A PROJECT REPORT

on

"Sales Quantity Forecasting using SARIMA and XGBoost"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN

COMPUTER SCIENCE AND ENGINEERING

BY

AKSHYAYANAND PANI 2105520

UNDER THE GUIDANCE OF

Dr. Abinas Panda

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

April 2025

PROJECT REPORT

on

“Sales Quantity Forecasting using SARIMA and XGBoost”

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR’S DEGREE IN
COMPUTER SCIENCE AND ENGINEERING

BY

AKSHYAYANAND PANI 2105520

UNDER THE GUIDANCE OF

Dr. Abinas Panda

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAE, ODISHA -751024

April 2025

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is to certify that the project entitled

"Sales Quantity Forecasting using SARIMA and XGBoost "

submitted by

AKSHYAYANAND PANI 2105520

is a record of Bonafide work carried out by him, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2024-2025, under our guidance.

Date: 9/04/2025

Dr. Abinas Panda
Project Guide

Acknowledgements

I am profoundly grateful to **Dr. Abinas Panda** of KIIT UNIVERSITY for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. .....

AKSHYAYANAND PANI

ABSTRACT

This project report outlines the design, development, and evaluation of an innovative prediction pipeline for forecasting product sales quantities. The pipeline integrates a classical statistical forecasting model (SARIMA) and a modern machine learning algorithm (XGBoost) through an ensemble averaging approach. Initially, raw sales data is rigorously cleaned and pre-processed. A SARIMA model is then employed to capture temporal patterns in the data, while an XGBoost model leverages advanced feature engineering techniques for prediction. The final output is generated by averaging the individual predictions from these two models, providing improved accuracy and robustness.

Keywords: Time Series Forecasting, SARIMA, XGBoost, Data Preprocessing, Ensemble Methods, Sales Prediction

Contents

Chapter 1

Introduction

In the rapidly evolving business environment, accurate demand forecasting plays a pivotal role in ensuring optimal inventory management, resource allocation, and production planning. The challenges associated with predicting future sales require both statistical and machine learning techniques, as each approach addresses different aspects of the data characteristics. In this project, a hybrid forecasting pipeline is developed by merging the capabilities of the Seasonal AutoRegressive Integrated Moving Average (SARIMA) model with those of the XGBoost regression model.

The report discusses the need to clean and preprocess raw sales data, a task vital for reliable model training. The SARIMA model efficiently identifies and exploits seasonal patterns within the time series data, while the XGBoost model leverages feature engineering techniques to capture non-linear interactions between predictors. The final prediction is achieved by averaging outputs from these methods, thus improving forecasting accuracy by mitigating model-specific biases. This integrated strategy forms the basis for a robust predictive system applicable to dynamic sales environments

Chapter 2

Basic Concepts/ Literature Review

This chapter provides a detailed overview of the underlying concepts, the rationale behind the methodological choices, and a review of relevant literature.

2.1 Forecasting Techniques

- **SARIMA (Seasonal ARIMA):**
  SARIMA extends the ARIMA model to include seasonal effects, which is crucial for time series data exhibiting periodic trends. The model involves identifying optimal parameters using diagnostic tools like the Autocorrelation (ACF) and Partial Autocorrelation (PACF) plots, and testing for stationarity using the Augmented Dickey-Fuller (ADF) test. Its strength lies in its interpretability and ability to model seasonality explicitly.
- **XGBoost (Extreme Gradient Boosting):**
  XGBoost is an advanced ensemble learning technique that builds on decision trees for regression tasks. It is particularly effective in handling datasets with complex feature interactions and non-linear relationships. Key advantages include its scalability, regularization capabilities to prevent overfitting, and high prediction speed. In this project, XGBoost incorporates feature scaling and encoding to ensure the robustness of predictions.

2.2 Data Preprocessing and Feature Engineering

- Effective data preprocessing is essential for model accuracy. The raw sales data, which includes inconsistencies such as extra spaces and varying date formats, is cleaned using Python's pandas library. Categorical variables are transformed using Label Encoding, while numerical variables are standardized using StandardScaler. This meticulous preparation not only improves model performance but also ensures that both SARIMA and XGBoost models are provided with coherent and clean data sets. Advanced feature generation, such as extracting month-year information for seasonality, plays a crucial role in enhancing the predictive capabilities of the models.

Chapter 3

Problem Statement / Requirement Specifications

3.1 Project Planning

The project is divided into four major phases:

1. **Data Cleaning:**
   Script: `Cleaning.py`
   The raw sales data is read, cleaned, and pre-processed. Key operations include date parsing, column name standardization, missing value treatment, and feature encoding.
2. **Time Series Forecasting using SARIMA:**
   Script: `Sarima.py`
   The cleaned data is aggregated on a daily basis. A SARIMA model is then fitted after performing stationarity tests, and 90-day forecasts are generated.
3. **Machine Learning Prediction using XGBoost:**
   Script: `XGB_P1.py` along with `Future_features_90.py`
   An XGBoost regression model is trained using engineered features from the cleaned data. Future feature generation is applied to predict sales for the next 90 days.
4. **Final Prediction through Averaging:**
   Script: `Final.py`
   The outputs of the SARIMA and XGBoost models are merged on their respective dates, and the final prediction is calculated by averaging the forecasted quantities from both models.

3.2 Project Analysis (SRS)

The project requirements include:

- **Accurate Data Cleaning:**
  Ensuring data integrity through thorough preprocessing.
- **Robust Forecasting Models:**
  Implementing both a statistical model (SARIMA) for its interpretability and a machine learning model (XGBoost) for its non-linear predictive power.
- **Seamless Integration:**
  Merging model outputs to obtain a final forecast that leverages the strengths of both approaches.
- **Scalability:**
  Designing a pipeline that can be extended or modified with minimal disruption.

3.3 System Design

3.3.1 Design Constraints

- **Data Quality:**
  The pipeline is designed to be robust against noise and inconsistencies in raw data.
- **Resource Efficiency:**
  Considering computational limitations, particularly when training the SARIMA model, the code is optimized for performance.
- **Modularity:**
  Each phase of the project is implemented as an independent module, facilitating debugging and future enhancements.

3.3.2 System Architecture (UML) / Block Diagram

A simplified block diagram of the system is as follows:

- **Input Layer:**
  Raw Sales Data (CSV file)
- **Preprocessing Module:**
  `Cleaning.py` cleans and preprocesses the data, outputting `cleaned_data.csv`.
- **Forecasting Modules:**
  - **Time Series Forecasting:**
    `Sarima.py` aggregates the data and generates 90-day forecasts.
  - **Machine Learning Prediction:**
    `XGB_P1.py` trains an XGBoost model and predicts future sales using feature data from `Future_features_90.py`.
- **Ensembling Module:**
  `Final.py` merges the forecasts from both models and outputs the final averaged prediction (`final_90.csv`).
- **Output Layer:**
  Forecast CSV files that record predictions at each stage.

Chapter 4

Implementation

This chapter details the end-to-end implementation of the forecasting pipeline, including technical methodology, testing, and result analysis.

4.1 Methodology / Proposal

- **Data Preprocessing:**
  The script `Cleaning.py` is responsible for reading the raw CSV, stripping extraneous spaces from column names, converting date formats, and encoding categorical variables. Numerical features are scaled to improve model performance. The cleaned dataset (`cleaned_data.csv`) is then stored for subsequent processing.
- **Time Series Forecasting with SARIMA:**
  The `Sarima.py` script aggregates the cleaned data to a daily frequency and checks for stationarity using the ADF test. Using ACF/PACF plots to inform model parameter selection, a SARIMA model is fitted on the data. The selected parameters (e.g., order (0, 1, 3) and seasonal order (0, 2, 2, 30)) are used to forecast the next 90 days, with the forecasts output to `sarima_output_90.csv`.
- **Machine Learning Forecasting with XGBoost:**
  In `XGB_P1.py`, the XGBoost model is configured with carefully tuned hyperparameters (including tree depth, learning rate, and regularization terms) to model complex relationships in the data. Future features required for predictions are generated by `Future_features_90.py`, ensuring the new data is pre-processed in the same way as the training data. The XGBoost predictions are saved as `xgboost_output_90.csv`.
- **Final Ensembling:**
  The `Final.py` script reads the prediction outputs from both the SARIMA and XGBoost models. By aligning predictions based on matching dates and averaging the forecasted sales quantities, a final refined forecast (`final_90.csv`) is produced.

4.2 Testing / Verification Plan

Testing was performed at various stages:

- **Unit Testing:**
  Individual scripts (e.g., date conversion in `Cleaning.py`, stationarity testing and model fitting in `Sarima.py`) were tested separately to ensure correctness.
- **Integration Testing:**
  Each module was integrated sequentially to ensure the correct flow of data from preprocessing to final prediction. Output CSVs were manually inspected for consistency and accuracy.
- **Performance Testing:**
  The execution time for SARIMA model training and prediction was logged to assess resource efficiency. Additionally, the convergence of the XGBoost model was monitored during training.

A sample test plan table:

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Data Cleaning	Raw CSV with format inconsistencies	Cleaned CSV with consistent date and numeric formats	A `cleaned_data.csv` with correct formatting
T02	SARIMA Forecast Generation	Daily aggregated data with seasonal effects	Forecast model generates 90-day predictions	Non-empty `sarima_output_90.csv` with plausible forecasts
T03	XGBoost Prediction	Preprocessed and encoded data	XGBoost produces 90-day forecast predictions	Non-empty `xgboost_output_90.csv`
T04	Final Output Merge	Matching date indices from both models	Correct merge and averaging of forecasts	A valid `final_90.csv` with averaged values

4.3 Result Analysis

- **SARIMA Model Results:**
  The time series plots showed the observed daily sales alongside the forecasted trend. Confidence intervals were displayed for the forecasts, giving insights into prediction uncertainty.
- **XGBoost Model Outcomes:**
  The regression output from the XGBoost model was evaluated against historical sales values. Scatter plots and residual graphs confirmed the model's accuracy and highlighted areas for further refinement.
- **Ensemble Forecast:**
  Final results obtained from averaging the predictions of the two models were compared with historical trends to ensure alignment. Visualizations in the report (screenshots/plots) illustrate the ensemble's improved performance over the individual models.

4.4 Quality Assurance

To ensure reliability, quality assurance steps included:

- Rigorous code review and adherence to coding standards.
- Validation of data consistency at each processing stage.
- Cross-verification of model predictions using statistical tests and visual diagnostics.
- Documentation of parameter tuning for both forecasting models, which facilitates reproduction of results in future iterations.

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

The project successfully demonstrates a hybrid approach to sales forecasting by integrating a classical time series model (SARIMA) with an advanced machine learning algorithm (XGBoost). The ensemble averaging strategy harnesses the strengths of both models, resulting in improved forecasting accuracy over a 90-day period. The pipeline developed is modular, scalable, and reproducible, enabling its potential adaptation to various forecasting scenarios in a dynamic business environment.

5.2 Future Scope

Future improvements could focus on:

- **Enhanced Hyperparameter Optimization:**
  Implementation of automated hyperparameter tuning (e.g., grid search, Bayesian optimization) to further refine model performance.
- **Incorporation of Exogenous Variables:**
  Integrating additional factors such as market trends, seasonal promotions, and economic indicators to better capture demand fluctuations.
- **Real-Time Forecasting Deployment:**
  Extending the pipeline into a real-time forecasting system with continuous data updates and automated re-training.
- **Advanced Ensembling Techniques:**
  Exploring weighted averaging or stacking ensembles to improve prediction accuracy further.