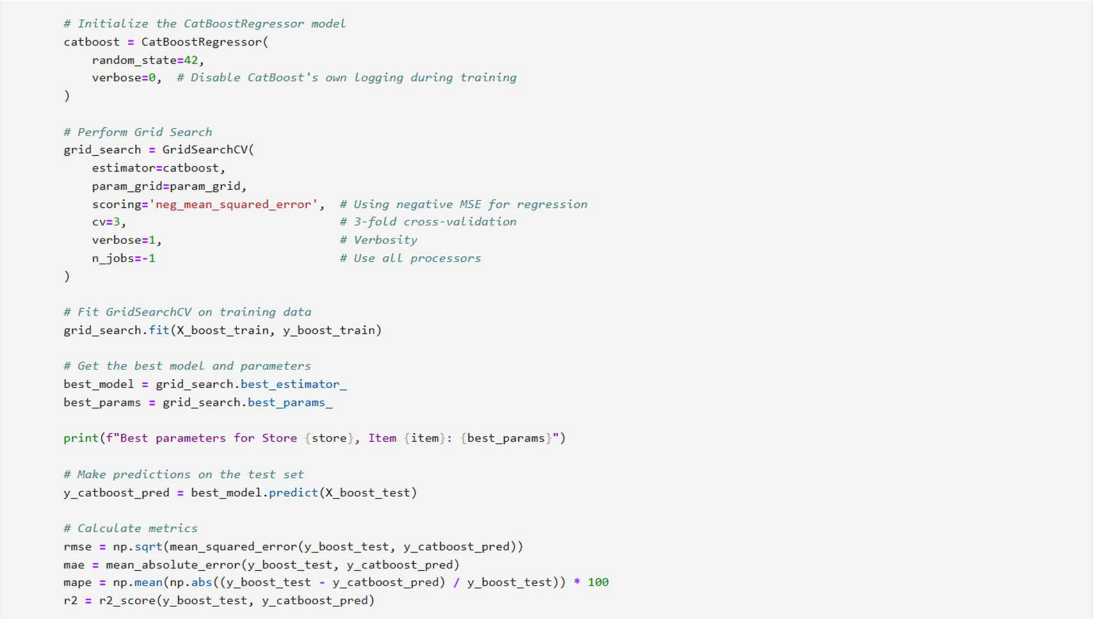


```
        # **Feature Scaling** (important for ANN)
        scaler = MinMaxScaler()
        X_train_scaled = scaler.fit_transform(X_train)
        X_test_scaled = scaler.transform(X_test)

        # Define the model
        model = Sequential([
            Dense(128, activation='tanh', kernel_regularizer=l2(0.001), input_shape=(X_train_scaled.shape[1],)),
            Dropout(0.2),
            Dense(64, activation='tanh', kernel_regularizer=l2(0.001)),
            Dropout(0.2),
            Dense(32, activation='tanh'),
            Dropout(0.2),
            Dense(1)  # Output layer for regression
        ])

        # Compile the model
        optimizer = Adam(learning_rate=0.0005)
        model.compile(optimizer=optimizer, loss='mse', metrics=['mae'])

        # **Train Model with Early Stopping**
        early_stopping = EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=True)

        history = model.fit(
            X_train_scaled, y_train,
            epochs=500, batch_size=16,
            validation_split=0.2,
            callbacks=[early_stopping],
            verbose=1
        )

        # **Predict**
        final_predictions = model.predict(X_test_scaled).flatten()
```

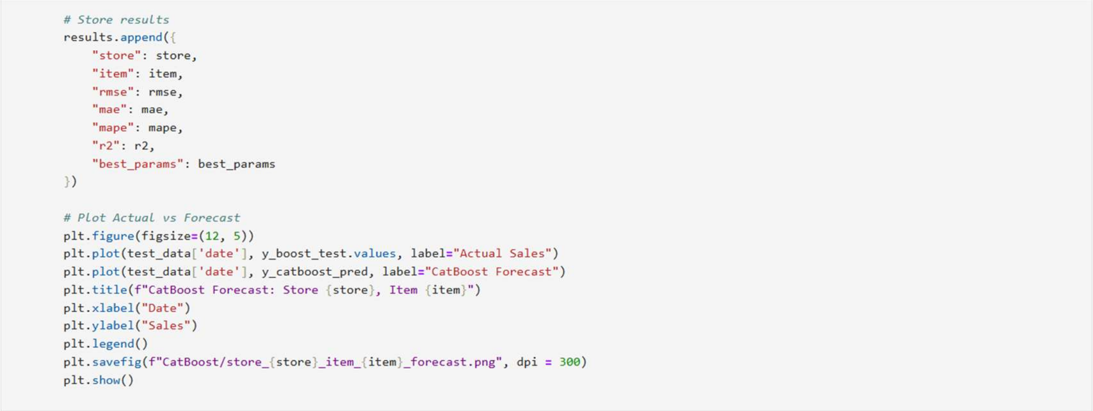
```
        # **Calculate Metrics**
        rmse = np.sqrt(mean_squared_error(y_test, final_predictions))
        mae = mean_absolute_error(y_test, final_predictions)
        mape = np.mean(np.abs((y_test - final_predictions) / (y_test + 1e-8))) * 100
        r2 = r2_score(y_test, final_predictions)

        # Store results
        ann_results.append({
            "store": store,
            "item": item,
            "rmse": rmse,
            "mae": mae,
            "mape": mape,
            "r2": r2
        })

        # Print results
        print(f"Store: {store}, Item: {item}")
        print(f"RMSE: {rmse:.4f}")
        print(f"MAE: {mae:.4f}")
        print(f"MAPE: {mape:.2f}%")
        print(f"R² Score: {r2:.4f}")

        # **Plot Training vs. Validation Loss**
        plt.figure(figsize=(8, 5))
        plt.plot(history.history['loss'], label='Training Loss')
        plt.plot(history.history['val_loss'], label='Validation Loss')
        plt.title(f'Training vs. Validation Loss for Store {store}, Item {item}')
        plt.xlabel('Epochs')
        plt.ylabel('Loss (MSE)')
        plt.legend()
        plt.grid(True)
        plt.show()
```

```
        # **Plot Actual vs Forecast**
        plt.figure(figsize=(12, 5))
        plt.plot(test_data['date'], y_test.values, label="Actual Sales")
        plt.plot(test_data['date'], final_predictions, label="ANN Forecast")
        plt.title(f"ANN Forecast: Store {store}, Item {item}")
        plt.xlabel("Date")
        plt.ylabel("Sales")
        plt.legend()
        plt.savefig(f"CatBoost/store_{store}_item_{item}_forecast.png", dpi = 300)
        plt.show()

# Print overall results
for result in ann_results:
    print(f"Store {result['store']}, Item {result['item']} -> RMSE: {result['rmse']:.4f}, "
          f"MAE: {result['mae']:.4f}, MAPE: {result['mape']:.2f}%, R²: {result['r2']:.4f}")
```

Figure 5 : CatBoost modelling with grid search technique

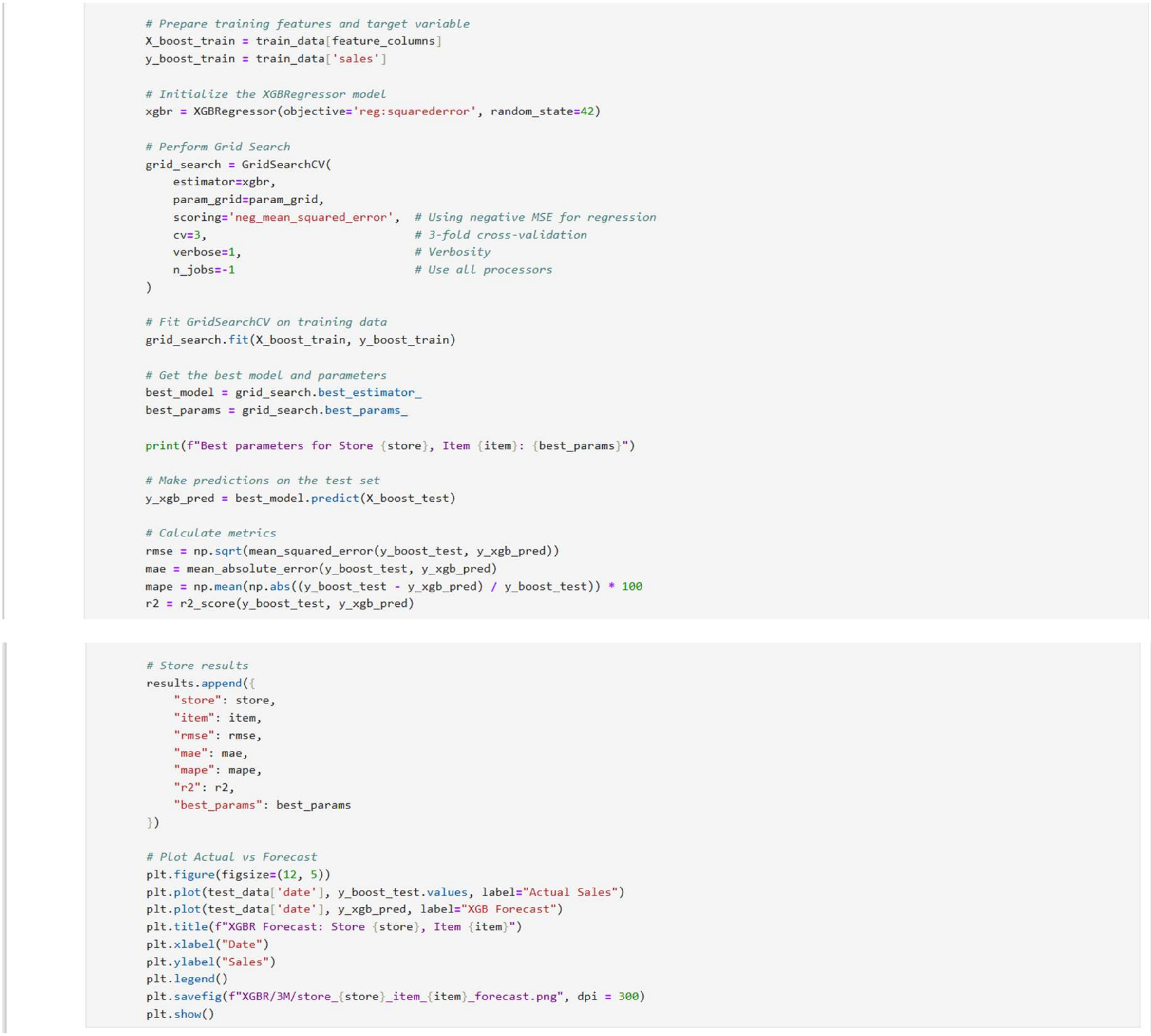


```
# Initialize the CatBoostRegressor model
catboost = CatBoostRegressor(
    random_state=42,
    verbose=0,  # Disable CatBoost's own logging during training
)

# Perform Grid Search
grid_search = GridSearchCV(
    estimator=catboost,
    param_grid=param_grid,
    scoring='neg_mean_squared_error',  # Using negative MSE for regression
    cv=3,                              # 3-fold cross-validation
    verbose=1,                         # Verbosity
    n_jobs=-1                          # Use all processors
)

# Fit GridSearchCV on training data
grid_search.fit(X_boost_train, y_boost_train)

# Get the best model and parameters
best_model = grid_search.best_estimator_
best_params = grid_search.best_params_

print(f"Best parameters for Store {store}, Item {item}: {best_params}")

# Make predictions on the test set
y_catboost_pred = best_model.predict(X_boost_test)

# Calculate metrics
rmse = np.sqrt(mean_squared_error(y_boost_test, y_catboost_pred))
mae = mean_absolute_error(y_boost_test, y_catboost_pred)
mape = np.mean(np.abs((y_boost_test - y_catboost_pred) / y_boost_test)) * 100
r2 = r2_score(y_boost_test, y_catboost_pred)
```



```
# Store results
results.append({
    "store": store,
    "item": item,
    "rmse": rmse,
    "mae": mae,
    "mape": mape,
    "r2": r2,
    "best_params": best_params
})

# Plot Actual vs Forecast
plt.figure(figsize=(12, 5))
plt.plot(test_data['date'], y_boost_test.values, label="Actual Sales")
plt.plot(test_data['date'], y_catboost_pred, label="CatBoost Forecast")
plt.title(f"CatBoost Forecast: Store {store}, Item {item}")
plt.xlabel("Date")
plt.ylabel("Sales")
plt.legend()
plt.savefig(f"CatBoost/store_{store}_item_{item}_forecast.png", dpi = 300)
plt.show()
```

Figure 6 : XGBR modelling with grid search technique



```
# Prepare training features and target variable
X_boost_train = train_data[feature_columns]
y_boost_train = train_data['sales']

# Initialize the XGBRegressor model
xgbr = XGBRegressor(objective='reg:squarederror', random_state=42)

# Perform Grid Search
grid_search = GridSearchCV(
    estimator=xgbr,
    param_grid=param_grid,
    scoring='neg_mean_squared_error',  # Using negative MSE for regression
    cv=3,                              # 3-fold cross-validation
    verbose=1,                         # Verbosity
    n_jobs=-1                          # Use all processors
)

# Fit GridSearchCV on training data
grid_search.fit(X_boost_train, y_boost_train)

# Get the best model and parameters
best_model = grid_search.best_estimator_
best_params = grid_search.best_params_

print(f"Best parameters for Store {store}, Item {item}: {best_params}")

# Make predictions on the test set
y_xgb_pred = best_model.predict(X_boost_test)

# Calculate metrics
rmse = np.sqrt(mean_squared_error(y_boost_test, y_xgb_pred))
mae = mean_absolute_error(y_boost_test, y_xgb_pred)
mape = np.mean(np.abs((y_boost_test - y_xgb_pred) / y_boost_test)) * 100
r2 = r2_score(y_boost_test, y_xgb_pred)
```

```
# Store results
results.append({
    "store": store,
    "item": item,
    "rmse": rmse,
    "mae": mae,
    "mape": mape,
    "r2": r2,
    "best_params": best_params
})

# Plot Actual vs Forecast
plt.figure(figsize=(12, 5))
plt.plot(test_data['date'], y_boost_test.values, label="Actual Sales")
plt.plot(test_data['date'], y_xgb_pred, label="XGB Forecast")
plt.title(f"XGBR Forecast: Store {store}, Item {item}")
plt.xlabel("Date")
plt.ylabel("Sales")
plt.legend()
plt.savefig(f"XGBR/3M/store_{store}_item_{item}_forecast.png", dpi = 300)
plt.show()
```

Major Project Report.pdf

Delhi Technological University

Document Details

Submission ID trn:oid:::27535:98118713	76 Pages
Submission Date May 28, 2025, 11:19 AM GMT+5:30	17,526 Words
Download Date May 28, 2025, 4:09 PM GMT+5:30	99,331 Characters
File Name Major Project Report.pdf	
File Size 2.5 MB	

6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 13 words)

Match Groups

37 Not Cited or Quoted 6%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

5%	Internet sources
2%	Publications
5%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **37** Not Cited or Quoted 6%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

5%	Internet sources
2%	Publications
5%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

#	Type	Source	Match
1	Internet	www.coursehero.com	<1%
2	Submitted works	dtusimilarity on 2024-05-29	<1%
3	Internet	dspace.dtu.ac.in:8080	<1%
4	Submitted works	University of Huddersfield on 2023-01-27	<1%
5	Internet	cwejournal.org	<1%
6	Internet	dtu.ac.in	<1%
7	Internet	www.dspace.dtu.ac.in:8080	<1%
8	Internet	neural-monkey.readthedocs.io	<1%
9	Internet	mafiadoc.com	<1%
10	Internet	www.bethanne.net	<1%

11	Internet	ir.uitm.edu.my	<1%
12	Submitted works	Wollega University on 2024-03-18	<1%
13	Internet	www.uyik.org	<1%
14	Submitted works	University of Birmingham on 2008-09-02	<1%
15	Internet	www.mdpi.com	<1%
16	Internet	rulrepository.ru.ac.bd	<1%
17	Internet	www.fastercapital.com	<1%
18	Submitted works	AAB College on 2024-09-17	<1%
19	Internet	repository.lcu.edu.ng	<1%
20	Internet	bengordonpalmbeach.com	<1%
21	Internet	nsuworks.nova.edu	<1%
22	Submitted works	Delhi Technological University on 2024-03-27	<1%
23	Submitted works	Liverpool John Moores University on 2023-01-17	<1%
24	Internet	es.slideshare.net	<1%

25	Submitted works	AlHussein Technical University on 2023-06-20	<1%
26	Submitted works	Texas State University- San Marcos on 2024-12-07	<1%
27	Submitted works	University of East London on 2023-05-03	<1%
28	Internet	dspace.bracu.ac.bd	<1%
29	Internet	ersilia.gitbook.io	<1%
30	Internet	tudr.thapar.edu:8080	<1%
31	Internet	www.thegriffinssociety.org	<1%

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis Machine Learning Approach For Retail Demand Forecasting: Integrating Feature Engineering With Stacked Tree-Based Ensemble And Deep Learning Model For Improved Accuracy

Total Pages: 76 Name of the scholar: Shishir Acharya

Supervisor(s): Dr. S. K. Garg

Department: Department of Mechanical Engineering

This is the report that the above thesis was scanned for similarity detection. The process and outcome are given below:

Software used: Turnitin Similarity Index: 6% Total Word Count: 17 526

Date: 28/05/2025

Candidate's Signature **Signature of Supervisor(s)**

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE OF FINAL THESIS SUBMISSION

(1) Name: **Shishir Acharya**
(2) Roll No.: **23/IEM/11**
(3) Thesis Title: **"Machine Learning Approach For Retail Demand Forecasting: Integrating Feature Engineering With Stacked Tree-Based Ensemble And Deep Learning Model For Improved Accuracy"**
(4) Degree for which the thesis is submitted: **M.Tech.**
(5) Faculty of the University to which the thesis is submitted: **Prof. S.K Garg**
(6) Thesis Preparation Guide was referred to for preparing the thesis: **YES**
(7) Specifications regarding thesis format have been closely followed: **YES**
(8) The contents of the thesis have been organized based on the guidelines: **YES**
(9) The thesis has been prepared without resorting to plagiarism: **YES**
(10) All sources used have been cited appropriately: **YES**
(11) The thesis has not been submitted elsewhere for a degree: **YES**
(12) Submitted 2 spiral bound copies plus one CD: **YES**

Signature of Supervisor

Dr. S.K. Garg

Signature of Candidate

Shishir Acharya

23/IEM/11

[redacted]

DELHI TECHNOLOGICAL UNIVERSITY

Shahbad Daulatpur,Main Bawana Road, Delhi-42

Proforma for Submission of M.Tech. Major Project

01. Name of the Student...Shishir Acharya

02. Enrolment No ...23/IEM/11.......

03. Year of Admission ...2023....................

04. Programme M.Tech., Branch....2023-25

05. Name of Department...Mechanical Engineering

06. Admission Category i.e. Full Time/ Full Time (Sponsored)/ Part Time:...ICCR............

07. Applied as Regular/ Ex-student....Regular...........

08. Span Period Expired on ....

09. Extension of Span Period Granted or Not Granted (if applicable)...................................

10.Title of Thesis/Major Project..Machine Learning Approach for Retail demand forecasting: Integrating Feature Engineering with stacked tree-based ensemble

11. Name of Supervisor...Dr. S.K. Garg...

12. Result Details (Enclose Copy of Mark sheets of all semesters) :

S. No.	Semester	Passing Year	Roll No.	Marks Obtained	Max. Marks	% of Marks	Details of Back Paper Cleared (if any)
01	1st	2024	23/IEM/11			9	
02	2nd	2024	23/IEM/11			8.88	
03	3rd	2025	23/IEM/11			10	
04	4th (P/T only)						
05	5th (P/T only)						

13. Fee Details (Enclose the Fee Receipt).

Amount Paid (in Rs.) ..3000............	Receipt No ..DU01225943..	Date ...28../.05../.25.

Signature of Student

It is certified that the name of Examiners for evaluation of the above thesis/ project have already been recommended by the BOS.

Signature of Supervisor

Signature of HOD with Seal

(Instructions for filling up the Form may see on back side please.)

e-Receipt for State Bank Collect Payment

REGISTRAR, DTU (RECEIPT A/C)

BAWANA ROAD, SHAHABAD DAULATPUR, , DELHI-110042
Date: 28-May-2025

SBCollect Reference Number :	DUO1225943
Category :	Miscellaneous Fees from students
Amount :	₹3000
University Roll No :	23/IEM/11
Name of the student :	Shishir Acharya
Academic Year :	2023-25
Branch Course :	Industrial Engineering and Management
Type/Name of fee :	Others if any
Remarks if any :	M.Tech Thesis Submission
Mobile No. of the student :	7011020052
Fee Amount :	3000
Transaction charge :	0.00
Total Amount (In Figures) :	3,000.00
Total Amount (In words) :	Rupees Three Thousand Only
Remarks :	M.Tech Thesis Submitted for M.Tech (Industrial Engineering

	and Management), Shishir Acharya (23/IEM/11)
Notification 1:	Late Registration Fee, Hostel Room rent for internship, Hostel cooler rent, Transcript fee (Within 5 years Rs.1500/- & $150 in USD, More than 5 years but less than 10 years Rs.2500/- & $250 in USD, More than 10 years Rs.5000/- & $500 in USD) Additional copies Rs.200/- each & $20 in USD each, I-card fee,Character certificate Rs.500/-.
Notification 2:	Migration Certificate Rs.500/-, Bonafide certificate Rs.200/-, Special certificate (any other certificate not covered in above list) Rs.1000/-,Provisional certificate Rs.500/-, Duplicate Mark sheet (Within 5 years Rs.2500/- & $250 in USD, More than 5 years but less than 10 years Rs.4000/- & $400 in USD, More than 10 years Rs.10000/- & $1000 in USD)

Thank you for choosing SB Collect. If you have any query / grievances regarding the transaction, please contact us

Toll-free helpline number i.e. 1800-1111-09 / 1800 - 1234/1800 2100

Email -: sbcollect@sbi.co.in

Delhi Technological University

(Formerly Delhi College of Engineering)

THE RESULT OF THE CANDIDATE WHO APPEARED IN THE FOLLOWING EXAMINATION HELD IN NOV 2023 IS DECLARED AS UNDER:-

Master of Technology(Industrial Engineering and Management), I-SEMESTER

Result Declaration Date : 04-03-2024

Notification No: 1660

IEM501 : Data Analytics IEM503 : Production & Operation Management IEM5205 : Principles of Managment IEM5305 : Total Quality Management IEM5407 : Product Design & Development

Sr.No	Roll No.	Name of Student	IEM501	IEM503	IEM5205	IEM5305	IEM5407	SGPA	TC	CGPA	Failed Courses
			4.00	4.00	2.00	3.00	4.00				
1	23/IEM/01	RAVI RANJAN	F	F	A+	O	A+	6.46	9	--	IEM501
2	23/IEM/02	AATIF AMEER	O	B+	A+	O	O	9.18	17	9.176	
3	23/IEM/03	MAHESH SAROHA	A+	A	A	A	O	8.71	17	8.706	
4	23/IEM/04	REDDI DUSHYANTH VENKATA SAI KRISHNA	A+	B+	A+	A	A	8.12	17	8.118	
5	23/IEM/05	DIVYANSH	C	C	A	C	B+	5.82	17	5.824	
6	23/IEM/06	RAJENDER	A+	B	A	A	A	7.76	17	7.765	
7	23/IEM/07	PIYUSH KUMAR	A+	B	A+	B	A	7.53	17	7.529	
8	23/IEM/08	ISHAN KOTNALA	C	F	B	C	B	4.18	13	--	IEM503
9	23/IEM/09	LOKESH KUMAR	A+	B+	A+	B+	A	7.94	17	7.941	
10	23/IEM/10	DHRUV SHANKAR SAXENA	A+	A	O	O	A+	9.06	17	9.059	
11	23/IEM/11	SHISHIR	A+	A+	A+	A+	A+	9	17	9	
12	23/IEM/12	MORIE MEYER KOUNA FERRAND	C	P	B+	B	B+	5.65	17	5.647	
13	23/IEM/13	FREDRICK KABWE	C	B	A	B+	B+	6.41	17	6.412	

IEM501 : Data Analytics IEM503 : Production & Operation Management

Sr.No	Roll No.	Name of Student	IEM501	IEM503	SGPA	TC	CGPA	Failed Courses
			4.00	4.00				
14	23/IEM/501	PRAMOD	C	F	2.5	4	--	IEM503

OIC (Results)

Controller of Examination

Note: Any discrepancy in the result in r/o name/roll no/registration/marks/grades/course code/title should be brought to the notice of Controller of Examination/OIC(Results) within 15 days of declaration of result in the prescribed proforma.

Delhi Technological University

(Formerly Delhi College of Engineering)

THE RESULT OF THE CANDIDATE WHO APPEARED IN THE FOLLOWING EXAMINATION HELD IN MAY 2024 IS DECLARED AS UNDER:-

Master of Technology(Industrial Engineering and Management), II-SEMESTER

Result Declaration Date : 16-07-2024

Notification No: 1691

IEM502 : OPERATIONS RESEARCH

Sr.No	Roll No.	Name of Student	IEM502	SGPA	TC	Failed Courses
			4.00			
1	23/IEM/501	PRAMOD	C	5	4	

IEM502 : OPERATIONS RESEARCH   IEM504 : SUPPLY CHAIN MANAGEMENT   IEM5210 : Contemporary Issues in Industrial Engineering and Managment   IEM5304 : International Logistics and Warehouse Managment   IEM5404 : INDUSTRY 4.0 & SMART MANUFACTURING

Sr.No	Roll No.	Name of Student	IEM502	IEM504	IEM5210	IEM5304	IEM5404	SGPA	TC	Failed Courses
			4.00	4.00	2.00	3.00	4.00			
2	23/IEM/01	RAVI RANJAN	O	B+	A+	A	O	8.82	17	
3	23/IEM/02	AATIF AMEER	A+	B+	O	A+	A+	8.65	17	
4	23/IEM/03	MAHESH SAROHA	O	A	A	A+	A+	8.88	17	
5	23/IEM/04	REDDI DUSHYANTH VENKATA SAI KRISHNA	A+	B	A	A	A+	8	17	
6	23/IEM/05	DIVYANSH	B+	P	B	B+	A+	6.65	17	
7	23/IEM/06	RAJENDER	A+	B	A	A	O	8.24	17	
8	23/IEM/07	PIYUSH KUMAR	A	B	A	A	O	8	17	
9	23/IEM/08	ISHAN KOTNALA	P	C	B+	B+	A	6.06	17	
10	23/IEM/09	LOKESH KUMAR	A+	A	A+	A+	A+	8.76	17	
11	23/IEM/10	DHRUV SHANKAR SAXENA	A+	A	O	O	O	9.29	17	
12	23/IEM/11	SHISHIR ACHARYA	A+	A	A	A+	O	8.88	17	
13	23/IEM/12	MORIE MEYER KOUNA FERRAND	A	C	A	A	A	7.29	17	
14	23/IEM/13	FREDRICK KABWE	A	B	A	A	A	7.53	17	

OIC (Results)

Controller of Examination

Note: Any discrepancy in the result in r/o name/roll no/registration/marks/grades/course code/title should be brought to the notice of Controller of Examination/OIC(Results) within 15 days of declaration of result in the prescribed proforma.

Delhi Technological University

(Formerly Delhi College of Engineering)

THE RESULT OF THE CANDIDATE WHO APPEARED IN THE FOLLOWING EXAMINATION HELD IN NOV 2024 IS DECLARED AS UNDER:-

Master of Technology(Industrial Engineering and Management), III-SEMESTER

Result Declaration Date : 12-03-2025

Notification No: 1798

IEM5205 : Principles of Managment IEM5305 : Total Quality Management IEM5407 : Product Design & Development

Sr.No	Roll No.	Name of Student	IEM5205	IEM5305	IEM5407	SGPA	TC	Failed Courses
			2.00	3.00	4.00			
1	23/IEM/501	PRAMOD	B+	C	B+	6.333	9	

IEM601 : MAJOR PROJECT I IEM6201 : E- Commerce IEM6305 : GLOBAL BUSINESS MANAGEMENT IEM6405 : Advanced Operation Research

Sr.No	Roll No.	Name of Student	IEM601	IEM6201	IEM6305	IEM6405	SGPA	TC	Failed Courses
			3.00	2.00	3.00	4.00			
2	23/IEM/01	RAVI RANJAN	A+	A+	A+	O	9.333	12	
3	23/IEM/02	AATIF AMEER	A+	A+	A+	B+	8.333	12	
4	23/IEM/03	MAHESH SAROHA	A	O	A+	O	9.250	12	
5	23/IEM/04	REDDI DUSHYANTH VENKATA SAI KRISHNA	A+	A+	A+	A+	9.000	12	
6	23/IEM/05	DIVYANSH	A+	A	B+	B	7.333	12	
7	23/IEM/06	RAJENDER	O	A+	A	A+	9.000	12	
8	23/IEM/07	PIYUSH KUMAR	A+	A+	B+	B+	7.833	12	
9	23/IEM/08	ISHAN KOTNALA	A+	A+	A+	B+	8.333	12	
10	23/IEM/09	LOKESH KUMAR	A+	A+	A+	B+	8.333	12	
11	23/IEM/10	DHRUV SHANKAR SAXENA	O	O	O	O	10.000	12	
12	23/IEM/11	SHISHIR ACHARYA	O	O	O	O	10.000	12	
13	23/IEM/12	MORIE MEYER KOUNA FERRAND	A+	A	A+	A	8.500	12	
14	23/IEM/13	FREDRICK KABWE	A+	A+	A+	A+	9.000	12	

OIC (Results)

Controller of Examination

Note: Any discrepancy in the result in r/o name/roll no/registration/marks/grades/course code/title should be brought to the notice of Controller of Examination/OIC(Results) within 15 days of declaration of result, in the prescribed proforma.

SHISHIR ACHARYA

CONTACT NUMBER – 7011020052 **EMAIL-ID – shishir_23iem11@dtu.ac.in**
ROLL NUMBER – 23/IEM/11

EDUCATION

M.Tech.(Industrial Engineering & Management)	2023-2025	Delhi Technological University, New Delhi	8.94 CGPA
B.Tech. (Mechanical Engineering)	2018-2022	Sharda University, Greater Noida	9.23 CGPA
HSEB Board (Class XII)	2016-2018	Prasadi Academy Secondary School, Nepal	81%
SLC Board (Class X)	2016	G. S. Niketan Ma Vi, Nepal	91.25%

ACADEMIC PROJECTS

- **Supply Chain Dashboard**
  - Created an interactive **Supply Chain Dashboard** using Power BI that provided insights into **total sales**, **order distribution by category**, **shipping performance**, and **sales patterns by year**
  - Presented **key KPIs** such as **total sales (25.50M)**, **shipping mode performance**, and **country-wise order distribution** for data-driven decision-making.
  - **Visualized historical sales patterns from 2015 to 2018** and **forecasted sales trends for the next quarter**.
- **Financial Resource Analysis of DTU |** January 2024 – May 2024
  - Conducted a financial analysis of DTU for **AY 2021-22 and 2022-23**, comparing its resources with **IIT Madras and Oxford University**.
  - Analyzed a budget exceeding **INR 900 crores**, identifying allocation inefficiencies and potential cost-saving opportunities.
  - Evaluated key financial metrics and trends to identify **DTU's strengths and improvement areas**, and proposed recommendations.
- **Water Quality Classification using python|** November 2023 – December 2023
  - Developed a **logistic regression model** to classify water quality into distinct categories based on chemical composition.
  - Processed a dataset containing **8,000 water samples** with **20 different chemical attributes**.
  - Achieved accuracy of **90%** and used cross validation for calculating accuracy.

PUBLICATIONS

- **Patents**
  - 342726-001,” Steering Assembly for Vehicle”, March 25, 2022
  - 344314-001,” Pedal Operated Forklift”, July 23, 2021
  - 345386-001,” Half Steering Wheel”, August 13, 2021
  - 345387-001,” Two Spoke Steering Wheel”, August 13, 2021
  - 345633-001,” Muffler Assembly”, September 10, 2021

ACADEMIC ACHIEVEMENTS

- **Secured a fully funded scholarship** from the Government of India for master’s studies.
- **Achieved the highest academic ranking** out of 52 students in the department, earning the prestigious **Vice-Chancellor's Gold Medal** at Sharda University.
- **Received a merit-based scholarship** for ranking among the top 5% of students during bachelor’s studies, recognizing exceptional academic performance.

ADDITIONAL

Technical Skills: Data Analysis, Data Visualization, Supply Chain Management

Tools: Power BI, SQL, Python, MS Office

Soft Skills: Teamwork, Creativity, Communication, Problem Solving