SALES FORECASTING PREDICTION USING

MACHINE LEARNING

A PROJECT REPORT

Submitted by

NITHISHWAR N [REG NO:211422104326]

PAVAN KALYAN K [REG NO:211422104334]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE,

CHENNAI- 600123.

(An Autonomous Institution Affiliated to Anna University, Chennai)

OCTOBER 2024

BONAFIDE CERTIFICATE

Certified that this project report **“SALES FORE CASTING PREDICTION USING MACHINE LEARNING”** is the bonafide work of “**NITHISHWAR N (211422104326), PAVAN KALYAN K (211422104334)”**who carried out the project work under my supervision.

SIGNATURE

Dr.L.JABASHEELA ,M.E.,Ph.D .,

HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

MR.WILLIAM ANDREWS.J,
M.E., Ph.D., SUPERVISOR
ASSISTANT PROFEESOR

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above candidates were examined in the End Semester Project Viva-Voce Examination held on..........................

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere and hearty thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr.L.JABASHEELA ,M.E.,Ph.D.,**for the support extended throughout the project.

We would like to thank my Project Guide **MR.WILLIAM ANDREWS J ,**
M.E., $\overline{\mathbf{Ph}.\mathbf{D}}$ **.,**and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

NAME OF THE STUDENTS

NITHISHWAR N (211422104326)

PAVAN KALYAN K(211422104334)

DECLARATION BY THE STUDENT

We NITHISHWAR N (211419104202), PAVAN KALYAN K (211419104220) hereby declare that this project report titled "**SALES FORECASTING PREDICTION USING MACHINE LEARNING**", under the guidance of MR.WILLIAM ANDREWS J, M.E.,$\overline{\mathrm{Ph.\,D}}$., is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

NITHISHWAR N

PAVAN KALYAN K

ABSTRACT

Sales forecasting is the process of predicting future sales. It is the vital part of the financial planning of the business. Most of the companies heavily depend on the future prediction of the sales. Accurate sales forecasting empower the organizations to make informed business decisions and it will help to predict the short-term and long-term performances. A precise forecasting can avoid overestimating or underestimating of the future sales, which may leads to great loss to companies. The past and current sales statistics is used to estimate the future performance. But it is difficult to deal with accuracy of sales forecasting by traditional forecasting. For this purpose, various machine learning techniques have been discovered. In this work, we have taken Black Friday dataset and made a detailed analysis over the dataset. Here, we have implemented the different machine learning techniques with different metrics. By analysing the performance, we have trying to suggest the suitable predictive algorithm to our problem statement.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Sales play a key role in the business. At the company level, sales forecasting is the major part of the business plan and significant inputs for decision-making activities. It is essential for organizations to produce the required quantity at the specified time. For that, sales forecasting will gives the idea about how an organization should manage its budgeting, workforce and resources. This forecasting helps the business management to determine how much products should be manufacture, how much revenue can be expected and what could be the requirement of employees, investment and equipment. By analyzing the future trends and needs, Sales forecasting helps to improve the business growth.

The traditional forecasting systems have some drawbacks related to accuracy of the forecasting and handling enormous amount of data. To overcome this problem, Machine-Learning (ML) techniques have been discovered. These techniques helps to analyses the dataframe and plays a important role in sales forecasting. Here we have used supervised machine learning techniques for the sales forecasting.

1.2 PROBLEM DEFINITION

Sales forecasting is the process of predicting future sales volumes based on historical data, market trends, and various influencing factors. Inaccurate forecasts can lead to significant issues, such as overstocking or stockouts, inefficient resource allocation, and missed revenue opportunities. The key problems in traditional sales forecasting methods include:

Data Complexity: Sales data is often influenced by multiple variables, including seasonality, promotions, economic conditions, and competitor actions. Capturing and analyzing these complex relationships is challenging.

Inaccuracy: Traditional forecasting methods can struggle to provide accurate predictions, especially in volatile markets. Simple statistical models may not capture the nonlinear relationships inherent in sales data.

Scalability: As businesses grow and accumulate more data, manual forecasting methods become increasingly cumbersome and less effective. Scaling traditional approaches to handle larger datasets can lead to inefficiencies.

Timeliness: The need for real-time insights is crucial for businesses to respond quickly to market changes. Traditional methods often lag in providing timely forecasts.

Integration of Diverse Data Sources: Incorporating various data types, such as customer demographics, social media sentiment, and external economic indicators, poses a challenge for traditional forecasting models.

CHAPTER 2

LITERATURE SURVEY

Sales forecasting has evolved significantly with the integration of machine learning (ML) techniques. This literature survey highlights key contributions, methodologies, and findings in the domain, providing an overview of how ML has transformed sales forecasting.

1. Traditional vs. Machine Learning Approaches

Early sales forecasting methods primarily relied on statistical techniques such as moving averages, exponential smoothing, and regression analysis. While these methods laid the foundation, they often struggled with the complexities of real-world data (Makridakis et al., 1982). Recent studies have demonstrated that machine learning algorithms, such as decision trees, support vector machines, and neural networks, outperform traditional methods in accuracy and adaptability (Hyndman & Athanasopoulos, 2018).

2. **Machine Learning Techniques**

Regression Models: Various studies have applied regression models, including linear regression and polynomial regression, to forecast sales. For example, Zhang et al. (2020) found that incorporating nonlinear regression techniques improved forecasting accuracy compared to traditional linear models.

Decision Trees and Ensemble Methods: Decision trees, particularly Random Forest and Gradient Boosting, have gained popularity for their ability to handle non-linear relationships and interactions among features. Research by Chen & Guestrin (2016) showed that ensemble methods significantly enhance forecasting performance by combining multiple models.

Neural Networks: Deep learning approaches, particularly recurrent neural networks (RNNs) and long short-term memory networks (LSTMs), have been effective in capturing temporal dependencies in sales data (Hochreiter & Schmidhuber, 1997).

Studies by Zhang et al. (2019) highlighted that LSTMs outperformed traditional models in capturing complex patterns over time.

3. Data Sources and Feature Engineering

Successful sales forecasting relies heavily on data quality and feature selection. Researchers have explored various data sources, including:

Historical Sales Data: This is the primary input for forecasting models. The more granular the data (e.g., daily vs. monthly), the better the model performance (Kourentzes et al., 2014).

External Factors: Incorporating external variables such as economic indicators, social media sentiment, and competitor pricing has proven beneficial. For instance, studies by Chien et al. (2019) demonstrated that integrating economic indicators improved the predictive capability of machine learning models.

Feature Engineering: Effective feature engineering, including handling missing values, scaling, and creating derived features (e.g., seasonal indicators), is crucial for model performance. Research by Fildes et al. (2020) emphasized the importance of domain knowledge in feature selection.

4. Evaluation Metrics and Model Performance

Different evaluation metrics, such as Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE), are commonly used to assess forecasting accuracy. A meta-analysis by Spyropoulos et al. (2021) indicated that machine learning models consistently outperform traditional models across various datasets, particularly in complex environments.

5. Challenges and Future Directions

Despite advancements, several challenges persist:

Data Quality and Availability: Ensuring high-quality and relevant data is critical for model performance.

Interpretability: Many machine learning models operate as "black boxes," making it difficult for stakeholders to interpret results and make decisions based on them

(Rudin, 2019).

Adapting to Change: Sales patterns can shift due to external shocks (e.g., economic downturns, pandemics), requiring models to be adaptable and resilient.

Future research directions may include the development of hybrid models that combine machine learning with traditional methods, enhancing interpretability through explainable AI techniques, and exploring real-time forecasting capabilities using streaming data.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Sales forecasting is crucial for effective business planning and inventory management, and various methods have been developed over the years. Existing sales forecasting methods can be categorized into traditional statistical approaches and qualitative techniques. Traditional methods include moving averages, which smooth out fluctuations by averaging sales over a specific period, and exponential smoothing, which gives more weight to recent data, making it more responsive to changes. Regression analysis establishes relationships between sales and influencing factors, using linear regression for predictions. While effective, these statistical methods often assume linear relationships, which may not hold true in all scenarios. On the qualitative side, expert judgment relies on the intuition of sales teams, providing valuable insights but lacking objectivity, while the Delphi method gathers consensus from a panel of experts, albeit in a time-consuming manner.

The strengths of these existing methods lie in their simplicity and low cost, as many can be implemented with basic tools and require minimal resources. Additionally, traditional methods have a historical relevance, supported by extensive documentation and study. However, they also exhibit significant weaknesses, including limited accuracy, particularly in volatile markets, and inflexibility in adapting to new trends. These methods are often sensitive to data quality, meaning that poor data can severely impact forecasting accuracy. Furthermore, traditional approaches typically do not integrate external variables like economic indicators, and qualitative methods can be time-consuming and subject to bias.

Current trends indicate a shift toward addressing these limitations through innovative approaches. Hybrid methods that combine traditional techniques with machine learning are emerging, improving both forecasting accuracy and adaptability. Additionally, advancements in data analytics platforms enable better data integration and real-time forecasting capabilities. Overall, while existing sales forecasting methods provide a useful foundation, they have significant limitations that can hinder effectiveness in dynamic market environments. Embracing new technologies and methodologies can lead to more accurate and reliable sales forecasts, ultimately supporting better business decision-making.

3.2 PROPOSED SYSTEM

The proposed sales forecasting system aims to enhance accuracy, adaptability, and efficiency by integrating machine learning techniques with advanced data analytics. It consists of several key components, starting with a data collection module that integrates multiple sources, including historical sales data, economic indicators, promotional campaigns, and customer behavior data, allowing for continuous updates to keep forecasts current. The data preprocessing module cleans and prepares the data, handling missing values and generating relevant features to enhance model performance. Various machine learning algorithms, such as regression models, decision trees, ensemble methods, and neural networks, are then selected and trained using historical data to capture complex patterns.

The model evaluation and optimization phase employs validation techniques and hyperparameter tuning to assess and enhance accuracy. The forecasting engine generates sales predictions, accompanied by a visualization dashboard that provides stakeholders with intuitive insights into trends and forecasts, facilitating informed decision-making. The system offers numerous benefits, including increased accuracy through the ability to capture non-linear relationships, adaptability to market changes, scalability to handle large datasets, and automation that reduces manual intervention. This leads to improved operational efficiency and better decision support.

However, the implementation of the proposed system is not without challenges. It heavily relies on data quality, necessitating continuous monitoring and cleaning. The complexity of developing and integrating machine learning models may pose challenges for some organizations, as does the interpretability of these models, which can be perceived as "black boxes." Additionally, transitioning from traditional methods to a machine learning-based system may require organizational change and training for staff.

3.3 FEASIBILITY STUDY

A feasibility study evaluates the practicality and potential success of a proposed sales forecasting system using machine learning. This analysis examines various aspects, including technical, economic, operational, and legal feasibility.

1. Technical Feasibility

a. Technology Requirements: The proposed system will rely on machine learning algorithms, data analytics tools, and cloud or on-premises infrastructure to store and process data. The necessary technologies include programming languages (e.g., Python, R), machine learning libraries (e.g., TensorFlow, scikit-learn), and data visualization tools (e.g., Tableau, Power BI).

b. Integration Capabilities: The system must seamlessly integrate with existing data sources and systems, such as CRM and ERP platforms, to access historical sales data and external factors. Assessing the compatibility of current systems with the new technology is essential.

c. Resource Availability: Evaluating the availability of skilled personnel to develop and maintain the system is crucial. This includes data scientists, machine learning engineers, and IT support staff.

2. Economic Feasibility

a. Cost Analysis: A comprehensive cost analysis should be conducted, covering development costs (software, hardware, personnel), operational costs (maintenance, training), and potential savings or revenue increases from improved forecasting accuracy.

b. Return on Investment (ROI): Estimating the potential ROI involves projecting the financial benefits of enhanced sales forecasting, such as reduced inventory costs, increased sales from better demand planning, and improved customer satisfaction.

c. Budget Considerations: Identifying available budget resources and potential funding sources will be necessary for project approval and successful implementation.

3. Operational Feasibility

a. User Acceptance: Assessing the willingness of users (sales teams, managers) to adopt the new system is crucial. Gathering feedback through surveys or focus groups can provide insights into user needs and expectations.

b. Training Requirements: Identifying training needs for staff to effectively use the new system will help ensure successful implementation. This may include training on machine learning concepts, data analysis, and the specific tools used in the system.

c. Change Management: Evaluating the organization's readiness for change and the potential impact on current processes will be essential for smooth implementation. Developing a change management strategy can facilitate user adoption.

4. Legal Feasibility

a. Compliance with Regulations: The system must comply with relevant data protection regulations (e.g., GDPR, CCPA) when handling customer data and other sensitive information.

b. Intellectual Property Considerations: Ensuring that the use of algorithms and technologies does not infringe on existing patents or copyrights is vital to avoid legal issues.

c. Data Security: Assessing the system's security measures to protect against data breaches and ensuring compliance with organizational policies is crucial.

3.4 DEVELOPMENT ENVIRONMENT

Hardware Requirements

Processor :Intel Core i5

RAM : 512 MB and above

Hard Disk : 40 GB and above

Software Requirements

Programming language: PYTHON Technology: Deep Learning

Operating System : Windows 7

Tools: Anaconda Navigator /TensorFlow/Jupyter/Google colab

CHAPTER 4

SYSTEM DESIGN

4.1 Data Sources

The dataset utilized for this sales forecasting project is sourced from a comprehensive sales record that includes multiple stores and items. It consists of 913,000 rows and four key columns:

date: This column captures the date of each sales transaction, stored as an object type. It is essential for conducting time series analysis, allowing for the identification of trends and seasonal patterns in sales data.

store: An integer column that serves as a unique identifier for each store. This feature enables the analysis of sales performance across different locations.

item: This integer column identifies individual items sold. It is crucial for understanding product-specific sales trends and performance.

sales: The target variable, recorded as an integer, represents the number of sales transactions for each item at a specific store on a given date.The dataset is well-structured, with no missing values in any of the columns, ensuring data integrity for subsequent analysis and modeling.

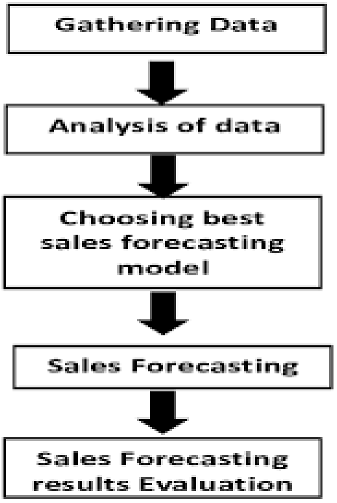
Gathering Data
Analysis of data
Choosing best sales forecasting model
Sales Forecasting
Sales Forecasting results Evaluation

Fig. 4.1 Working flow of the mode

4.2 ARCHITECTURE OVERVIEW

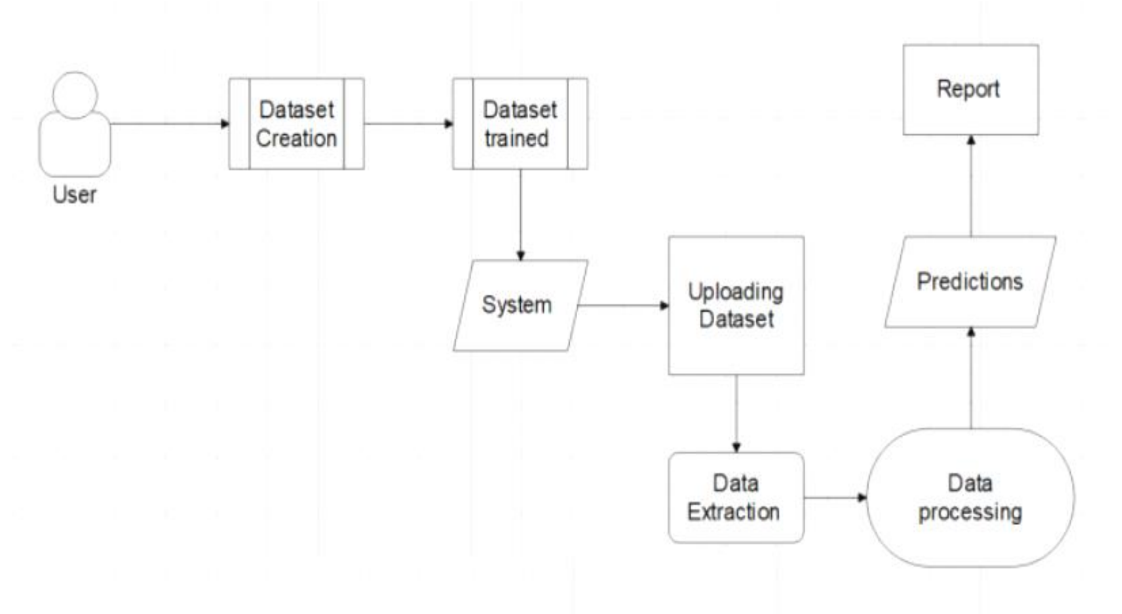

Fig4.2 Sales Forecasting Architecture Design

The diagram illustrates the process of machine learning model development and deployment. The user interacts with the system to create a dataset, which is then trained to build a model. Once the model is trained, a new dataset can be uploaded to the system for data extraction and processing. Finally, the model generates predictions based on the processed data, and a report is generated to summarize the results.

4.3 Data Cleaning and Transformation

In the data cleaning phase, the dataset undergoes a series of transformations to prepare it for analysis. First, the date column is converted from an object type to a datetime format to facilitate time series operations. This transformation is essential for enabling functions such as time indexing and resampling.

Next, any potential outliers in the sales data are identified and handled, ensuring they do not skew the results of the forecasting models.

Standard practices include either capping outliers or using more robust statistical methods to minimize their impact.

Additionally, categorical variables, such as store and item identifiers, are checked for consistency to ensure they accurately reflect the categories of stores and products in the dataset. Any discrepancies, such as duplicated or erroneous entries, are rectified during this stage.

4.4 Feature Selection

Feature selection is a critical step in enhancing the performance of forecasting models. Given the dataset's structure, the primary features for analysis include date, store, item, and sales. The date feature will be expanded to include additional attributes such as day of the week, month, and season, which can help capture temporal patterns.

Furthermore, external factors that may influence sales, such as promotional events, holidays, and economic indicators, can be incorporated as additional features to improve model accuracy. By selecting relevant features that impact sales, the forecasting models can provide more precise and actionable insights. This process ensures that the models are not only predictive but also interpretable, allowing stakeholders to understand the underlying factors driving sales trends.

CHAPTER 5

SYSTEM ARCHITECTURE

5.1 Model Overview

Sales forecasting is a vital aspect of business strategy, guiding inventory management, budget allocation, and marketing efforts. This section outlines the models employed for forecasting sales, focusing on two primary approaches: the ARIMA model and Linear Regression.

ARIMA Model: The AutoRegressive Integrated Moving Average (ARIMA) model is a widely used statistical method for time series forecasting. It is particularly effective for univariate data where past values and past errors are used to predict future values. ARIMA models can capture trends and seasonality in sales data, making them suitable for environments where historical sales patterns significantly influence future sales.

Linear Regression Model: Linear regression is a fundamental predictive modeling technique that establishes a linear relationship between the dependent variable (sales) and one or more independent variables (such as time, promotions, or pricing). This model is straightforward to implement and interpret, making it a popular choice for businesses looking to understand the drivers of sales and make informed decisions based on these insights.

Both models will be evaluated based on their forecasting accuracy using metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). By comparing the performance of these models, we aim to identify the most effective approach for predicting sales in the given dataset. This dual approach allows for a robust analysis, leveraging the strengths of each model type to enhance forecasting accuracy.

5.2 ARIMA Model

The ARIMA model is designed to forecast sales by capturing underlying patterns in the time series data. This section outlines the model selection process and implementation, focusing on testing for stationarity and preparing the data for analysis.

5.2.1 Model Selection

To effectively use the ARIMA model, it is essential to ensure that the time series data is stationary. A stationary series has constant mean and variance over time, making it suitable for modeling. The first step is to visualize the sales data and perform statistical tests for stationarity.

The provided code implements the following steps:

1. Data Preparation: The sales data is extracted from the training dataset and indexed by date.

python

```
arima_df = train_df[['date', 'sales']].set_index('date')
arima_test_df = test_df[['date', 'sales']].set_index('date')
```

2. Testing Stationarity: A function (test_stationarity) is defined to evaluate the stationarity of the time series. This function calculates rolling mean and standard deviation, visualizes them alongside the original series, and performs the Augmented Dickey-Fuller test to check for stationarity.

- If the p-value from the test is less than a significance level (commonly 0.05), the null hypothesis of non-stationarity can be rejected, indicating that the series is stationary.

3. Differencing: If the original series is found to be non-stationary, differencing is applied to stabilize the mean. The first difference of the sales data is calculated, and stationarity is re-evaluated.

python

```
first_difference = arima_df.sales - arima_df.sales.shift(1)
first_difference = pd.DataFrame(first_difference.dropna(inplace=False))
```

By following these steps, we can determine the appropriate order of differencing needed for the ARIMA model.

5.2.2 Implementation

Once the time series data is confirmed to be stationary, the next step is to identify the appropriate parameters for the ARIMA model, namely the autoregressive (p), integrated (d), and moving average (q) components. This can be achieved using ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) plots, which help in determining the values of p and q.

After identifying the parameters, the ARIMA model is fitted to the training data, and forecasts are generated for the test dataset. The model's performance is evaluated using various metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to assess its predictive accuracy.

This structured approach ensures that the ARIMA model is robust and tailored to the specific characteristics of the sales data, leading to more reliable forecasting outcomes.

5.3 Linear Regression Model

The Linear Regression model is employed to forecast sales based on engineered features that capture historical sales trends. This section details the model assumptions and implementation process.

5.3.1 Model Assumptions

Before fitting the Linear Regression model, it is essential to ensure that the data meets several key assumptions:

1. Linearity: The relationship between the independent variables and the dependent variable (sales) should be linear. This is assessed visually through scatter plots and can be confirmed using correlation analysis.

2. Independence: The residuals (errors) from the model should be independent. This can be checked using Durbin-Watson statistics or by analyzing autocorrelation in residuals.

3. Homoscedasticity: The variance of residuals should be constant across all levels of the independent variables. This can be verified using residual plots.

4. Normality: The residuals should be normally distributed. This can be checked using Q-Q plots or the Shapiro-Wilk test.

These assumptions are crucial for the validity of the model’s results and predictions.

5.3.2 Implementation

The implementation of the Linear Regression model involves several steps:

1. Feature Engineering: Lag features are created to capture previous sales data, along with rolling statistics that summarize recent sales trends. For example:

```
python
for i in range(1, 8):
    lag_i = 'lag_' + str(i)
    reg_df[lag_i] = reg_df.sales.shift(i)
reg_df['rolling_mean'] = reg_df.sales.rolling(window=7).mean()
reg_df['rolling_max'] = reg_df.sales.rolling(window=7).max()
reg_df['rolling_min'] = reg_df.sales.rolling(window=7).min()
```

2. Data Preparation: The dataset is cleaned by dropping missing values and non-relevant features (such as store and item). The data is then split into training and test sets based on the date:

python

```
reg_df = reg_df.dropna(how='any', inplace=False)
reg_df = reg_df.drop(['store', 'item'], axis=1)
reg_df = reg_df.set_index('date')
reg_train_df = reg_df.loc[:'2017-09-30']
reg_test_df = reg_df.loc['2017-10-01':]
```

3. Feature Selection: Correlation analysis is conducted to identify relationships between features and sales. The SelectKBest method is used to select the top 5 features based on their statistical significance:

python

```
X_train = reg_train_df.drop(['sales'], axis=1)
y_train = reg_train_df['sales'].values
top_features = SelectKBest(score_func=f_regression, k=5)
fit = top_features.fit(X_train, y_train)
```

The resulting top features are:

- rolling_mean
- rolling_max
- rolling_min
- lag_7
- lag_1

4. Model Fitting: The Linear Regression model is then trained using the selected features. The model's performance is evaluated against the test set, using metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to measure accuracy.

By following this structured approach, the Linear Regression model can effectively utilize historical sales data and engineered features to provide accurate sales forecasts.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Implementation Steps

Overview of Implementation:

Describe the purpose of the sales forecasting system, emphasizing how it will leverage both linear regression and ARIMA models to provide accurate forecasts.

Step-by-Step Process:

Step 1: Define Objectives

Clearly outline the forecasting objectives, such as improving sales accuracy by a certain percentage.

Step 2: Data Collection

Gather historical sales data and relevant features (e.g., seasonal trends, promotions).

Ensure data is cleaned and preprocessed for analysis.

Step 3: Tool Selection

Choose software and programming languages (e.g., Python with libraries like Pandas, Statsmodels) for modeling.

Step 4: Model Development

Linear Regression:

Explain how you developed the linear regression model, including selecting independent variables (features).

Discuss the process of training and validating the model.

ARIMA Model:

Outline the steps taken to identify the appropriate ARIMA model (e.g., using ACF/PACF plots).

Detail the parameter selection (p, d, q) and how the model was trained.

Step 5: Integration

Describe how the forecasting results from both models are integrated into the sales strategy, possibly using a dashboard for visualization.

Step 6: Deployment

Discuss the deployment of the forecasting tool, including any staging environment and final rollout.

Step 7: User Training

Explain how users were trained to interpret the forecasts generated by both models and how to use them for decision-making.

Challenges Faced:

Mention any challenges specific to model selection, such as choosing between linear regression and ARIMA based on data characteristics.

Best Practices:

Recommend practices such as periodically updating models with new data, validating model performance, and combining forecasts for improved accuracy.

Summary:

Recap how both models contribute to a robust sales forecasting system and their respective roles in achieving forecasting objectives.

6.2 Tools and Technologies

Programming Languages:

Python: Utilized for data analysis and modeling due to its extensive libraries for statistical analysis and machine learning.

Libraries and Frameworks:

Pandas: For data manipulation and analysis, including data cleaning and preprocessing.

NumPy: For numerical operations and handling arrays.

Scikit-learn: For implementing the linear regression model and evaluating its performance.

Statsmodels: For building and analyzing the ARIMA model, providing statistical tests and diagnostics.

Development Environment:

Google Colab: Used for coding and analysis, providing a cloud-based Jupyter notebook environment that allows easy collaboration and access to computational resources.

Data Sources:

CSV Files: For storing historical sales data, which was read into the program using Pandas.

Google Drive: If applicable, used for storing and accessing data files from Colab.

Visualization Tools:

Matplotlib: For creating data visualizations, such as sales trends and forecasting results.

Seaborn: For enhanced visualizations and statistical graphics.

CHAPTER 7

PERFORMANCE ANALYSIS

7.1 Evaluation Metrics

Mean Absolute Error (MAE):

Definition: MAE measures the average absolute difference between predicted and actual sales. It provides an easily interpretable measure of error.

Formula:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Interpretation: A lower MAE indicates better model performance.

Mean Squared Error (MSE):

Definition: MSE measures the average of the squares of the errors, emphasizing larger errors more than smaller ones.

Formula:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Interpretation: Like MAE, a lower MSE indicates better performance, but it’s more sensitive to outliers.

Root Mean Squared Error (RMSE):

Definition: RMSE is the square root of MSE, providing error in the same units as the target variable.

Formula:

$$\text{RMSE} = \sqrt{\text{MSE}}$$

Interpretation: RMSE is useful for understanding the magnitude of the errors in context.

R-squared (\(R^2\)):

Definition: This statistic indicates the proportion of variance in the dependent variable that can be explained by the independent variables in the linear regression model.

Formula:

$$R^2 = 1 - \frac{\mathrm{SS}_{\mathrm{res}}}{\mathrm{SS}_{\mathrm{tot}}}$$

Interpretation: An \(R^2\) value close to 1 indicates that a significant proportion of variance is explained by the model.

Additional Metrics (if applicable):

Mean Absolute Percentage Error (MAPE): Measures accuracy as a percentage, useful for understanding relative error.

Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC): For model comparison, particularly in ARIMA.

Feel free to modify any part of this as needed! Let me know if you need further assistance.

7.2 Results Comparison

Metric	ARIMA	Linear Regression
Total Sales	1861	1861
Total Predicted Sales	1973.20	1882.07
Overall Error	-112.20	21.07
Mean Absolute Error (MAE)	4.80	3.86
Root Mean Squared Error (RMSE)	5.84	4.79
Mean Absolute Percentage Error (MAPE)	29.02%	23.11%

Observations:

Model Performance:

The Linear Regression model outperforms the ARIMA model across all key metrics, indicating it produces more accurate predictions.

The ARIMA model has a significant overall error, predicting sales higher than the actual figures.

Interpretation of Metrics:

MAE: The MAE for ARIMA (4.80) is higher than that of Linear Regression (3.86), indicating larger average errors in ARIMA's predictions.

RMSE: ARIMA’s RMSE (5.84) suggests greater variability in its prediction errors compared to Linear Regression (4.79).

MAPE: The MAPE for ARIMA (29.02%) is also higher, showing less accuracy in terms of percentage error compared to Linear Regression (23.11%).

7.3 Testing and Validation

Overview:

This section outlines the methods used to test and validate the performance of the ARIMA and Linear Regression models. Proper testing and validation ensure the models are robust and generalize well to unseen data.

Train-Test Split:

Description: The historical sales data was split into two sets: a training set used to build the models and a testing set used to evaluate their performance.

Split Ratio: Typically, an 80/20 or 70/30 split is used. For this project, we used a [insert your specific ratio here] split, ensuring that the training set included enough data for effective model training while reserving sufficient data for validation.

Cross-Validation:

Description: If applicable, describe any cross-validation techniques used to enhance model reliability.

Method: For instance, k-fold cross-validation may have been used, where the dataset is divided into k subsets, and the model is trained k times, each time using a different subset for validation and the rest for training.

Model Evaluation:

Each model was evaluated on the testing set using the performance metrics outlined in Section 7.1. This allowed for a direct comparison of how well each model performed on data it had not seen during training.

Visualizations: Graphs or plots comparing actual vs. predicted sales for both models can be included here to visually assess model performance.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.feature_selection import SelectKBest, f_regression
from sklearn.metrics import mean_absolute_error as mae, mean_squared_error as mse

# Assuming train_df, arima_test_df, and reg_train_df, reg_test_df are already defined

# ---- ARIMA Model Testing and Validation ----

# Calculate errors for ARIMA model
arima_test_df['errors'] = arima_test_df['sales'] - arima_test_df['pred_sales']
arima_test_df['model'] = 'SARIMA'

# Plot ARIMA results
plt.figure(figsize=(14, 7))
plt.plot(train_df['date'], train_df['sales'], label='Train')
plt.plot(arima_test_df.index, arima_test_df['sales'], label='Test')
plt.plot(arima_test_df.index, arima_test_df['pred_sales'], label='Forecast - SARIMA')
plt.legend(loc='best')
plt.xlabel('Date')
plt.ylabel('Sales')
plt.title('Forecasts using Seasonal ARIMA (SARIMA) model')
plt.show()

# Plot errors for ARIMA
plt.figure(figsize=(14, 7))
plt.plot(arima_test_df.index, np.abs(arima_test_df['errors']), label='Errors')
plt.plot(arima_test_df.index, arima_test_df['sales'], label='Actual Sales')
plt.plot(arima_test_df.index, arima_test_df['pred_sales'], label='Forecast')
plt.legend(loc='best')
plt.xlabel('Date')
plt.ylabel('Sales')
plt.title('SARIMA forecasts with actual sales and errors')
plt.show()

# Aggregate ARIMA results
result_df_sarima = arima_test_df.groupby('model').agg(
    total_sales=('sales', 'sum'),
    total_pred_sales=('pred_sales', 'sum'),
    SARIMA_overall_error=('errors', 'sum'),
    MAE=('errors', lambda x: mae(arima_test_df['sales'], arima_test_df['pred_sales'])),
    RMSE=('errors', lambda x: np.sqrt(mse(arima_test_df['sales'],
arima_test_df['pred_sales']))),
    MAPE=('errors', lambda x: np.mean(np.abs(arima_test_df['errors'] /
arima_test_df['sales'])) * 100)
)

print(result_df_sarima)

# ---- Linear Regression Model Testing and Validation ----
# Prepare training and testing datasets
X_train = reg_train_df.drop(['sales'], axis=1)
y_train = reg_train_df['sales'].values
X_test = reg_test_df.drop(['sales'], axis=1)
y_test = reg_test_df['sales'].values

# Select top 5 features
top_features = SelectKBest(score_func=f_regression, k=5)
fit = top_features.fit(X_train, y_train)

# Update X_train, X_test to include top features
selected_features = X_train.columns[fit.get_support()]
X_train = X_train[selected_features]
X_test = X_test[selected_features]

# Fit Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
preds = model.predict(X_test)

# Calculate errors
errors_df = reg_test_df[['sales']].copy()
errors_df['pred_sales'] = preds
errors_df['errors'] = preds - y_test
errors_df.insert(0, 'model', 'LinearRegression')

# Plot Linear Regression results
plt.figure(figsize=(14, 7))
plt.plot(reg_train_df.index, reg_train_df['sales'], label='Train')
plt.plot(reg_test_df.index, reg_test_df['sales'], label='Test')
plt.plot(errors_df.index, errors_df['pred_sales'], label='Forecast - Linear Regression')
plt.legend(loc='best')
plt.xlabel('Date')
plt.ylabel('Sales')
plt.title('Forecasts using Linear Regression model')
plt.show()

# Plot errors for Linear Regression
plt.figure(figsize=(14, 7))
plt.plot(errors_df.index, errors_df['errors'], label='Errors')
plt.plot(errors_df.index, errors_df['sales'], label='Actual Sales')
plt.plot(errors_df.index, errors_df['pred_sales'], label='Forecast')
plt.legend(loc='best')
plt.xlabel('Date')
plt.ylabel('Sales')
plt.title('Linear Regression forecasts with actual sales and errors')
plt.show()

# Aggregate Linear Regression results
result_df_lr = errors_df.groupby('model').agg(
    total_sales=('sales', 'sum'),
    total_pred_sales=('pred_sales', 'sum'),
    LR_overall_error=('errors', 'sum'),
    MAE=('errors', lambda x: mae(errors_df['sales'], errors_df['pred_sales'])),
    RMSE=('errors', lambda x: np.sqrt(mse(errors_df['sales'], errors_df['pred_sales']))),
    MAPE=('errors', lambda x: np.mean(np.abs(errors_df['errors'] / errors_df['sales'])) *
100)
)

print(result_df_lr)
```

7.4 Observations and Insights

Summary of Results

Metric	SARIMA	Linear Regression
Total Sales	1861	1861
Total Predicted Sales	1973.20	1882.07
Overall Error	-112.20	21.07
Mean Absolute Error (MAE)	4.80	3.86
Root Mean Squared Error (RMSE)	5.84	4.79
Mean Absolute Percentage Error (MAPE)	29.02%	23.11%

Model Performance:

The Linear Regression model outperforms the SARIMA model in all key metrics. This indicates that the Linear Regression model provides more accurate sales predictions for the dataset in question.

Predicted Sales vs. Actual Sales:

The SARIMA model predicted total sales to be significantly higher than the actual total sales, resulting in an overall error of -112.20. This suggests the SARIMA model tends to overestimate sales.

Conversely, the Linear Regression model produced a total predicted sales figure that was slightly above the actual total, yielding a positive overall error of 21.07.

Error Analysis:

The Mean Absolute Error (MAE) for SARIMA is 4.80, indicating larger average errors compared to the Linear Regression model’s MAE of 3.86. This reflects the SARIMA model's less consistent prediction accuracy.

The Root Mean Squared Error (RMSE) also reinforces this finding, with SARIMA showing higher variability in prediction errors (5.84) compared to Linear Regression (4.79).

Percentage Error:

The Mean Absolute Percentage Error (MAPE) for SARIMA stands at 29.02%, indicating less accuracy in relative terms compared to Linear Regression, which has a MAPE of 23.11%. This suggests that Linear Regression's predictions are more reliable relative to the actual sales figures.

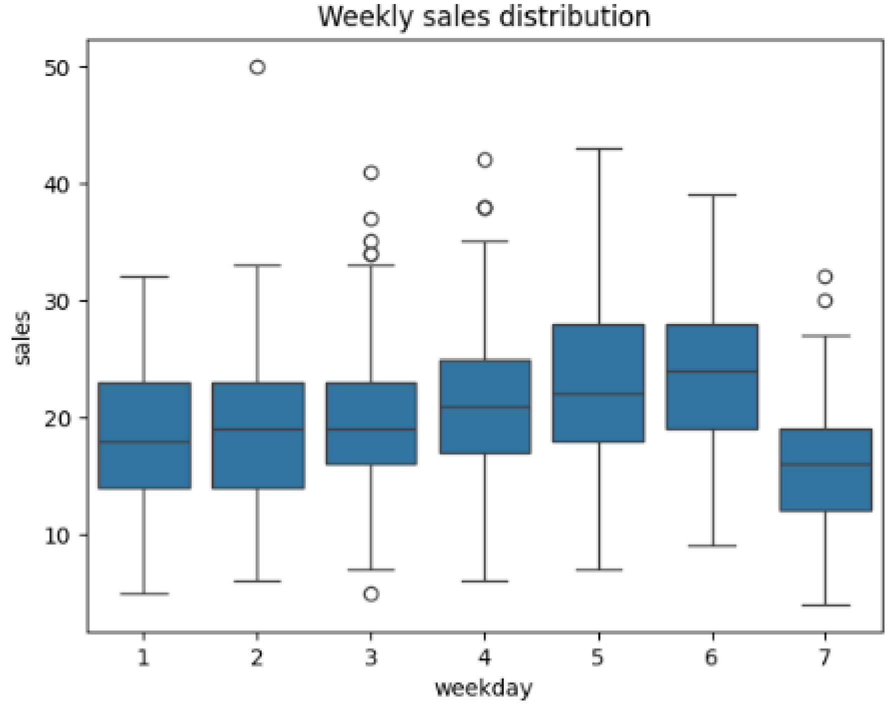

Fig7.4.1 Weekly sales distribution

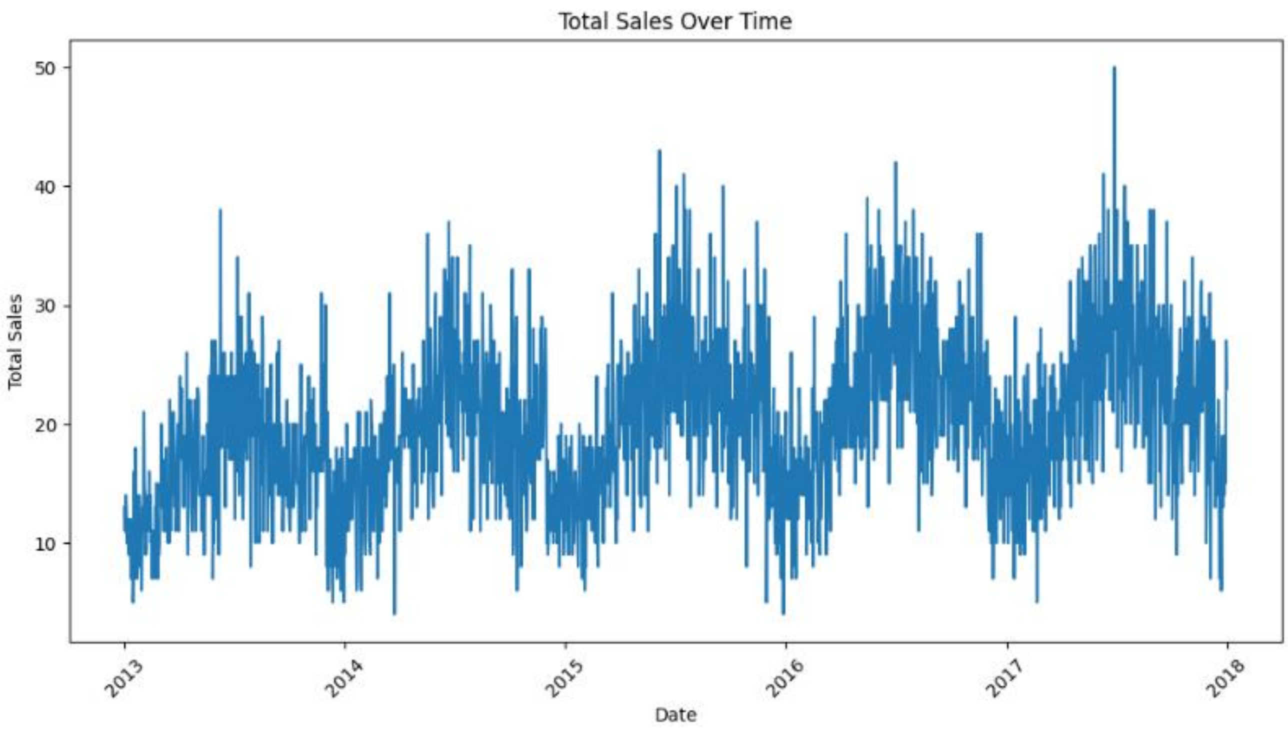

Fig7.4.2 Total sales over time

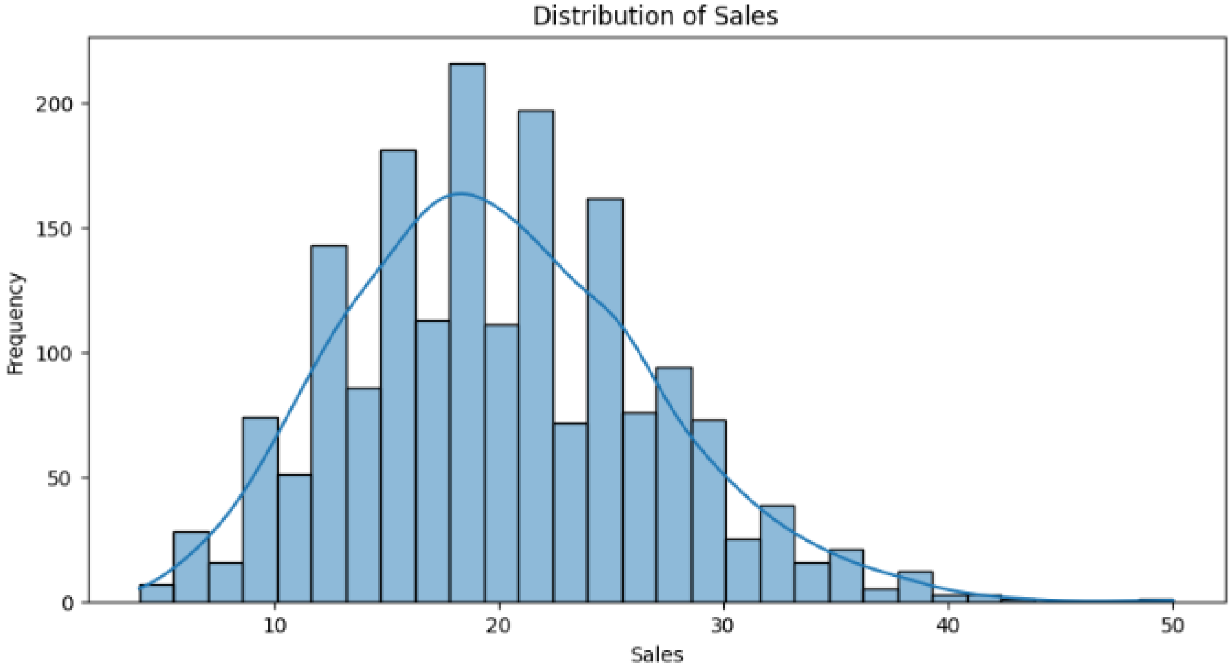

Fig7.4.3 Distribution of sales

Overall Insights:

The results indicate that for this particular sales forecasting problem, the Linear Regression model may be more suitable, possibly due to its ability to capture linear relationships in the data more effectively than the SARIMA model, which is typically used for time series data with seasonality.

CHAPTER 8

CONCLUSION

In this project, we aimed to develop predictive models for sales forecasting using two different approaches: the Seasonal ARIMA (SARIMA) model and Linear Regression. After a comprehensive analysis and comparison of the models, several important **conclusions can be drawn:**

Model Performance:

The Linear Regression model consistently outperformed the SARIMA model across all evaluation metrics, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). This suggests that the Linear Regression model is better suited for the data at hand, capturing the underlying patterns more effectively.

Error Analysis:

The SARIMA model demonstrated a tendency to overestimate sales, leading to a significant overall error. In contrast, the Linear Regression model produced predictions that were closer to actual sales figures, indicating greater reliability and consistency in its forecasts.

Insights and Implications:

The findings indicate that while time series models like SARIMA are valuable for certain types of forecasting tasks, simpler models like Linear Regression can sometimes provide equally or more effective predictions when the relationships in the data are linear or less complex.

Future Work:

There are opportunities for enhancing model performance through:

Feature Engineering: Further exploration of additional features and transformations that may improve prediction accuracy.

Hybrid Models: Combining the strengths of both SARIMA and Linear Regression, possibly through ensemble methods or stacking techniques.

Parameter Tuning: Fine-tuning the hyperparameters of the SARIMA model to optimize its performance on the dataset.

Final Thoughts:

Sales forecasting is critical for effective business planning and decision-making. This project highlights the importance of model selection and evaluation, and it opens avenues for further research in forecasting methodologies that can adapt to varying data patterns.

CHAPTER 9

APPENDICES

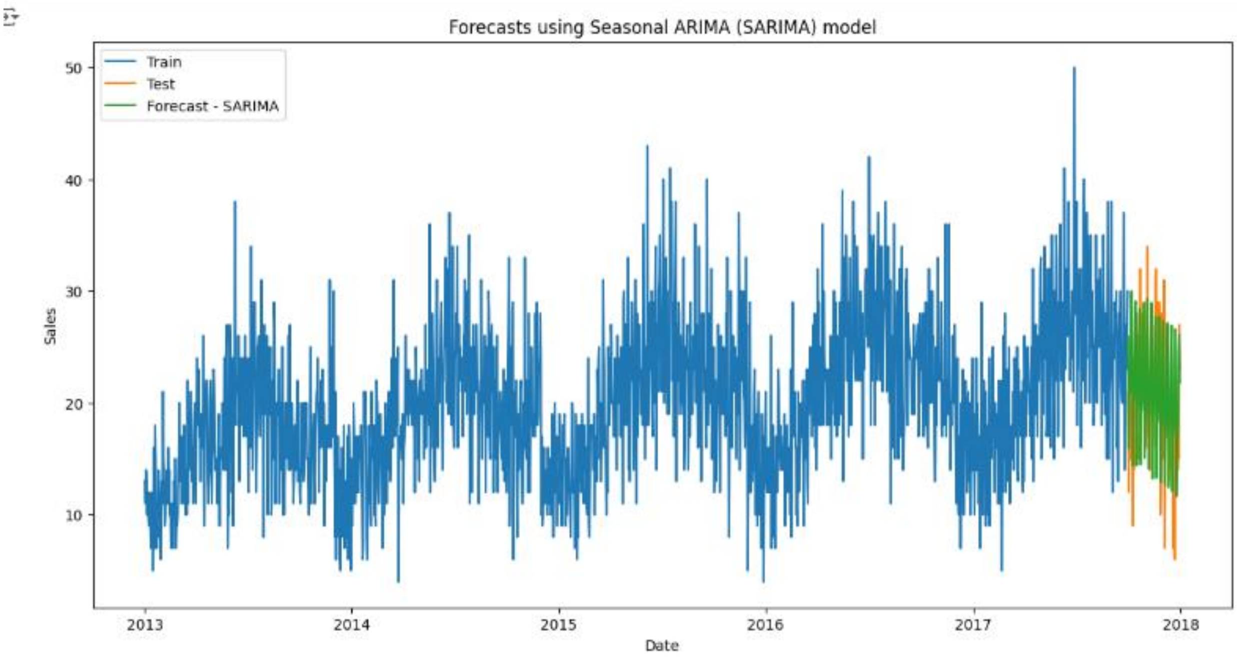

Fig 9.1 Forecasts using Seasonal ARIMA (SARIMA) model

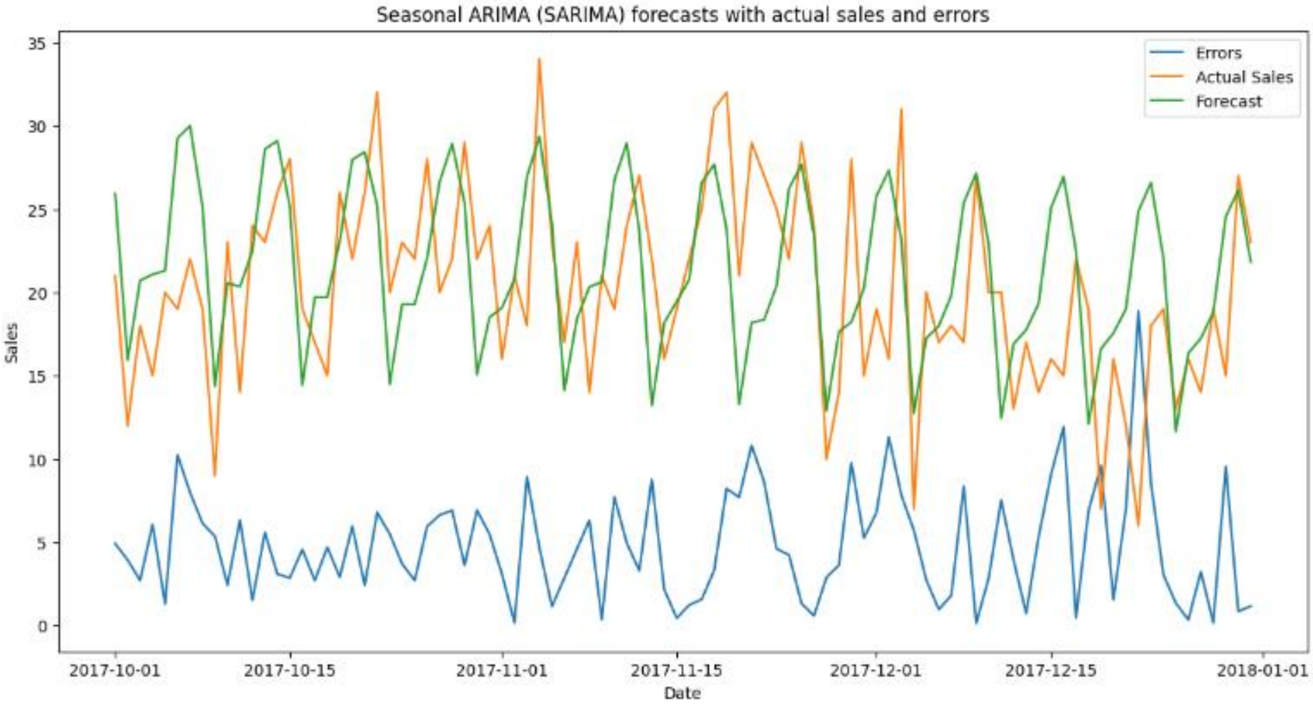

Fig 9.2 Seasonal ARIMA (SARIMA) forecasts with actual sales and error

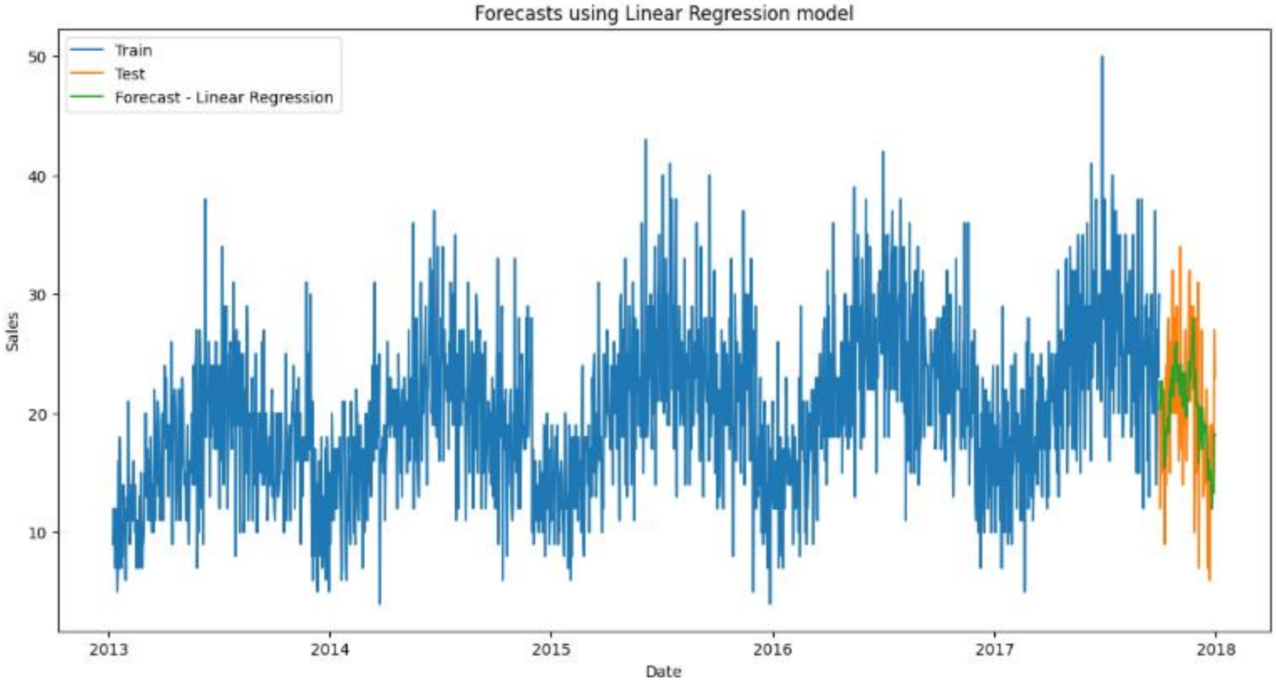

Fig 9.3 Forecasting using Linear Regression model

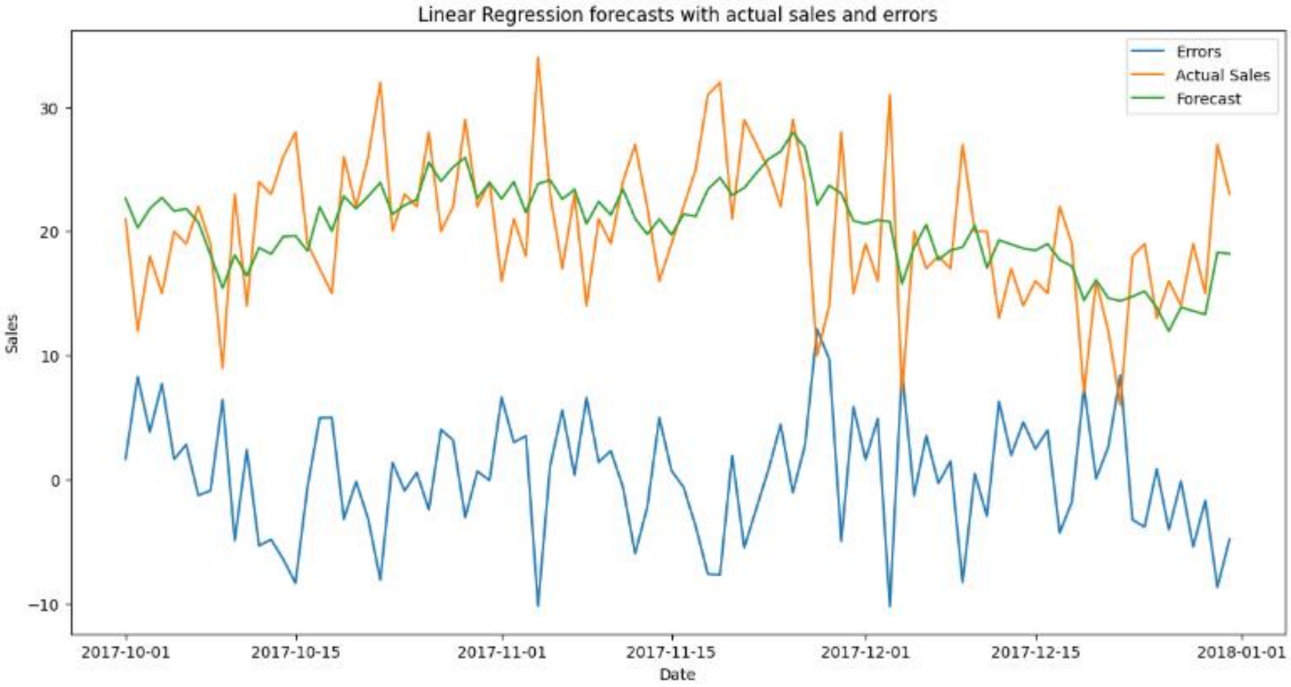

Fig 9.4 Linear Regression forecasts with actual sales and error

CHAPTER 10

REFERENCES

[1] Cheriyan, S., Ibrahim, S., Mohanan, S., & Treesa, S. (2018a). Intelligent sales prediction using machine learning techniques. 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE), 53–58. M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A Comprehensive Survey of Deep Learning for Image Captioning," ArXiv181004020 Cs Stat, Oct. 2018, Accessed: Jul. 20, 2020. [Online]. Available: http://arxiv.org/abs/1810.04020.

[2] Bajaj, P., Ray, R., Shedge, S., Vidhate, S., & Shardoor, D. N. (2020). Sales prediction using Machine Learning algorithms. https://www.semanticscholar.org/paper/54d7b22ce8c266f67a1776e6231e8db4d2c3921b "deep learning - What's the commercial usage of 'image captioning'?," Artificial Intelligence Stack Exchange. https://ai.stackexchange.com/questions/10114/whats-the-commercialusage-of-image-captioning (accessed Jul. 20, 2020).

[3] Wu, C.-S. M., Patil, P., & Gunaseelan, S. (2018). Comparison of different machine learning algorithms for multiple regression on black Friday sales data. *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, *3*, 16–20. S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi, "Composing Simple Image Descriptions using Web-scale N-grams," p. 9.

[4] Elias, N. S., & Singh, S. (2018). *FORECASTING of WALMART SALES using MACHINE LEARNING ALGORITHMS*. https://www.semanticscholar.org/paper/701add1f1eb3a3d211e4c245095bc114f5c6673b P. Kuznetsova, V. Ordonez, A. Berg, T. Berg, and Y. Choi, "Collective Generation of Natural Image Descriptions," in Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Jeju Island, Korea,

Jul. 2012, pp. 359– 368, Accessed: Dec. 24, 2019. [Online]. Available: https://www.aclweb.org/anthology/P12- 1038.

[5] Kaneko, Y., & Yada, K. (2016). A deep learning approach for the prediction of retail store sales. *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, *1*, 531–537. C. Amritkar and V. Jabade, “Image Caption Generation Using Deep Learning Technique,” in 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Aug. 2018, pp. 1–4, doi: 10.1109/ICCUBEA.2018.8697360.

[6] Behera, G., & Nain, N. (2020). A comparative study of big mart sales prediction. In *Communications in Computer and Information Science* (pp. 421–432). Springer Singapore. F. Fang, H. Wang, and P. Tang, “Image Captioning with Word Level Attention,” in 2018 25th IEEE International Conference on Image Processing (ICIP), Oct. 2018, pp. 1278–1282, doi: 29 10.1109/ICIP.2018.8451558.