# || Jai Baba Ki || || Jai Shri RadheyKrishna ||

Machine Learning-Based Sales Forecast and Prediction

Submitted by:   
ANMOL BANSAL

Enrolment no.: E23CSEU0431

Submitted to:  
Dr. Sushmita Das(Assistant Professor)

July-Dec 2024 (odd semester)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING



**Candidate’s Declaration:**

**I hereby declare that the project report entitled "Machine Learning-Based Sales-forecast and Predictions submitted for partial fulfillment of the requirements for the degree/diploma of [Degree/Program Name] is my original work. This work has been carried out under the guidance of Dr. Susmita Das, Assistant Professor.**

**I affirm that this report has not been submitted for any other degree or diploma at any other university or institution. All sources of information and references used in this project are duly acknowledged.**

**I understand that any act of plagiarism, if detected, will result in disqualification and disciplinary action as per the regulations.**

Date: 17th November 2024

Student’s Name: Anmol Bansal

Enrolment number: E23CSEU0431

**Acknowledgment:**

I would like to express my sincere gratitude to Dr. Susmita Das, Assistant Professor, for her invaluable guidance, encouragement, and support throughout the duration of this project. Her expertise and insights have been instrumental in shaping the direction of my work and enhancing my understanding of machine learning applications in sales forecast and prediction. I am deeply thankful for her constructive feedback, which has helped me improve and refine my project. This project would not have been possible without her mentorship, and I am grateful for the opportunity to work under her supervision.

Project Report Index

1. Abstract

* Brief Overview of the project objectives, methodology, and key results.

1. Introduction
2. Related Work

* Overview of previous studies or projects on sales predictions (if applicable), and comparison with your approach.

1. Methodology
2. Hardware and Software Requirements
3. Experimental Results
4. Conclusion
5. Future Scope
6. GitHub Repository

Link to The Complete Project code: [Github](https://github.com/AnmolBansal10/Sales-Forecasting-and-Predictions.git)

# Abstract

Accurate sales forecasting plays a pivotal role in ensuring efficient resource management, strategic decision-making, and overall business success. This project implements a machine learning-based approach to predict future sales trends using historical data. The model transforms raw sales data into a supervised learning problem through advanced feature engineering techniques and uses regression modeling to uncover patterns in the data. Evaluation metrics such as Mean Squared Error (MSE) and R-squared (R²) provide insights into the model's performance, ensuring reliability. The system is designed to adapt to seasonal and temporal dependencies, making it a robust tool for businesses seeking data-driven strategies to optimize operations and maximize profitability.

# Introduction

Sales forecasting is a cornerstone of modern business analytics, enabling organizations to anticipate future demand, manage inventory effectively, and plan marketing campaigns. Traditional forecasting methods often struggle to capture complex, non-linear relationships in data, limiting their accuracy and applicability in dynamic market environments. By leveraging machine learning techniques, this project addresses these limitations, creating a model capable of handling time series data with intricate dependencies. The project focuses on preprocessing historical sales data, engineering features to highlight trends and seasonality, and applying a regression-based approach for prediction. This work bridges the gap between traditional statistical methods and modern machine learning, delivering a solution that is both practical and scalable for real-world applications.

# Related Work

Numerous studies have focused on Sales prediction using machine learning algorithms like Neural Networks, SARIMAX etc. In this project, we compare models like Linear Regression, Random Forest Regressor, and XGBoost. Through, Flask web interface, we also deploy our model for practical usage.

# Methodology

## 1. Import Libraries and Load Data

The code begins by importing essential libraries for data manipulation, preprocessing, visualization, and modeling. These libraries include:  
- numpy: For numerical operations.  
- pandas: For data manipulation and loading.  
- sklearn: For preprocessing and evaluation metrics.  
- matplotlib: For data visualization.  
  
The dataset is then loaded using pandas from a CSV file (train.csv') and parsed as a time series by setting the 'date' column as the index.

## 2. Feature Engineering

To improve model accuracy, the code performs feature engineering by transforming the time series data into a supervised learning format. Specifically, it calculates the difference between consecutive sales values to stabilize the series and make it stationary. This is done by:  
  
- Calculating the difference between current and previous month's sales.  
- Removing NaN values introduced by the differencing.  
- Creating a supervised learning problem using a specified lag (12 months in this case) through the create\_supervised function.  
  
The create\_supervised function shifts the series to generate lagged columns, enabling the model to learn patterns from previous sales values.

## 3. Train-Test Split and Scaling

The transformed data is divided into training and testing sets to allow evaluation of model performance. The last 12 months are used as the test set, and the remaining data is used for training. MinMaxScaler is then applied to scale the features to a range of -1 to 1, which helps in stabilizing the model training and often improves accuracy.

## 4. Define Dates for Predictions and Actual Sales

To compare model predictions with actual sales, dates corresponding to the test set period are extracted from the dataset. These dates are stored in a DataFrame, along with actual sales values for the last 13 months, for easy plotting and visualization.

## 5. Model Training (Example - Linear Regression)

A linear regression model is initially trained on the dataset. The model learns to predict sales based on past values and is evaluated using common metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared. The predictions are inverse-transformed to the original scale to interpret the results correctly.

## 6. Model Training (Alternative - Random Forest and XGBoost)

The code can be modified to use other models such as Random Forest Regressor or XGBoost Regressor, which are non-linear models that often capture complex patterns more effectively than linear regression. These models can provide higher accuracy on time series with non-linear trends.  
  
To switch models, the relevant library (e.g., XGBoost) needs to be imported, and the code block for model initialization and training needs to be updated.

## 7. Model Evaluation

Once predictions are generated, they are evaluated against the actual sales data using several metrics:  
- Root Mean Squared Error (RMSE): Measures the standard deviation of residuals (prediction errors).  
- Mean Absolute Error (MAE): Measures the average magnitude of errors in a set of predictions.  
- R-squared: Represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.  
  
These metrics provide insight into the model's accuracy and predictive power.

## 8. Visualization

The code uses Matplotlib to visualize the model's performance by plotting the actual vs. predicted sales values. This visual comparison helps in quickly understanding how well the model fits the observed data.  
  
A line plot is created, with actual sales represented as a continuous line and predictions as a dashed line, clearly showing any discrepancies.

future sales values. The project emphasizes the importance of data preprocessing, feature engineering, and model evaluation in building an effective forecasting model.

# Hardware/Software Required

Software

* Python 3.x: For scripting and development.
* Flask: For creating the web application.

Hardware

* A standard PC or laptop with atleast 8GB RAM and a dual-core processor is sufficient to run the model and application.

## Experimental Results

model evaluation: Once predictions are generated, they are evaluated against the actual sales data using several metrics: RMSE, mae and r-squared.

visualization: A line plot is created, with actual sales represented as a continuous line and predictions as a dashed line, clearly showing any discrepancies.

## Conclusion

The sales forecasting provides a structured approach to predicting future sales based on past data. It demonstrates the importance of feature engineering, data scaling, and model selection in building an effective time series model. The code can be further enhanced by experimenting with different models or optimizing model parameters to improve forecasting accuracy.

This project demonstrates a structured approach to time series forecasting using machine learning. The Random Forest Regressor captures patterns from historical sales data to predict

## Future Scope

1. STRENGTH lr: lINEAR REGRESSION IS MOST EFFECTIVE FOR IDENTIFYING ADN MODELING LINEAR RELATIONSHIPS BETWEEN VARIABLES.
2. EXPLORING HYBRID MODELS: FUTURE RESEARCH SHOULD FOCUS ON HYBRID MODELS THAT COMBINE STRENGETHS OF BOTH LINEAR REGRESSION AND RANDOM FOREST FOR BETTER ACCURACY.
3. INTEGRATING DATA SOURCES: INCORPORATING DIVERSE DATA SOURCES CAN SIGNIFICANTLY ENHANCE PREDICTION ACCURACY AND MODEL ROBUSTNESS.

## GitHub Link

For the Complete Project, Including code, data and model files. Please visit the GitHub Repository: [Sales-Forecasting and Prediction](https://github.com/AnmolBansal10/Sales-Forecasting-and-Predictions.git).