**Group Number: COMPARATIVE ANALYSIS BETWEEN DIFFERENT TIME-SERIES FORECASTING MODELS**

Your submissions:

* Group number\_Report.pdf
* Group number\_Codes.ipynb (with necessary comments)
* Group number\_Codes.html (converted from the ipynb file above)
* Group number\_Slide.pdf (your presentation slide)

Notes

* No extension to the deadline
* Each team can only submit one copy by a single member, just list all of your members in the report
* use RED font for the parts that you revised according to the feedbacks in your presentation

|  |  |  |
| --- | --- | --- |
| First Name | Last Name | Email address |
| **ABDUL** | **KARIM ALI** | **aali62@hawk.iit.edu** |
| **ARYAN** | **THAKRAN** | **alnu5@hawk.iit.edu** |
| **RAHUL** | **POOJARI** | **rpoojari1@hawk.iit.edu** |
| **SHRAYAN** | **CHAUDHURI** | **schaudhuri@hawk.iit.edu** |

Table of Contents

[**1. Introduction** 2](#_Toc56029461)

[**2. Data** 2](#_Toc56029462)

[**3. Problems and Solutions** 2](#_Toc56029463)

[**4. KDD** 2](#_Toc56029464)

[4.1. Data Processing 2](#_Toc56029465)

[4.2. Data Mining Methods and Processes 2](#_Toc56029466)

[**5. Evaluations and Results** 2](#_Toc56029467)

[5.1. Evaluation Methods 2](#_Toc56029468)

[5.2. Results and Findings 2](#_Toc56029469)

[**6. Conclusions and Future Work** 2](#_Toc56029470)

[6.1. Conclusions 2](#_Toc56029471)

[6.2. Limitations 2](#_Toc56029472)

[6.3. Potential Improvements or Future Work 2](#_Toc56029473)

# **1. Introduction**

# Forecasts form the bedrock of decision-making in diverse spheres. Governments project economic growth, scientists anticipate future population trends, and businesses, particularly brick-and-mortar grocery stores, rely on forecasting to oversee product demand. Accurate sales predictions are pivotal for grocers navigating the delicate balance of inventory management. Overestimation leads to surplus perishable stock, while underestimation results in depleted items, revenue loss, and dissatisfied customers. Leveraging machine learning in forecasting presents a promising avenue for retailers to optimize operations, ensuring optimal stock levels to meet customer needs while streamlining operations.

# Conventional subjective methods in retail forecasting often lack empirical backing and pose challenges in automation. The complexity amplifies as retailers expand into new territories with distinctive demands, introduce novel products, grapple with ever-evolving seasonal preferences, and navigate unpredictable marketing strategies.

# Our project, "COMPARATIVE ANALYSIS BETWEEN DIFFERENT TIME-SERIES FORECASTING MODELS," utilizes data from a Kaggle competition focusing on store sales time series forecasting. This project harnesses machine learning and predictive analytics to confront the intricate challenges encountered by retail enterprises. It underscores the importance of deciphering sales patterns as guiding beacons, offering indispensable insights crucial for merchants navigating the intricacies of the retail sphere.

# At the core of our project lies the ambition to employ a range of advanced time series forecasting models to extract actionable insights from the provided sales dataset. Our comprehensive exploration and comparison of these models aim to discern the most effective techniques for accurately predicting sales trends. This comparative analysis serves as a compass for businesses, enabling them to make informed decisions and craft resilient strategies within the dynamic retail environment.

# Aligned with the Kaggle competition's commitment to fostering innovation and excellence in predictive modeling, our project endeavors to deliver robust solutions, equipping businesses with the foresight necessary to excel in an ever-evolving marketplace.

# **2. Data**

# The dataset used for training and testing the model comprises a substantial volume of data, totaling 3,000,888 rows. It includes diverse information encompassing dates, store specifics, product details, promotion indicators, and sales figures. Supplementary files complement the dataset, offering additional information crucial for model development.

# train.csv: This file constitutes the training data, presenting time series data featuring key attributes such as store\_nbr, family, onpromotion, and the target variable, sales.

# store\_nbr: Identifies the store where the products are sold.

# family: Represents the product type sold.

# sales: Indicates the total sales for a particular product family at a specific store on a given date. It's noteworthy that fractional values are plausible, considering products might be sold in fractional units (e.g., 1.5 kg of cheese).

# onpromotion: Denotes the count of items in a product family that were under promotion at a store on a given date.

# stores.csv: This file contains metadata about stores, comprising information on city, state, type, and cluster.

# cluster: Represents a grouping classification of similar stores.

# oil.csv: Daily oil price data is included, encompassing values within both the train and test data timeframes. This dataset is especially relevant since Ecuador, the context for this project, heavily relies on oil as a crucial element in its economy. The economic health of the country is highly susceptible to fluctuations in oil prices.

# 4. holidays\_events.csv: This dataset encompasses holidays and events classified into three types: local events celebrated city-wide, regional holidays celebrated state-wide, and national holidays observed nationwide. This information on events and holidays could potentially influence sales patterns and should be considered when building models.

# **3. Problems and Solutions**

# Addressing the challenges encountered, we've also explored potential solutions:

# Data Aggregation and Integration Solutions: Overcoming the complexity of assembling a cohesive dataset from disparate CSV files necessitated meticulous data processing techniques. To streamline this, we implemented robust data merging strategies, parsing through raw data to efficiently combine and integrate information from multiple sources. The solution involved meticulous extraction and integration methodologies for holiday data, ensuring a seamless merge with the training set while deciphering its impact on sales.

# Feature Importance Extraction Strategy: Recognizing the significance of feature importance, the utilization of the XGBoost algorithm was instrumental. Leveraging XGBoost enabled the extraction of critical feature importance, providing valuable insights for subsequent feature selection methods. This approach empowered the enhancement of model performance by focusing on influential variables and eliminating redundant or less impactful ones.

# Temporal (Time Series) Cross-Validation Approach: Addressing disruptions caused by traditional k-fold cross-validation methods, we explored innovative techniques such as temporal cross-validation. Solutions like Forward Chaining or Walk-Forward Validation, Rolling Window Cross-Validation, and Time-based Splits were evaluated and implemented. These alternatives ensured the preservation of time-related patterns in the dataset, fostering a more consistent validation framework.

1. Seasonality Issue: - One significant challenge of the ARIMA model is effectively handling seasonality in time series data. ARIMA is designed to capture linear trends but struggles with complex seasonal patterns. Seasonal variations may result in residual errors that the model fails to account for, leading to inaccurate predictions. To address this, more advanced versions like Seasonal ARIMA (SARIMA) or alternative models incorporating seasonality, such as SARIMA-X, may be preferred. These models integrate seasonal components, allowing for a more accurate representation of periodic patterns, making them suitable for datasets with pronounced seasonal fluctuations that traditional ARIMA may inadequately capture.

# **4. KDD**

## 4.1. Data Processing

In the data processing phase, we initiated by conducting a comprehensive check for null values across all dataframes. Addressing null values within the 'oil' dataframe, we opted for a forward-fill approach, assuming continuity in oil prices between consecutive days. Subsequently, the 'oil' dataframe underwent merging with the training dataframe, employing a left join operation based on the date column.

Delving into insights from the 'holidays.csv' dataset, we strategically segmented the dataset into three distinct dataframes: one each for national, regional, and local holidays. These segmented dataframes were then seamlessly integrated with the training datasets via a left join, utilizing date and locale as the primary merging keys. Introducing three binary features denoting national, regional, or local holidays, this approach aimed to unravel potential patterns showcasing the impact of sales concerning different holiday types.

Expanding the dataset integration process, we merged the 'stores' data frame with the training data using a left join operation, aligning based on the store number. Subsequently, the 'transactions' dataframe was integrated into the dataset through a left join, incorporating date and store number as merging keys.

Post the meticulous data processing steps, our final training data frame culminated with 3,000,888 rows and 15 columns, paving the way for subsequent modeling and analysis.

## 4.2. Data Mining Methods and Processes

To enhance the dataset's time-related information, we derived additional date-time features such as the day of the month, month, year, and day of the week. This approach aimed to provide a granular understanding of timestamps, as direct utilization of date-time features proved inadequate for our modeling purposes.

Addressing categorical features including product family, state, city, and day of the week, we implemented one-hot encoding techniques. This transformation converted these categorical attributes into binary features, facilitating their integration into our models effectively.

For feature selection, our strategy focused on retaining a substantial number of features, considering the requirement of feeding a neural network with comprehensive data. Employing XGBoost's feature importance method, we gauged the significance of various features within the dataset. This approach guided us in identifying and preserving the most influential features relevant to our modeling objectives.

Additionally, scaling operations were selectively applied to the dataframe when deemed necessary. This process ensured uniformity and compatibility of feature magnitudes within the dataset, contributing to improved model performance.

# **5. Evaluations and Results**

## 5.1. Evaluation Methods

1. XGBoost Model Evaluation:

Validation Method: Hold-out validation with an 80-20 split for training and validation, respectively.

Hyperparameters: Set with specific values such as n\_estimators=10, max\_depth=4, min\_child\_weight=1, learning\_rate=0.1, subsample=0.8, colsample\_bytree=0.8, and tree\_method='hist'.

Validation metric : RMSLE value = 6.72

1. LSTM Model Evaluation:

Validation Method: Hold-out validation with a 70-20-10 split for training, validation, and test sets.

Data Preparation: Setting time\_steps=33, creating X\_tensor and y\_tensor by reshaping the data for the LSTM model. The X\_predict dataset is also prepared for predicting the next 1 row.

Model Definition: Utilizes a Sequential model with specific architecture settings. It includes Bidirectional LSTM layers with units=128 and 64, Dropout layer for regularization, Dense layers with units=32, and a final output layer, used Adam optimizer with learning rate of 0.001.

Validation metric : RMSLE value = 1.12

1. GRU Model Evaluation:

Validation Method: Similar to LSTM, using hold-out validation with the same dataset preparation settings.

Model Definition: Utilizes a Sequential model with a different architecture. It comprises GRU layers with units=64 and 32, along with a Dropout layer for regularization and a Dense output layer, used SGD optimizer with learning rate of 0.001.

Validation metric : RMSLE value = 1.39

1. ARIMA Model Evaluation:

Validation Method: Hold-out validation with 80 %– 20% split for training, and testing sets.

Evaluation Method: -

We initially assessed the dataset's stationarity through visual inspection and the Augmented Dickey Fuller test, identifying non-stationarity. To address this, we applied the differencing technique, subtracting each value from its preceding one, making the data more stationary.

The Auto ARIMA model was employed to automatically determine the optimal order (p, d, q) for ARIMA. The chosen model was (5, 0, 5). Subsequently, the model was fitted using the training set, and predictions were made for the testing set.

Validation metric : RMSE value = 43.52

Evaluation Methods after feedback:  
1. XGBoost Model Evaluation:

Validation Method: We used Time Series data split using TimeSeriesSplit() and created 18 splits with test size 30 days and a gap of 1 day. Forward Chaining Cross was applied on these train and test sets.

Hyperparameters: Set with specific values such as n\_estimators=100, max\_depth=4, min\_child\_weight=5, learning\_rate=0.01, subsample=1.0, colsample\_bytree=0.8, and tree\_method='hist'.

Validation metric : RMSLE value = 2.93

1. LSTM Model Evaluation:

Validation Method: We used forward chaining cross validation where we divided the dataset into 18 parts and Hold-out cross validation with a 70-20-10 split for training, validation, and test sets of each split where we joined each split 1 by 1 for better model training.

Data Preparation for each split: Setting time\_steps=30, creating X\_tensor and y\_tensor by reshaping the data for the LSTM model. The X\_predict dataset is also prepared for predicting the next 1 row.

Model Definition: Utilizes a Sequential model with specific architecture settings. It includes Bidirectional LSTM layers with units=32 and 64, Dropout layer for regularization, Dense layers with units=32, and a final output layer, used Adam optimizer with learning rate of 0.001.

Validation metric : RMSLE value = 0.88

1. ARIMA Model Evaluation:

Validation Method: Hold-out validation with 80 %– 20% split for training, and testing sets.

Evaluation Method: -

This time we directly ran the auto ARIMA function on the data without checking the stationarity test and differencing technique.

The Auto ARIMA model was employed to automatically determine the optimal order (p, d, q) for ARIMA. The chosen model was (5, 1, 5). Subsequently, the model was fitted using the training set, and predictions were made for the testing set.

Validation metric : RMSE value = 41.51

## 5.2. Results and Findings

|  |  |  |
| --- | --- | --- |
| Model | Result with hold out validation | Results with forward chaining |
| XGBoost (RMSLE) | 6.72 | 2.93 |
| LSTM (RMSLE) | 1.12 | 0.88 |
| ARIMA (RMSE) | 43.51 | 41.52 |

# **6. Conclusions and Future Work**

## 6.1. Conclusions

Our project, "COMPARATIVE ANALYSIS BETWEEN DIFFERENT TIME-SERIES FORECASTING MODELS," focused on leveraging machine learning for retail sales forecasting. Through model comparisons, we highlighted the importance of accurate predictions for inventory management, emphasizing the significance of data-driven approaches in the retail sector.

## The evaluation of models, such as XGBoost, LSTM, GRU, and ARIMA, provided valuable insights into their strengths and limitations in handling the intricacies of retail sales forecasting. Notably, the LSTM model exhibited superior performance, showcasing its potential in capturing intricate temporal patterns within the dataset.

## 6.2. Limitations

While our project aimed to address several challenges encountered in retail forecasting, certain limitations persisted. The project's scope was primarily confined to the provided dataset, potentially limiting the generalizability of our findings to diverse retail settings. Additionally, despite employing robust techniques, the models might still face challenges in accurately capturing unpredictable external factors influencing sales patterns.

## 6.3. Potential Improvements or Future Work

6.3.1. Incorporation of Unsupervised Learning Techniques

To gain deeper insights into inherent data patterns, employing unsupervised learning methodologies like k-means clustering could provide a more granular understanding of customer behavior, product segmentation, or store categorization. Uncovering hidden patterns through clustering techniques might uncover latent structures within the data, potentially aiding in more refined forecasting models.

6.3.2. External Factors Integration

Expanding the scope by integrating external data sources, such as weather patterns, social media sentiments, or economic indicators, could enhance model robustness. Incorporating these diverse factors into the forecasting process might better capture the complexities of real-world retail dynamics.

6.3.3. Advanced Time-Series Models Exploration

Further exploration into advanced time-series models, such as Prophet, Gaussian Process Regression, or hybrid models combining neural networks and traditional approaches, could offer improved accuracy in capturing non-linear trends and seasonality present in retail sales data.