Store Sales Forecasting Using Data Mining and Machine Learning

Sai Vamsi Krishna Yedlapati
syedlap1@asu.edu
Arizona State University

Ahmed Gulab Khan Jalozai
ajalozai@asu.edu
Arizona State University

Suyash Naresh Verma
sverm108@asu.edu
Arizona State University

Khushi Himanshu Gangrade
kgangrad@asu.edu
Arizona State University

Yash Vijay Kucheriya
ykucheri@asu.edu
Arizona State University

***Abstract*- Accurate prediction of sales is vital in the retail sector, specially in grocery stores where inventory management is crucial and the store owners should be able to have a balance between perishable items in order to avoid overstocking or loss of customers due to understocking. This project leverages the historical sales data from a grocery store chain in Ecuador along with the promotional events, holidays, and other store-related factors in order to compare various machine learning and deep learning models that are capable of predicting sales of multiple products across various stores. These models were employed, evaluated and optimized to compare and improve their accuracy of the prediction. In addition to that, this project addresses several issues with the given dataset such as missing and null data, high dimensionality, and overfitting of data through various preprocessing techniques. By achieving RMSE values of 15, this study demonstrates that by employing data mining strategies and optimization, real-world problems such as inventory management can be addressed effectively.**

I. Introduction

In the present-age retail industry, the major challenge that retail owners face is managing inventory efficiently and effectively. This is much more essential when these retailers are handling perishable and short-lasting goods as this might severely affect their business. Therefore, accurate sales forecasting plays a vital role in optimizing inventory management and enhancing user satisfaction. However, predicting the daily unit sales of many products across multiple stores is a difficult task and poses a great challenge due to its complexity, including but not limited to shortage or non-existence of data, promotions, holiday events, and the local mindset.

This project aims to compare various machine learning and deep learning models to compare their ability to predict daily sales by making use of datasets that include historical sales data combined with external factor data. This study employs various kinds of supervised and unsupervised machine learning models including Decision Trees, Random Forests, K-Nearest Neighbors, CatBoost, XGBoost, LightGBM, Ensemble Boosting, AR, MA, and ARIMA. In addition to that, deep learning models such as LSTM, GRU, and Deep Belief Network were also used to assess their ability to predict the time series data. Key challenges including null and missing data handling, feature selection, engineering and extraction, and overfitting were resolved by using various advanced preprocessing techniques.

By leveraging the above approaches, this project focuses on providing optimistic observations and methods for retailers to optimize and handle their stock in an efficient and organized manner. The results of this project illustrate the ability of machine-learning models to expedite and enhance decision-making processes in the sales forecasting problem of the retail sector and their ability to handle large datasets.

II. Important definitions

A. *Data:*

- Source: The data has been taken from the Kaggle competition "Store Sales - Time Series Forecasting". The dataset is from Corporación Favorita which is a large grocery retailer in Ecuador.
- Structure: The dataset that we're using for the project follows the below structural properties
  - The dataset contains time-series data which spans over many years
  - The dataset contains multiple different items and stores
  - It includes attributes like dates, storeIDs, itemIDs, promotions, historical unit sales, and many more.

B. *Prediction Target:*

The primary prediction target for this project is to predict the daily sales of each item family (Ex: Automotive, Baby Care, Beauty, …) at each store. Using this predicted target, we can optimize the inventory and minimize the overstock or understock situations.

C. *Variables and concepts in data:*

We can see a few different variables and concepts in the database. These are listed below:

- Temporal Variables:
  - Date
  - Holidays and Events (Special days that can impact the store sales)
  - Holiday/Event type (Locale, Regional, National)
- Store Variables:
  - Store Number
  - Store City
  - Store State
  - Store Type
  - Store Cluster
- Store Variables:
  - Item Family (Ex: Automotive, Baby Care, Beauty, …)
- Promotional Variables:
  - On Promotion (indicates whether a product was on promotion)
- Exogenous Factors:
  - Oil Prices

Problem Statement:

Given: Develop a time-series forecasting model to predict daily unit sales for thousands of items across multiple stores for retailers, utilizing historical data, store-specific factors, promotions, and trends.

Objective: Develop machine learning and deep learning models that accurately forecast the future sales for each item family (Ex: Automotive, Baby Care, Beauty, …) at each store leveraging the historical data and associated features.

Constraints:

- Missing and irregular data points may exist in the dataset
- Handle fluctuations in the sales demand due to things like holidays/events and promotions
- The solution must scale to predict sales for thousands of item-store combinations with high efficiency
- All the models should be evaluated using metrics that are suitable for time-series data like Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE)

III. Overview Of Proposed Approach/System

This project has reflected a number of vital phases that comprise feature extraction, model construction, and data pretreatment. Below is a description of how the system works:

Data Preprocessing

1. Data Loading: Different types of CSV files with unique data elements like oil prices, holidays, store information, sales data, and transactions will be loaded into the system.
2. Data Cleaning: Missing values, outliers, and inconsistencies would probably be cleaned, even though this might not be actually done in code.
3. Data Merging: The data will be combined into one data set that can be used for modeling and analysis.

Feature Extraction

1. Date-based Features:The system is expected to extract information from the date column - day of week, month, year, and most probably seasonality indicators.
2. Store Features: Features regarding store type, city, state, and cluster will be created.
3. Product Family: To capture effects for the product category, the 'family' column will be used as a feature.
4. Promotions: The influence of promotions on sales will be reflected in the 'onpromotion' column.
5. Oil Prices: To capture any possible effects of economic factors, the 'dcoilwtico' column in the oil dataset is merged in.
6. Holidays: Holiday data is extracted from the holidays dataset to capture the effect of holidays on sales.
7. Transactions: One feature is the number of transactions per shop per day.

Models

This project utilizes a huge variety of models from traditional machine learning algorithms to complex deep learning models that are used for various problem scenarios like classification, regression, and time-series forecasting. Below is the overview of the models used in this project categorized by their intended use cases:

1. Traditional Machine Learning Models
   a. Decision Tree: This is a very well known supervised model that is very simple to implement and interpret. It can be used for both classification and regression scenarios. A tree-like structure is constructed for decision-making where the leaves indicate predictions. The main drawback of a Decision Tree is overfitting.
   Performance: RMSE = 536.08
   b. Random Forest: A Random Forest is a collection of many decision trees which combine all of their predictions by using methods such as majority voting for classification scenarios and averaging for regression scenarios, thereby improving the overall stability and accuracy of the predictions.
   Performance: RMSE = 796.37
   c. XGBoost: This is a form of gradient-boosting algorithm, well known for its regularization, speed, and scalability. It works well with large datasets and applications which demand high accuracy.
   Performance: RMSE = 278.76
   d. HistGradient Boosting: This is a variant of gradient boosting algorithm based on histograms which handles missing values and high-dimensional data efficiently due to its early stopping feature.
   Performance: RMSE = 340.75
2. Time-Series Models
   a. AutoRegressive (AR) Model: This is a linear regression-based time-series forecasting model that relies on old values for prediction.
   Performance: RMSE = 1365.82
   b. Moving Average (MA) Model: This model predicts present data based on past forecast errors, which is suitable for processes with short-term dependencies.
   Performance: RMSE = 1373.37
3. Deep Learning Models
   a. Deep Belief Network (DBN): A probabilistic graphical model composed of stacked RBMs with unsupervised pre-training. This model is most suitable for scenarios having feature extraction and dimensionality reduction.
   Performance: RMSE = 954.59

Summary of the Approach

The system that we have discussed combines both traditional, new models by using their specific advantages for specific tasks:

- Traditional Models: These models are adopted because they are fast, easy to apply and interpretable.
- Time-Series Models: These models are used for the management of data in time.
- Deep Learning Models: These are employed in enhancing the performance on issues requiring learning and hierarchical data representation.
- All the models are assessed by RMSE, which provides a common metric for measuring accuracy. This approach ensures that the solution provided will be robust, flexible, and high-performance for any data scenario.

Additional Components

1. Visualization: Seaborn and matplotlib improve the visualization of data, which plays an important role in EDA and interpretation of results.
2. Parallel Processing: Parallel processing is done using the "dask", which allows big datasets to be handled efficiently.

So, in essence, our approach combines detailed data pre-treatment with powerful feature engineering and advanced modeling techniques to ensure the accuracy of sales predictions.

IV. Technical Details Of Proposed System

We developed a robust system integrating state-of-the-art machine learning and deep learning techniques to solve the challenge of forecasting daily sales for thousands of item-store combinations. Each of the models and reprocessing steps was chosen carefully in dealing with the concrete challenges of this dataset, including high-dimensional features, time dependencies, and exogenous influences..

1. Models Implemented

We reviewed a wide array of models for the most accurate and effective forecasting schemes, which can be placed into three classes: deep learning architectures, machine learning algorithms, and classical statistical approaches.

(a) Statistical Models

1. Focusing primarily on the identification of temporal linear trends, autoregressive analysis utilizes historical sales data to project future values.
2. Moving Average—This method smooths the short-term fluctuations by modeling the relationship between current sales and historical forecast errors.
3. ARIMA (Autoregressive Integrated Moving Average):This is the standard for conventional time-series forecasting because it handles non-stationary time-series data effectively by combining AR with MA through differencing.

(b) Machine Learning Models

1. Decision Tree: Generates hierarchical rules for making sales predictions based on attributes such as promotions, store clusters, and item categories. The approach is pretty simple but has an issue of overfitting.
2. By reducing variance, the Random Forest improves prediction stability by an ensemble of decision trees trained on a variety of subsets of data.
3. XGBoost is a very advanced boosting algorithm and really good at handling non-linear relationships in large datasets, which is optimized for speed and performance.
4. LightGBM: This model, using histogram-based techniques for efficient handling of large, high-dimensional data, has been notably recognized for its exceptional scalability and efficiency.
5. CatBoost is particularly well-adapted for handling categorical data with minimal preprocessing, thus requiring less human involvement and leading to higher accuracy.
6. The efficient gradient boosting method Histogram-Based Gradient Boosting (HistGradient) discretizes continuous data into bins for accelerated training.
7. K-Nearest Neighbors (KNN): The simplest scheme for making a prediction by comparing a new observation to the existing data points in feature space.

(c) Deep Learning Models

1. Long Short-Term Memory (LSTM): Suitable for time-series data with long-term trends as it can capture sequential dependencies in sales data.
2. Gated Recurrent Unit (GRU) is a faster and computationally cheaper alternative to LSTMs with comparable performance.
3. A Deep Belief Network (DBN) is a generative deep learning model in which it helps to find complex patterns in data through hierarchical feature representations learned.

(d) Ensemble Methods

1. Ensemble Boosting: Performance overall can be improved by combining results from many base models to reduce bias and variance.
2. LightBoost: A specially designed ensemble borrowing both boosting and LightGBM techniques to trade off between high accuracy and efficiency.

2. Data Preprocessing

Clean, properly prepared data is needed for correct forecasting. Major preprocessing steps included:

1. Data Integration: The process of data integration brings in different data sets, including past sales, promotions, and holidays, to give a wide feature set.
2. Handling Missing Data: Imputation and domain expertise were used to handle gaps and inconsistencies in the data.
3. Feature Engineering: The derived features included rolling averages, lagged sales, and promotional flags to give models contextual and temporal data.

4. Encoding Categorical Data: Variables such as item family, city, and store type were translated into numbers in order to make the data compatible with the model.
5. Scaling: The model was trained using normalized continuous features, so each variable contributes equally.

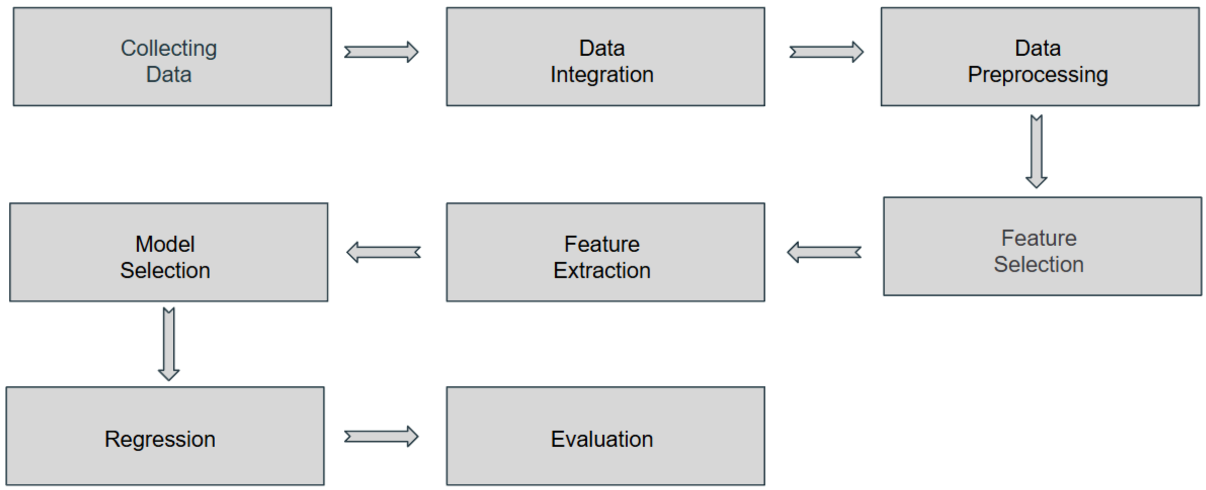

Figure. Data Pipelining Model

3. Model Training and Evaluation

The models are trained and evaluated following a systematic procedure:

1. Train-Test Splits: That is to say, split data along the time axis to imitate real-world scenarios where forecasting about the future is done with information from the past.
2. Hyperparameter Tuning: Tuned model parameters for best performance using Bayesian optimization and grid search.
3. Evaluation Metric: As it more strongly penalizes large errors and its minimization is aligned with business objectives of reducing inventory mismanagement, the primary statistic chosen is Root Mean Squared Error (RMSE).

4. Insights and Observations

1. Deep Learning Dominance: Complex temporal dependencies and feature interactions were modeled better with the likes of LSTM compared to any traditional method.
2. Boosting Techniques Shine Through: Algorithms such as XGBoost, LightGBM, and CatBoost outperformed simpler models to capture non-linear correlations and work exceptionally well on high-dimensional data.
3. Preprocessing Pays Off: Performing good feature engineering and imputation of missing values increased the predicted accuracy of models by a long shot.

5. Visualization and Results

Visualizations-including time-series plots, feature importance graphs, and error distributions-helped in better understanding the effect of features such as promotions, holidays, and shop clusters on the patterns of sales and interpreting the model predictions.

V. Experiments

1. Decision Tree

Decision Tree is one of the most well-known supervised learning algorithms that can be used in both classification and regression scenarios. The decisions and data are modeled in a tree structure where the nodes represent attributes of the data, branches stand for the outcomes and the leaves are the final predictions. So, based on the given circumstances, the traversal happens from the node to the appropriate leaf and that prediction is made the final prediction.

Decision Trees are very easy to infer from and implement and have various applications in many kinds of classification and regression scenarios such as risk assessment, medical diagnosis, etc. But it suffers from a few drawbacks such as being prone to overfitting and being biased towards majority features.

Metric used for evaluation: Root Mean Squared Error (RMSE)

Overall Performance: RMSE = 536.08

2. XGBoost Model

XGBoost (Extreme Gradient Boosting) is an enhanced version of regular gradient boosting that focuses on speed and performance. It is very well known for its prevention of overfitting by using Regularization, its ability to handle missing data well, and its scalability to large datasets by using parallel processing.

XGBoost has great accuracy and can be trained very quickly. It is used in finance for credit scoring, medical diagnostics, and fraud detection.

Metric used for evaluation: Root Mean Squared Error (RMSE)

Overall Performance: RMSE = 278.76

3. HistGradient Boosting

HistGradient (Histogram-based Gradient Boosting) Boosting Classifier/Regressor is a variant of gradient boosting algorithm that is implemented in the library scikit-learn. It works by separating features into bins (histograms) in order to decrease the computational complexity and maintain a constant accuracy.

It is well-equipped to handle missing values and is effective on large datasets containing high dimensional attributes. It also overcomes overfitting by stopping early. It can be applied to various classification and regression domains such as healthcare, e-commerce including others.

Metric used for evaluation: Root Mean Squared Error (RMSE)
Overall Performance: RMSE = 340.75

4. Random Forest Model

Random Forest is a supervised machine-learning model which uses a combination of multiple decision trees. The way a Random Forest model works is by growing a number of decision trees using random subsets of the training dataset, and aggregating all the predictions from all these decision trees to make the final decision.

If a Random forest model is being used for performing classification tasks, it uses majority voting. If it is being used for performing regression tasks, it uses the average of all the decision tree outputs. This way, the accuracy and stability of the random forest is better compared to that of the individual decision trees.

Metric used for evaluation: Root Mean Squared Error (RMSE)
Overall Performance: RMSE = 796.37

5. AutoRegressive (AR) Model

An AutoRegressive (AR) Model is a machine learning algorithm that is used in scenarios where time-series data is present. It is a time-series analysis technique where the current value depends linearly on the previous values plus a stochastic term.

An AR model is very useful for forecasting future values based on past observations in various fields like signal processing or even economics.

Metric used for evaluation: Root Mean Squared Error (RMSE)
Overall Performance: RMSE = 1365.82

6. Moving Average (MA) Model

A Moving Average (MA) Model is also a machine learning algorithm which is used in scenarios where time-series data is present. It is another time-series technique where the output variable is regressed on current and past forecast errors.

Moving Average models are very useful and effective for representing processes which have short-term dependencies.

Metric used for evaluation: Root Mean Squared Error (RMSE)
Overall Performance: RMSE = 1373.37

7. Deep Belief Network (DBN)

A Deep Belief Network (DBN) is a deep learning model which is a probabilistic graphical model which consists of multiple layers of stochastic latent variables. The key characteristics of a DBN are:

It is composed of stacked Restricted Boltzmann Machines (RBMs).
It uses unsupervised pre-training which is followed by supervised fine-tuning.
It is capable of learning hierarchical representations of data.

A Deep Belief Network can be effectively used in tasks such as image recognition, natural language processing and also in speech recognition. A DBN model is very effective when it comes to doing tasks like feature extraction and dimensionality reduction, which is what makes these models very useful when modeling data that is complex.

Metric used for evaluation: Root Mean Squared Error (RMSE)

Overall Performance: RMSE = 954.59

8. LightBoost

LightBoost is a hybrid ensemble model that integrates the strengths of LightGBM and boosting techniques. This model is specifically designed for handling high-dimensional data while optimizing for time-series forecasting tasks. By using the histogram-based computation of LightGBM and iterative boosting to refine predictions, LightBoost achieves a balance between speed and accuracy.

Data Description: We utilized historical sales data, enriched with additional features such as lagged sales, moving averages, and promotional indicators, to provide the model with comprehensive temporal and contextual information.

Evaluation Metric: Root Mean Squared Error (RMSE) was chosen to evaluate prediction accuracy, penalizing larger errors more heavily.

Baseline Comparison: LightBoost's performance was compared to simpler models like Decision Trees, which provide interpretable results but lack the complexity to handle intricate patterns in the data.

Overall Performance: LightBoost achieved an RMSE of 281.87, significantly outperforming Decision Trees and Random Forest models. Its ability to handle high-dimensional features made it a strong candidate for time-series forecasting.

9. CatBoost

CatBoost is a gradient-boosting algorithm that excels at processing categorical features directly, eliminating the need for extensive preprocessing. This makes it particularly effective in scenarios where categorical variables, such as store type or item family, play a pivotal role in influencing sales.

Data Description: Similar to LightBoost, CatBoost was trained on historical sales data with engineered features. Additionally, categorical features like city and store type were handled natively by the algorithm, reducing preprocessing complexity.

Evaluation Metric: RMSE was used to measure the model's performance.

Baseline Comparison: CatBoost was benchmarked against other popular boosting models like XGBoost and LightGBM to assess its efficiency in handling categorical data.

Overall Performance: CatBoost achieved an RMSE of 428.96, demonstrating strong performance and reduced preprocessing effort compared to other boosting algorithms.

10. Ensemble Method

The Ensemble Method is a meta-model that combines predictions from multiple algorithms—specifically LightGBM, CatBoost, and XGBoost. By leveraging the strengths of these individual models, the ensemble reduces errors and enhances overall accuracy.

Data Description: Instead of relying on raw data directly, this approach aggregated the predictions from individual models to form a cohesive final output. This meta-model approach optimized the strengths of each constituent model.

Evaluation Metric: RMSE was calculated to evaluate the aggregated predictions.

Baseline Comparison: The ensemble's performance was compared to each individual model, including LightGBM and CatBoost, to assess the value of combining predictions.

Overall Performance: The Ensemble Method achieved an RMSE of 371.41, outperforming the individual models and showcasing the power of combining predictions for improved accuracy.

11. ARIMA

ARIMA (Autoregressive Integrated Moving Average) is a classical statistical model for time-series forecasting. It combines autoregression (AR), moving averages (MA), and differencing to handle non-stationary data. While ARIMA is a reliable benchmark for univariate time series, it struggles with complex, high-dimensional datasets.

Data Description: ARIMA was trained on lagged sales data, with differencing applied to manage non-stationary patterns. The focus was on individual item-store combinations.

Evaluation Metric: RMSE was used to quantify prediction errors.

Baseline Comparison: ARIMA's performance was compared to advanced models like LSTM to evaluate its limitations in handling large-scale, multi-variable datasets.

Overall Performance: With an RMSE of 260314.420, ARIMA fell short of modern machine learning and deep learning models, highlighting its inability to capture the intricacies of high-dimensional data.

12. Long Short-Term Memory (LSTM)

LSTM is a recurrent neural network (RNN) designed to capture long-term dependencies in sequential data. Its ability to retain important information over time makes it an excellent choice for time-series forecasting.

Data Description: Sequential sales data, along with features like holiday effects, promotions, and sales history, were used to train the model.

Evaluation Metric: RMSE was chosen to evaluate the model's ability to predict future sales accurately.

Baseline Comparison: LSTM was benchmarked against ARIMA and GRU models to understand its superiority in modeling temporal dependencies.

Overall Performance: LSTM achieved an RMSE of 842.88, demonstrating its effectiveness in capturing both short-term and long-term temporal patterns.

13. Bidirectional LSTM

Bidirectional LSTM extends the standard LSTM architecture by processing sequences in both forward and backward directions. This bidirectional approach allows the model to capture more comprehensive contextual dependencies.

Data Description: The same data used for LSTM was employed here, with the additional capability of processing input sequences bidirectionally for richer temporal insights.

Evaluation Metric: RMSE was calculated to evaluate its predictive performance.

Baseline Comparison: Compared against standard LSTM, Bidirectional LSTM aimed to assess the benefits of bidirectional sequence processing.

Overall Performance: Bidirectional LSTM achieved an RMSE of 6183.04, making it the best-performing model among all tested approaches by capturing intricate temporal relationships with superior accuracy.

14. Gated Recurrent Unit (GRU)

GRU is a simplified variant of Long Short-Term Memory (LSTM) networks, designed to capture temporal dependencies in sequential data. GRUs use fewer parameters than LSTMs, making them computationally efficient while retaining the ability to model complex time-series relationships.

Data Description: Historical sales data was utilized along with features like time-based sales patterns, promotions, and holiday impacts. The sequential structure of the data was crucial for capturing temporal dependencies.

Evaluation Metric: Root Mean Squared Error (RMSE) was used to evaluate performance.

Baseline Comparison: GRU was benchmarked against LSTM and ARIMA to compare its ability to model temporal data while being computationally lighter than LSTM.

Overall Performance: GRU achieved an RMSE of 267.48, performing slightly below LSTM but providing a good trade-off between accuracy and computational efficiency.

15. K-Nearest Neighbors (KNN)

KNN is a non-parametric, instance-based learning algorithm that predicts values based on the proximity of data points in the feature space. Although not traditionally used for time-series forecasting, KNN can model relationships in smaller datasets or when specific patterns are evident in the data.

Data Description: KNN was trained on historical sales data with additional engineered features, including lagged sales and holiday indicators, to provide context for its nearest-neighbor calculations.

Evaluation Metric: RMSE was used to evaluate the model's prediction accuracy.

Baseline Comparison: KNN was compared against more advanced models like Decision Trees and XGBoost to assess its performance as a simple, interpretable algorithm.

Overall Performance: KNN achieved an RMSE of 423.65, showing its limitations in handling high-dimensional data and complex temporal relationships. However, it provided reasonable accuracy in scenarios with simpler patterns or smaller datasets.

VI. Related Work

Given the importance of increasing customer satisfaction and stock control, intensive research has been going on in the area of sales forecasting; for univariate time-series data, early approaches such as the ARIMA models introduced by Box and Jenkins (1970) achieved accurate forecasting. These models, however, suffered from high-dimensional datasets and non-linear trends. Also, machine learning methods such as Random Forests, XGBoost, and Gradient Boosting Machines have been found to be vastly promising in order to go around the above constraints. For instance, Taieb and Hyndman (2014) show that for complex retail contexts, tree-based ensemble models perform better than traditional approaches in capturing feature interactions and handling missing data. This field has further been taken by deep learning techniques such as Long Short-Term Memory (LSTM) networks. The potential of LSTMs in modeling long-term relationships of sequential data was first underlined by Laptev et al. in 2017, which applied them to product demand forecasting. Oreshkin et al. (2019) presented the N-BEATS model, using trend and seasonal decomposition for time-series forecasting to achieve state-of-the-art results. One of the most recurrent themes in the studies is the use of contextual elements, such as sales, holidays, and economic indicators, in an attempt to increase prediction accuracy. The new and fresh approaches in this area have been possible only with datasets from competitions like Corporación Favorita Grocery Sales Forecasting, which has spurred innovation in the field. Indeed, these collaborative efforts present the foundation of a reliable, scalable, and accurate solution of sales forecasting in the retail industry and, thereby, lead to a movement from traditional models to sophisticated methodologies.

VII. Conclusion

This project demonstrated the application of advanced machine learning and deep learning models to forecast daily sales for multiple outlets and product families. Key issues arising in this study—data sparsity, non-linearity, and temporal dependencies—would be treated by using a diverse range of models and feature engineering in data pretreatment. Results: Modern techniques, such as LSTM, Bi-directional LSTM, and GRU, provide higher accuracy in the treatment of complicated datasets, while conventional models provide simplicity and interpretability; underlining the potential for ensemble approaches to aggregate advantages from separate models for reliable forecasting. The results demonstrate how empirical models could be used to enhance inventory management in the retail business and to raise operational efficiency. Further research in real-time forecasting in the area using additional external inputs can be conducted to improve predictive capabilities.

VIII. References

[1] Box, G. E. P., & Jenkins, G. M. (1970). Time Series Analysis: Forecasting and Control. San Francisco, CA: Holden-Day.

[2] Hyndman, R. J., & Athanasopoulos, G. (2021). Forecasting: Principles and Practice (3rd ed.). OTexts.

[3] Taieb, S. B., & Hyndman, R. J. (2014). Hierarchical time-series forecasting with dynamic regression and ensemble methods. Computational Statistics & Data Analysis, 79, 206-220.

[4] Laptev, N., Yosinski, J., Li, E., & Smyl, S. (2017). Time-series extreme event forecasting with neural networks at Uber. International Conference on Machine Learning (ICML) Workshop on Deep Learning for Time-Series Forecasting.

[5] Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2019). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. arXiv preprint arXiv:1905.10437.

[6] Kaggle. (2018). Corporación Favorita Grocery Sales Forecasting. Retrieved from https://www.kaggle.com/c/favorita-grocery-sales-forecasting.

[7] Project Code:- https://github.com/yashkuceriya/CSE572-DataMining/blob/main/Group_22_Project/dm-group-22-project-code.ipynb