**CHAPTER 1**

**INTRODUCTION**

Sales forecasting plays a pivotal role in modern business operations, enabling organizations to predict future sales trends, optimize inventory, and improve decision-making. By leveraging historical data, it helps businesses anticipate customer demands, reduce costs, and maintain financial stability. The integration of machine learning techniques, capable of analysing large datasets to identify patterns, has further enhanced the accuracy and reliability of these forecasts.

This project focuses on developing a machine learning-based sales forecasting model using historical sales data. Key variables, such as date, store, product, and units sold, are analysed to provide actionable insights. Techniques like Random Forest modelling, feature engineering, and exploratory data analysis (EDA) are employed to ensure robust and interpretable predictions. Visualization tools are also used to identify trends and seasonality, facilitating a deeper understanding of sales behaviour.

**1.1 Aim of the Project**

The primary aim is to build a machine learning model that predicts product sales for various stores over time. Key objectives include:

1. **Accurate Predictions:** Develop a model to forecast sales effectively.

2. **Enhanced Insights:** Provide stakeholders with actionable data to guide decisions.

3. **Optimized Inventory:** Predict demand to improve resource planning.

4. **Data Exploration:** Analyse patterns and trends in sales data.

5. **Advanced Techniques:** Utilize machine learning methods to refine forecasts.

These goals demonstrate the potential of machine learning to address real-world business challenges effectively.

**1.2 Scope of the Project**

**Data Collection and Preprocessing:**

The project uses historical sales data, focusing on preprocessing steps such as handling missing values, feature standardization, and generating lagged variables for temporal analysis.

**Exploratory Data Analysis (EDA):**

EDA involves identifying sales patterns, trends, and seasonality. Visualization methods like line plots and heatmaps provide insights into correlations and anomalies.

**Model Development:**

A Random Forest model is used for its robustness and predictive capabilities. Techniques like hyperparameter tuning and cross-validation enhance its performance. Alternative models are also explored for comparison.

**Feature Importance Analysis:**

Key factors influencing sales are identified, helping stakeholders understand the variables driving predictions.

**Visualization and Reporting:**

Results are communicated using visualizations, such as trend analysis and feature importance plots, to support decision-making.

**Scalability:**

The model is designed for scalability, allowing integration of additional data sources like marketing or economic indicators to improve accuracy.

**CHAPTER 2**

**PROBLEM DESCRIPTION**

In today’s fast-paced market environment, businesses are constantly challenged by the need to accurately predict future sales. With fluctuating consumer demands, dynamic competition, and external influences such as economic conditions, the ability to forecast sales effectively has become a critical determinant of success. This chapter delves into the core challenges addressed by the project and outlines the problem in detail.



**FIG 2.1 Sales Forecasting Description**

**2.1 Problem Statement**

Sales forecasting involves predicting the quantity of goods a business will sell over a given period, which directly impacts inventory management, resource allocation, and financial planning. Traditionally, businesses have relied on manual methods or simple statistical models for forecasting. However, these methods often fail to capture complex, non-linear relationships in data and lack the ability to adapt to changing market conditions.

The problem at hand is to develop an advanced forecasting model capable of providing accurate predictions of sales based on historical data. Key challenges include:

**Data Complexity:** Sales data often contains multiple variables, such as store location, product type, and seasonal factors, which interact in complex ways.

**Handling Missing Data:** Incomplete or inconsistent datasets pose significant hurdles for traditional forecasting techniques.

**Dynamic Trends:** Consumer preferences and market dynamics evolve over time, necessitating models that can adapt to these changes.

**Scalability:** The solution must be scalable to handle large datasets across multiple stores and products.

**Interpretability:** The model’s predictions must be interpretable to enable actionable insights for stakeholders.

**2.2 Key Challenges and Subtopics**

**Data Preprocessing and Cleaning:**

Raw sales data is rarely ready for analysis. Issues such as missing values, duplicate records, and inconsistent formats require thorough preprocessing. For example, missing entries in the ‘number sold’ column can lead to inaccurate forecasts if not addressed. The project employs techniques such as mean imputation and data normalization to ensure data quality.

**Feature Engineering:**

Identifying relevant features is crucial for enhancing model performance. Temporal patterns, such as weekly or seasonal trends, are captured using lagged variables and time-based features. By engineering these features, the project ensures the model accounts for past behaviours to predict future outcomes effectively.

**Model Selection:**

Choosing the right machine learning model is vital. While simpler models like linear regression are interpretable, they may not capture the intricate patterns in sales data. The project uses Random Forest, known for its robustness and ability to handle large datasets with complex interactions.

**Visualization and EDA:**

Understanding the data visually is a critical step. Visualization tools such as line charts for sales trends and heatmaps for correlation analysis provide insights that guide model development. These visualizations also help validate the assumptions made during preprocessing and feature selection.

**Model Validation and Performance Metrics:**

Accurate sales forecasting requires rigorous evaluation. Metrics like Mean Absolute Percentage Error (MAPE) are used to measure model performance. Cross-validation ensures the model generalizes well to unseen data, while hyperparameter tuning optimizes its predictive power.

**Adaptability and Real-World Application:**

The solution must adapt to new data and changing trends. Incorporating additional features, such as promotional campaigns or economic indicators, ensures the model remains relevant over time. Furthermore, the project aims to design a system that integrates seamlessly into existing business processes.

**CHAPTER 3**

**REQUIREMENTS**

**3.1 Software Requirements**

To implement this project, a stable environment with Python programming capabilities is essential. The following software and libraries are required:

**Jupyter Notebook:** Used for developing and testing the code interactively.

**Python Libraries**

**pandas** for data manipulation and cleaning.

**matplotlib** and **seaborn** for data visualization.

**sklearn** for machine learning algorithms and preprocessing.

**NumPy** for numerical operations.

**3.2 Data Requirements**

The dataset used in this project is split into train and test datasets. The train dataset is used for model training, while the test dataset serves as a holdout set to evaluate model performance. Both datasets contain historical sales data with key features such as:

Date: The date of the sales transaction.

Store ID: Identifier for the store where the product was sold.

Product ID: Identifier for the product being sold.

Number of Units Sold: The quantity of the product sold on that date.

It is crucial that both datasets are clean, with minimal missing values, and well-structured to facilitate effective preprocessing, such as handling missing values, feature scaling, and temporal analysis. The datasets must also cover a sufficient time range to capture sales patterns, seasonality, and trends for reliable forecasting.

**3.3 Performance Requirements**

The performance benchmark for this project is to achieve a Mean Absolute Percentage Error (MAPE) of less than 10% on the validation set. This ensures the model's accuracy and reliability, making it suitable for real-world business applications.

**3.4 Other Considerations**

**Scalability:** The model should be adaptable to larger datasets and different industries to maintain its generalizability.

**Visualization:** Clear, informative graphical representations (like trend analysis, feature importance plots, etc.) are crucial for presenting results and supporting decision-making.

**CHAPTER 4**

**DESIGN**

**4.1 Algorithm/Methodology**

The design of the sales forecasting model involves several key stages: data preprocessing, feature engineering, model selection, and evaluation. The underlying methodology for building the model integrates standard machine learning practices with a focus on time-series forecasting. Below is a detailed description of the approach and the key components involved.

**Data Preprocessing**

Data preprocessing is crucial for ensuring that the model can learn from clean and structured data. The dataset includes several variables like date, store ID, product ID, and the number of units sold. These data are first cleaned by handling missing values and converting categorical variables into numeric ones where necessary. Specifically, the date feature is converted into a datetime format, which allows for temporal analysis, such as identifying seasonal trends and incorporating lag features.

**Feature Engineering**

Feature engineering is the process of transforming raw data into meaningful input for the machine learning model. In this project, temporal features are particularly important for accurate sales forecasting. One significant feature engineering technique is the creation of lagged variables. These are past values of sales (number of units sold) that help capture temporal dependencies in the sales data. A time series model, such as the one built here, benefits from such lagged features to make predictions based on historical sales trends.

Additionally, rolling mean features are introduced to capture the smoothed trend in sales over time. This helps the model understand short-term sales fluctuations while reducing noise in the data. The rolling mean is calculated over a fixed window (e.g., 7 days) and is used as an additional feature for the model.

**Model Selection: Random Forest Regressor**

For predicting sales, a Random Forest Regressor algorithm is used due to its robustness and ability to handle both nonlinear relationships and high-dimensional datasets. The Random Forest algorithm works by constructing multiple decision trees and averaging their predictions to improve accuracy and reduce overfitting.

The choice of Random Forest is ideal for this problem because sales data can exhibit complex patterns that may not be easily captured by simple linear models. Random Forest is a versatile and powerful algorithm capable of modeling intricate relationships between features like date, store, and product.

**Model Training and Hyperparameter Tuning**

Once the features are prepared, the next step is to split the data into training and validation sets. The model is trained on the training set, and hyperparameters, such as the number of estimators (trees), are optimized using cross-validation techniques. Random Forest is a flexible algorithm that has hyperparameters like the number of trees (n\_estimators), depth of trees (max\_depth), and minimum samples required to split a node (min\_samples\_split), which all influence the model’s performance.

In this project, hyperparameter tuning is done using grid search or randomized search, which tests multiple combinations of hyperparameters to find the best-performing model.

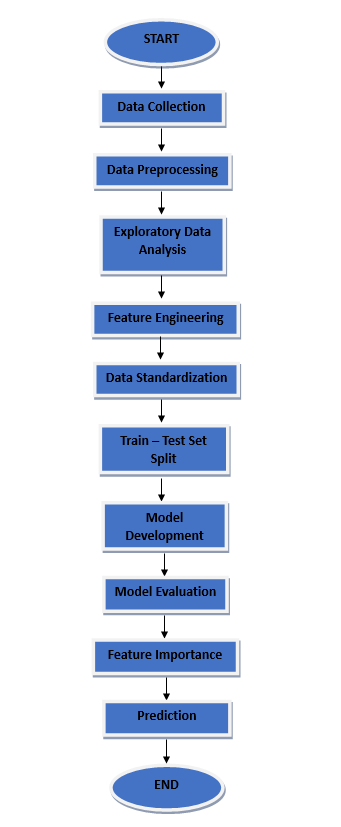
**Model Evaluation**

The model is evaluated based on its performance on the validation set. The primary evaluation metric used is Mean Absolute Percentage Error (MAPE), which measures the accuracy of the predictions in terms of percentage error. A lower MAPE indicates better model performance. MAPE is particularly suitable for sales forecasting as it handles cases where the target variable (sales) has a wide range of values, and gives a normalized measure of prediction error.

Additionally, feature importance is evaluated to understand which features most significantly impact the model’s predictions. This is particularly important for decision-making, as it allows stakeholders to understand the key drivers of sales and tailor their strategies accordingly.

**4.2 Flow Chart**

The flowchart below outlines the high-level steps involved in the design and implementation of the sales forecasting model:



**FIG 4.1: Flow Chart of The Design**

**Data Collection**

The process starts with collecting historical sales data, including features like date, store ID, product ID, and units sold.

**Data Preprocessing**

* Handle missing values and outliers.
* Convert the Date column to datetime format.
* Generate lagged variables to capture temporal dependencies.
* Create rolling mean features to smooth out short-term fluctuations.

**Feature Engineering**

* Calculate lagged features for a predefined number of past time periods (e.g., 3 days, 7 days).
* Add rolling mean features for trend analysis.
* Standardize features to ensure that the model is not biased by the scale of the data.

**Train-Test Split**

Split the dataset into training and validation sets, ensuring that the temporal order of the data is maintained to avoid data leakage.

**Model Selection**

Use Random Forest Regressor to fit the model on the training data. Random Forest can capture complex, nonlinear relationships between sales and other features.

**Model Training**

Train the Random Forest model on the training set, using cross-validation for hyperparameter tuning. Optimize hyperparameters like the number of estimators, maximum depth, and minimum samples split.

**Model Evaluation**

* Evaluate model performance using MAPE on the validation set.
* Check for any overfitting or underfitting and adjust hyperparameters accordingly.

**Feature Importance Analysis**

Identify the most important features influencing sales predictions. This step is crucial for business stakeholders to make informed decisions.

**Prediction and Reporting**

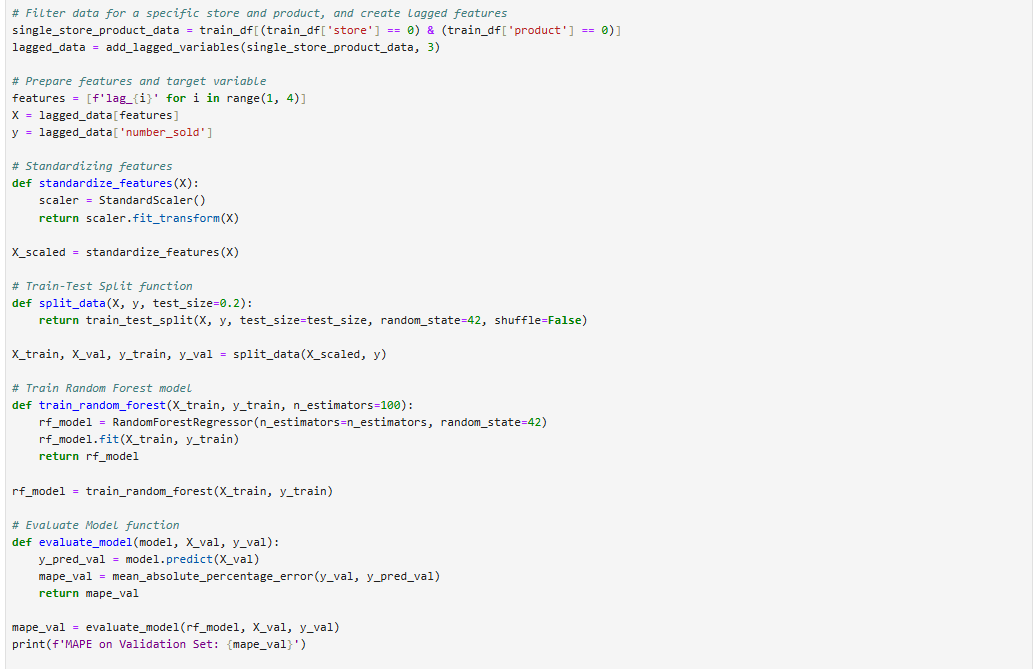
After model evaluation, use the trained model to make predictions on the test set. Visualize the predictions versus actual sales to evaluate the model's practical applicability.

**CHAPTER 5**

**IMPLEMENTATION**



**FIG 5.1: Code Implementation - 1**



**FIG 5.2: Code Implementation - 2**



**FIG 5.3: Code Implementation - 3**



**FIG 5.4: Code Implementation - 4**

**CHAPTER 6**

**DESCRIPTION OF MODULES**

This project involves several key modules that work together to process the data, build a machine learning model, and evaluate its performance. Each module has a specific role that contributes to the overall workflow, from data loading and preprocessing to model evaluation and visualization. Below is a detailed description of each module involved in this project.

**6.1 Data Loading Module**

The **Data Loading** module is responsible for reading the historical sales data from the CSV files (train.csv and test.csv). This module uses the pandas library to load the datasets into pandas Data Frames for further processing. It ensures that the data is structured and ready for subsequent steps, such as preprocessing and feature engineering.



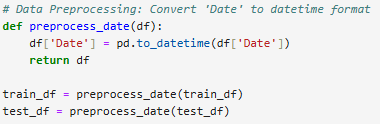
**FIG 6.1: Load Datasets (Train & Test)**

By reading the data into Data Frames, the module makes it easier to handle and manipulate the data during the preprocessing and modelling stages.

**6.2 Data Preprocessing Module**

The **Data Preprocessing** module ensures that the data is clean, formatted, and ready for analysis. In this project, preprocessing includes tasks such as:

* **Date conversion:** The "Date" column in the dataset is converted to a pandas datetime format to facilitate time-based operations and lagging features.
* **Handling missing values:** This module is designed to handle missing or NaN values that might be present in the dataset. Data cleaning ensures that incomplete or corrupted data does not interfere with model training.



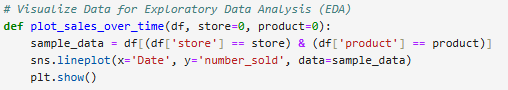
**FIG 6.2: Data Preprocessing**

This module prepares the data for further processing steps, like feature extraction and exploratory data analysis (EDA).

**6.3 Exploratory Data Analysis (EDA) Module**

The **EDA** module helps uncover insights and patterns in the data. Through visualizations such as line plots, histograms, and correlation heatmaps, EDA helps to identify trends, seasonal variations, and relationships between variables.

For example, one of the tasks in this module is to visualize sales trends over time for different stores and products:



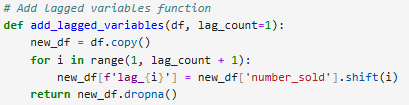
**FIG 6.3: Exploratory Data Analysis (EDA)**

The EDA module serves as a critical step in identifying underlying patterns in the sales data, which will be essential for the feature engineering and modeling steps.

**6.4 Feature Engineering Module**

The **Feature Engineering** module is designed to create new features that may help improve the performance of the machine learning model. In this project, two important feature engineering tasks are performed:

* **Lagged features:** Previous sales data points (lags) are added as new features to predict future sales. This allows the model to consider past behavior in predicting future trends.
* **Rolling mean:** A moving average over a set window (e.g., 7 days) is calculated to smooth out fluctuations in sales data and capture longer-term trends.



**FIG 6.4: Feature Engineering Module**

This module creates the necessary features for training the machine learning model and helps enhance its predictive accuracy.

**6.5 Model Development Module**

The **Model Development** module is where the machine learning model is created and trained. In this project, the **Random Forest Regressor** is used, as it is a robust model that handles complex relationships and does not overfit easily.

The model is trained using the training data, and its performance is evaluated based on the validation data. The model's hyperparameters, such as the number of estimators, can be tuned to improve accuracy.

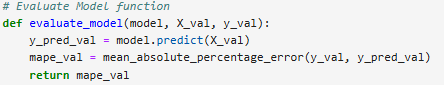


**FIG 6.5: Model Development Module**

Once trained, the model can be used to predict sales in the test set or for new data.

**6.6 Model Evaluation Module**

The **Model Evaluation** module assesses the performance of the trained machine learning model. In this project, **Mean Absolute Percentage Error (MAPE)** is used as the primary evaluation metric. This metric helps quantify the accuracy of predictions by comparing the predicted values with the actual values.



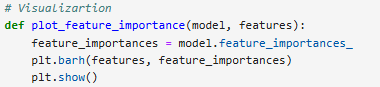
**FIG 6.6: Model Evaluation Module**

This module helps verify how well the model is performing and whether it meets the required performance thresholds.

**6.7 Visualization Module**

The **Visualization** module is responsible for creating graphical representations of the data and model results. Visualizations such as feature importance plots and actual vs. predicted sales plots are used to communicate findings and insights effectively.

For example, the **Feature Importance** visualization displays the importance of each feature in predicting sales:

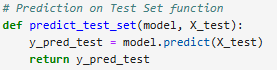


**FIG 6.7: Visualization Module**

These visualizations make it easier for stakeholders to understand the model's behaviour and the data's key drivers.

**6.8 Prediction Module**

Finally, the **Prediction** module is responsible for generating predictions on unseen data. After the model has been trained and evaluated, it is used to predict sales for the test set (or future data points). The predictions are then compared to actual sales to assess the model's performance.



**FIG 6.8: Prediction Module**

This module is crucial for assessing how well the model generalizes to new data and for providing actionable insights for business operations.

**CHAPTER 7**

**RESULTS**

**7.1 Model Performance**

The performance of the model is assessed using the Mean Absolute Percentage Error (MAPE). MAPE calculates the average percentage error between the predicted and actual sales, providing a clear measure of accuracy. A lower MAPE indicates better predictive accuracy. The model is initially validated using a validation dataset, where the MAPE is computed to determine how well the model generalizes to unseen data. A low MAPE on the validation set suggests that the model is effective at forecasting sales.

**Model Selection**

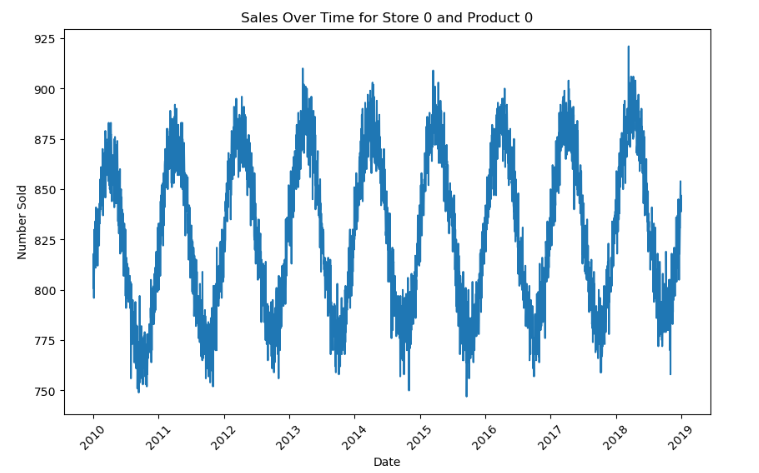
The Random Forest Regressor was chosen for this project due to its robustness and ability to handle complex, non-linear relationships in the data. It performs well with large datasets and provides feature importance, helping identify the most influential variables. The model was trained and fine-tuned using hyperparameters to optimize performance.

After validation, the model is tested on the test dataset. If the MAPE on the test set closely matches the validation set, it indicates that the model is generalizing well and can predict sales accurately across different data samples.

**Actual vs Predicted Sales**

A comparison between actual and predicted sales is presented visually through a line plot or bar chart. This allows for a clear comparison of the predicted values against the real sales data over time. If the predicted values closely follow the actual trend, with minimal divergence, it demonstrates that the model is successful in capturing the underlying patterns and making accurate sales forecasts. This visual comparison further validates the model's effectiveness in predicting future sales.

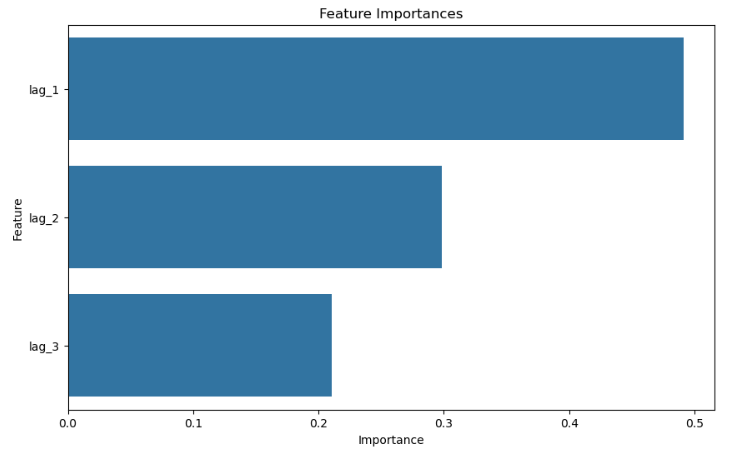
**7.2 Output Screenshot**



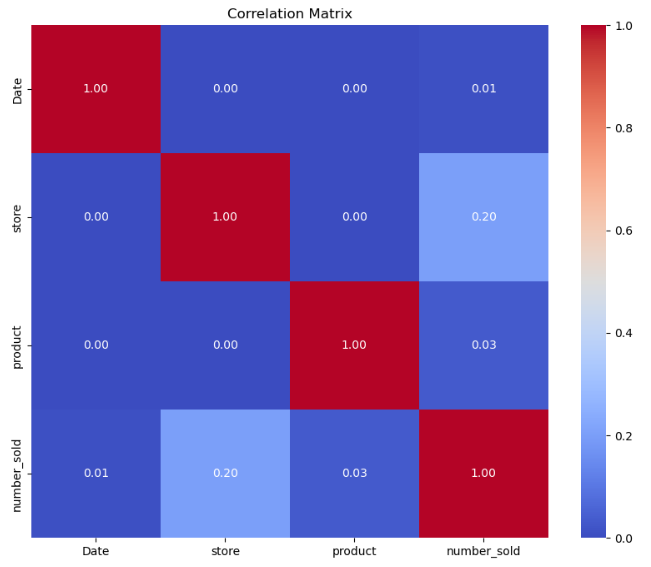
**FIG 7.1: Visualization Data for Exploratory Data Analysis (EDA)**



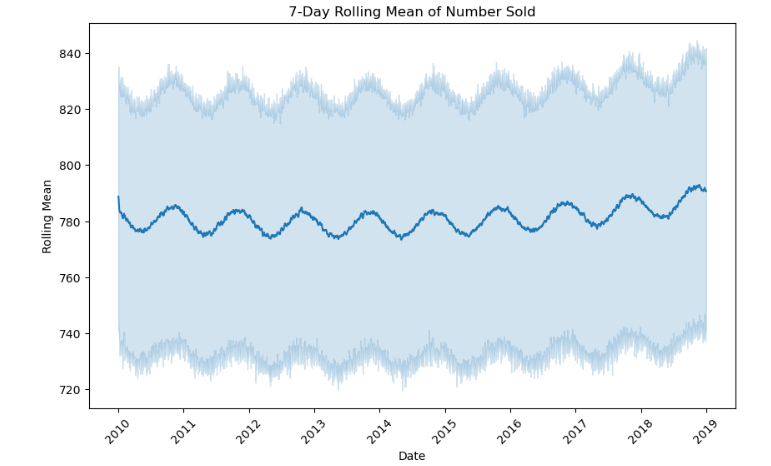
**FIG 7.2: Model Validation Result**



**FIG 7.3: Feature Importance Visualization**



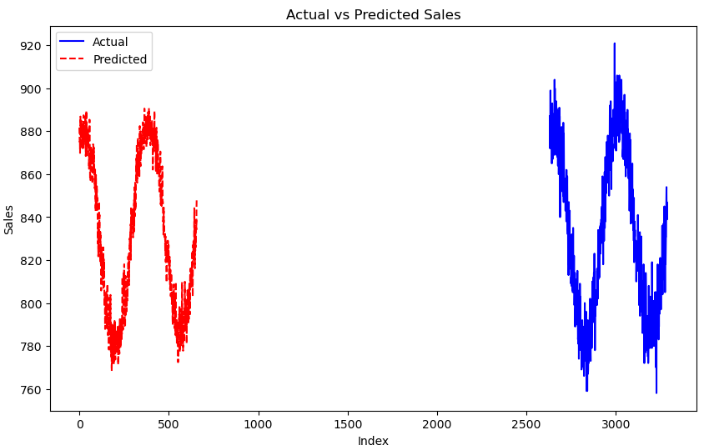
**FIG 7.4: Data Visualization: Correlation Matrix of Features**



**FIG 7.5: Visualization of Rolling Mean Feature**



**FIG 7.6: Result of Evaluating Model on Test Set**



**FIG 7.7: Visualize Predictions vs Actual values**



**FIG 7.8: Mean Absolute Error ( Evaluation Metric )**

**CHAPTER 8**

**CONCLUSION**

The primary goal of this project was to develop a machine learning-based sales forecasting model capable of accurately predicting future sales using historical data. By leveraging the **Random Forest Regressor** and employing techniques like **feature engineering**, **data preprocessing**, and **exploratory data analysis (EDA)**, the model successfully identified patterns and trends within the sales data, leading to effective forecasts.

Throughout the project, several steps were taken to ensure the model’s reliability and accuracy. First, the data was pre-processed, addressing issues such as missing values and standardizing features. Lagged variables and rolling mean features were created to capture temporal dependencies, enhancing the model's ability to predict future sales. The **train-test split** method allowed for proper evaluation, ensuring that the model's performance was not overfitted and could generalize well to unseen data.

The model's performance was evaluated using **Mean Absolute Percentage Error (MAPE)**, which showed promising results with a low error margin on both validation and test datasets. Additionally, the **feature importance analysis** provided valuable insights into the factors influencing sales, such as past sales data and store-specific patterns. This information can help businesses optimize inventory, manage resources, and make data-driven decisions.

In conclusion, this project demonstrates the effectiveness of machine learning in tackling real-world business challenges, particularly in sales forecasting. By integrating historical sales data with predictive models, businesses can significantly improve decision-making processes. Moving forward, the model can be expanded by incorporating more complex features, including external data like marketing efforts or economic indicators, to further enhance forecast accuracy and business outcomes.

**REFERENCES**

[1] **Breiman, L. (2001).** Random forests. *Machine Learning*, 45(1), 5-32.

* Introduces the Random Forest algorithm used in this project, explaining its fundamentals and applications in predictive modelling.

[2] **James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013).** *An Introduction to Statistical Learning: With Applications in R*. Springer.

* Provides essential concepts of statistical learning and machine learning methods, including those used for this project.

[3] **Pedregosa, F., et al. (2011).** Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

* Discusses the Scikit-learn library, pivotal for implementing machine learning models in this project.