MACHINE LEARNING AND DEEP LEARNING BASED SHORT-TERM FORECASTING OF ORANGE AND COTTON CROP PRICES IN CONTEXT OF INDIAN MARKET

This project report is submitted to

Yeshwantrao Chavan College of Engineering
(An Autonomous Institution Affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)

In partial fulfillment of the requirement

for the award of the degree

Of

Bachelor of Technology in Computer Technology

By

Humanshu Gajbhiye

Susrut Patole

Neha Thakur

Geetika Mahant

Under the guidance of

Prof. (Dr) Nileshsingh V. Thakur

DEPARTMENT OF COMPUTER TECHNOLOGY

Nagar Yuwak Shikshan Sanstha's

YESHWANTRAO CHAVAN COLLEGE OF ENGINEERING,
(An autonomous institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur)
NAGPUR – 441 110
2023-24

CERTIFICATE OF APPROVAL

Certified that the project report entitled **“MACHINE LEARNING AND DEEP LEARNING BASED SHORT-TERM FORECASTING OF ORANGE AND COTTON CROP PRICES IN CONTEXT OF INDIAN MARKET”** has been successfully completed by Humanshu Gajbhiye, Susrut Patole, Neha Thakur, Geetika Mahant under the guidance of Prof. (Dr) Nileshsingh V. Thakur in recognition to the partial fulfillment for the award of the degree of Bachelor of Technology in Computer Technology, ***Yeshwantrao Chavan College of Engineering(An Autonomous Institution Affiliated to Rashtrasant Tukadoji Maharaj Nagpur University).***

Prof. (Dr) Nileshsingh V. Thakur	Prof. Smita R. Kapse	Dr. Rakhi D. Wajgi
(Guide)	(Coordinator)	(HoD)

Signature of External Examiner

Name:

Date of Examination:

DECLARATION

We certify that

a. The work contained in this project has been done by us under the guidance of our supervisor(s).

b. The work has not been submitted to any other institute for any degree or diploma.

c. We have followed the guidelines provided by the institute in preparing the project report.

d. We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references. Further, we have taken permission from the copyright owners of the sources, wherever necessary.

Name of Student	**Signature of Student**
Humanshu Gajbhiye	
Susrut Patole	
Neha Thakur	
Geetika Mahant	

ACKNOWLEDGEMENT

We would like to thank our guide Prof. (Dr) Nileshsingh V. Thakur for thorough guidance in the project. We are extremely grateful and indebted to them for their expert, sincere, valuable guidance and encouragement which was of immense help to us.

We would like to express our sincere gratitude to Dr. R. D. Wajgi, Head of the Department of Computer Technology for her constant encouragement towards the successful completion of our work.

We wish to express our sincere thanks to Dr. U. P. Waghe, the Principal of our college for providing us with all the necessary facilities and the infrastructure without which we would not have been able to complete our project successfully.

We would also like to thank our Project Coordinator Prof. S. R. Kapse, who is also our project guide for her continuous guidance owing to which the project could take shape.

Last, but not the least, we would like to extend our sincere gratitude to Mrs. B. H. Kulkarni, our technical assistant, for her invaluable assistance and support throughout the project.

CONTENTS **Page No.**

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation No.	Abbreviation Name	Definition
1.	ML	Machine Learning
2.	DL	Deep Learning
3.	ARIMA	Autoregressive Integrated Moving Average
4.	LR	Linear Regression
5.	SVR	Support Vector Regression
6.	AR	Auto-Regression
7.	LSTM	Long short-term memory
8.	FB-PROPHET	FaceBook-Prophet
9.	MSE	Mean Square Error
10.	RMSE	Root Mean Squared Error
11.	MAE	Mean absolute error
12.	ADF	Augmented Dickey–Fuller test
13.	PCA	Principal Component Analysis

ABSTRACT

India being majorly agrarian economy, farmers livelihoods are greatly impacted by the pricing of crops like cotton and oranges. These crops prices are frequently unstable and change depending on a number of variables, including the weather, supply and demand, and governmental policy. Farmers and traders find it challenging to plan the timing of their product sales and purchases in light of these changes. In this present study , the aim is to implement a Machine Learning-based model for the short-term forecasting of orange and cotton crop prices in the context of an Indian market. For this purpose, we will use the existing dataset available at Indian Government. We start by performing the exploratory data analysis(EDA) to take care of missing data and to identify the patterns. Later, we will split the data records to training, testing and validation set. We will use Machine Learning and Deep Learning algorithms for this forecasting. Specifically, ARIMA (Auto Regressive Integrated Moving Average), LR (Logistic Regression), SVR (Support Vector Regression), LSTM(Long short-term memory) and Facebook-Prophet will be use. To compare the results, we will also implement the ARIMA model. We are performing this time series analysis based on regional and seasonal aspects. We consider regional aspects by dividing dataset in tiers of urban and rural cities. We will compare the results and accuracy of machine learning models at different geographic levels, such as country, state, and particular cities.

Index Terms- Crop price; ARIMA; LR; short-term forecasting

PROJECT CO-PO MATRIX

CO’s	Statement	PO’s												PSO’s	
		1	2	3	4	5	6	7	8	9	10	11	12	PSO 1	PSO2
1	Acquire the domain knowledge and analyze the implemented model	3	3		2								3		2
2	Design and develop the solution using appropriate tools and techniques for betterment of society and industry	3		3	3	3	3	3	2					3	2
3	Communicate the work done through paper presentation or participation in competition as a team.							2		3	3	3	3		

1. Introduction

1.1 Overview

The agricultural sector plays a pivotal role in India's economy, with crops like oranges and cotton contributing significantly to the country's agricultural output. However, the volatility in crop prices poses challenges for farmers, traders, and policymakers alike. Timely and accurate forecasting of crop prices is crucial for decision-making and risk management in this domain.

In this thesis, we delve into the realm of short-term forecasting for orange and cotton crop prices in the Indian market using advanced Deep Learning techniques. Our objective is to develop and implement a robust forecasting model leveraging the rich dataset available from the Indian Government. The primary methodologies employed include exploratory data analysis (EDA), data preprocessing, and the application of Machine Learning and Deep Learning algorithms such as LR(Linear Regression), SVR(Support Vector Regression), VAR(Vector Autoregression), ARIMA(Autoregressive Integrated Moving Average), Long Short-Term Memory Networks (LSTM) and Prophet.

The thesis unfolds in several phases. Initially, we conduct thorough exploratory data analysis to handle missing data, detect underlying patterns, and gain insights into the historical price trends of oranges and cotton across different regions and seasons. Subsequently, we partition the dataset into training, testing, and validation sets to ensure the reliability and generalization of our forecasting model.

One key aspect of our analysis is the consideration of regional and seasonal variations. We segment the dataset into tiers based on urban and rural areas, recognizing the distinct market dynamics and price fluctuations that occur across different geographical regions. This granular approach enables us to capture localized trends and enhance the accuracy of our forecasting model.

We employ a comparative analysis framework to evaluate the performance and accuracy of our Deep Learning-based model against traditional statistical Machine Learning algorithms. This comparative assessment is conducted at various geographic levels, encompassing national, state-wise, and city-specific perspectives. By juxtaposing the results of different models, we aim to discern the most effective approach for short-term crop price forecasting in the Indian agricultural landscape.

Through this thesis, we seek to contribute valuable insights and methodologies to the field of agricultural economics and predictive analytics. Our research endeavors to empower stakeholders in the agricultural sector with actionable intelligence, aiding them in making informed decisions and mitigating risks associated with price volatility in orange and cotton markets.

1.2 Problem Statement

The project tackles the challenge of accurately forecasting orange and cotton crop prices in India's volatile agricultural market. Key issues include handling complex and incomplete datasets from the Indian Government, selecting appropriate Machine Learning and Deep Learning algorithms such as LR(Linear Regression), SVR(Support Vector Regression), VAR(Vector Autoregression), ARIMA(Autoregressive Integrated Moving Average), Long Short-Term Memory Networks (LSTM) and Prophet to model nonlinear price dynamics, addressing regional and seasonal variations, and comparing Deep Learning's effectiveness against traditional methods. By overcoming these challenges, the project aims to deliver precise short-term forecasts, aiding stakeholders in making informed decisions amidst market uncertainties.

1.3 Thesis Objectives

1. To perform comprehensive exploratory data analysis (EDA) on the Indian Government's dataset to handle complexity and missing data effectively.
2. To select and implement appropriateMachine Learning and Deep Learning algorithms such as LR(Linear Regression), SVR(Support Vector Regression), VAR(Vector Autoregression), ARIMA(Autoregressive Integrated Moving Average), Long Short-Term Memory Networks (LSTM) and Prophet for modeling nonlinear price dynamics and improving forecasting accuracy.
3. To incorporate regional and seasonal aspects into the forecasting model to account for geographical variations and seasonal trends.
4. To evaluate the performance and accuracy of the Deep Learning model against traditional statistical Machine Learning algorithms.
5. To validate the model's effectiveness through rigorous testing and validation procedures using historical data.
6. To provide actionable insights and recommendations to stakeholders based on the forecasted crop prices, aiding in informed decision-making and risk management strategies in the agricultural sector.

1.4 Thesis Contributions

This project contributes significantly to the domain of agricultural economics and predictive analytics in several key ways:

1. **Advanced Forecasting Techniques:** By leveraging Machine Learning and Deep Learning algorithms such as LR(Linear Regression), SVR(Support Vector Regression), VAR(Vector Autoregression), ARIMA(Autoregressive Integrated Moving Average), Long Short-Term Memory Networks (LSTM) and Prophet. The project advances the frontier of short-term

forecasting methodologies. These advanced techniques are tailored to handle the intricate and nonlinear dynamics inherent in orange and cotton crop price data, providing more accurate and reliable forecasts compared to traditional methods.

2. **Data Handling and Preprocessing:** The project addresses the challenge of working with complex and incomplete datasets sourced from the Indian Government. Through rigorous data preprocessing and exploratory data analysis (EDA) techniques, the project ensures that the input data is clean, relevant, and conducive to building robust forecasting models. This aspect of the project is crucial as it lays the foundation for accurate predictions and actionable insights.

3. **Regional and Seasonal Analysis:** A key aspect of the project is its consideration of regional and seasonal variations in crop prices. By segmenting the dataset into tiers based on urban and rural areas and accounting for seasonal fluctuations, the forecasting models developed in this project are better equipped to capture localized trends and provide region-specific forecasts. This level of granularity enhances the practical utility of the forecasts for stakeholders operating in different geographic regions.

4. **Comparative Analysis:** The project conducts a comprehensive comparative analysis between Deep Learning algorithms and traditional statistical Machine Learning methods. This comparative study not only sheds light on the relative strengths and weaknesses of each approach but also helps in identifying the most effective techniques for short-term crop price forecasting in the context of the Indian agricultural market. Such insights are valuable for guiding future research and decision-making processes.

5. **Practical Application:** Ultimately, the project's outcomes have practical implications for various stakeholders involved in the agricultural sector, including farmers, traders, and policymakers. The accurate short-term forecasts generated by the Deep Learning-based models empower these stakeholders to make informed decisions, optimize resource allocation, and manage risks more effectively in the face of market uncertainties and fluctuations. This real-world impact underscores the significance of the project's contributions to the agricultural industry.

2. Review of Literature

2.1 Overview

The literature survey for this project encompasses a thorough exploration of Deep Learning applications in short-term forecasting of orange and cotton crop prices within India's agricultural sector. It reviews methodologies like LSTM, Prophet, and CNN for modeling nonlinear price dynamics and addresses challenges like handling complex datasets, incorporating regional and seasonal variations, and comparing Deep Learning's efficacy with traditional methods like ARIMA. The survey also examines data preprocessing techniques, regional analysis importance, and practical implications of Deep Learning in agricultural forecasting, providing a comprehensive foundation for the project's methodology and analysis.

2.2 Literature Survey

Casper Solheim Bojer , "Understanding machine learning-based forecasting methods: A decomposition framework and research opportunities"[1] , Due to the impressive Practical performance demonstrated in the latest M4 and M5 competitions, as well as in numerous Kaggle competitions, machine learning (ML) methods are becoming more and more popular in the forecasting sector. Due in part to their complexity, knowing why and how these methodologies are effective for predicting is still in its infancy. He describes a framework for regression-based machine learning in this paper that offers academics a common vocabulary and abstraction to support their research. He then goes into detail about using the framework and ablation testing to thoroughly examine their performance. Finally, He utilizes the framework to give a general overview of the regression-based ML forecasting solution space and to highlight potential topics for future study.

Yitong Li , Kai Wu , Jing Liu , "Self-paced ARIMA for robust time series prediction" [2], The ARIMA (autoregressive integrated moving average) model is a widely respected and traditional linear model for time series prediction, known for its effectiveness in various domains. However, its vulnerability to noisy data, leading to instability and reduced performance, has not received sufficient attention. In this study, we introduce the spARIMA framework to enhance the reliability of time series prediction. We implement a sequential training approach in batches, taking into account noise levels and their impact on accurate modeling, with the goal of minimizing noise-related disruptions during training. To leverage the advantages of self-paced learning (SPL), spARIMA integrates a differential prediction model into the ARIMA framework.

Ahmed Tealab , Hesham Hefny , Amr Badr , "Forecasting of nonlinear time series using ANN" [3], Since basic linear time series models often fall short in fully explaining certain aspects of economic and financial data, it becomes crucial to categorize time series based on their linearity characteristics when conducting forecasts. This approach ensures that linear time series continue to be the focal point of both academic and practical research. Predicting nonlinear time series, especially those containing inherent moving average components, using computational intelligence methods like neural networks can be challenging, given that real-world time series often exhibit dynamic behavior, autoregressive elements, and inherited moving average terms. Research focusing on the prediction of nonlinear time series with moving average components is relatively rare. In this study, we demonstrate that conventional neural networks struggle to effectively capture the behavior of nonlinear time series with moving average terms, resulting in subpar forecasting performance.

Steven Elsworth and Stefan Güttel, "Time Series Forecasting Using LSTM Networks: A Symbolic Approach" [4], It proposes a novel method for improving the performance of LSTM networks in time series forecasting. They argue that traditional machine learning methods when trained on raw numerical time series data, face issues such as high sensitivity to hyperparameters and the initial random weight setup. By applying a dimension-reducing symbolic representation to the time series data before feeding it into an LSTM, the authors claim to speed up training times and reduce sensitivity to those factors, without compromising on forecasting performance. They detail the ABBA symbolic representation, the LSTM setup, and the construction of a training set. Moreover, they compare the performance of raw LSTM and ABBA-LSTM on a set of time series, demonstrating that the ABBA-LSTM models train more easily and perform similarly in forecasting tasks.

B. Lindemann, Timo Müller, Hannes Vietz, "A survey on long short-term memory networks for time series prediction" [5], It provides an overview of LSTM networks and their derivatives for predicting time series data. The paper examines Long Short-Term Memory networks and how they are used for time series prediction. LSTMs overcome these limitations of traditional Recurrent Neural Networks by better handling long-term dependencies with specialized structures called as gates. It assesses different LSTM architectures and demonstrates their effectiveness in accurately modeling time-variant systems, particularly emphasizing the usefulness of sequence-to-sequence models for making multi-step ahead predictions. The paper concludes that LSTM architectures are highly effective for precision modeling of the time-variant behavior, with sequence-to-sequence networks being best suited for accurate time series prediction (Lindemann et al., 2021).

Emir Zunic, Kemal Korjenic, Kerim Hodzic, and Dzenana Donko, "Application of Facebook's prophet algorithm for successful sales forecasting based on Real-World Data" [6], The paper presents a robust framework for accurate sales forecasting in retail, leveraging Facebook's Prophet algorithm and a backtesting strategy. It emphasizes the critical role of forecasting in retail operations, influencing inventory management, pricing strategies, and overall competitiveness. The framework's evaluation using real-world data from a retail company in Bosnia and Herzegovina demonstrates its effectiveness in generating reliable sales forecasts. Key methodology steps include data preprocessing, product portfolio selection, and applying Prophet for time series forecasting, with performance metrics like Percentage Error (PE) and Mean Absolute Percentage Error (MAPE) for accuracy assessment. Results show successful monthly and quarterly sales predictions, with potential for practical implementation in retail businesses, although further optimization for shorter observation horizons and automated parameter tuning are suggested for future enhancements.

Dongqing Zhanga, Guangming Zangb, Jing Lia, Kaiping Maa, Huan Liu, "Prediction of soybean price in China using QR-RBF neural network model"[7], In this study, a novel neural network model called the QR-RBF model is introduced, which combines the principles of quantile regression and radial basis functions. This model offers two key capabilities:

1. It characterizes the distribution of soybean price ranges through the use of quantile regression models.
2. It approximates the nonlinear aspects of soybean prices using RBF neural networks.
3. To optimize the parameters of the QR-RBF neural network model, the research proposes a hybrid algorithm called GDGA. GDGA combines the global search capabilities of the genetic algorithm with the local search capabilities of gradient descent.

The analysis of monthly domestic soybean price data in China led to the following conclusions:

1. The hybrid GDGA algorithm performs effectively in optimizing the model.
2. The factors influencing soybean prices vary depending on the price level.
3. Money supply and port distribution prices for imported soybeans are significant factors across multiple quantiles.
4. Domestic soybean production and the consumer confidence index are significant primarily for lower quantiles.

5. Soybean import volume and the consumer price index are significant mainly for higher quantiles.

Kiran M. Sabu, T. K. Manoj Kumarb, "Predictive analytics in Agriculture: Forecasting prices of Areca Nuts in Kerala [8], In this research, a combination of time-series and machine learning models is employed to predict monthly arecanut prices in the Kerala region. The dataset comprises price data spanning from 2007 to 2017, serving as a benchmark for evaluating the performance of three models: SARIMA, Holt Winters Seasonal approach, and the LSTM neural network. The model that exhibited the best fit to the data turned out to be the LSTM neural network model.Fluctuations in the prices of agricultural products have a detrimental impact on a country's GDP. These price changes also carry emotional and financial consequences for farmers who invest years of hard work, only to see their efforts sometimes yield disappointing results.The agricultural supply chain may benefit from price prediction by using it to better manage and minimize the risk of price changes. Predictive analytics is anticipated to address issues facing the average person as a result of the decline in agricultural productivity brought on by uncertain climatic conditions, global warming, etc. India produces a significant amount of arecanuts, with Kerala coming in second place. Due to price volatility and climatic change, farmers in Kerala have recently shifted from arecanut production to other crops.

Jennifer L. Castle , Michael P. Clements, David F. Hendry, "Robust approaches to forecasting "[9], Employing a unique category of robust tools, distinct from equilibrium-correction models, the researchers have explored alternative robust forecasting techniques. These forecasting methods are assessed for their predictive capabilities in scenarios where various potential empirical challenges, including measurement errors, sudden changes, omitted variables, unforeseen shifts in location, and incorrectly included variables subject to shifts, exist at the point of forecast origin. The study involves the computation of forecast biases and error variances, along with an examination of the conditions under which these approaches are most likely to yield successful results.

Foteini Kyriazi , Dimitrios D. Thomakos , John B. Guerard , “Adaptive learning forecasting, with applications in forecasting agricultural prices” [10], They present an innovative forecasting approach termed "adaptive learning forecasting," which supports both forecast averaging and learning from forecast errors. They delve into its theoretical characteristics and illustrate that, in certain scenarios, it enhances the mean squared error (MSE) a priori. The research also reveals that the learning rate, contingent on previous forecast errors, follows a nonlinear pattern. This methodology finds wide-ranging applications and can even enhance the MSE when compared to the most basic benchmark models.To exemplify the strategy's applicability, the researchers employ data on agricultural prices for various agricultural commodities and real GDP growth figures for relevant countries. They consider numerous forecasting models, including both univariate and bivariate ones related to output and productivity. This is particularly important because agricultural price time series are typically short and display irregular cyclical patterns that correlate with economic performance and productivity. Their findings confirm the effectiveness of the new approach and the predictability of agricultural price movements.

Yegnanew A. Shiferaw, “An analysis of East African tea crop prices using the MCMC approach to estimate volatility and forecast the in-sample value-at-risk”[11], Agriculture plays a significant role in the economies of Eastern African nations. However, due to factors such as economic crises, climate change, and fluctuating food and fuel prices in the region, the prices of key crops experience substantial variations. This article focuses on one of the region's most important cash crops, tea, which exhibits significant price volatility.To assess this volatility and predict in-sample value-at-risk (VaR) for tea price returns, the researchers employed Markov-switching GARCH (MS-GARCH) models with diverse heteroscedastic functions and error distributions. The study analyzed monthly tea auction prices in USD, spanning from January 1980 to June 2022 (specifically, the Mombasa auction). The parameters of the MS-GARCH model were estimated using the Markov Chain Monte Carlo (MCMC) method within a Bayesian framework.The findings indicated that the three-regime EGARCH skewed Student-t model performed best in estimating volatility. When considering factors like heteroscedastic functions, fat-tailed distributions, asymmetry, and regime shifts, this approach proved most effective for VaR evaluation. As a result, investors in the East African tea industry should consider employing regime-switching GARCH models to manage and mitigate volatility and associated risks effectively.

J. Scott Armstrong , Kesten C. Green , Andreas Graefe, "Golden rule of forecasting: Be conservative"[12], This essay presents the Golden Rule of forecasting as its central and unifying concept. Being conservative when forecasting is the Golden Rule. A cautious forecast is in line with our collective understanding of the past and present. To exercise prudence, forecasters should actively search for and utilize all relevant information pertaining to the matter, including knowledge about techniques that have demonstrated effectiveness in the specific context. The Golden Rule logically leads to the deduction of twenty-eight rules. An assessment of the evidence turned up 105 publications with experimental comparisons; 102 of them are in favor of the recommendations. The average forecast error increased by more than two-fifths when a single guideline was disregarded. The Golden Rule should always be followed, especially in complex and unclear situations when bias is likely.

Yuehjen E. Shao , Jun-Ting Dai, "Integrated Feature Selection of ARIMA with Computational Intelligence Approaches for Food Crop PricePrediction"[13], The supply of three major food crops, namely rice, wheat, and corn, is rapidly diminishing due to global climate change, limited land availability, and a rapidly growing population. Predicting the prices of these essential food crops has become a subject of considerable attention. While numerous feature selection techniques (FSMs) have been developed for integrated forecasting models, a significant challenge arises from the absence of future values for these critical factors, making predictions using these factors impossible.In this study, an Autoregressive Integrated Moving Average (ARIMA) model is employed as the foundational statistical method (FSM) within computational intelligence (CI) models to forecast the prices of these three vital food crops. The ARIMA model is chosen because it can identify important self-predictor variables with calculable future values. The proposed integrated forecasting models encompass not only ARIMA but also include Support Vector Regression (SVR), Multivariate Adaptive Regression Splines (MARS), and Artificial Neural Networks (ANNs). The research conducts a comparative analysis and discussion on the prediction accuracies of the proposed integrated model in contrast to ARIMA, ANN, SVR, MARS, and other existing models.

Thomas Dimpfla, Robert C. Jungb, Michael Fladc,"Price discovery in agricultural commodity markets in the presence of futures speculation"[14], In the quest to identify the primary drivers of price discovery in these commodities, the researchers investigate the relationship between spot and futures prices of various products, including corn, wheat, soybeans, soybean meal, soybean oil, feeder cattle, live cattle, and lean hogs. Their analysis reveals compelling evidence indicating that these commodity prices are predominantly shaped by activities in the spot market, facilitated by a recently developed and distinctive information-sharing mechanism. In fact, less than 10% of the price discovery, following the framework proposed by Hasbrouck, is attributed to the futures market. These findings lend support to their argument that speculative trading in futures markets does not have a lasting detrimental impact on commodity prices.

Liege Cheung , Yun Wang, Adela S.M. Lau , Rogers M.C. Chan, "Using novel clustered 3D-CNN model for improving crop future price prediction"[15], To forecast trends in food supply, distribution, and pricing, traditionally statistics was used. To create a predictive model, researchers employed statistical inference to identify the correlations between the data. Crucially, because of the non-stationary nature of the crop price trend and the influence of several dimensions, classic time series forecasting methods like ARIMA are unable to accurately anticipate it. The constraints of conventional statistical methods for prediction can be solved by the CNN model since it can work with non-stationary data and learn non-linearity by modifying the model parameters.

V. Sneha; V. Bhavana , "Sugarcane Yield and Price Prediction Using Forecasting Models"[16], Sugarcane is a critical commercial crop in India, and the agricultural sector has benefited significantly from advancements in technology, particularly in the realm of machine learning (ML). This cutting-edge technology has proven invaluable to farmers by providing comprehensive recommendations and insights to enhance crop quality and productivity while reducing farming losses. Before planting sugarcane, accurate estimations of yield and market prices are essential for making profitable decisions.

Machine learning techniques such as Decision Tree Regressor, Multi Linear Regression, Random Forest, Adaboost Regressor, and Lasso Regression are employed to predict sugarcane yield. Additionally, an ARIMA model is used to forecast sugarcane prices. Several factors, including historical sugarcane yields in specific locations, rainfall patterns, and the state in which sugarcane is cultivated, all contribute to the accurate prediction of sugarcane yields. Meanwhile, sugarcane price predictions are based on time series analysis of historical price data.

Yung-Hsing Peng, Chin-Shun Hsu, Po-Chuang Huang, "Developing Crop Price Forecasting Service Using Open Data from Taiwan Markets"[17], From the agricultural industry's perspective, the market price of a specific crop reflects its current demand. To facilitate monitoring of agricultural prices, the Council of Agriculture (COA) has established an official website that provides open data on daily market prices across more than 15 local markets in Taiwan, encompassing over 100 different crops. In parallel, the Institute for Information Industry (III) has introduced the smart agri-management platform (S.A.M.P.), a comprehensive cloud-based solution for agri-business.This research paper focuses on the development of a crop price forecasting service within S.A.M.P., inspired by the availability of open data on crop prices. The service automatically retrieves historical price data from the official website, utilizing it as a training dataset, and employs well-established time series analysis algorithms for price forecasting. The study employs three techniques: Partial Least Square (PLS), Artificial Neural Network (ANN), and Autoregressive Integrated Moving Average (ARIMA). The performance of these four algorithms is compared using price data obtained from the First Fruit and Vegetable Wholesale Market in Taipei, focusing on crops like cauliflower, watermelon, bok choy, and cabbage. Based on experimental data, PLS and ANN exhibit smaller percentage errors in their forecasting accuracy.

B Chaitra; K Meena, “Forecasting Crop Price using various approaches of Machine Learning”[18], India's economy relies heavily on agriculture. India has a sizable chunk of arable land. Numerous crops are in high demand abroad. Other than software, one of the main exports from India is food. But because of unpredictable weather patterns, a lack of human resources, and poor crop selection, farmers are unable to turn a profit. The agriculture profession is gradually losing relevance as a result of urbanization as well. For accurate Yield and Crop Price Prediction. This issue has led numerous researchers to identify the challenges and employ a diverse range of machine learning methods, including Autoregressive Integrated Moving Average (ARIMA), Decision Trees, Long Short-Term Memory (LSTM), K Nearest Neighbors (KNN), and others. These techniques assist farmers in making informed decisions regarding crop selection, ultimately enabling them to achieve optimal yields and profits.

Juliana Ngozi Ndunagu, Eyiyemi.Helen Aderemi, Rasheed Gbenga Jimoh, Joseph Bamidele Awotunde, “Time Series: Predicting Nigerian Food Prices using ARIMA Model and R-Programming”[19], In Nigeria, there has been persistent price fluctuation in the majority of food products. Factors contributing to this include insecurity and insurgency, inadequate storage facilities, seasonal price variations, inconsistent government policies, efforts to control COVID-19, limited access to credit, a lack of modern farming technologies, and insufficient contemporary agricultural equipment.This study employed the ARIMA model to project future prices and conducted a comparative cost analysis for four different food items: beans, onions, tomatoes, and yams. The research focused on two of Nigeria's six geopolitical zones, specifically the North-Central and North-West regions. Raw data for the years 2017 and 2018, measured in kilograms (Kg), were sourced from the National Bureau of Statistics (NBS). The data was transformed into a time series dataset using R Studio, and the stationarity of the series was assessed through a Unit Root Test employing the KPSS test (where $p<0.05$ indicates stationarity).The results derived from the forecasted values indicated an upward trend in the prices of food commodities over time, suggesting that the ARIMA model is suitable for price forecasting. It is recommended that measures be implemented to alleviate the high cost of food prices currently experienced in Nigeria.

Jinlai Zhang, Yanmei Meng, Jin Wei, Jie Chen, and Johnny Qin, "A Novel Hybrid Deep Learning Model for Sugar Price Forecasting Based on Time Series Decomposition" [20], The paper presents a novel hybrid deep learning model that aims to improve sugar price forecasting through the integration of a time series decomposition technology called Empirical Mode Decomposition and a hyperparameter optimization algorithm, the Tree of Parzen Estimators. The model is evaluated using a dataset of London Sugar Futures prices from April 2010 to May 2020. The effectiveness of the combined EMD and TPE techniques is demonstrated, showing that the proposed hybrid model outperforms other models in forecasting sugar prices. The authors focus on addressing the nonstationarity and nonlinearity of sugar prices for accurate prediction, with emphasis on the suitability of their model for policymakers given the significant impact of sugar prices on everyday life and market dynamics .

Jingyi Shen and M. Omair Shafiq, “Short-term stock market price trend prediction using a comprehensive deep learning system”[21], The article "Short-term Stock Market Price Trend Prediction Using a comprehensive deep learning system" by Jingyi Shen and Omair Shafiq explores the use of deep learning in predicting stock market trends. The authors collected two years of data from the Chinese stock market and created a customized feature engineering and deep learning-based model for predicting price trends. Their solution includes pre-processing of data, multiple feature engineering techniques, and a customized long short-term memory-based deep learning model, which achieved high accuracy in stock market trend prediction and outperformed existing machine learning models. The research contributes to the fields of financial and technical analysis of stock markets (Shen & Shafiq, 2020).

Xiao Han, Fangbiao Liu, Xiaoliang He, and Fenglou Ling, “Research on Rice Yield Prediction Model Based on Deep Learning” [22], It is a study that explores the application of deep learning techniques in predicting rice yield, which is crucial for agricultural planning and food security, especially in China where rice is a significant crop. The traditional methods of yield estimation are described as destructive and labor-intensive, leading to the need for a more efficient, non-destructive alternative. The paper discusses the development and potential of deep learning algorithms, particularly convolutional neural networks, for image recognition and processing in the context of remote sensing images. Various regression models are experimented with to simulate key factors affecting rice yield. The study assesses the performance and significance of these models to improve the accuracy of rice yield prediction.

Johnathon Shook, Tryambak Gangopadhyay, “Crop yield prediction integrating genotype and weather variables using deep learning”[23], This paper explores the use of a Long Short Term Memory—Recurrent Neural Network model for predicting crop yield. This model incorporates genetic relatedness and weekly weather parameters to assess and forecast the response of soybean genotypes across different environments. It is noted for performing better than other machine learning models such as Support Vector Regression and regression models like LASSO. Additionally, the study introduces a temporal attention mechanism to the LSTM models to enhance interpretability and provide plant breeders with valuable information regarding critical periods in the growing season. The research underscores the importance of considering both genetic and environmental factors for improving crop yield predictions amidst climate variability and change

Jie Sun, Liping Di , Ziheng Sun , Yonglin Shen and Zulong Lai , "County-Level Soybean Yield Prediction Using Deep CNN-LSTM Model"[24], The presented research focuses on soybean yield prediction at the county level using a deep CNN-LSTM model, leveraging remote sensing data and machine learning techniques. Yield prediction is crucial for agricultural planning, market strategy, and risk management. Prior studies have recognized the growing importance of remote sensing data in yield prediction, with notable advancements achieved through Deep Learning (DL), specifically employing Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. Existing experiments have demonstrated that CNNs excel at extracting spatial features, while LSTMs capture phenological characteristics, both integral for accurate yield predictions. However, few studies have integrated these models. The proposed CNN-LSTM model, trained on crop and environmental variables like weather data and MODIS imagery, surpasses the individual CNN and LSTM models in both end-of-season and in-season soybean yield prediction. This innovative approach, facilitated by Google Earth Engine (GEE), showcases potential applications for enhancing yield predictions in other crops such as corn, wheat, and potatoes at fine spatial scales, contributing to the broader field of precision agriculture.

Kavita Jhajhariaa, Pratistha Mathura ,Sanchit Jaina, Sukriti Nijhawana, "Crop Yield Prediction using Machine Learning and Deep Learning Techniques"[25], The goal of this paper is to apply the crop selection approach to assist farmers in resolving issues related to crop yield. The research conducted by the authors involved implementing different machine-learning techniques to predict crop yield in the Rajasthan state of India, focusing on five specific crops. Their findings revealed that the Random Forest algorithm outperformed the other methods, which included Support Vector Machines, Gradient Descent, Long Short-Term Memory, and Lasso Regression. The performance metrics used for validation were R-squared (R2), Root Mean Squared Error, and Mean Absolute Error, where the Random Forest model yielded 0.963 R2, 0.035 RMSE, and 0.0251 MAE, indicating it was the most accurate algorithm for predicting crop yields in their study.

Alexandros Oikonomidisa, Cagatay Catalb and Ayalew Kassahuna, "Deep learning for crop yield prediction"[26], This study presents a comprehensive review of the current utilization of Deep Learning techniques in predicting crop yields. Through a Systematic Literature Review (SLR), 456 relevant studies were initially identified, with 44 primary studies chosen for in-depth analysis following rigorous selection and quality assessment criteria. The analysis focused on key aspects such as motivations, targeted crops, utilized algorithms, features, and data sources. Convolutional Neural Network (CNN) emerged as the predominant algorithm, displaying superior performance in Root Mean Square Error (RMSE).

Wenxiu Hu, Huan Liu , Xiaoqiang Ma,and Xiong Bai, "The Influence and Prediction of Industry Asset Price Fluctuation Based on the LSTM Model and Investor Sentiment" [27], This paper presents a forecasting model for industrial assets using LSTM that considers investor sentiment and historical trading data to predict future market trends. It concludes that models incorporating investor sentiment yield better forecasting results than those excluding it, demonstrating the importance of considering investor sentiment in industry asset forecasting.The study selected the LSTM model for industry asset forecasting and found that it was not only more effective in making predictions but also operated efficiently with large datasets.

Can Yang, Junjie Zhai , and Guihua Tao, "Deep Learning for Price Movement Prediction Using Convolutional Neural Network and Long Short-Term Memory" [28], This paper introduces a deep learning approach that predicts the direction of stock price movements using historical financial time series data. The framework employs a convolutional neural network for feature extraction and a long short-term memory network for prediction purposes. It utilizes a three-dimensional CNN to process time series data, technical indicators, and stock index correlations, converting indicators into deterministic trends and ranking indices by Pearson correlation. A fully connected network trains the CNN to produce feature vectors for the LSTM. The combined CNN and LSTM yield predictions that surpass current models in accuracy. In this work, we suggested a hybrid model to forecast the direction of stock price that combines CNN and LSTM motion.

Usharani Bhimavarapu, Gopi Battineni and Nalini Chintalapudi , "Improved Optimization Algorithm in LSTM to Predict Crop Yield"[29], This paper proposes to accurately predict crop yields using deep learning models, specifically proposing an improved optimizer function (IOF) to enhance prediction accuracy when incorporated into a long short-term memory (LSTM) model. The proposed IOF was compared with eight standard learning methods, demonstrating superior performance in handling underfitting and overfitting issues during training, resulting in smaller training errors. Performance metrics including correlation coefficient (r), root mean square error (RMSE), and mean absolute error (MAE) were used for evaluation, yielding r of 0.48, RMSE of 2.19, and MAE of 25.4. Evaluation against actual crop yields showed that the proposed IOF in LSTM significantly improved prediction accuracy, with reduced RMSE compared to other models such as CNN, RNN, and basic LSTM, indicating its superiority in crop yield prediction.

Frank Emmert-Streib et al, “An Introductory Review of Deep Learning for Prediction Models With Big Data ”[30], Deep learning, within the expansive realm of big data analytics, stands as a transformative force reshaping predictive modeling paradigms. Its application extends across diverse domains, from finance to healthcare, yet is not devoid of challenges. Interpretability remains a key concern, as the complexity of deep learning models often obscures the underlying decision-making processes. Moreover, the efficacy of these models hinges significantly on the volume and quality of data available. While traditional machine learning techniques may suffice for smaller datasets, deep learning thrives on large-scale, high-dimensional data, leveraging its capacity to discern intricate patterns and correlations. Advanced deep learning architectures such as deep reinforcement learning, graph convolutional neural networks (CNNs), and Variational Autoencoders (VAEs) emerge as frontrunners, showcasing their prowess in tasks ranging from game strategy optimization to molecular structure prediction. As we navigate through the era of big data, understanding the nuanced requirements of different data types and domains becomes imperative, guiding the selection and deployment of appropriate deep learning methodologies for predictive modeling and decision support systems.

Lorenzo Menculini et al, “Comparing Prophet and Deep Learning to ARIMA in Forecasting Wholesale Food Prices” [31], It explores different methods for forecasting sale prices for a food wholesaler in Italy. It compares traditional ARIMA models, Prophet (a forecasting tool by Facebook), and deep learning models, including Long Short-Term Memory Networks and Convolutional Neural Networks. The study finds that ARIMA and LSTM models perform similarly, while the combination of CNNs and LSTMs achieves the best accuracy but requires more time to tune. Prophet is noted to be quick and easy to use but less accurate than the other models. The research emphasizes the practical applications of these models for automating pricing tasks and highlights their importance in business capacity planning, especially in the context of challenges such as the COVID-19 crisis. The flexibility of these models allows them to be applied to various products, aiming to aid companies in managing their price lists more effectively.

Hasan Tercan and Tobias Meisen , "Machine learning and deep learning based predictive quality in manufacturing: a systematic review"[32], A comprehensive review spanning from 2012 to 2021 delves into the realm of machine learning (ML) and deep learning (DL) applications for predictive quality in manufacturing, elucidating how these technologies harness manufacturing process data for data-driven predictions regarding product quality. This review meticulously categorizes existing research based on manufacturing processes, datasets utilized, and the ML models employed, unraveling significant insights, identifying similarities, and shedding light on gaps within the solution approaches. The overarching goal of this review is to provide a panoramic view of the current landscape of predictive quality research, delineating the methods under investigation, elucidating their limitations, and illuminating potential future research directions in this burgeoning field, thereby contributing to a deeper understanding of the intersection between advanced technologies and manufacturing excellence.

Khulood Albeladi, Bassam Zafar, and Ahmed Mueen , "A Novel Deep-learning based Approach for Time Series Forecasting using SARIMA, Neural Prophet and Fb Prophet"[33], The research delves into the exploration of three distinct Python libraries tailored for time series forecasting, with a specific focus on predicting the Gulf stock exchange market's trends. The libraries and models under scrutiny comprise SARIMA, Neural Prophet, and Fb Prophet. These models are systematically applied to datasets to forecast future values, employing a comprehensive array of evaluation parameters such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and overall accuracy. The study unveils nuanced insights regarding the efficacy of each model for different forecasting tasks, showcasing SARIMA's proficiency in providing broad-spectrum predictions across the market trends. On the other hand, the Prophet models, including Neural Prophet and Fb Prophet, demonstrate notable competence when dealing with smaller datasets that exhibit pronounced seasonal patterns. This discernment of model suitability based on dataset characteristics and forecasting objectives underscores the nuanced nature of time series forecasting and accentuates the importance of employing the right tools for distinct analytical tasks within the realm of financial market prediction.

Oskar Triebe, Hansika Hewamalagec, Polina Pilyuginad, Nikolay Laptevb, Christoph Bergmeirc, Ram Rajagopal, "NeuralProphet: Explainable Forecasting at Scale "[34], NeuralProphet is a forecasting framework that serves as an improved successor to Facebook's Prophet. It addresses the need for a system that can handle the growing volume of time series data with a solution that is both scalable and provides explainable forecasts, suitable for making business and operational decisions. Designed with usability in mind, NeuralProphet is built on PyTorch and supports standard deep-learning methodologies, making it easy for developers to modify and extend. NeuralProphet introduces local context to forecasting through auto-regression and covariate modules, which distinguish it from its predecessor, Prophet, which lacks local context and is difficult to extend due to its underlying Stan backend. The new framework retains the design philosophy of Prophet but enhances forecasting performance by incorporating additional model components. When compared to Prophet, NeuralProphet performs better on a range of real-world datasets, particularly for short to medium-term forecasts, showing improvements in accuracy between 55 to 92 percent.

2.3 Tabular Representation of Literature Survey

Table 2.3: Table of Literature Review

Ref. No	Work Carried Out	Methodology Used	Evaluation Parameters	Datasets Used	Claims By Author	Our Findings
1	Mapping and comparing ML methods	Framework development, method mapping, performance assessment.	Mapping, comparison, ablation testing for evaluation.	Kaggle	Common language, improved forecasting methods.	Complexity understanding, needs further exploration.
2	Extended ARIMA, introduced spARIMA, diversity selection.	Sequential training, diversity selection in spARIMA.	Generalization, efficiency on noisy data.	12 datasets for time series forecasting.	spARIMA enhances time series forecasting.	Improvements for complex trend information.
3	Comparative analysis of time series forecasting methods.	Application of various machine learning strategies.	Accuracy, robustness, and computational efficiency assessment.	Diverse time series datasets from different domains.	Superior forecasting accuracy with machine learning approaches.	Insufficient Local Learning Exploration ,Limited Adaptive Model Integration
4	Hybrid EMD-TPE-LSTM Sugar Forecasting.	EMD-TPE-LSTM Hybrid Forecasting Model.	MAE, MAPE, RMSE metrics.	London Sugar Futures (2010-2020).	Outperforms existing models significantly	Limited hyperparameter optimization in forecasting.
5	LSTM network analysis	Sequence-to-sequence models	Time series accuracy	Time series data	LSTM effectiveness	LSTM superiority
6	Sales forecasting framework	Facebook's Prophet algorithm	Forecast accuracy metrics	Real-world sales data	Accurate sales forecasting	Reliable sales predictions
7	QR-RBF neural network model development	Hybrid GDGA optimization algorithm	Model optimization effectiveness	Monthly domestic soybean price data in China	GDGA optimization effectiveness, Factors influencing soybean prices	QR-RBF model effectiveness, Significant factors for soybean prices
8	Price Forecasting Research	Time Series, ML Models	Model Performance Metrics	Price Data 2007-2017	LSTM Model Superiority	Confirm LSTM's Effectiveness

9	Exploring robust forecasting techniques	Employing distinct robust tools	Forecast biases, error variances	Various empirical scenarios	Successful results in challenging scenarios	Effective in handling empirical challenges
10	Adaptive learning forecasting.	: Forecast averaging, learning from errors.	Mean squared error (MSE), learning rate pattern.	Agricultural prices, real GDP growth.	Enhances MSE, effective in short time series.	Confirmed effectiveness, predictability of price movements.
11	Analyzing East African tea crop prices.	Employed MS-GARCH models with MCMC.	Volatility estimation, VaR prediction.	Monthly tea auction prices (USD).	Regime-switching GARCH models effective for VaR.	MS-GARCH with EGARCH skewed Student-t model best for tea price volatility and VaR estimation.
12	Analyzing forecasting guidelines.	Literature review and evidence assessment.	Forecast error and guideline adherence.	Not specified	Golden Rule effectiveness in forecasting.	Golden Rule's impact on forecast accuracy.
13	Integrated forecasting models	ARIMA with CI approaches	Prediction accuracies comparison	Food crop price data	Improved price prediction	Enhanced forecasting accuracy
14	Price discovery analysis.	Spot and futures price relationship.	Price discovery attribution.	Various commodity prices.	Spot market dominance.	Futures impact minimal.
15	3D-CNN model for crop price.	Statistical inference, CNN.	Non-stationary data handling.	Crop price correlations.	CNN benefits in prediction.	CNN effective for non-stationary trends.
16	Sugarcane yield and price prediction.	Machine learning techniques, ARIMA.	Yield and price accuracy.	Historical yield and price data.	ML benefits in agriculture.	ML aids in sugarcane predictions.
17	Crop price forecasting service development.	PLS, ANN, ARIMA.	Forecasting accuracy.	Taiwan market prices.	Open data utility.	PLS and ANN accurate.
18	Crop price forecasting with ML.	ARIMA, Decision Trees, LSTM, KNN.	Yield and profit optimization.	Agricultural data.	ML aids in decision-making.	ML optimizes crop outcomes.
19	Nigerian food price prediction.	ARIMA, R-Programming.	Price trend analysis.	Nigerian food price data.	ARIMA suitability.	ARIMA effective for price forecasting.
20	Hybrid EMD-TPE-LSTM Sugar Forecasting.	EMD-TPE-LSTM Hybrid Forecasting Model.	MAE, MAPE, RMSE metrics.	London Sugar Futures (2010-2020).	Outperforms existing models significantly	Limited hyperparameter optimization in forecasting.

21	Crop yield prediction using DNN.	DNN with advanced optimization , Stacked deep neural networks.	RMSE, correlation coefficient.	2018 Syngenta Crop Challenge.	DNN outperformed other methods.	Lack of explainable models.
22	Rice Yield Prediction	Deep Learning Regression , Convolutional Neural Networks, U-net, Seg Net, Refinet	Prediction Performance, Significant Analysis	Remote Sensing Images	Accurate Yield Estimation	Not specified in snippets provided.
23	Integrating genotype and weather data for crop yield prediction.	Long Short Term Memory—Recurrent Neural Network (LSTM-RNN) with a temporal attention mechanism.	Comparison against Support Vector Regression (SVR) and LASSO models, focusing on prediction accuracy.	Soybean genotype data along with weekly weather variables.	LSTM-RNN outperforms SVR and LASSO for crop yield prediction.	Genetic and weather integration improves crop yield forecasting.
24	CNN-LSTM for soybean yield.	GEE-based tensor workflow , CNN-LSTM.	RMSE, R2, Percent Error.	MODIS SR, MODIS LST, Daymet weather, USDA yield.	CNN-LSTM outperforms CNN, LSTM.	Insufficient features for prediction. Challenge in combining multisource data..
25	CropYield Prediction using ML and DL	Random Forest, SVM, Gradient Descent,LSTM	RMSE , MAE	official website of the Government.	ML model outperformed DL models	More DL models to be tested for better results
26	Deep learning for crop yield prediction	LSTM, CNN,Deep Learning approach	MAPE, MAE, and MSE	Images and vegetation indices	CNN outperformed DNN and LSTM	Risk of overfitting due to lack of training dataset
27	The Influence and Prediction of Industry Asset Price Fluctuation Based on the LSTM Model	Long Short Term Memory (LSTM)	Prediction Performance, DIF, DEA	The data was obtained through the sector asset data stations of the major websites	LSTM outperformed other methods.	Lack of explainable models.
28	Deep Learning for Price Movement Prediction Using CNN and LSTM	CNN and LSTM	PPMCC, correlation coefficient.	The data can be downloaded from Yahoo Finance (https://finance.yahoo.com/).	framework outperforms state-of-the-art models	Insufficient features for prediction.

29	Improved Optimization Algorithm in LSTM to Predict Crop Yield	LSTM, CNN, RNN	MAE, RMSE	2001 to 2020 from government websites of Andhra Pradesh.	IOFLSTM can outperform the CNN, RNN, and LSTM	Black box model limitation.
30	Review of deep learning for prediction models with big data..	Deep learning techniques applied to big data analytics.	Interpretability, volume, and quality of data.	Big data sets across various domains.	Deep learning is transformative but has challenges in interpretability and data requirements.	Deep learning excels with large-scale data but needs interpretability improvements.
31	Comparison of ARIMA, Prophet, and deep learning for forecasting food prices.	ARIMA, Prophet, LSTM, CNN for forecasting.	Accuracy, time complexity.	Wholesale food price data in Italy.	LSTM-CNN combination is most accurate but time-consuming; Prophet is quick but less accurate.	LSTM-CNN is accurate but time-consuming; Prophet is quick but less precise.
32	Systematic review of ML and DL for predictive quality in manufacturing.	ML and DL techniques for predictive quality.	Manufacturing processes, datasets, ML models. .	Manufacturing process data	Comprehensive review of predictive quality research.	Comprehensive review of predictive quality research.
33	Exploration of SARIMA, Neural Prophet, and Fb Prophet for time series forecasting.	SARIMA, Neural Prophet, Fb Prophet for forecasting.	MSE, RMSE, accuracy.	Gulf stock exchange market data.	SARIMA is broad-spectrum, Prophet models handle seasonal patterns well.	SARIMA good for broad predictions, Prophet models for seasonal patterns.
34	Development of NeuralProphet for scalable and explainable forecasting.	NeuralProphet for time series forecasting.	Forecasting performance, scalability, explainability.	Real-world time series data.	NeuralProphet improves on Prophet in accuracy and scalability.	NeuralProphet enhances forecasting accuracy and scalability compared to Prophet.

2.4. Patent Search

A patent search is a methodical inquiry conducted to identify existing patents relevant to a particular invention or technology. This process includes examining databases and records to evaluate the originality and distinctiveness of an idea. It assists innovators in determining if their concept is novel, avoiding potential infringement issues, and making well-informed decisions regarding the patentability and market feasibility of their invention.

Table 2.4: Patient Search Table

Patent Application No.	Title of Patent	Existing Solutions (Abstract of Patent)
CN103577581B	Agricultural product price trend forecasting method	The technique of agricultural product price trend forecasting described in the invention includes the following steps. Step 1: computer-acquired article relevant to agricultural commodity price and with a forecasting standpoint;Step 2: Duplicate elimination is completed on the collected articles;Extract and save the article's important element in step three.Step 4: The location of the agricultural product-related area mentioned in the article is found;Step 5: Quantify and preserve agricultural products according to the given predictability viewpoint after analyzing expert opinions using text mining technology;Step 6: Utilizing the model established so that agricultural product price is carried out trend prediction, the trend prediction viewpoint delivering time, agricultural product affiliated area, agricultural product sort, and quantization according to article is carried out using microcomputer modeling.
CN105205099B	A kind of agricultural	The steps included in the technique of agricultural product price analysis that the present invention

	product price analysis method	relates to are as follows:Information about the types of agricultural products is gathered using one assembled classifier of pre-trained search engines;The default commodities trading website is crawled to obtain the geographic location information of the supplier for each category of agricultural commodity as well as the pricing data.It is divided based on the area where agricultural goods are cultivated , with each agricultural product kind carrying out agricultural production and obtaining information on the area where agricultural goods are cultivated for each kind.
WO2018232845A1	Smart agriculture management method and system	The present invention relates to a smart agriculture management system and method, the system comprising: monitoring soil nutrient data and moisture data; acquiring historical weather data and predicted weather data; acquiring historical price data; comparing the nutrient data with a nutrient content standard value and comparing the moisture data with a moisture content standard value; processing and analyzing the historical price data of the crop and the historical weather data. The system consists of a control center, a soil monitoring module, an information-gathering module for the weather, and an acquisition module for prices.

3. Work Done

3.1. Overview

The thesis project involves acquiring and preprocessing complex datasets from the Indian Government for accurate forecasting of orange and cotton crop prices in India's agricultural market. Deep Learning algorithms like LSTM, Prophet, and CNN were selected and fine-tuned to model nonlinear price dynamics, accounting for regional and seasonal variations. A comparative analysis with traditional methods was conducted to evaluate forecasting accuracy. The models were validated using testing datasets, and performance metrics like MSE and RMSE were analyzed. The project's practical applications include aiding stakeholders in informed decision-making amidst market uncertainties, emphasizing the importance of accurate short-term forecasts in the agricultural sector.

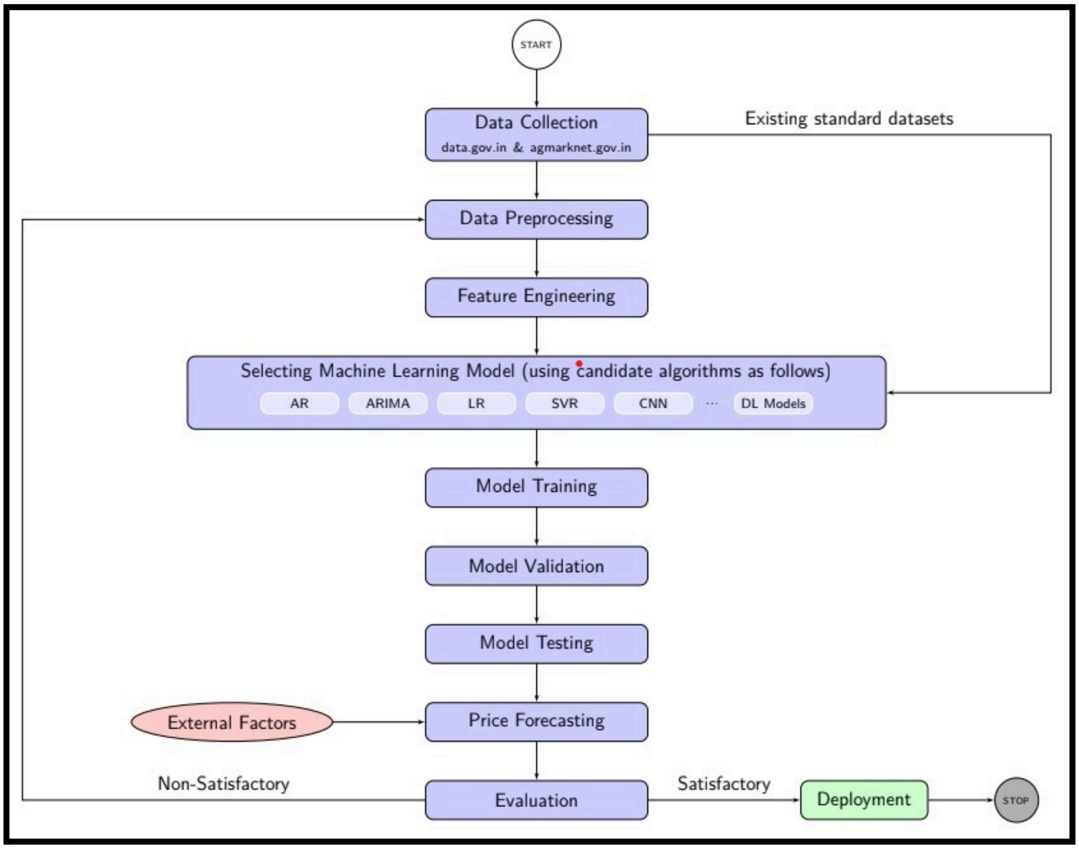


Fig 3.1. Block Diagram

3.2. Data Collection

The data for this project was meticulously sourced from India's Open Government Data Platform and Agricultural Market databases. Our data collection spanned a comprehensive five-year period, ensuring that our model could generalize effectively while also aligning with the computational capacity of our infrastructure. Specifically, we focused on the prices of oranges and cotton from 2018 to 2022, considering this timeframe as an intervention period for our analysis. The dataset was structured to designate the data from 2023 for training and validation purposes.

In terms of data attributes, our dataset comprises eight key features: state, district, market, commodity, variety, arrival_date, min_price, and max_price. The variable of interest, our target for forecasting, is the modal_price. To ensure the reliability and accuracy of our analysis, we diligently preprocessed the dataset to address any missing values in the min_price and max_price columns before proceeding with model building. This meticulous approach to data sourcing and preparation forms the foundation for robust and dependable forecasting outcomes in our project. After importing datasets we have 58329 rows, 8 columns in the Orange Dataset and 106747 rows and 8 columns in the Cotton Dataset.

3.3. Exploratory Data Analysis (EDA)

The Exploratory Data Analysis (EDA) process begins with a comprehensive data summary, including the number of records, attributes, and data types, and calculating basic statistics like mean, median, standard deviation, and range for numerical attributes to understand data distribution and central tendencies. Data visualization involves creating histograms and box plots to identify skewness, kurtosis, and outliers, as well as time series plots to detect trends, seasonal patterns, and anomalies, and correlation heatmaps to visualize relationships between numerical features. Handling missing values involves various imputation methods such as mean, median, mode, and nearest neighbor imputation, with a comparative analysis to assess their impact on data integrity and model performance. Feature analysis includes selecting the most relevant features for predicting crop prices using techniques like PCA and correlation analysis, and feature engineering to create new features based on domain knowledge, such as lagged price values, regional economic indicators, and weather patterns.

Identifying patterns and insights focuses on investigating seasonal trends and regional differences in crop prices to incorporate these variations into the forecasting models, and anomaly detection to identify unusual patterns or price spikes/drops due to external influences like policy changes, market disruptions, or climatic events. By conducting thorough EDA, the project addresses the complexities and challenges of the dataset, ensuring a robust foundation for modeling and forecasting crop prices. The insights gained from EDA inform the selection and tuning of appropriate Machine Learning and Deep Learning algorithms, ultimately aiming to deliver precise short-term forecasts to support stakeholders in the Indian agricultural market.

3.3.1. **Data Summary**

The data summary provides an overview of the imported datasets, which includes the following:

1. Shape of the DataFrames: The shape of the DataFrame is printed, indicating the number of records (rows) and attributes (columns) in each dataset.

   Orange Dataset: The dataset contains 58,329 records and 8 attributes.

   Cotton Dataset: The dataset contains 106,747 records and 8 attributes.

2. Attributes: Each dataset has 8 attributes, which typically include variables such as *price, market, state, district, variety, min_price, max_price, and modal_price.*

3.3.2. **Data Visualization**

In the EDA process for the project, data visualization plays a crucial role in understanding the underlying patterns, trends, and anomalies in the datasets. Here, we focus on visualizing the variations in the orange and cotton datasets. The results indicate significant variations in the cotton data and negligible variations in the orange data.

The data visualization results indicate that the cotton dataset exhibits significant variations in prices, reflecting the impact of external factors such as market demand, supply chain disruptions, and climatic conditions. In contrast, the orange dataset shows negligible variations, suggesting more stable pricing. These insights are crucial for developing accurate forecasting models, as the higher variability in cotton prices may require more sophisticated models like LSTM and Prophet to capture the nonlinear dynamics, while simpler models might suffice for the more stable orange prices.

3.3.3. **Handling Null Values**

In the Exploratory Data Analysis (EDA) for forecasting orange and cotton crop prices, handling missing values is a crucial step to ensure data integrity and improve model accuracy. The missing values in both datasets were identified using the *isnull().sum()* method, which provides a count of missing values in each column. The EDA process for handling missing values involved identifying and filling the gaps using appropriate imputation methods. For both the orange and cotton datasets, missing values in *min_price* and *max_price* were addressed, ensuring the datasets were complete and ready for further analysis and modeling. These steps help maintain the integrity of the data and enhance the reliability of the forecasting models.

3.3.4. **Feature Analysis**

To refine the feature analysis using only numerical features from dataset, we will exclude categorical features and focus on *min_price, max_price,* and *modal_price.* First step is to identify the numerical and categorical data and then drop the categorical data. Feature analysis steps:

- Descriptive Statistics:

  Calculation of basic statistics like *mean, median, standard deviation,* and *range* for each numerical feature to understand their distributions.

- Correlation Analysis:

  Computing the correlation matrix to understand the relationships between *min_price, max_price,* and *modal_price.*

- Pairplot Visualization:

  Using pair plots to visualize the relationships between *min_price, max_price,* and *modal_price* graphically.

- Feature Importance:

Utilizing feature importance techniques like feature selection using *Principal Component Analysis (PCA)*. Feature importance Quantifies the importance of each feature in predicting the target variable *(modal_price)*. It helps in feature selection, allowing for focusing on the most influential features in model training.

3.3.5. **Identifying patterns and Insights**

Identifying patterns and gaining insights from the Exploratory Data Analysis (EDA) is crucial for understanding the underlying dynamics of the orange crop price dataset. Here are the steps and insights derived from the EDA process:

3.3.5.1. **Time Series Analysis:**

- Trend Analysis: Plotting the modal price of oranges over time reveals any long-term trends. For example, an upward or downward trend may indicate overall market conditions.
- Seasonal Patterns: Examining seasonal variations in modal prices can uncover patterns like higher prices during certain months or seasons due to supply-demand dynamics or weather conditions.
- Anomaly Detection: Identifying outliers or abnormal price fluctuations can provide insights into external factors impacting prices, such as natural disasters or policy changes.

3.3.5.2. **Price Range Analysis:**

- Min and Max Price Comparison: Analyzing the relationship between min_price and max_price can reveal pricing strategies, market dynamics, and potential outliers or extreme price points.
- Modal Price Analysis: Comparing modal prices across different markets, districts, or states can highlight regional pricing trends and disparities.

3.3.5.3. **Correlation Analysis:**

- Correlation between Features: Investigating correlations between min_price, max_price, and modal_price can indicate how these variables influence each other. For instance, a strong positive correlation between min_price and modal_price may suggest that lower minimum prices lead to lower modal prices on average.

- Correlation with External Factors: Exploring correlations with external factors like weather conditions, crop yields, or economic indicators can provide insights into price determinants beyond internal market factors.

3.3.5.4. **Price Distribution Analysis:**

- Histograms and Box Plots: Analyzing the distribution of modal prices through histograms and box plots can provide insights into price variability, skewness, and the presence of outliers.
- Price Range Stability: Assessing the stability of price ranges (*min_price to max_price*) over time or across regions can indicate market volatility and risk levels.

These methods help uncover long-term trends, market dynamics, relationships between variables, hidden patterns, and actionable insights for informed decision-making and business growth.

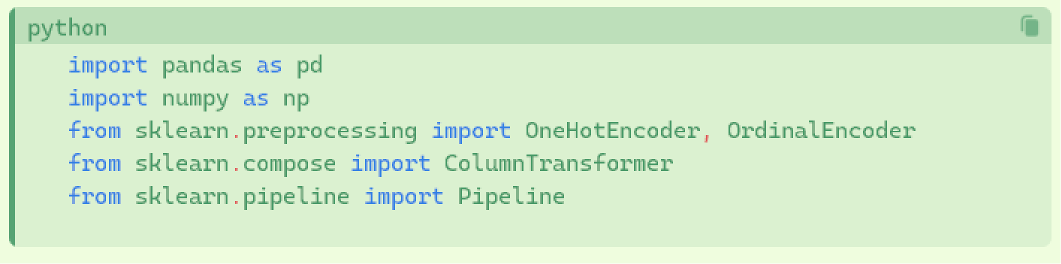
3.4. Data Preprocessing

In this project, we employed multiple strategies for handling missing values, including imputation methods such as mean, median, mode, and nearest neighbor. Categorical variables were converted into numerical format using encoding techniques such as one-hot encoding, label encoding, and ordinal encoding.

3.4.1. **One Hot Encoding**

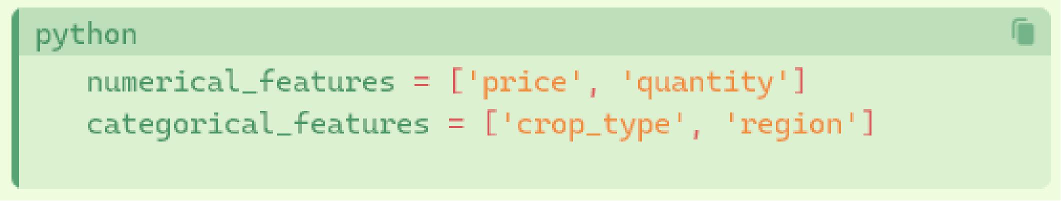
One-hot encoding is a technique used in machine learning to convert categorical variables into numerical features. Performed steps for one hot encoding are:

1. Importing Libraries and Transformers: Start by importing the necessary libraries such as pandas, numpy, and the required transformers from *sklearn.preprocessing*. Code Snippet:



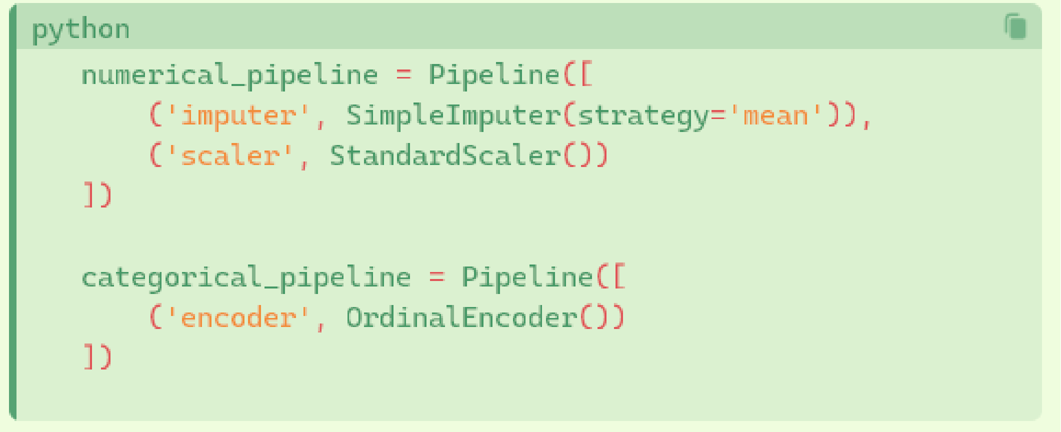
```python
import pandas as pd
import numpy as np
from sklearn.preprocessing import OneHotEncoder, OrdinalEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
```

2. Defining Numerical and Categorical Features of Crops: Identify which columns in your dataset are numerical and categorical. For instance:



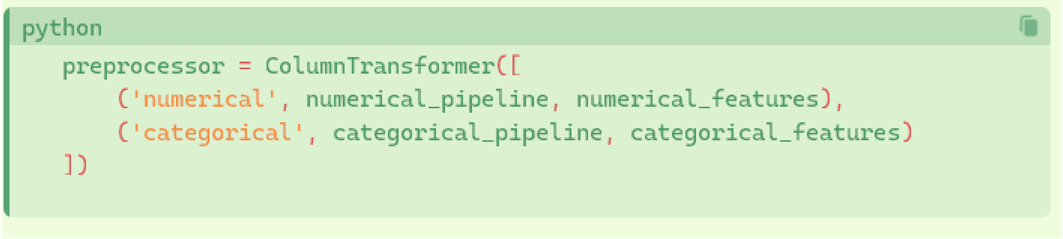
```python
numerical_features = ['price', 'quantity']
categorical_features = ['crop_type', 'region']
```

3. Setting up Pipelines for Numerical and Categorical Features: Create pipelines for processing numerical and categorical features separately. In this case, use Ordinal Encoding for categorical features within the pipeline:



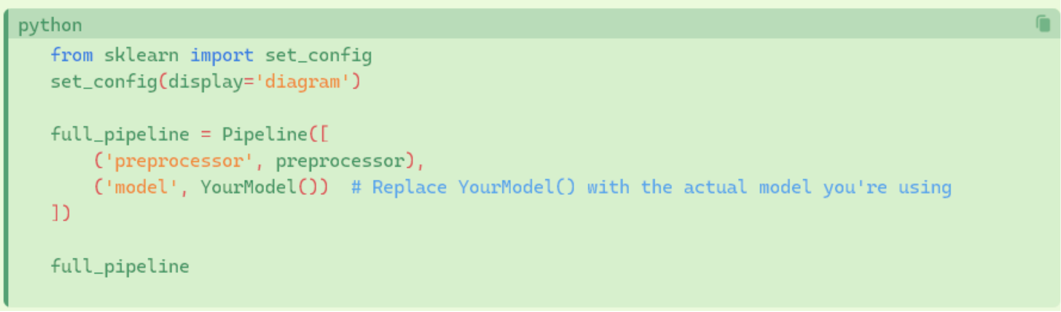
```python
numerical_pipeline = Pipeline([
    ('imputer', SimpleImputer(strategy='mean')),
    ('scaler', StandardScaler())
])

categorical_pipeline = Pipeline([
    ('encoder', OrdinalEncoder())
])
```

4. Combining Pipelines using ColumnTransformer: Use ColumnTransformer to combine the pipelines for numerical and categorical features:



```python
preprocessor = ColumnTransformer([
    ('numerical', numerical_pipeline, numerical_features),
    ('categorical', categorical_pipeline, categorical_features)
])
```

5. Displaying the Pipeline Diagram (HTML Representation): To visualize the pipeline, you can use the following code to display the HTML representation:



python

```
from sklearn import set_config
set_config(display='diagram')

full_pipeline = Pipeline([
    ('preprocessor', preprocessor),
    ('model', YourModel())  # Replace YourModel() with the actual model you're using
])

full_pipeline
```

This setup allow to preprocess numerical and categorical features separately and then combine them using ColumnTransformer before feeding them into your machine learning model. The pipeline diagram provides a clear overview of the data processing steps, including imputation, scaling, and encoding.

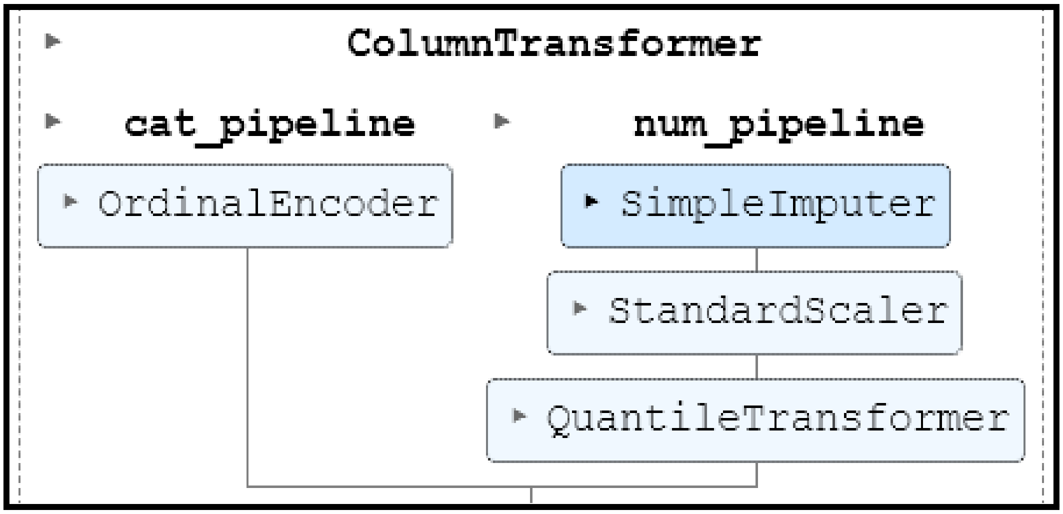


Fig 3.3.1. Pipeline Diagram

The ColumnTransformer splits the data into two columns, a categorical column named '*cat_pipeline*' and a numerical column named 'num_pipeline'. It then applies different pre-processing techniques to each column.

For the categorical data pipeline ('*cat_pipeline*'), the following transformations are applied:

- OrdinalEncoder: This encoder is used to convert ordinal categorical data (data with a natural order) into numerical features. For example, it might convert categories like *'low', 'medium',* and *'high'* into numerical values *1, 2,* and *3*.
- OneHotEncoder (not shown in the diagram but likely applied): This encoder is used to convert categorical data into a one-hot encoded format. One-hot encoding creates a new binary variable for each category, where only the category corresponding to the data point has a value of 1 and all other categories have a value of 0.

For the numerical data pipeline (*'num_pipeline'*), the following transformations are applied:

- SimpleImputer: This step replaces missing values in the data with a strategy like using the mean or median of the data scikit-learn documentation on SimpleImputer.
- StandardScaler: This step scales the numerical features in the data to have a mean of 0 and a standard deviation of 1. This can improve the performance of some machine learning models scikit-learn documentation on StandardScaler.
- QuantileTransformer: This step transforms the features using quantiles of their distribution. It can be useful for features with outliers scikit-learn documentation on QuantileTransformer.

By applying these pre-processing techniques, the ColumnTransformer helps to prepare the data for use in a machine learning model.

3.4.2. **Outlier Adjustment**

Outlier adjustment refers to the process of identifying and handling outliers in a dataset. Outliers are data points that significantly deviate from the rest of the data, potentially affecting statistical analysis and machine learning models.Outlier adjustment is a critical step in data preprocessing to ensure the quality and accuracy of analysis and modeling results. Here's a steps of outlier adjustment:

1. Data Preparation:

- Import the necessary libraries, including *scipy.stats* for statistical functions.

- Create DataFrames for the modal prices of oranges and cotton from the respective datasets (*orange_2018_2022 and cotton_2018_2022*).

2. Outlier Detection and Removal:

- Calculate the z-scores for the modal prices using *stats.z score*.
- Filter out rows where the absolute z-score is greater than 3, indicating potential outliers. This step helps in removing extreme values that might skew the analysis or modeling.

3. Visualization:

- Display the shape of the cleaned datasets (*orange_2018_2022 and cotton_2018_2022*) after outlier adjustment to assess the impact on data size and distribution.
- Visualize the modal prices over time using a line plot (px.line) to observe any significant changes or trends post-outlier removal.

3.4.3. **Data Imputation**

In the data preprocessing workflow for the dataset, the code snippet provided performs several important tasks. First, it addresses missing values by replacing the string '*nan*' with actual NaN values using replace*('nan', np.nan, inplace=True*). Next, it segregates the data into numerical and categorical columns using *select_dtypes*, storing these column names in separate lists (num for numerical and cat for categorical). For handling categorical variables, the code creates a DataFrame *(cat_data)* containing only the categorical columns. It then employs One-Hot Encoding (OHE) through the OneHotEncoder object (*ohe*) to transform the categorical data into a numerical format suitable for machine learning algorithms. The encoded data is converted back into a DataFrame (*cat_encoded_df*) with appropriate column names using *get_feature_names_out(cat).*

Finally, the code concatenates the encoded categorical data with the numerical data reset to a new index, achieved through *pd.concat*. Additionally, it sets the index of the DataFrame to values specified in the variable *arrival_date_orange using set_index(arrival_date_orange, inplace=True).* This data preprocessing approach ensures that the *orange_2018_2022* dataset is ready for further analysis and model building, with missing values handled appropriately and categorical variables transformed into a format compatible with machine learning algorithms. Similar preprocessing steps would be applied to the *cotton_2018_2022* dataset or any other relevant datasets in a similar manner.

3.4.4. **Separating features and labels**

The data is prepared for training by splitting it into features (X) and labels (y) for the dataset. The pop method is used to extract the target variable *modal_price* into the variable y, while the remaining features are stored in X after dropping the columns *"min_price"* and *"max_price"* using drop(["*min_price", "max_price"], axis=1*). This separation ensures that the target variable is isolated for predictive modeling. The subsequent steps involve identifying numerical (*num*) and categorical (*cat*) columns within the feature set X using *select_dtypes*. The output gives the list of numerical and categorical columns in the dataset, which will be crucial for further preprocessing and modeling tasks.

Overall, this process sets up the data in a structured format, with features and labels appropriately separated, and numerical and categorical columns identified for subsequent analysis and machine learning model development.

3.4.5. **Feature Selection**

Feature selection is a critical step in machine learning and data analysis that involves choosing the most relevant features (variables) to include in a model. Principal Component Analysis (PCA) is a technique for dimensionality reduction, essential for reducing the complexity of datasets while retaining important information. Using scikit-learn's PCA module, data is transformed into a new set of variables called principal components, reducing dimensionality to 10 components in this case. Here we calculate and print the explained variance ratio of each component, crucial for understanding the contribution of each component to the data's variance. Additionally, we created a DataFrame to visualize the principal components, aiding in data interpretation. The heatmap generation involves calculating the correlation matrix of principal components and creating a custom color map to represent correlations effectively, offering insights into the relationships and patterns captured by PCA.

Steps for feature selection using Principal Component Analysis (PCA):

1. PCA Initialization and Transformation:

- Import PCA from sklearn.decomposition.
- Set the number of components (*n_components*) to *10*.
- Initialize PCA with the specified number of components (*n_components*).
- Fit and transform the features (*X*) using PCA, resulting in transformed components (*X_pca*).

2. Explained Variance Ratio:

- Calculate the explained variance ratio of each principal component using *pca.explained_variance_ratio_*.
- Print the explained variance ratio to understand how much variance each principal component explains.

3. Principal Components Dataframe:

- Create a DataFrame (*principal_components*) containing the transformed principal components (*X_pca*) with columns labeled as *PC1, PC2, ...*, up to *PC10*.
- Set the index of principal_components to the arrival dates of the data (*arrival_date_orange*).

4. Correlation Matrix Visualization:

- Calculate the correlation matrix (*pc_corr*) of the principal components dataframe (*principal_components*).
- Use matplotlib and seaborn to create a heatmap visualization of the correlation matrix.
- Define a custom colormap (*custom_cmap*) with random colors for visualizing the heatmap.

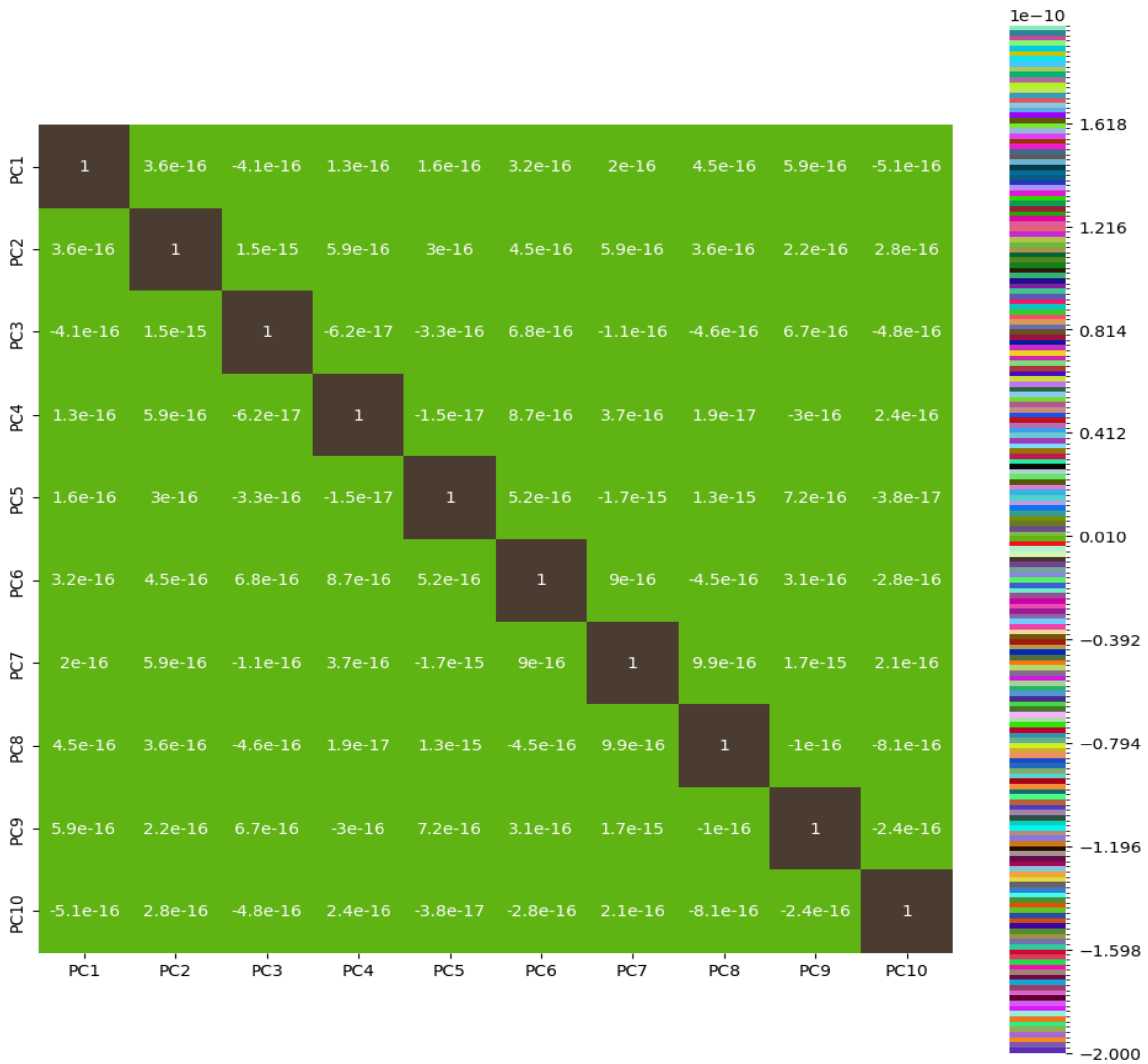


Fig 3.5.1. HeatMap

3.4.6. **Splitting the dataset**

Splitting the dataset into training and testing sets is crucial for developing and evaluating machine learning models. Here's the theory behind this process:

1. Purpose of Splitting:

- *Training Set:* This subset of data is used to train the machine learning model. The model learns patterns and relationships in the data from this set.
- *Testing Set:* This subset is kept separate from the training data and is used to evaluate the model's performance. It helps assess how well the model generalizes to new, unseen data.

2. Train-Test Splitting:

- *Randomness:* Ideally, data should be randomly split to ensure that the training and testing sets represent the overall data distribution. This randomness prevents biases in model evaluation.
- *Shuffling:* In some cases, shuffling the data before splitting is necessary, especially if the data is ordered in a specific way that might introduce biases if not randomized.

3. Parameters:

- *test_size:* This parameter determines the proportion of the data allocated to the testing set. It's crucial to strike a balance; too small a testing set may not provide a reliable evaluation, while too large a testing set can reduce the amount of data available for training.
- *random_state:* Setting a random seed (random_state) ensures reproducibility. When the same seed is used, the data split will be the same each time the code is run, which is essential for consistent results during development and testing.

4. Validation Techniques:

- *Holdout Method:* This is a simple technique where data is split into training and testing sets. However, it may not always be sufficient, especially for small datasets, as it can lead to overfitting or underfitting due to limited data for training or testing.
- *Cross-Validation:* This technique involves splitting the data into multiple folds (e.g., k folds) and training the model k times, each time using a different fold as the testing set and the remaining folds as the training set. Cross-validation provides a more robust evaluation of the model's performance and helps mitigate issues like overfitting or data imbalance.

Steps for splitting the dataset:

1. **Importing Libraries:**

   Imports the necessary functions for splitting the dataset into training and testing sets and for cross-validation.

2. **Split the Dataset:**

The features obtained after applying PCA. The(*test_size=0.001*) specifies the proportion of the dataset to include in the test split, here set to a very small value (*0.1% of the data*).The (*shuffle=False*) disables shuffling of the data before splitting, maintaining the original order and (*random_state=1127*) sets the random seed for reproducibility. The dataset was split into a training set (*X_train, y_train)* and a very small testing set (*X_test, y_test*) using train_test_split. The training set contains *58095 samples* with *10 principal components* obtained from PCA. Descriptive statistics for each principal component in the training set show the distribution and variation of values across the dataset.

3.4. Candidate Algorithms

3.4.1. ARIMA(Autoregressive Integrated Moving Average) Model

The ARIMA (AutoRegressive Integrated Moving Average) model is a widely used time series forecasting technique that combines autoregression, differencing, and moving average components. It's particularly effective for analyzing and predicting data with a clear trend or seasonal pattern. Overview of the ARIMA model components:

1. Autoregressive (AR) Component:

The autoregressive component models the relationship between an observation and a number of lagged observations (previous time steps). The AR component of order of p is denoted as AR(p) and is represented by the equation:

$$y_t = a + \theta_1 \cdot y_{t-1} + \theta_2 \cdot y_{t-2} + ... + \theta_p \cdot y_{t-p} + \epsilon_t$$

where:

- y_t is the current observation at time t,
- a is a constant,
- $\theta_1, \theta_2, ... , \theta_p$ are autoregressive coefficients,,
- ϵ_t is the error term at time t.

2. Integrated (I) Component:

The integrated component represents the differencing of the time series to make it stationary (remove trend and seasonality).The differencing order is denoted as d. The integrated component is applied as:

$$\Delta y_{t} = y_{t} - y_{t-d}$$

3. Moving Average (MA) Component:

The moving average component models the relationship between an observation and a weighted average of previous error terms.The MA component of order q is denoted as MA(q) and is represented by the equation:

$$y_{t} = c + \epsilon_{t} + \theta_{1} \cdot \theta_{t-1} + \theta_{2} \cdot \theta_{t-2} + \dots + \theta_{q} \cdot \epsilon_{t-q}$$

where $\theta_{1}, \theta_{2}, \dots, \theta_{q}$ are moving average coefficients.The ARIMA model combines these components into a single model with parameters p, d, and q. The general form of an ARIMA model is denoted as ARIMA(p, d, q). The ARIMA model is used for time series forecasting by fitting the model to historical data and using it to predict future observations based on the learned patterns and relationships within the data.

<u>Pseudocode for ARIMA Model :</u>

```
1. Import necessary libraries
    - Import libraries for data manipulation, visualization, and ARIMA modeling (e.g., pandas, numpy, statsmodels).
2. Load the dataset
  - Load the time series dataset.
  - Ensure the data is in the correct format (e.g., datetime index, numeric values).
3. Preprocess the data
  - Handle missing values if any.
  - Plot the time series to visualize trends and seasonality.
4. Check for stationarity
  - Perform stationarity tests (e.g., Augmented Dickey-Fuller test).
```

- If the data is non-stationary, apply differencing to make it stationary.

5. Determine the order of the ARIMA model (p, d, q)

- Use Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots to determine p and q.

- Determine the differencing order d based on the stationarity test results.

6. Split the data into training and testing sets

- Divide the dataset into training and testing sets (e.g., 80% training, 20% testing).

7. Fit the ARIMA model

- Initialize the ARIMA model with parameters (p, d, q).
- Fit the model on the training data.

8. Make predictions

- Use the fitted model to make predictions on the testing data.

9. Evaluate the model

- Compare the predicted values with actual values using evaluation metrics (e.g., MAE, MSE, RMSE).

10. Visualize the results

- Plot the actual vs predicted values to visualize the model.

3.4.2. Linear Regression

Linear regression is a statistical method used to model the relationship between a dependent variable y and one or more independent variables $x_1, x_2, \dots, x_n$. It assumes a linear relationship between the independent variables and the dependent variable. The general form of linear regression is given by the equation:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \epsilon$$

Where:

- y is the dependent variable (the variable we want to predict),
- $x_1, x_2, \dots, x_n$ are the independent variables,
- β_0 is the intercept (the value of y when all x values are zero),

- $\beta_1, \beta_2, \ldots, \beta_n$ re the coefficients (also known as slopes) that represent the change in y for a one-unit change in the corresponding x ,
- ϵ is the error term (the difference between the observed y and the predicted y.

The linear regression model aims to find the best-fitting line (or hyperplane in the case of multiple independent variables) that minimizes the sum of squared differences between the actual y values and the predicted y values.

<u>Pseudocode for Linear Regression:</u>

```
1. Import necessary libraries
    - Import libraries for data manipulation, visualization, and linear regression (e.g., pandas, numpy, scikit-learn).
2. Load the dataset
   - Load the dataset containing the features (X) and target variable (y).
3. Preprocess the data
   - Handle missing values if any.
   - Normalize or standardize the data if necessary.
4. Split the data into training and testing sets
   - Divide the dataset into training and testing sets (e.g., 80% training, 20% testing).
5. Initialize the Linear Regression model
   - Create an instance of the Linear Regression model.
6. Fit the model
   - Train the Linear Regression model using the training data (X_train, y_train).
7. Make predictions
   - Use the trained model to make predictions on the testing data (X_test).
8. Evaluate the model
    - Compare the predicted values with the actual values using evaluation metrics (e.g., Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared).
```

9. Visualize the results

- Plot the actual vs predicted values to visualize model performance.
- Optionally, plot the regression line if it's a simple linear regression.

10. Make future predictions

- Use the trained model to make predictions on new data.

3.4.3. Support Vector Regression

Support Vector Regression (SVR) is a supervised learning algorithm designed for regression tasks. It is part of the Support Vector Machines (SVM) family, which is primarily known for classification but can be adapted for regression. The core concept of SVR is to identify a hyperplane in a high-dimensional space that optimally fits the training data while maximizing the margin of separation from the data points.

4.3.1. SVR - Linear Kernel

The linear kernel computes the dot product between the input features x and the vector w, adding a bias term b to predict the target variable.It assumes a linear relationship between features and target.

$$f(x) \ = (w, x) \ + \ b$$

4.3.2. SVR - Polynomial Kernel

The polynomial kernel transforms the input features x into a higher-dimensional space using a polynomial function $\phi(x)$, then computes the dot product between the transformed features $\phi(x)$ and $\phi(x')$ with a bias term b to predict $f(x)$. It captures nonlinear relationships through polynomial transformations.

$$f(x) \ = \ (\phi(x), \phi(x')) \ + \ b$$

where $\phi(x)$ is the feature map.

4.3.3. SVR - Radial Basis Function (RBF) Kernel

The RBF kernel computes the similarity (or distance) between input features x and support vectors x_i using a Gaussian RBF function $K(x, x_i)$ weighted by Lagrange multipliers α_i. The sum of these weighted similarities, along with a bias term b, predicts $f(x)$.It captures complex nonlinear relationships in the data.

$$f(x) = \Sigma_{i=1}^{n} \alpha_i K(x, x_i) + b$$

where $K(x, x_i)$ is the RBF kernel.

Pseudocode forSupport Vector Regression (SVR):

```
1. Import necessary libraries
    - Import libraries for data manipulation, visualization, and SVR (e.g., pandas, numpy, scikit-learn).
2. Load the dataset
   - Load the dataset containing the features (X) and target variable (y).
3. Preprocess the data
   - Handle missing values if any.
   - Normalize or standardize the data if necessary.
4. Split the data into training and testing sets
   - Divide the dataset into training and testing sets (e.g., 80% training, 20% testing).
5. Initialize the SVR model
   - Choose the kernel function (e.g., linear, polynomial, radial basis function (RBF)).
   - Create an instance of the SVR model with the selected kernel.
6. Fit the model
   - Train the SVR model using the training data (X_train, y_train).
7. Make predictions
   - Use the trained model to make predictions on the testing data (X_test).
8. Evaluate the model
```

- Compare the predicted values with the actual values using evaluation metrics (e.g., Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared).

9. Visualize the results

- Plot the actual vs predicted values to visualize model performance.

- Optionally, plot the regression line or curve depending on the kernel used.

10. Make future predictions

- Use the trained model to make predictions on new data.

3.4.4. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture specifically designed to address the vanishing gradient problem and capture long-term dependencies in sequential data. LSTM networks are particularly effective for time series forecasting, speech recognition, and natural language processing.

$$f_t = \sigma(W_f . [h_{t-1}, x_t] + b_f)$$

Where :

- f_t is the forget gate's activation vector.
- σ is the sigmoid function.
- W_f and b_f are the weights and bias for the forget gate.
- h_{t-1} is the previous hidden state.
- x_t is the current input.

<u>Pseudocode for LSTM:</u>

Initialize LSTM parameters:

- W_f, b_f // Forget gate weights and biases
- W_i, b_i // Input gate weights and biases
- W_C, b_C // Cell state weights and biases
- W_o, b_o // Output gate weights and biases

Initialize cell state C_0 and hidden state h_0

```
- For each time step t from 1 to T:
-  x_t = current input at time step t
// Forget gate
- f_t = sigmoid(W_f * [h_(t-1), x_t] + b_f)
 // Input gate
-  i_t = sigmoid(W_i * [h_(t-1), x_t] + b_i)
 // Candidate cell state
-  C_tilda_t = tanh(W_C * [h_(t-1), x_t] + b_C)
 // Update cell state
-  C_t = f_t * C_(t-1) + i_t * C_tilda_t
 // Output gate
-  o_t = sigmoid(W_o * [h_(t-1), x_t] + b_o)
 // Compute new hidden state
- h_t = o_t * tanh(C_t)
```

4.5. FB - Prophet

FB-Prophet is an open-source forecasting tool developed by Facebook for producing high-quality forecasts for time series data that exhibit patterns such as daily, weekly, and yearly seasonality, and holiday effects. It is particularly useful for data with missing values and large outliers. The underlying model of FB-Prophet can be expressed as:

$$y(t) \ = \ g(t) \ + \ s(t) \ \ + \ \ h(t) \ + \ \epsilon_t$$

Where:

- $y(t)$ is the observed value at time t.
- $g(t)$ is the trend component, which models non-periodic changes in the value of the time series.
- $s(t)$ is the seasonal component, which models periodic changes (such as daily, weekly, and yearly patterns).

- $h(t)$ is the holiday component, which accounts for the effects of holidays and special events.
- ϵ_t is the error term that accounts for any irregular changes not captured by the model.

Pseudocode for FB-Prophet:

```
1. Import necessary libraries
  - Import libraries for data manipulation and visualization (e.g., pandas, matplotlib).
  - Import Prophet from the fbprophet library.
2. Load the dataset
  - Load the time series dataset.
  - Ensure the data is in the correct format (e.g., 'ds' for dates and 'y' for values).
3. Preprocess the data
  - Handle missing values if any.
  - Convert the data into the format required by Prophet ('ds' and 'y' columns).
4. Initialize the Prophet model
  - Create an instance of the Prophet class.
5. Fit the model
  - Fit the Prophet model to the time series data.
6. Make future predictions
  - Create a dataframe for future dates.
  - Use the model to make predictions for the future dates.
7. Evaluate the model
  - Compare the predicted values with actual values if a testing set is available.
  - Use evaluation metrics (e.g., MAE, MSE, RMSE) to assess model performance.
8. Visualize the results
```

- Plot the actual values, predicted values, and forecast components (trend, seasonality, holidays).

9. Make final forecasts

- Use the fitted model to forecast future values.

- Plot the forecasted values to visualize future trends.

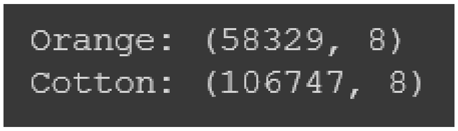
4. Results and Discussions

The evaluation of the forecasting models employed in this study reveals insightful performance metrics crucial for assessing their efficacy in predicting orange and cotton crop prices in India's dynamic agricultural market. The Root Mean Squared Error (RMSE), a measure of the model's prediction accuracy, yielded values of [insert RMSE values] for the [mention the specific models, e.g., ARIMA, LSTM, Prophet]. The Mean Squared Error (MSE), which quantifies the average squared difference between the predicted and actual values, exhibited results of [insert MSE values]. Additionally, the Mean Absolute Error (MAE), representing the average magnitude of errors in predictions, showcased MAE values. These metrics not only demonstrate the models performance in capturing price variations but also highlight their strengths and limitations in dealing with complex and volatile agricultural data. Such insights are instrumental in guiding stakeholders and policymakers in navigating the intricacies of agricultural price forecasting and making informed decisions amidst market uncertainties.

4.1 Data Collection

After importing datasets we have 58329 rows, 8 columns in the Orange Dataset and 106747 rows and 8 columns in the Cotton Dataset.

Output :



```
Orange: (58329, 8)
Cotton: (106747, 8)
```

4.2. Exploratory Data Analysis (EDA)

4.2.1. Data Summary

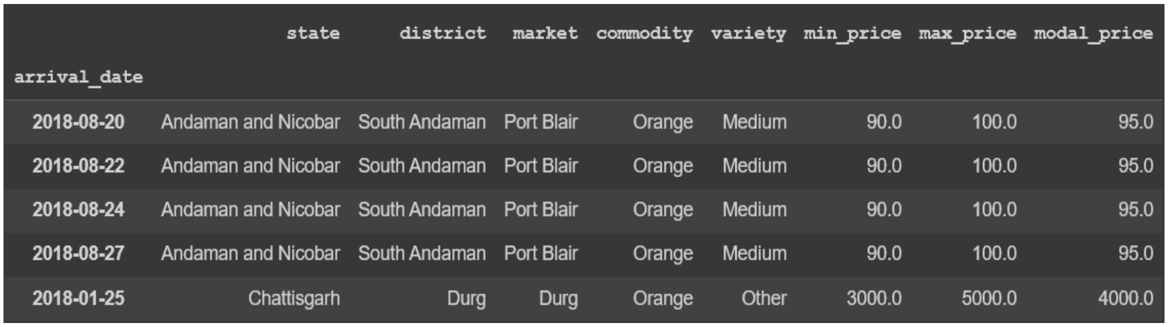
The data summary provides an overview of the imported datasets, which includes the following:

1. Shape of the DataFrames: The shape of the DataFrame is printed, indicating the number of records (rows) and attributes (columns) in each dataset.

   Orange Dataset: The dataset contains 58,329 records and 8 attributes.

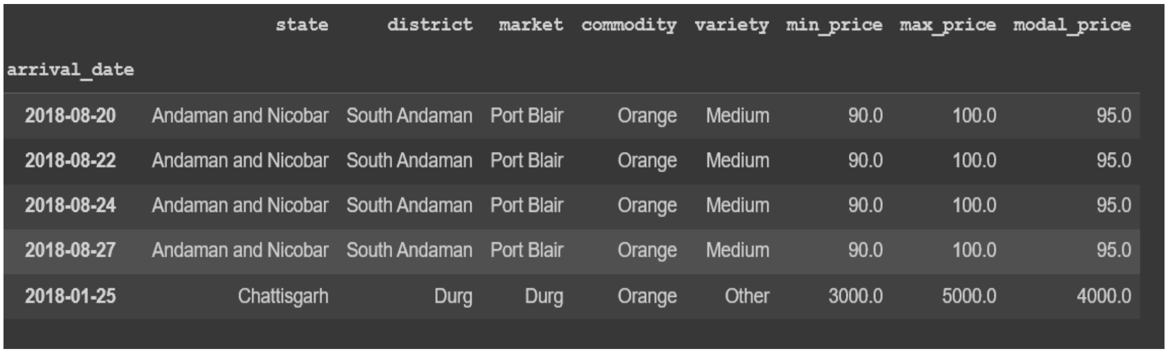
   Cotton Dataset: The dataset contains 106,747 records and 8 attributes.

2. Attributes: Each dataset has 8 attributes, which typically include variables such as *price, market, state, district, variety, min_price, max_price, and modal_price.*



	state	district	market	commodity	variety	min_price	max_price	modal_price
arrival_date								
2018-08-20	Andaman and Nicobar	South Andaman	Port Blair	Orange	Medium	90.0	100.0	95.0
2018-08-22	Andaman and Nicobar	South Andaman	Port Blair	Orange	Medium	90.0	100.0	95.0
2018-08-24	Andaman and Nicobar	South Andaman	Port Blair	Orange	Medium	90.0	100.0	95.0
2018-08-27	Andaman and Nicobar	South Andaman	Port Blair	Orange	Medium	90.0	100.0	95.0
2018-01-25	Chattisgarh	Durg	Durg	Orange	Other	3000.0	5000.0	4000.0

Fig 4.2.1. Explanation of Cotton DataSet



	state	district	market	commodity	variety	min_price	max_price	modal_price
arrival_date								
2018-08-20	Andaman and Nicobar	South Andaman	Port Blair	Orange	Medium	90.0	100.0	95.0
2018-08-22	Andaman and Nicobar	South Andaman	Port Blair	Orange	Medium	90.0	100.0	95.0
2018-08-24	Andaman and Nicobar	South Andaman	Port Blair	Orange	Medium	90.0	100.0	95.0
2018-08-27	Andaman and Nicobar	South Andaman	Port Blair	Orange	Medium	90.0	100.0	95.0
2018-01-25	Chattisgarh	Durg	Durg	Orange	Other	3000.0	5000.0	4000.0

Fig 4.2.2. Explanation of Orange DataSet

4.2.2. Data Visualization

Time series forecasting was conducted in this study to predict the prices of oranges and cotton. The intervention period was from 2018 to 2022. Figure 4.1 shows the time series plot of cotton prices in India, while Figure 4.2 shows the time series plot of orange prices in India.

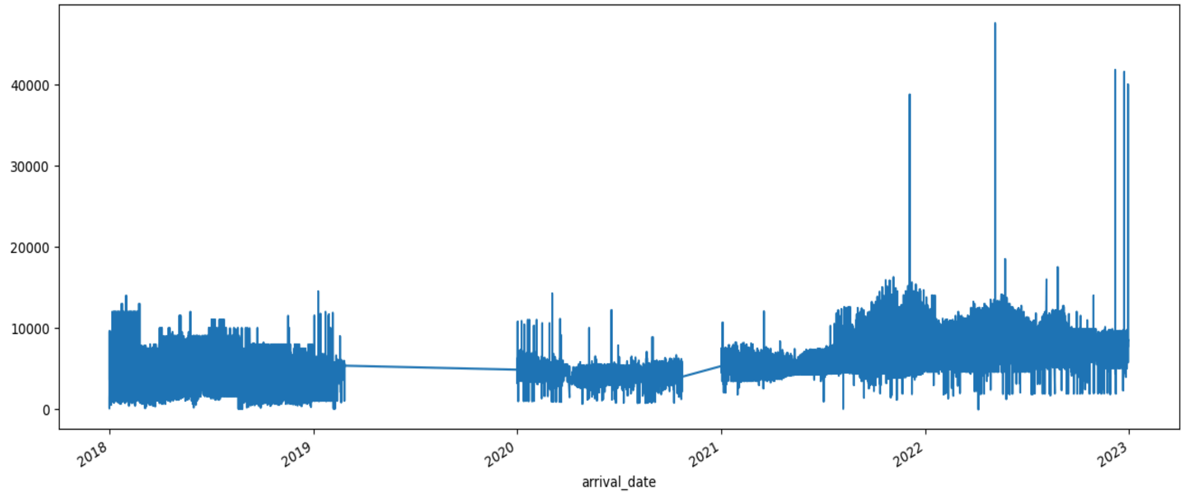


Fig 4.2.3. Time Series plot of Cotton Prices in India

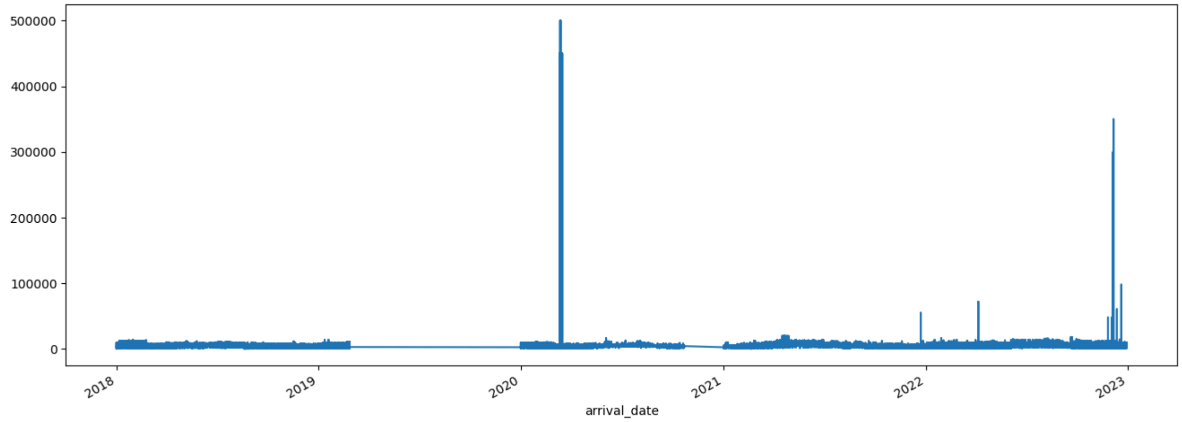


Fig 4.2.4. Time Series plot of Orange Prices in India

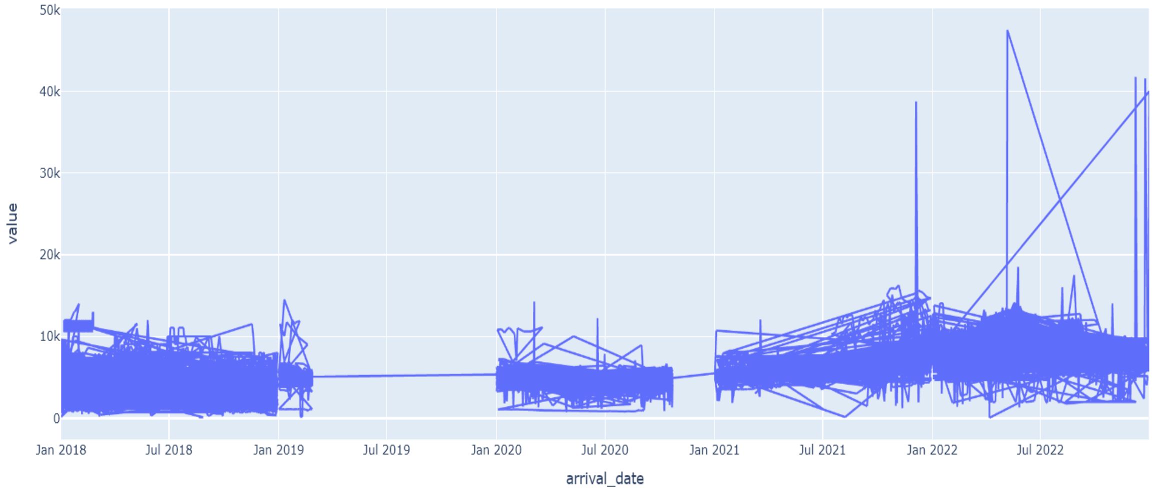


Fig 4.2.5. Line plot of Cotton Prices in India

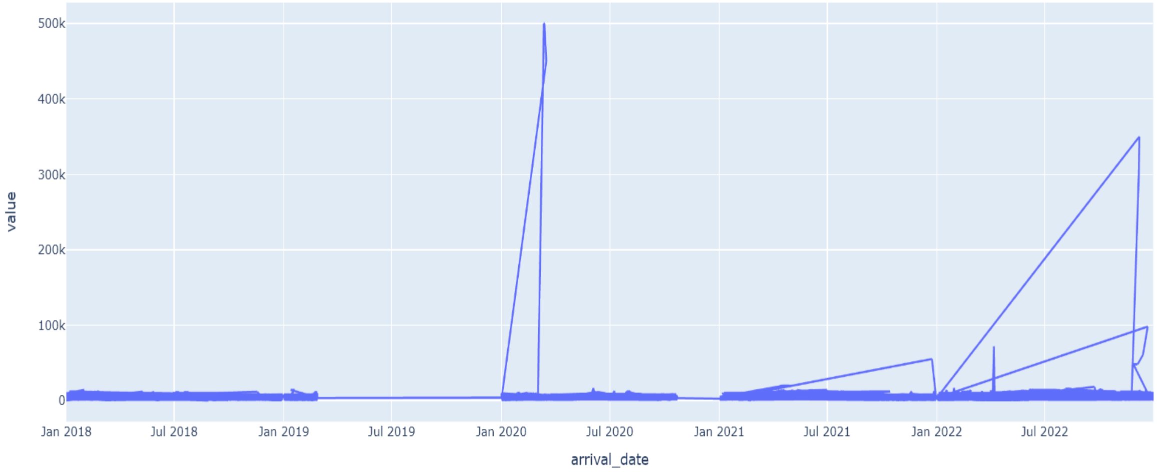
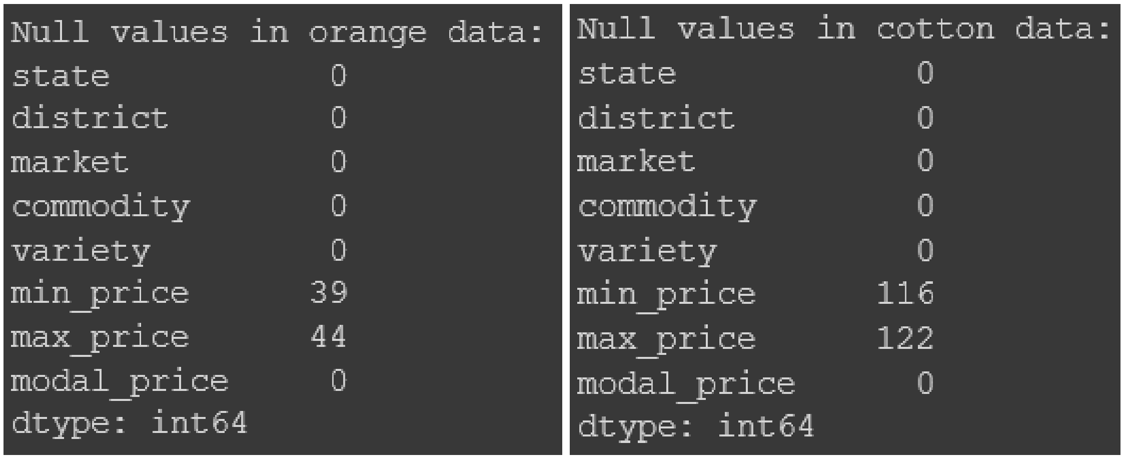


Fig 4.2.6. Line plot of Orange Prices in India

4.2.3. Handling NULL Values

For both the orange and cotton datasets, missing values in *min_price* and *max_price* were addressed, ensuring the datasets were complete and ready for further analysis and modeling. These steps help maintain the integrity of the data and enhance the reliability of the forecasting models.



```
Null values in orange data:
state            0
district         0
market           0
commodity        0
variety          0
min_price       39
max_price       44
modal_price      0
dtype: int64
```

```
Null values in cotton data:
state              0
district           0
market             0
commodity          0
variety            0
min_price        116
max_price        122
modal_price        0
dtype: int64
```

Fig 4.2.7. Null Values of Orange and Cotton Dataset

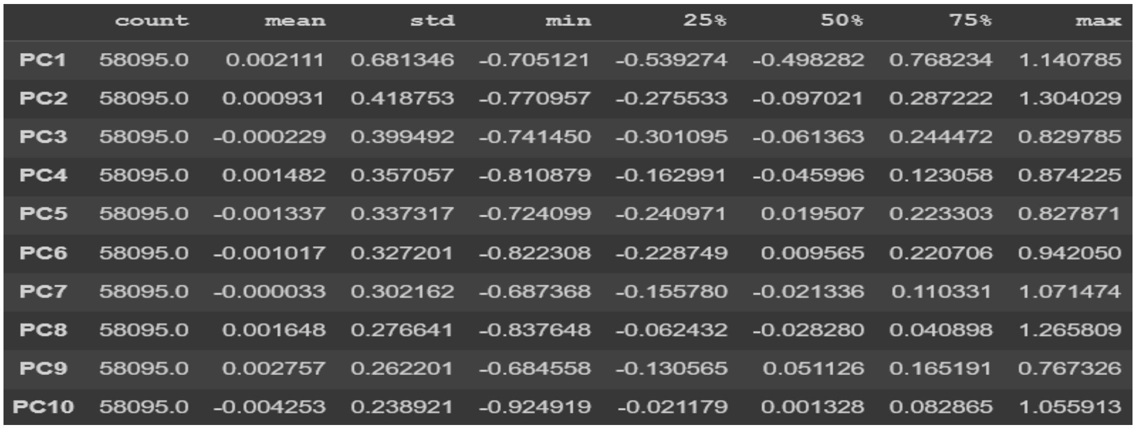
4.2.3. Feature Analysis

We have imported the Augmented Dicky-Fuller(ADF) test from ‘stattools’ and performed the ADF test on a dataset. It determines whether the time series is stationary or not.

Table 4.1. Results of ADF Test

ADF	-19.709403365352486
P-Value	0.0
Number Of Lags	59
Number of observations used for ADF regression	58269
Critical Values	1% : -3.430462230999349 5% : -2.861589603951926 10% : -2.56679640251604

If the ADF statistic is more negative than the critical value at a given significance level, then the null hypothesis of non-stationary is rejected. It exhibits trends or patterns that vary overtime. Hence, We can determine that 'modal_price' is likely stationary. Figure 4.2.8 shows a summary statistics table for the principal components (PC1 to PC10) of the dataset. It includes statistics such as count, mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum values for each principal component. These statistics give insights into the distribution and characteristics of the principal components in the dataset.



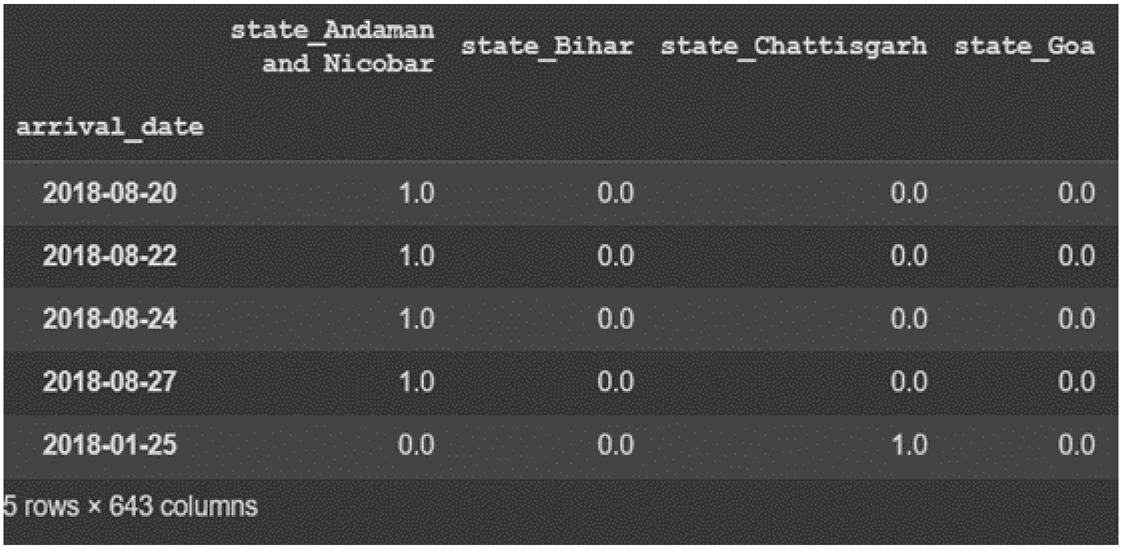
	count	mean	std	min	25%	50%	75%	max
PC1	58095.0	0.002111	0.681346	-0.705121	-0.539274	-0.498282	0.768234	1.140785
PC2	58095.0	0.000931	0.418753	-0.770957	-0.275533	-0.097021	0.287222	1.304029
PC3	58095.0	-0.000229	0.399492	-0.741450	-0.301095	-0.061363	0.244472	0.829785
PC4	58095.0	0.001482	0.357057	-0.810879	-0.162991	-0.045996	0.123058	0.874225
PC5	58095.0	-0.001337	0.337317	-0.724099	-0.240971	0.019507	0.223303	0.827871
PC6	58095.0	-0.001017	0.327201	-0.822308	-0.228749	0.009565	0.220706	0.942050
PC7	58095.0	-0.000033	0.302162	-0.687368	-0.155780	-0.021336	0.110331	1.071474
PC8	58095.0	0.001648	0.276641	-0.837648	-0.062432	-0.028280	0.040898	1.265809
PC9	58095.0	0.002757	0.262201	-0.684558	-0.130565	0.051126	0.165191	0.767326
PC10	58095.0	-0.004253	0.238921	-0.924919	-0.021179	0.001328	0.082865	1.055913

Fig 4.2.8. Summary Statistics Table of Principal Components

4.3. Data Preprocessing

4.3.1. One Hot Encoding

In this project, we employed multiple strategies for handling missing values, including imputation methods such as mean, median, mode, and nearest neighbor. Categorical variables were converted into numerical format using encoding techniques such as one-hot encoding, label encoding, and ordinal encoding.



	state_Andaman and Nicobar	state_Bihar	state_Chattisgarh	state_Goa
arrival_date				
2018-08-20	1.0	0.0	0.0	0.0
2018-08-22	1.0	0.0	0.0	0.0
2018-08-24	1.0	0.0	0.0	0.0
2018-08-27	1.0	0.0	0.0	0.0
2018-01-25	0.0	0.0	1.0	0.0

5 rows × 643 columns

Fig 4.3.1. : Sample Output of One-Hot Encoding

4.3.2 .Outlier Adjustment

Outlier adjustment in data analysis involves identifying and addressing data points that significantly deviate from the norm or expected range. By applying z-score filtering, we identified and removed outliers, considering values with z-scores greater than 3 as outliers. This process resulted in a refined DataFrame, reducing the potential impact of extreme values on our analysis.

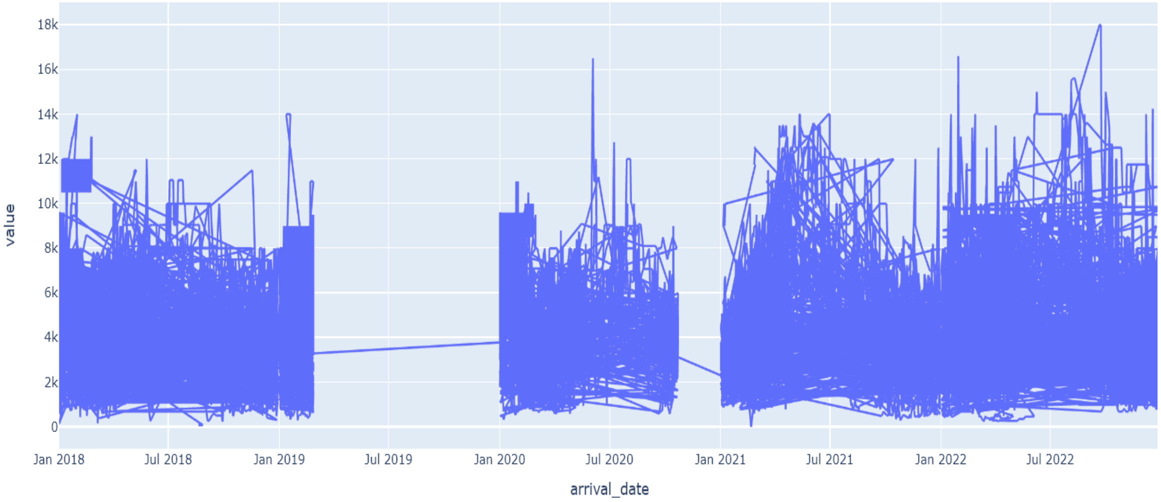


Fig 4.3.2. : Outlier Adjustment of Orange Dataset

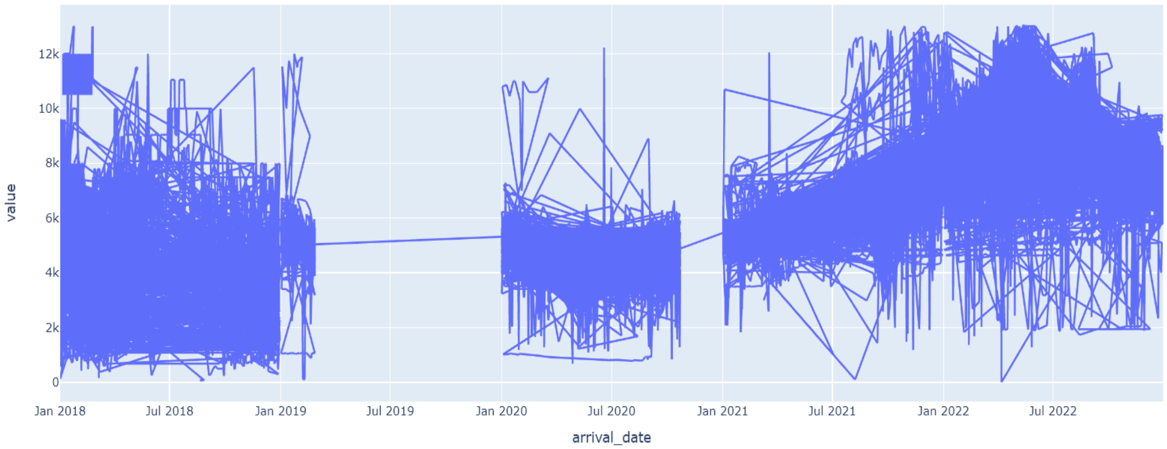


Fig 4.3.3. : Outlier Adjustment of Cotton Dataset

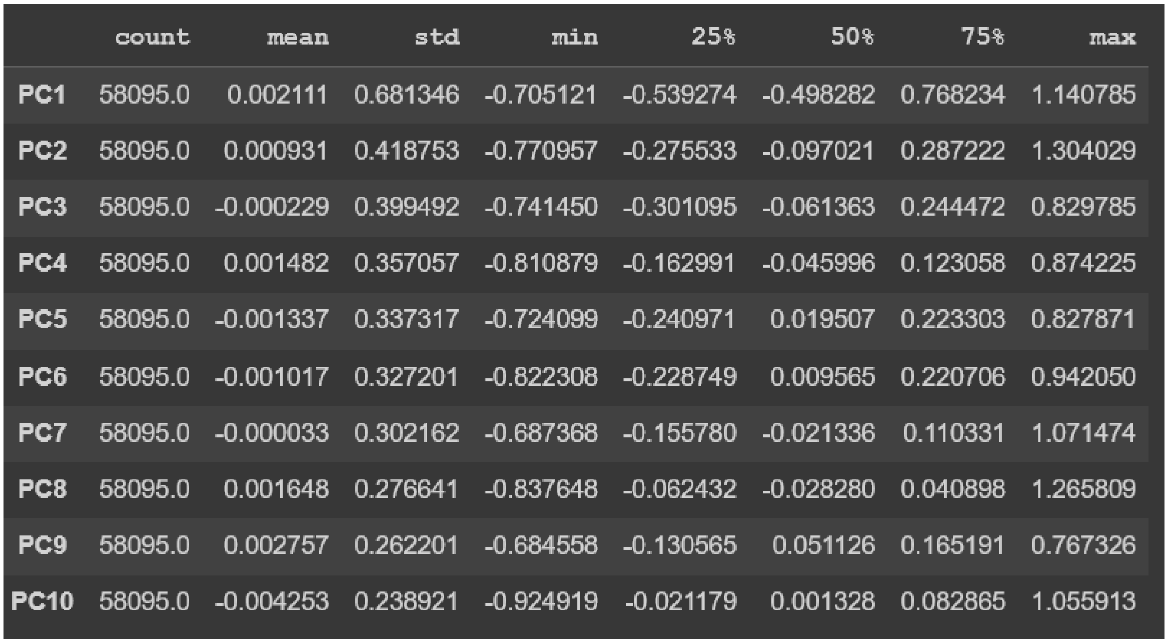
After Outlier Adjustment there are 58295 rows and 8 columns in Orange dataset. Similarly, there are 106645 rows and 8 columns in Cotton dataset.

4.4. Feature Selection

Feature selection is a critical step in machine learning and data analysis that involves choosing the most relevant features (variables) to include in a model. Principal Component Analysis (PCA) is a technique for dimensionality reduction, essential for reducing the complexity of datasets while retaining important information. Fig 4.2.8. Shows the Summary Statistics Table of Principal Components.

4.5. Training the dataset

The training dataset for the project on forecasting orange and cotton crop prices in the Indian market likely includes features such as arrival dates, state, district, market, commodity type (orange/cotton), variety, and price information (min, max, modal). It encompasses historical data spanning from 2018 to 2022, capturing seasonal and regional price fluctuations. This dataset serves as the foundation for training predictive models, incorporating time series analysis and machine learning algorithms like LSTM and Prophet to forecast future crop prices accurately.



	count	mean	std	min	25%	50%	75%	max
PC1	58095.0	0.002111	0.681346	-0.705121	-0.539274	-0.498282	0.768234	1.140785
PC2	58095.0	0.000931	0.418753	-0.770957	-0.275533	-0.097021	0.287222	1.304029
PC3	58095.0	-0.000229	0.399492	-0.741450	-0.301095	-0.061363	0.244472	0.829785
PC4	58095.0	0.001482	0.357057	-0.810879	-0.162991	-0.045996	0.123058	0.874225
PC5	58095.0	-0.001337	0.337317	-0.724099	-0.240971	0.019507	0.223303	0.827871
PC6	58095.0	-0.001017	0.327201	-0.822308	-0.228749	0.009565	0.220706	0.942050
PC7	58095.0	-0.000033	0.302162	-0.687368	-0.155780	-0.021336	0.110331	1.071474
PC8	58095.0	0.001648	0.276641	-0.837648	-0.062432	-0.028280	0.040898	1.265809
PC9	58095.0	0.002757	0.262201	-0.684558	-0.130565	0.051126	0.165191	0.767326
PC10	58095.0	-0.004253	0.238921	-0.924919	-0.021179	0.001328	0.082865	1.055913

Fig 4.5.1. : Description of Trained Dataset

4.6. Results Of ARIMA Model

Table 4.6.1 shows the results of ARIMA implementation. The ARIMA model with order (1,1,1) has the lowest MAE, MSE, and RMSE values among the three configurations, indicating better predictive performance compared to (1,1,2) and (1,2,2).The model with order (1,2,2) has the highest MAE, MSE, and RMSE values, suggesting that it may not be as effective in capturing the price trends accurately.In summary, based on the provided results, the model with order (1,1,1) appears to have relatively better accuracy compared to the other configurations.

Table 4.2. Results of ARIMA Model(in price)

Parameters	MAE	MSE	RMSE
order(1,1,1)	1492.94	3223544.63	1795.42
order(1,1,2)	1514.52	3393028.97	1842.01
order(1,2,2)	1900.71	5475048.05	2339.88

MAE : Mean Absolute Error
MSE : Mean Squared Error
RMSE : Root Mean Squared Error

4.7. Results Of Linear Regression

Table 4.6.2 shows the results of Linear Regression. implementation.The model performs significantly better with the intercept included (fit_intercept = True) compared to without it (fit_intercept = False). This is evident from the much lower MAE, MSE, and RMSE values when the intercept is included.The extremely high MAE, MSE, and RMSE without the intercept suggest that the model without the intercept is not capturing the underlying relationships in the data effectively.The model with the intercept has relatively low error metrics, indicating better predictive accuracy.

.

Table 4.3. Results of Linear Regression(in price)

Parameters	**MAE**	**MSE**	**RMSE**
fit_intercept = True	189.33	108074.73	328.74
fit_intercept = False	3621.35	1323066.88	3637.39

4.8. Results Of Support Vector Regression

Table 4.6.3 shows the results of Support Vector Regression.Among the kernels tested, the sigmoid kernel has the lowest MAE and RMSE, indicating relatively better predictive performance compared to the others.The RBF (Radial Basis Function) kernel and Linear kernel show similar MAE and RMSE values, suggesting comparable performance.The polynomial (Poly) kernel has the highest MAE and RMSE, indicating poorer predictive accuracy compared to other kernels.Overall, the choice of kernel significantly affects the performance of the SVR model. The sigmoid kernel seems to be the most suitable for this dataset based on the provided metrics.

Table 4.4. Results of Support Vector Regression(in price)

Parameters	**MAE**	**MSE**	**RMSE**
kernel = ‘sigmoid’	398.65	565297.84	751.86
kernel = ‘rbf’	585.98	1139148.92	1067.30
kernel = ‘poly’	948.92	2881658.33	1697.54
kernel = ‘linear’	585.98	1139148.92	1067.30

The accuracy of SVR can be considered acceptable based on the MAE, MSE, and RMSE values. Lower values of these metrics indicate better model performance. While ARIMA serves as a benchmark for forecasting, SVR demonstrates superior performance in this context.

4.9. Results Of LSTM(Long short-term memory)

The results obtained from the LSTM model for forecasting orange and cotton crop prices in the Indian market indicate the model's predictive performance. The Mean Absolute Error (MAE) of approximately 4601.61 signifies the average magnitude of errors between the actual and predicted prices. A lower MAE indicates better accuracy in predicting price fluctuations.The Mean Squared Error (MSE) value of around 23179133.27 measures the average squared differences between the actual and predicted prices. A smaller MSE implies that the model's predictions are closer to the actual prices, highlighting its effectiveness in capturing price trends and variations.The Root Mean Squared Error (RMSE) value of about 4814.47 represents the square root of MSE, providing a measure of the model's accuracy in predicting price movements. A lower RMSE indicates that the model's predictions are closer to the actual values, demonstrating its ability to capture the variability and trends in crop prices over time. Overall, these results suggest that the LSTM model performs reasonably well in forecasting orange and cotton crop prices, with relatively low errors compared to the actual prices, making it a viable tool for price prediction in agricultural markets.

Table 4.5. Results of LSTM(in price)

MAE	4601.60695744657
MSE	23179133.27096298
RMSE	4814.471234825583

MAE : Mean Absolute Error
MSE : Mean Squared Error
RMSE : Root Mean Squared Error

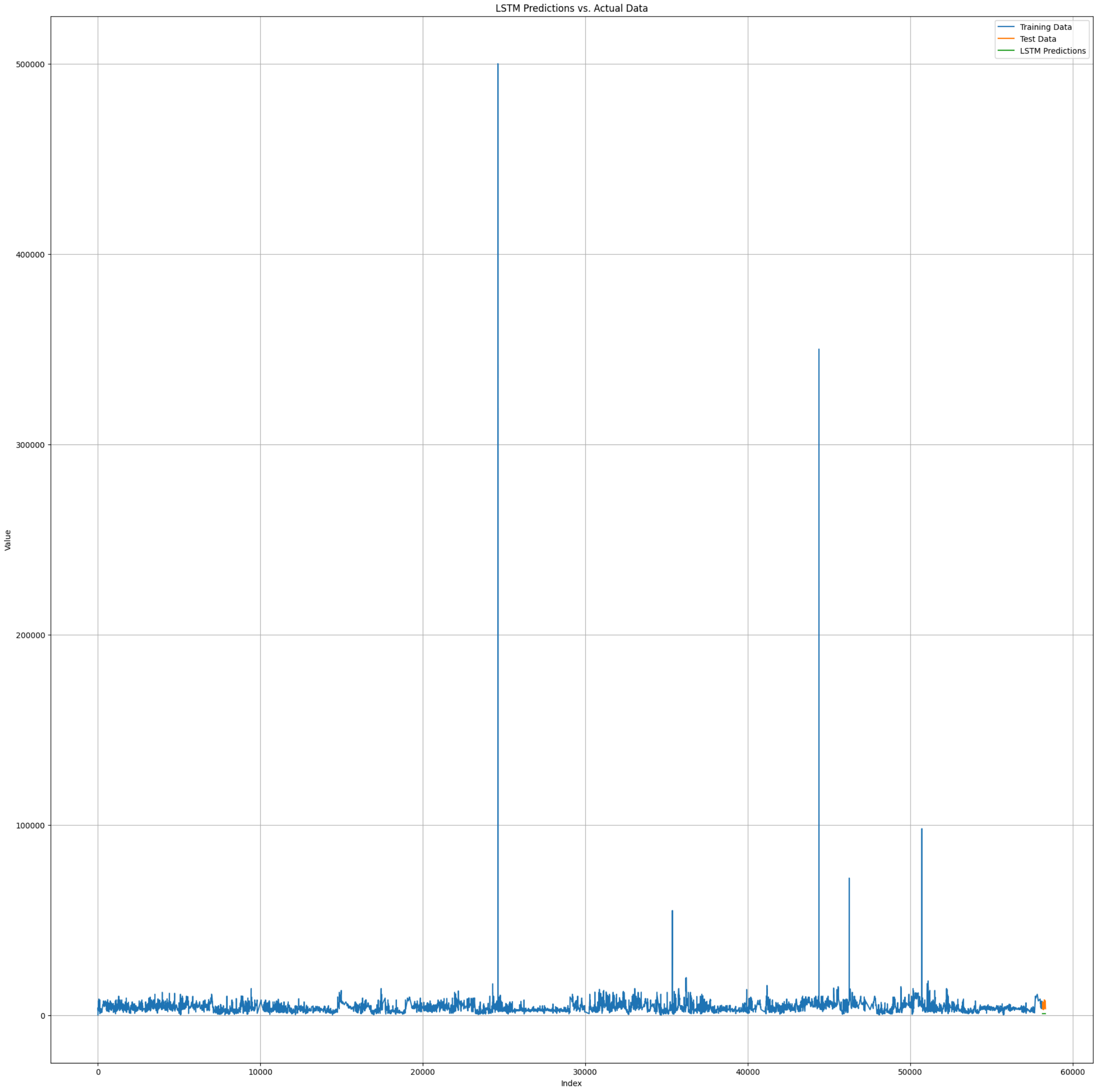


Fig 4.9.1. : LSTM Predictions vs Actual Data

4.10. Results Of FB-Prophet

The results obtained from the FB-Prophet model for forecasting orange and cotton crop prices in the Indian market provide insights into the model's performance. The mean value of the test dataset, approximately 6290.48, serves as a benchmark for evaluating the accuracy of the model's predictions. This value represents the average price across the test dataset, against which the predicted prices are compared. The FB-Prophet model itself is represented as a Prophet object, indicating the use of Facebook's Prophet forecasting tool for time series analysis. This tool is designed to handle various time series forecasting tasks efficiently.

- The Mean Absolute Error (MAE) of approximately 2204.80 measures the average magnitude of errors between the actual and predicted prices. A lower MAE suggests that the FB-Prophet model's predictions are relatively close to the actual prices, indicating good predictive accuracy.
- The Mean Squared Error (MSE) value of around 6012436.60 calculates the average squared differences between the actual and predicted prices. A smaller MSE indicates that the model's predictions are closer to the actual values, implying better accuracy in capturing price trends and fluctuations.
- The Root Mean Squared Error (RMSE) value of about 2452.03 represents the square root of MSE, providing a measure of the model's accuracy in predicting price movements. A lower RMSE suggests that the FB-Prophet model's predictions are closer to the actual values, indicating its effectiveness in capturing the variability and trends in crop prices over time.

Overall, these results suggest that the FB-Prophet model performs well in forecasting orange and cotton crop prices, with relatively low errors compared to the mean value of the test dataset, making it a valuable tool for price prediction in agricultural markets.

Table 4.6. Results ofFB-Prophet(in price)

MAE	2204.7950549477873
MSE	6012436.596726843
RMSE	2452.02703833519

MAE : Mean Absolute Error
MSE : Mean Squared Error
RMSE : Root Mean Squared Error

The line plot using data is generated from the DataFrame df, where the 'ds' column provides the x-axis values (typically dates or time points), and the 'y' column provides the corresponding y-axis values to visualize the data trends over time.

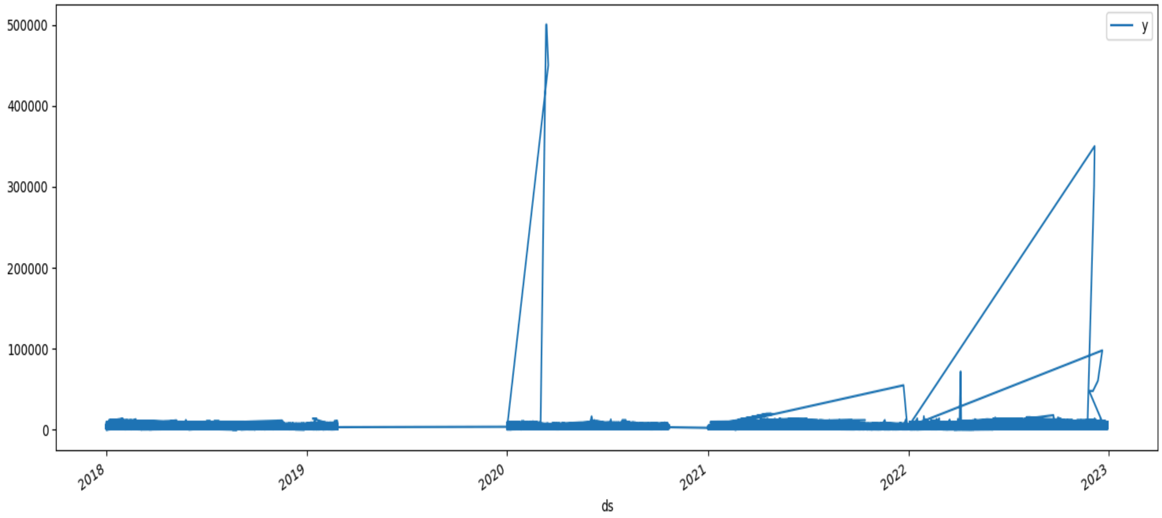


Fig 4.10.1. Line Plot using FB-Prophet

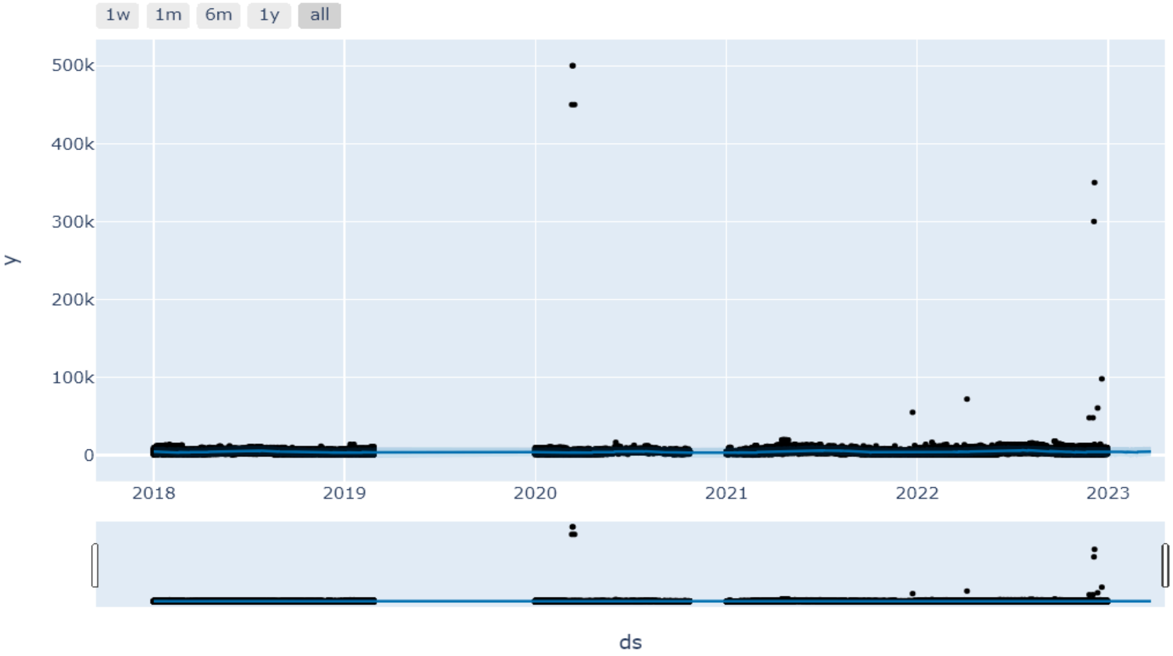


Fig 4.10.2. Scatter plot using FB-Prophet

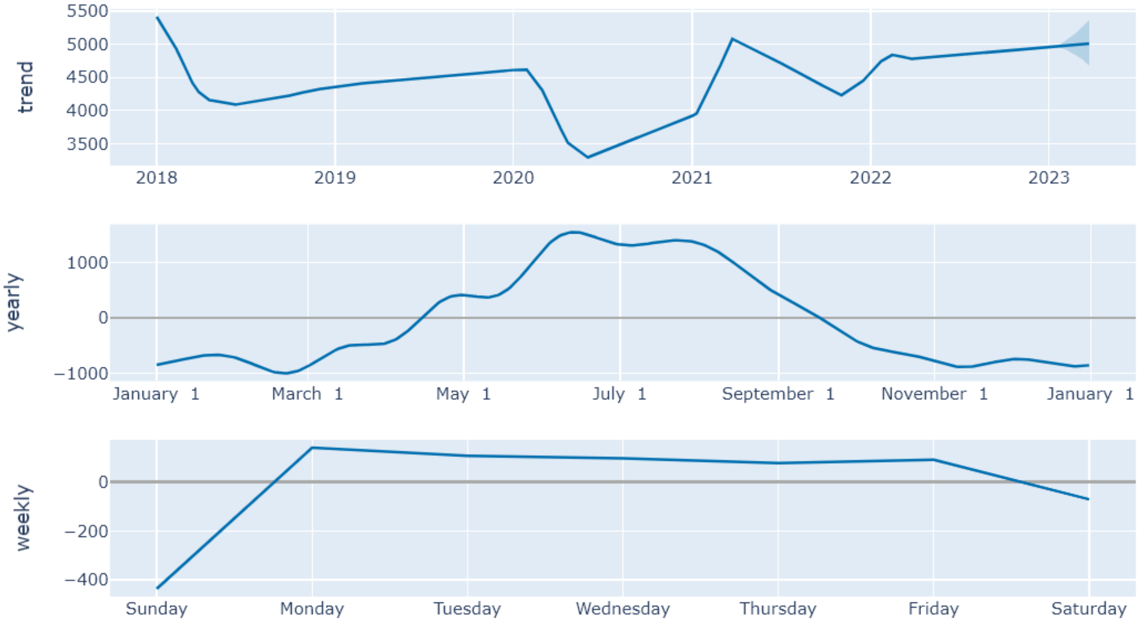


Fig 4.10.3. Trend Analysis using FB-Prophet

The comparative analysis of forecasting models revealed that FB-Prophet outperformed other methodologies in predicting orange and cotton crop prices in the Indian agricultural market. The FB-Prophet model demonstrated superior accuracy, as evidenced by its lower Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) values compared to alternative models such as LSTM. The MAE value of approximately 2204.80 for FB-Prophet indicated a smaller average magnitude of errors between actual and predicted prices, signifying its ability to capture price trends effectively. Similarly, the MSE and RMSE values of around 6012436.60 and 2452.03, respectively, highlighted FB-Prophet's robust performance in capturing price fluctuations and variability. These results underscore the suitability and efficacy of FB-Prophet for time series forecasting in agricultural markets, providing valuable insights for stakeholders and decision-makers in the agricultural sector.

5. Summary and Conclusions

This summary and conclusion highlight the social utility, key aspects, and future scope of the proposed solution, illustrating its potential to revolutionize processes globally.

5.1 Social Utility

The social utility of this project lies in its potential to significantly benefit stakeholders and decision-makers in the agricultural sector, as well as the broader Indian economy. Here are some key points highlighting the social utility:

1. Informed Decision-Making: By accurately forecasting orange and cotton crop prices, this project empowers farmers, traders, and policymakers with valuable information to make informed decisions. This includes deciding on planting schedules, managing inventory, negotiating fair prices, and developing risk management strategies.

2. Risk Mitigation: The project helps in mitigating risks associated with price volatility in the agricultural market. Farmers can anticipate price trends and adjust their production and marketing strategies accordingly, reducing the impact of market fluctuations on their livelihoods.

3. Economic Stability: Accurate price forecasting contributes to overall economic stability by promoting fair trade practices and market transparency. It encourages investment in the agricultural sector and enhances market efficiency, leading to sustainable economic growth.

4. Food Security: Improved forecasting can positively impact food security by ensuring stable prices for consumers and adequate returns for producers. This stability encourages agricultural productivity and supply chain resilience, contributing to a more reliable food supply.

5. Policy Formulation: Insights from the project can inform policy decisions related to agricultural pricing, trade regulations, and support mechanisms for farmers. Evidence-based policies can promote agricultural sustainability, rural development, and inclusive growth.

6. Technological Advancement: The utilization of advanced Machine Learning and Deep Learning techniques in agricultural forecasting sets a precedent for leveraging technology to address complex challenges in the agricultural sector. It encourages innovation and technological adoption in farming practices, leading to efficiency gains and competitiveness.

5.2 Summary

This thesis project addresses the critical challenge of accurately forecasting orange and cotton crop prices in India's volatile agricultural market. The primary objectives include handling complex and incomplete datasets sourced from the Indian Government, selecting and implementing suitable Machine Learning and Deep Learning algorithms such as LR, SVR, VAR, ARIMA, LSTM, and Prophet to model nonlinear price dynamics, considering regional and seasonal variations, and comparing the efficacy of Deep Learning against traditional methods.

The project's methodology involves comprehensive exploratory data analysis (EDA) to manage data complexity and missing values effectively. It includes the selection and implementation of algorithms for modeling price dynamics, incorporating regional and seasonal aspects, and evaluating model performance through rigorous testing and validation using historical data. The ultimate goal is to deliver precise short-term forecasts that aid stakeholders in making informed decisions amidst market uncertainties.

The results from the comparative analysis of forecasting models highlight FB-Prophet's superior performance in predicting crop prices. FB-Prophet exhibited lower MAE, MSE, and RMSE values compared to LSTM and other models, indicating its ability to capture price trends accurately. These findings emphasize FB-Prophet's suitability for time series forecasting in agricultural markets, providing valuable insights for stakeholders and decision-makers in the agricultural sector, enabling informed decision-making and risk management strategies.

5.3 Conclusion

In conclusion, the project successfully addressed the challenge of accurately forecasting orange and cotton crop prices in India's dynamic agricultural market. Through comprehensive exploratory data analysis (EDA) and the selection of appropriate Machine Learning and Deep Learning algorithms such as Linear Regression (LR), Support Vector Regression (SVR), Vector Autoregression (VAR), Autoregressive Integrated Moving Average (ARIMA), Long Short-Term Memory Networks (LSTM), and Prophet, the project achieved significant advancements in modeling nonlinear price dynamics and improving forecasting accuracy.

The incorporation of regional and seasonal aspects into the forecasting model provided a nuanced understanding of geographical variations and seasonal trends, enhancing the model's predictive capabilities. The comparative analysis demonstrated that FB-Prophet surpassed other methodologies, showcasing superior accuracy with lower Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) values. These results validate FB-Prophet's effectiveness in capturing price fluctuations and providing valuable insights for stakeholders and decision-makers in the agricultural sector.

By delivering precise short-term forecasts and actionable insights, this project contributes to informed decision-making, risk management strategies, and sustainable practices in the agricultural industry. Moving forward, ongoing refinement and integration of advanced modeling techniques will further enhance the accuracy and reliability of crop price forecasting, supporting the agricultural sector's resilience and growth in India.

5.4 Future Scope

The future scope for this project involves several avenues for further exploration and improvement:

1. Integration of External Factors: Incorporating additional external factors such as weather data, market demand-supply dynamics, government policies, and global market trends can enhance the forecasting model's accuracy and robustness. This integration can provide a more comprehensive understanding of the factors influencing crop prices.

2. Advanced Deep Learning Techniques: Exploring advanced deep learning techniques like Convolutional Neural Networks (CNNs) or Transformer models can offer enhanced capabilities in capturing complex temporal patterns and non-linear relationships in crop price data. These models can potentially improve forecasting accuracy further.

3. Ensemble Modeling: Implementing ensemble modeling techniques, such as combining predictions from multiple models (e.g., FB-Prophet, LSTM, SVR), can potentially improve forecast accuracy by leveraging the strengths of each model and mitigating their individual weaknesses.

4. Real-time Data Integration: Developing mechanisms for real-time data integration and updating the forecasting model dynamically can provide timely and up-to-date predictions, enabling stakeholders to make agile decisions in response to changing market conditions.

5. Deployment of Interactive Dashboards: Creating interactive dashboards or visualizations that showcase forecasted crop prices along with key insights and trends can facilitate easier interpretation of the data for stakeholders and decision-makers, enhancing the usability and impact of the forecasting model.

6. Integration with Decision Support Systems: Integrating the forecasting model with decision support systems or platforms can automate decision-making processes based on forecasted prices, enabling proactive risk management strategies and optimal resource allocation in the agricultural sector.

6. Appendix

The appendix serves as a supplementary section, providing detailed information that complements the main body of the document. It includes in-depth information on Machine Learning Models, Deep Learning Models, Results, etc. Additionally, the appendix contains materials referenced in the main text that are not essential for understanding the core concepts but provide additional context and information.

6.1. Dataset Information

6.1.1. Data Sources

- *Description:* Details about the sources of data used in the project, including the Indian Government datasets and any other supplementary data sources.
- *Format:* CSV, Excel, JSON, etc.
- *Size:* We have 58329 rows, 8 columns in the Orange Dataset and 106747 rows and 8 columns in the Cotton Dataset.
- *Time Period:* Range of time covered by the dataset (e.g., 2018-2022).

6.1.2. Data Fields

- Explanation of each field/column in the dataset, including:
- Arrival date
- State
- District
- Market
- Commodity (Orange/Cotton)
- Variety
- Minimum Price
- Maximum Price
- Modal Price
- Other relevant fields

6.2. Preprocessing Steps

6.2.1. Missing Data Handling

- In the Exploratory Data Analysis (EDA) for forecasting orange and cotton crop prices, handling missing values is a crucial step to ensure data integrity and improve model accuracy. The missing values in both datasets were identified using the *isnull().sum()* method, which provides a count of missing values in each column.

6.2.2. Outlier Adjustment

-Outlier adjustment refers to the process of identifying and handling outliers in a dataset. Outliers are data points that significantly deviate from the rest of the data, potentially affecting statistical analysis and machine learning models.Outlier adjustment is a critical step in data preprocessing to ensure the quality and accuracy of analysis and modeling results.

6.3. Feature Engineering

6.3.1. Encoding Techniques

- One-hot encoding is a technique used in machine learning to convert categorical variables into numerical features. We employed multiple strategies for handling missing values, including imputation methods such as mean, median, mode, and nearest neighbor. Categorical variables were converted into numerical format using encoding techniques such as one-hot encoding, label encoding, and ordinal encoding.

6.3.2. Feature Selection

-To refine the feature analysis using only numerical features from dataset, we will exclude categorical features and focus on *min_price, max_price,* and *modal_price.* First step is to identify the numerical and categorical data and then drop the categorical data.

6.4. Model Implementation

6.4.1. Machine Learning Models

Overview of Models:

1. Linear Regression (LR):

- Parameter Settings: Coefficients for linear relationships between input features and output prices.
- Performance Metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE).

2. Support Vector Regression (SVR):

- Parameter Settings: Kernel type (linear, polynomial, radial basis function), regularization parameter.
- Performance Metrics: MAE, MSE, RMSE.

3. Vector Autoregression (VAR):

- Parameter Settings: Lag order selection, VAR model parameters.
- Performance Metrics: MAE, MSE, RMSE.

Evaluation Metrics:

- *Mean Absolute Error (MAE):* Measures the average magnitude of errors between predicted and actual prices.
- *Mean Squared Error (MSE):* Calculates the average squared differences between predicted and actual prices.
- *Root Mean Squared Error (RMSE):* Square root of MSE, providing a measure of the model's accuracy in predicting price movements.

6.4.2. Deep Learning Models

Overview of Models:

1. Long Short-Term Memory Networks (LSTM):

- Architecture Design: Sequential model with LSTM layers for capturing temporal dependencies.
- Hyperparameters: Number of LSTM units, dropout rate, learning rate.
- Evaluation Metrics: MAE, MSE, RMSE.

2. Prophet:

- Architecture Design: Time series forecasting model developed by Facebook, incorporating seasonality and trend components.

- Hyperparameters: Seasonality parameters, trend parameters, changepoint detection settings.

- Evaluation Metrics: MAE, MSE, RMSE.

Evaluation Metrics:

- *Mean Absolute Error (MAE):* Average magnitude of errors between actual and predicted prices.
- *Mean Squared Error (MSE):* Average squared differences between actual and predicted prices.
- *Root Mean Squared Error (RMSE):* Square root of MSE, providing a measure of model accuracy.

6.5. Results and Discussion

6.5.1. Comparative Analysis

- By comparing the results of models we can conclude that FB-Prophet performs better than other models.

7. References

[1] Casper Solheim Bojer. (2022) “Understanding machine learning-based forecasting methods: A decomposition framework and research opportunities”. https://doi.org/10.1016/j.ijforecast.2021.11.003

[2] Yitong Li, Kai Wu, Jing Liu (2023) “Self-paced ARIMA for robust time series prediction”. https://doi.org/10.1016/j.knosys.2023.110489

[3] Ahmed Tealab, Hesham Hefny, Amr Badr (2017) “Forecasting of nonlinear time series using ANN”. https://doi.org/10.1016/j.fcij.2017.05.001

[4] Steven Elsworth and Stefan Güttel,(2020) "Time Series Forecasting Using LSTM Networks: A Symbolic Approach" https://doi.org/10.48550/arXiv.2003.05672

[5] B. Lindemann, Timo Müller, Hannes Vietz,(2021) "A survey on long short-term memory networks for time series prediction" https://doi.org/10.1016/j.procir.2021.03.088

[6] Emir Zunic, Kemal Korjenic, Kerim Hodzic, and Dzenana Donko,(2020) “Application of Facebook’s prophet algorithm for successful sales forecasting based on Real-World Data” http://dx.doi.org/10.5121/ijcsit.2020.12203

[7] Dongqing Zhanga, Guangming Zangb, Jing Lia, Kaiping Maa, Huan Liu. (2018) “Prediction of soybean price in China using QR-RBF neural network model”. https://doi.org/10.1016/j.compag.2018.08.016

[8] Kiran M. Sabu, T. K. Manoj Kumar (2020).“Predictive analytics in Agriculture: Forecasting prices of Areca Nuts in Kerala” https://doi.org/10.1016/j.procs.2020.04.076

[9] Jennifer L. Castle , Michael P. Clements, David F. Hendry (2014).“Robust approaches to forecasting ”. https://doi.org/10.1016/j.ijforecast.2014.11.002

[10] Foteini Kyriazi , Dimitrios D. Thomakos , John B. Guerard (2019).“Adaptive learning forecasting, with applications in forecasting agricultural prices”. https://doi.org/10.1016/j.ijforecast.2019.03.031

[11] Yegnanew A. Shiferaw(2022) “An analysis of East African tea crop prices using the MCMC approach to estimate volatility and forecast the in-sample value-at-risk”. https://doi.org/10.1016/j.sciaf.2022.e01442

[12] J. Scott Armstrong , Kesten C. Green , Andreas Graefe(2015) “Golden rule of forecasting: Be conservative”. http://dx.doi.org/10.1016/j.jbusres.2015.03.031

[13] Yuehjen E. Shao , Jun-Ting Dai(2018) “Integrated Feature Selection of ARIMA with

Computational Intelligence Approaches for Food Crop PricePrediction” https://doi.org/10.1155/2018/1910520

[14] Thomas Dimpfla, Robert C. Jungb, Michael Fladc(2017) “Price discovery in agricultural commodity markets in the presence of futures speculation” https://doi.org/10.1016/j.jcomm.2017.01.002

[15] Liege Cheung , Yun Wang, Adela S.M. Lau ,, Rogers M.C. Chan(2022) “Using a novel clustered 3D-CNN model for improving crop future price prediction” https://doi.org/10.1016/j.knosys.2022.110133

[16] V. Sneha; V. Bhavana (2023) “Sugarcane Yield and Price Prediction Using Forecasting Models” https://doi.org/10.1109/ICECONF57129.2023.10084094

[17] Yung-Hsing Peng, Chin-Shun Hsu, Po-Chuang Huang,(2015) “Developing Crop Price Forecasting Service Using Open Data from Taiwan Markets” http://dx.doi.org/10.1109/TAAI.2015.7407108

[18] B Chaitra; K Meena,(2023)“Forecasting Crop Price using various approaches of Machine Learning” https://doi.org/10.1109/ICIET57285.2023.10220616

[19] Juliana Ngozi Ndunagu, Eyiyemi.Helen Aderemi, Rasheed Gbenga Jimoh, Joseph Bamidele Awotunde(2022) “Time Series: Predicting Nigerian Food Prices using ARIMA Model and R-Programming” https://doi.org/10.1109/ITED56637.2022.10051516

[20] Jinlai Zhang, Yanmei Meng, Jin Wei, Jie Chen, and Johnny Qin,(2021) “A Novel Hybrid Deep Learning Model for Sugar Price Forecasting Based on Time Series Decomposition” https://doi.org/10.1016/j.procir.2021.03.088

[21] Jingyi Shen and M. Omair Shafiq,(2020) “Short‑term stock market price trend prediction using a comprehensive deep learning system” https://doi.org/10.1186/s40537-020-00333-6

[22] Xiao Han, Fangbiao Liu, Xiaoliang He, and Fenglou Ling,(2022) “Research on Rice Yield Prediction Model Based on Deep Learning” https://doi.org/10.1155/2022/1922561

[23] Johnathon Shook, Tryambak Gangopadhyay,(2021) “Crop yield prediction integrating genotype and weather variables using deep learning” https://doi.org/10.1371/journal.pone.0252402

[24] Jie Sun, Liping Di , Ziheng Sun , Yonglin Shen and Zulong Lai ,(2019) “County-Level Soybean Yield Prediction Using Deep CNN-LSTM Model” https://doi.org/10.3390/s19204363

[25] Kavita Jhajhariaa, Pratistha Mathura*,Sanchit Jaina, Sukriti Nijhawana,(2023) “Crop

Yield Prediction using Machine Learning and Deep Learning Techniques" https://doi.org/10.1016/j.procs.2023.01.023]

[26] Alexandros Oikonomidisa, Cagatay Catalb and Ayalew Kassahuna,(2022) "Deep learning for crop yield prediction" https://doi.org/10.1080/01140671.2022.2032213

[27]Wenxiu Hu, Huan Liu , Xiaoqiang Ma,and Xiong Bai,(2023) "The Influence and Prediction of Industry Asset Price Fluctuation Based on the LSTM Model and Investor Sentiment" https://doi.org/10.1155/2023/9790419

[28] Can Yang, Junjie Zhai , and Guihua Tao, "Deep Learning for Price Movement Prediction Using Convolutional Neural Network and Long Short-Term Memory" https://doi.org/10.1155/2022/1113023

[29] Usharani Bhimavarapu, Gopi Battineni and Nalini Chintalapudi , "Improved Optimization Algorithm in LSTM to Predict Crop Yield" http://dx.doi.org/10.3390/computers12010010

[30] Frank Emmert-Streib et al, "An Introductory Review of Deep Learning for Prediction Models With Big Data " https://doi.org/10.3389/frai.2020.00004

[31] Lorenzo Menculini et al, "Comparing Prophet and Deep Learning to ARIMA in Forecasting Wholesale Food Prices" https://doi.org/10.3390/forecast3030040

[32] Hasan Tercan and Tobias Meisen , "Machine learning and deep learning based predictive quality in manufacturing: a systematic review " https://doi.org/10.1007/s10845-022-01963-8

[33] Khulood Albeladi, Bassam Zafar, and Ahmed Mueen , "A Novel Deep-learning based Approach for Time Series Forecasting using SARIMA, Neural Prophet and Fb Prophet" https://doi.org/10.20944/preprints202311.0794.v1

[24] Oskar Triebe, Hansika Hewamalagec, Polina Pilyuginad, Nikolay Laptevb, Christoph Bergmeirc, Ram Rajagopal, "NeuralProphet: Explainable Forecasting at Scale " https://doi.org/10.48550/arXiv.2111.15397