Interdisciplinary Journal of Information, Knowledge, and Management

An Official Publication of the Informing Science Institute
InformingScience.org

IJIKM.org

Volume 17, 2022

MODELING THE IMPACT OF COVID-19 ON THE FARM PRODUCE AVAILABILITY AND PRICING IN INDIA

Niharika Prasanna Kumar — Department of ISE, RV Institute of Technology and Management, Bangalore, India — niharikapk.rvitm@rvei.edu.in

ABSTRACT

Aim/Purpose	This paper aims to analyze the availability and pricing of perishable farm produce before and during the lockdown restrictions imposed due to Covid-19. This paper also proposes machine learning and deep learning models to help the farmers decide on an appropriate market to sell their farm produce and get a fair price for their product.
Background	Developing countries like India have regulated agricultural markets governed by country-specific protective laws like the Essential Commodities Act and the Agricultural Produce Market Committee (APMC) Act. These regulations restrict the sale of agricultural produce to a predefined set of local markets. Covid-19 pandemic led to a lockdown during the first half of 2020 which resulted in supply disruption and demand-supply mismatch of agricultural commodities at these local markets. These demand-supply dynamics led to disruptions in the pricing of the farm produce leading to a lower price realization for farmers. Hence it is essential to analyze the impact of this disruption on the pricing of farm produce at a granular level. Moreover, the farmers need a tool that guides them with the most suitable market/city/town to sell their farm produce to get a fair price.
Methodology	One hundred and fifty thousand samples from the agricultural dataset, released by the Government of India, were used to perform statistical analysis and identify the supply disruptions as well as price disruptions of perishable agricultural produce. In addition, more than seventeen thousand samples were used to implement and train machine learning and deep learning models that can predict and guide the farmers about the appropriate market to sell their farm produce. In essence, the paper uses descriptive analytics to analyze the impact of COVID-19 on agricultural produce pricing. The paper explores the usage of prescriptive analytics to recommend an appropriate market to sell agricultural produce.

Accepting Editor Vishal Shah | Received: June 12, 2021 | Revised: October 21, December 7, December 28, 2021 | Accepted: January 3, 2022.
Cite as: Kumar, N. P. (2022). Modeling the impact of Covid-19 on the farm produce availability and pricing in India. *Interdisciplinary Journal of Information, Knowledge, and Management*, *17,* 35-63.
https://doi.org/10.28945/4897

Contribution	Five machine learning models based on Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Random Forest, and Gradient Boosting, and three deep learning models based on Artificial Neural Networks were implemented. The performance of these models was compared using metrics like Precision, Recall, Accuracy, and F1-Score.
Findings	Among the five classification models, the Gradient Boosting classifier was the optimal classifier that achieved precision, recall, accuracy, and F1 score of 99%. Out of the three deep learning models, the Adam optimizer-based deep neural network achieved precision, recall, accuracy, and F1 score of 99%.
Recommendations for Practitioners	Gradient boosting technique and Adam-based deep learning model should be the preferred choice for analyzing agricultural pricing-related problems.
Recommendations for Researchers	Ensemble learning techniques like Random Forest and Gradient boosting perform better than non-Ensemble classification techniques. Hyperparameter tuning is an essential step in developing these models and it improves the performance of the model.
Impact on Society	Statistical analysis of the data revealed the true nature of demand and supply and price disruption. This analysis helps to assess the revenue impact borne by the farmers due to Covid-19. The machine learning and deep learning models help the farmers to get a better price for their crops. Though the dataset used in this paper is related to India, the outcome of this research work applies to many developing countries that have similar regulated markets. Hence farmers from developing countries across the world can benefit from the outcome of this research work.
Future Research	The machine learning and deep learning models were implemented and tested for markets in and around Bangalore. The model can be expanded to cover other markets within India.
Keywords	Machine Learning, Classification, Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Random Forest, Gradient Boosting, Deep Learning, Artificial Neural Network, Deep Neural Network, Covid-19, Agriculture

INTRODUCTION

Agriculture plays a significant role in Indian Society. Agriculture and allied services are the largest sources of employment. Close to 54.6 % of the workforce is engaged in agriculture. It adds close to 17.8 % to India's Gross Domestic Product (Government of India, 2021). The Indian government supports the farmers by procuring select produce at a minimum support price (MSP) to ensure that farmers get up to one and a half times the production cost of their crops.

Covid-19 dealt a blow to the farming sector in India. Lockdowns during the Covid-19 pandemic led to supply disruptions for some of the farm produce. Loss of employment led to decreased spending and that in turn, led to demand destruction leading to lower price realization and hence a revenue loss for many farmers. Lockdown was imposed around the harvest season of the Rabi crop leading to a shortage of equipment and manpower in the hinterlands of India (Das & Mohanty, 2021). Hence the bumper Rabi crop could not be harvested in time. Due to the lockdowns, the transportation of farm produce got impacted as well (Maggo, 2020). Not just the farmers, the pandemic left a scar on the physical, social, economic, and emotional wellbeing of all stakeholders in the entire agricultural system (Cariappa et al., 2021). To overcome these challenges, experts in the field of

agriculture are stressing on the need to have a nationwide real-time price information system and to allow the farmers to sell their produce in any part of the country (Kumar et al., 2020).

India is a diverse country. Farmers in India cultivate and harvest a multitude of crops. However, in India, agricultural marketing is regulated by the government. Hence, farmers are mandated to sell their agricultural produce in government notified local markets. Due to the Covid-19 pandemic, there was an impact on the pricing of these crops in the local markets. The demand and supply dynamics got impacted as well. There is no literature available on the effect of Covid-19 on the local market. Within these local markets, there could have been noticeable variations in the price of farm produce. For example, the price of Cauliflower in the local markets in Bangalore could be different from the price in adjoining markets in Channapatna (a town about 50 kilometers from Bangalore). Currently, there is no mechanism for the farmers to assess the suitable market for their farm produce. Farmers sell their farm produce in the nearest markets without realizing that they could have gotten a better price for their farm produce had they sold their crop in adjacent markets where the demand for the farm produce could be higher.

Research Objective

There is a need for a detailed analysis of the impact of Covid-19. Specifically, there is a need to analyze the impact of Covid-19 on the farm produce availability as well as gyration in the price of farm produce during the pandemic. Moreover, when the farmers as faced with severe price fluctuations at the Mandis due to demand supply dynamics as seen during the Covid-19 Pandemic, they need a mechanism to determine the right market to sell their produce. Hence, there is a need for research on optimal tools that shall aid the farmer in determining the appropriate market for their produce.

The objective of this research work is to address these two aspects of farmers' needs during the Covid-19 pandemic. Firstly, the work proposes a descriptive analysis of the price of the farm produce before and during the lockdown. The descriptive analysis relies on the dataset released by the Government of India (2020a). One hundred and fifty thousand samples from this dataset spanning different geographies were analyzed. The sections titled "Lockdown and Supply disruption" and "Price Disruption" provide the results of this analysis. The analysis clearly shows that some produce like Ladies Finger witnessed a drop in price across large and small markets in India. The results also reveal that certain vegetables like French beans saw their price increase due to supply constraints. This behavior is visible across the geographically distributed markets in India.

To achieve the second objective the paper proposes data models developed using supervised machine learning methods. These models help the farmers in determining the appropriate market to sell their produce. The section titled "Machine Learning Models" proposes these models. Five supervised machine learning models, namely, Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Random Forest, and Gradient Boosting were implemented. Among the five models, the Gradient boosting model was able to accurately predict the appropriate market for the farmer to sell the produce. Artificial Neural Network-based models are being used in various machine learning-based solutions. This paper proposes three deep learning models based on artificial neural networks that aim to help the farmer to select the appropriate market. The section titled "Deep Learning Models" explores these models.

This paper has been divided into the following sections. The section titled "Literature Review and Related Work" delves into the relevant research work in the field of price discovery of farm produce. Researchers have explored ARIMA, Machine Learning, and Deep Learning-based models in the area of Agricultural pricing. This section explores the research work carried out in these three fields. The section titled "Impact of Covid-19 on Farm Produce" analyzes the price movement of perishables across different markets in India during the first half of the year 2020. This section describes the results of the statistical analysis of price disruption as well as supply disruption faced in the Agricultural sector due to Covid. The section titled "Machine Learning Models" proposes five supervised

models to help farmers decide on the optimal market to sell their produce. The models are graded based on four performance metrics, namely, precision, recall, accuracy, and F1-Score. This section also describes the various Hyper-parameter tuning procedures that were applied to improve the performance of these five models. The subsequent section titled "Deep Learning Models" proposes multi-layered artificial neural networks (Deep Neural Network) models to predict the target market for the farmers. Neural network models based on Stochastic Gradient Descent (SGD), Adagrad, and Adam optimizers are evaluated against the four performance metrics. The final section of this paper provides a conclusion of the research work with a proposal for future work based on the current research work.

LITERATURE REVIEW AND RELATED WORK

HISTORY OF AGRICULTURAL PRICING IN INDIA

India is one of the few countries that have a regulated market for agricultural produce. The regulations date back to the pre-Independence era when the Government of India, under the rule of the erstwhile East India Company, produced a report on agriculture (Government of India, 1928). The report emphasized the need to curb malpractices by the private procurement agencies so that farmers can realize better returns. In 1955, Independent India enacted the Essential Commodities Act (Government of India, 1955) that gave the government power to regulate the storage, transport, import, export of crops. The intent was to curb the practice of hoarding agricultural produce. The act was supposed to help consumers procure the produce at an affordable cost. The agricultural ecosystem saw few defining laws being enacted by the respective states in India in the 1960s and 70s. These laws are collectively called the Agricultural Produce Market Committee (APMC) Act (Agricultural Produce Market Committee, 2021). These laws paved the way for the creation of local markets (called Mandis) in cities and towns across India. The farmers would bring their produce and auction them in these mandis. Over the past few decades, the Indian government has procured farm products like wheat and paddy above a Minimum Support Price (MSP). MSP is the price that is fixed for each of the scheduled crops every year. MSP is usually one to one and half times the production cost of the crops (Tripathi, 2012). These laws and statutes are intended to save the farmers from the local dealers who supposedly force farmers to sell their produce at lower costs.

Over the years, these farm laws stifled the independence of the farmers. The farm laws led to a situation where the first sale of all agricultural products had to happen in these APMC markets. The farm produce had to be sold only to a select few people called licensed commission agents. The farmers also had to pay hefty fees that led to further erosion in their income. To help the farmers from this trap of commission agents, the government of India notified three laws (Government of India, 2020b, 2020c, 2020d). These laws enable the farmers to sell the farm produce beyond the APMC markets in their area. They can now sell it in any market in any part of India. The laws also empower private participants to buy the produce directly from the farmers. These new laws bring the agricultural produce out of the purview of the essential commodities act. The private players can build storage facilities to store the items procured from the farmers. These new laws intend to help farmers realize a higher price for their goods.

ARIMA MODEL BASED AGRICULTURAL PRICING

Predicting the future price of farm produce is an important area that has seen interest from the industry and government. Autoregressive Integrated Moving Average, or ARIMA, is one of the tools used to predict farm pricing. Often referred to as Box and Jenkins Model (Box & Jenkins, 1976), ARIMA uses three parameters (p, d, q) to define the model. "p" represents the number of lag observations, "d" represents the number of times the raw observations are differenced and "q" represents the size of the moving average window. In the Indian context, ARIMA has been used to forecast the price of Coriander in Rajasthan (Verma et al., 2016). Authors predicted a linear price increase from

Rs. 9677 to Rs. 9909 between July 2015 to December 2015. ARIMA (0, 1, 1) model was the most reliable model with the lowest values of Akaike's Information Criteria (AIC) and Schwartz Bayesian Criteria (SBC/BIC). ARIMA has also been used to forecast milk production in India (Deshmukh & Paramsivam, 2016). The authors used multi-decade data sets from organizations like the United Nations Food and Agricultural Organization database (FAOSTAT) and National Dairy Development Board (NDDB) to estimate the Milk production for 2017. ARIMA (1, 1, 1) model predicted the production volume with the lowest values for AIC, SBC, R2, and Mean Absolute Percentage Error (MAPE). ARIMA has also found application in forecasting the price of Paddy in India (Darekar & Reddy, 2017). The authors propose a state (or region) specific ARIMA model that was able to predict the price of Paddy for the Karif season of 2017. The model predicted that farmers would be able to realize a price of Rs. 1,600 – 2,200 per quintal. The models were evaluated against AIC, MAPE, and SBC. ARIMA was also used to forecast the green gram prices in Maharashtra (Chaudhari & Tingre, 2014). The authors evaluated different ARIMA models and concluded that ARIMA (0, 1, 0) was the optimal model with the lowest values for MAPE, R2, BIC. The model predicted a price range of Rs. 4,646 to Rs. 4,729 for Green Gram from October 2012 to February 2013. An ARIMA model that could forecast the onion prices in the Kolhapur district in Maharashtra (Darekar et al., 2016) has also been developed. Auto Correlation Function (ACF) and Partial Auto Correlation Function (PACF) values were observed to determine the values for "d", "p", and "q" values of the ARIMA model. ARIMA (1, 1, 1) model was found to be the optimal model to predict the price of the onion. ARIMA has also been used to forecast the Basmati rice crop in Haryana (Sain et al., 2020). Authors computed ACF and PACF for the input data and observed that both ARIMA (0, 1, 0) and ARIMA (1, 1, 2) were able to predict the price of Basmati rice crop with the lowest values for Root Mean Square Error (RMSE), MAPE and BIC.

Machine Learning Model Based Agricultural Pricing

Supervised machine learning models are being used in many research problems that require prescriptive analytics. Some of the well-known and widely used machine learning models employed for classification problems are described below:

Logistic regression

Pierre François Verhulst introduced the concept of logistic regression in the early 1830s. Logistic regression is a statistical model that uses the sigmoid function (Equation 1) to perform the classification of sample space.

$$S(mX + b) = \frac{1}{1 + e^{-(mX+b)}} \quad (1)$$

K-Nearest neighbor

Fix, and Hodges (1951) introduced the concept of K-Nearest Neighbors. K-Nearest Neighbor (Cover & Hart, 1967) is a non-parametric classification method that uses the "K" closest samples to determine the class membership of the test sample. The test sample shall belong to the class that is most common among the "K" samples.

Support vector machine

Boser et al. (1992) introduced the concept of support vector machines. Cortes and Vapnik (1995) extended this concept to non-separable data. Support vector machine is a non-probabilistic supervised machine learning method that classifies "n" dimensional samples into classes. This is achieved by drawing a higher dimensional hyperplane that maximizes the margin (i.e., Euclidean distance) between the nearest samples to the hyperplane (i.e., support vectors).

Random forest

Random Forest (Breiman, 1999) is an ensemble learning method (Polikar, 2006). Ensemble learning aims to build a predictive model by integrating multiple models (Rokach, 2010). Random Forest finds application in classification problems that strive to create many decision trees during the training phase to classify the samples into one target class. A mode of these decision trees performs the classification.

Gradient boosting

Kearns (1988) proposed the idea of hypothesis boosting. It heightens a weak hypothesis to transform it into a better one. Schapire (1990) proposed a method to convert the weakly learnable algorithm, which is slightly better than random guessing, into a system that can generate a highly accurate hypothesis. Freund and Schapire (1996) introduced Adaptive Boost, i.e., AdaBoost. Adaboost was one of the first successful boosting algorithms developed. Friedman (2001) further developed this framework and introduced Gradient Boosting Machine. Gradient boosting combines many weak learners, one at a time (i.e., iterative fashion), to generate a strong learner. The output thus generated aims to reduce the mean square error between the predicted and the observed values.

Machine learning models used in agriculture pricing

Machine learning models have been widely used to predict the pricing of agricultural products. Agrawal (2020) proposes a logistic regression-based agriculture pricing model. This model predicts the minimum support price for the crop based on the previous year's minimum support price. Additionally, the model guides the Farmer to choose the appropriate crop for the current season. Wang (2016) explores using logistic regression with features consisting of the price of the futures contract for different durations (5-day, 10-day, 15-day, and 20-day contracts). The model attempts to predict the price of agricultural products like corn, soybean, etc.

The state government of Madhya Pradesh in India commissioned a study (Atal Bihari Vajpayee Institute of Good Governance and Policy Analysis [AIGGPA], 2020) to predict the farm price for seven crops, namely Soybean, Bengal Gram, Mustard, Lentil, Maize, Red Gram, and Black Gram. The model uses features like arrival rate, the area under cultivation, the product-wise yield of the crop, minimum support price set by the government, weather data, spot price of the goods to predict the value of the crops fifteen days in advance. Four different machine learning techniques, namely, Random Forests, Support Vector Machine, LASSO, and Generalized Linear Model, were used to develop the model. The model was able to achieve an accuracy of 95 %.

Random Forest and Gradient boosting models have been used to predict the impact on the price of Maize based on the quantity of Maize production in North America (Zelingher et al., 2021). Using a Random forest with 500 trees and a gradient boosting model, authors could predict that an 8% increase in North American output of maize results in a 7% drop in the global price of Maize.

Deep Learning Model Based Agricultural Pricing

Over the past decade, Deep Neural Networks (DNN) based models have gained popularity in various aspects of machine learning. A Deep Neural Network is a type of deep learning architecture that involves computing units called the neurons, as shown in Figure 1.

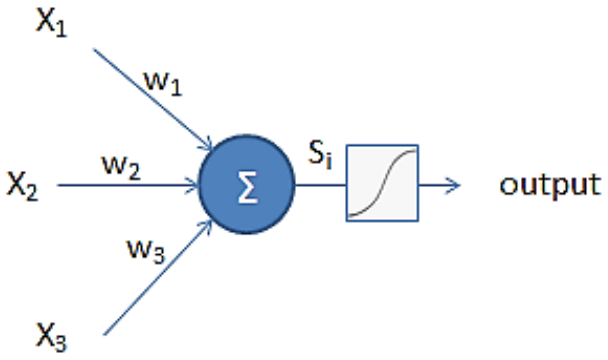

Figure 1: Artificial Neuron

A neuron computes a weighted sum of the inputs to generate an output (Equation 2).

$$S_i = \sum_{i=1}^{n} w_i x_i = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n \quad (2)$$

Many such artificial neurons work in parallel to form a layer. Many such layers process data sequentially (or sometimes in parallel) to create an artificial neural network called DNN (as shown in Figure 2). A neural network attempts to solve complex problems, such as speech recognition, drug design, computer vision, natural language processing, etc.

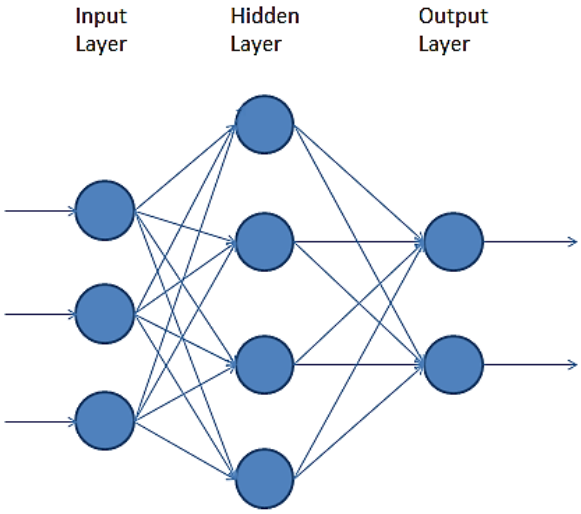

Figure 2: Artificial Neural Network

In addition to being deployed widely in the field of computing applications, Artificial Neural Networks (ANN) have also been employed to model and predict agricultural pricing. Unlike ARIMA, ANN can generate a model that can handle non-linear data. In the Indian context, ANN was employed to predict the price of Potato (Choudhary et al., 2019). Authors propose that the agricultural price series data which is generally nonlinear and non-stationary, be broken down using adaptive time series decomposition (like empirical mode decomposition or EMD) to generate independent intrinsic mode function and residues. The residues are then fed to artificial neural networks to predict the price of crops like potatoes. The authors propose a feed-forward ANN with a single hidden layer. To determine the number of nodes in the hidden layers the authors experimented with values between 2 to 20 and found optimal results with 17 nodes in the hidden layer. This EMD based model was able to forecast the short-term price of potatoes with an RMSE of about 23. ANN has also been used to forecast the price of oilseeds (Jha & Sinha, 2014). Authors use a feed-forward time-delay neural network (TDNN) to predict the price of oil seeds. ANN has also found application in predicting the

price of soybean and rapeseed-mustard (Jha & Sinha, 2013). The authors propose a hybrid strategy of using ARIMA and ANN together. The data was divided into time series (i.e., linear data) and non-linear data. The time-series data was passed through ARIMA and non-linear data was passed through a TDNN. ARIMA (1, 1, 0) was used for Soybean, and ARIMA (2, 1, 0) was used for rapeseed-mustard. The ANN is comprised of two input nodes and eight hidden nodes. The output of ARIMA and ANN models was combined to generate a single result. The results were evaluated using the RMSE (Equation 3) and Mean Absolute Deviation (MAD) (Equation 4). For Soybean, this hybrid model achieved an RMSE of 31.50 and MAD of 25.60 for a 12-month ahead prediction.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \quad (3)$$

$$MAD = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t| \quad (4)$$

ANN has also been used to forecast rice exports in Thailand (Co & Boosarawongse, 2007). The performance of ANN was better when compared against the ARIMA model. Backpropagation and Genetic Algorithm based ANN have also been used to predict the pricing of agricultural products (Subhasree & Priya, 2016). Backpropagation ANN was able to achieve an accuracy of 79%. Genetic Algorithm based ANN generated an accuracy of 89%.

Limitations of Existing work

ARIMA is inherently an autoregression model that relies on the dependent relationship between an observation and the other lagged observations. Autoregression (i.e., "AR" in ARIMA), Integrated (i.e., "I" in ARIMA), and the Moving Average (i.e., "MA" in ARIMA) are related, in some form or the other to the past data. However, in exigency situations like the Covid-19, the demand-supply dynamics keep changing every day. This change in dynamics results in non-linear data. In such a scenario, ARIMA may not be able to predict the price of the farm produce with reasonable accuracy. Hence, ARIMA would not help the farmers in realizing optimal prices for their goods.

Even the ANN-based approach, used to predict the pricing for the farm produce, does not consider the dynamic nature of the farm prices. Moreover, there isn't enough literature on employing ANN for dynamic farm produce pricing in the Indian context. The ANN models available in the current literature do not consider the market dynamics of these local markets in regulated countries, for example, the auction prices and transportation modalities. These models use the constant price to predict the future price of the crops.

This paper tries to address the gap by using the farm price of agricultural produces before and during the Covid-19 pandemic to predict the optimal market for the farmers to sell their produce. The model thus generated takes into account the daily auction price of the source and destination markets as well as other dynamic parameters that are local, for example, the transportation cost structure and the transportation model. The models generate a timely prediction so that the farmers can decide whether to sell their produce in the local market or transport them to a larger city market. In addition, the paper analyzes the impact of the Covid-19 pandemic on perishable food items like vegetables and provides insights to the researchers on the price gyrations that the agricultural system had to face due to the pandemic.

Impact of Covid-19 on Farm Produce

About one hundred and fifty thousand data samples from the dataset released by the Government of India (2020a) were analyzed. The intent was to observe and draw inferences regarding the price movement of some of the perishables. The timeframe covers three months before the Covid-induced lockdown and the subsequent three months during the lockdown.

Analysis of the data reveals two types of disruptions, namely, supply disruptions and price disruptions. The subsections below delve into the details for both these disruptions.

Lockdown and Supply Disruption

Though the lockdown was in force for many weeks, essential services like pharmaceutical supplies and farm produce did not witness extended lockdown. Figure 3 captures the supply disruption for farm produce in few large cities across India. In bigger cities, the markets would have seen supply disruption of anywhere between seven to ten days.

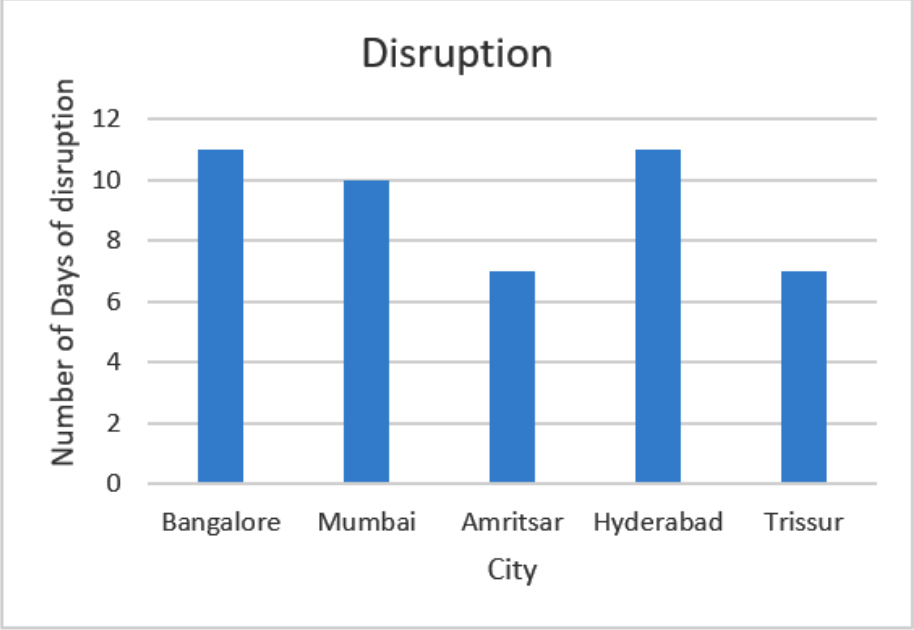

Figure 3: Supply disruption in major cities during the lockdown

Price Disruption

Due to the lockdown, there were challenges in transportation, storage, auction, and distribution of farm produce. The demand-supply dynamics changed overnight. Hence, lack of supply of certain types of vegetables would have resulted in a price rise at the Mandi. On the other hand, lack of demand or oversupply of few fruits and vegetables would have resulted in a price drop. The heterogeneous nature of the Indian population coupled with the varying demand-supply dynamics led to diverse price movements for perishables across different regions in India.

Price drop scenario

Analysis of price movements reveals that vegetables like Ladies Finger saw their prices drop during the lockdown. The average price of vegetables like Ladies' Finger was higher before lockdown. During the lockdown, the average price dropped by about 20-25%. This price drop continued throughout the lockdown. Figure 4 depicts the price movement in Ladies' Finger for Bangalore, Hyderabad, and Amritsar. The latter half of the graph shows the drop in price during the lockdown.

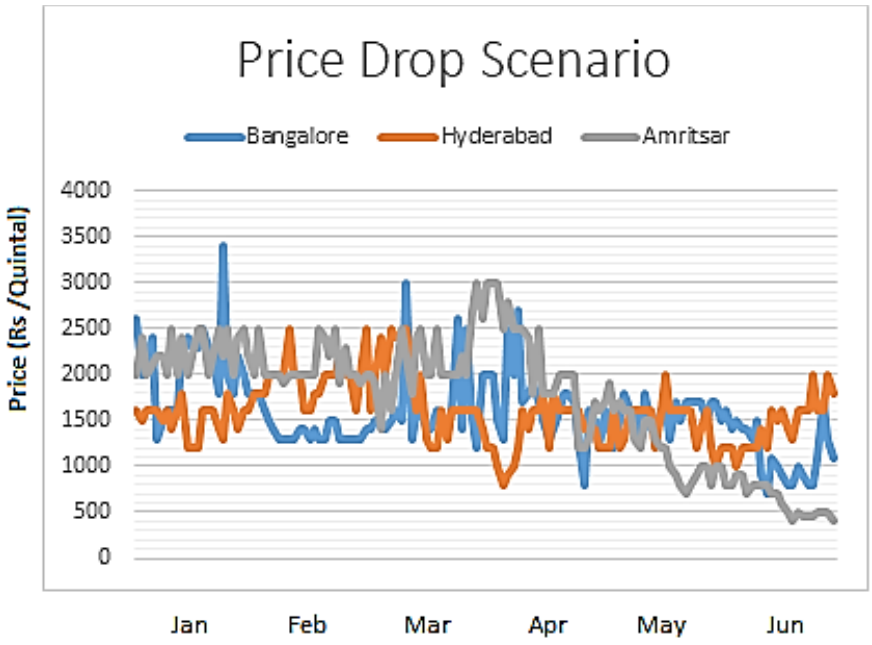

Figure 4: Price drop observed for Ladies' Finger in Bangalore, Hyderabad and Amritsar

Kolkata saw a dramatic drop in prices during the week of lockdown, and these lower prices sustained for many weeks into the lockdown. Figure 5 depicts the price drop for Kolkata.

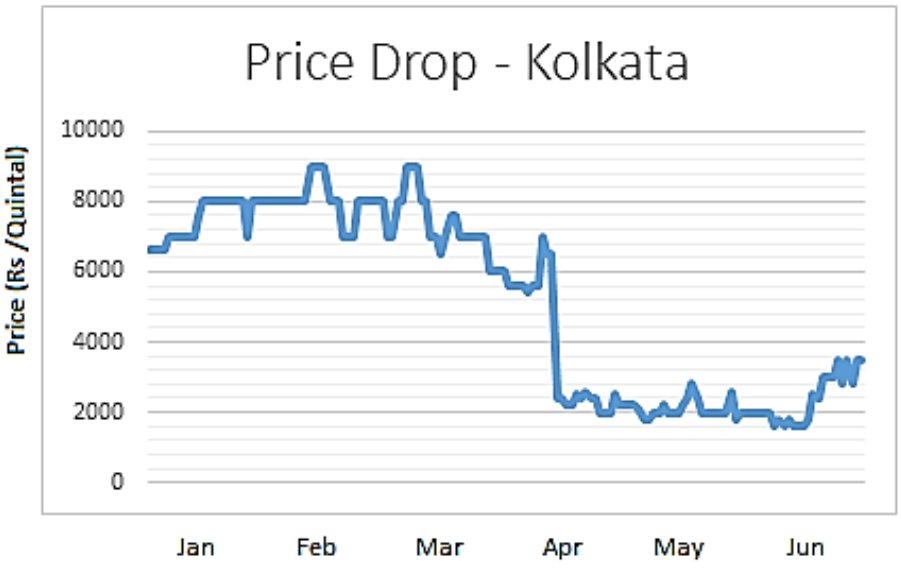

Figure 5: Price drop in Kolkata for Ladies' Finger during the first half of 2020

Table 1 captures the price change of Ladies' Finger for a few more cities during the lockdown. The uniform drop in price across different cities and towns indicates either a lack of demand for Ladies' Finger or oversupply during the lockdown. There were few exceptions though, cities like Trissur witnessed a price increase.

Table 1: Price change for Ladies' Finger in other cities

City	Price Change During Lockdown
Ambala	Price dropped
Ahmedabad	Price dropped
Surat	Price dropped
Cuttack (Banki)	Price dropped
Trissur	Price increased

Price rise scenario

Some vegetables like French Beans experienced a rise in price during lockdown compared to their price before lockdown. This price increase could be due to either a supply shortage or an increase in demand. This price rise was sustained throughout the lockdown. Figure 6 shows the price rise for French Beans in Mumbai, Surat, and Hyderabad. The latter half of the graph depicts the price rise due to the lockdown.

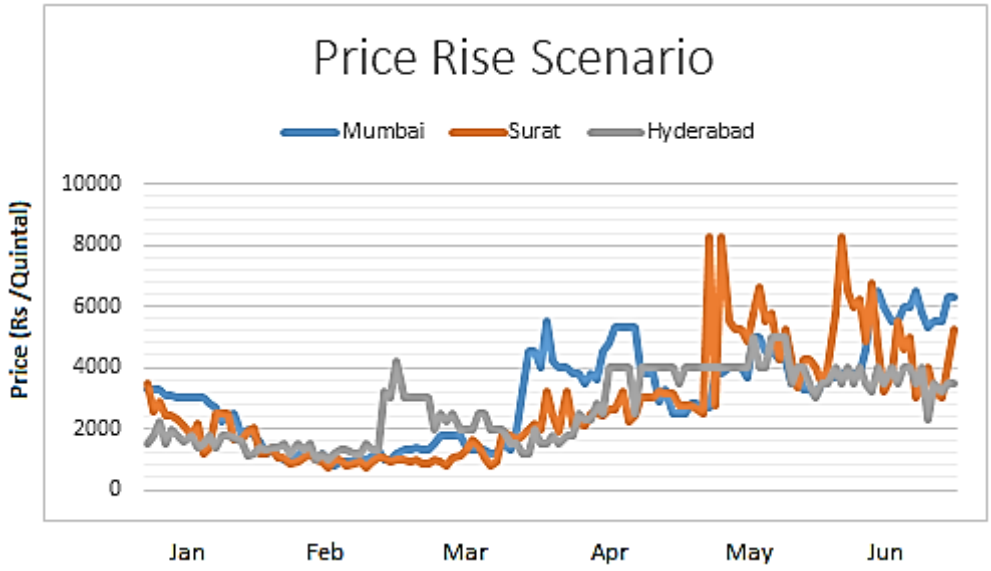

Figure 6: Price rise observed for French Beans in Mumbai, Surat, Hyderabad

Stable price scenario

Some vegetables like Cabbage did not see a noticeable increase or decrease in their price during the lockdown. The demand for such vegetables seemed to have gotten managed with the available supply. Moreover, vegetables like Cabbage generally have a relatively long shelf life. Hence, these vegetables can be stocked for a longer duration to mitigate the sudden increase in demands. There were intermittent spikes in the price which, returned to their mean price within a few days. Figure 7 shows the price for Cabbage in Amreli, Hyderabad, and Burdwan.

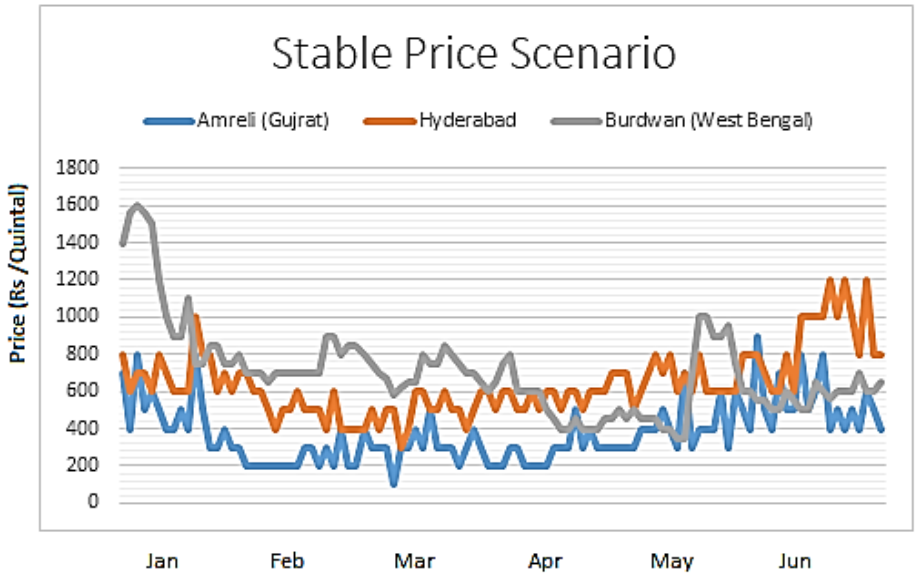

Figure 7: Stable price scenario for Cabbage in Amreli, Hyderabad and Burdan

MACHINE LEARNING MODELS

In the first half of 2020, farmers in India would have experienced volatility in the auction price for their produce in their local markets (i.e., Mandis). To recover their costs and to maximize the returns for their crop, data models have been proposed in this paper. These models inform the farmers whether it is profitable to sell the produce in the local markets of their respective towns or transport them to the larger cities and auction them in the city markets. These models can be used post Covid-19 pandemic as well thereby making them an efficient price discovery tool for the farmers.

DATASET

A dataset created from the government data available at https://data.gov.in/resources-from-web-service/3670701 (Government of India, 2020a), comprising 14,179 samples, was used to create and test the machine learning models. Bangalore's Binny Mills was chosen as the destination market of choice, and local markets in towns around Bangalore like Ramanagara, Chikkaballapura, and Channapatna were considered as the source markets.

SAMPLE SELECTION

These markets trade a wide variety of farm produce like fruits, vegetables, pulses, and cereals. Generally, cereals like wheat and rice, tubers like Potatoes, and bulbs like onions have a longer shelf life. Such farm produce may not be auctioned daily at the markets. Hence, such items are excluded from this analysis. Some vegetables and fruits like Mango are seasonal and omitted from the analysis process. Perishables like vegetables are auctioned daily and sold throughout the year. Hence such farm produce are considered for the analysis.

FEATURE SELECTION

The features shortlisted to train the classification model are shown in Table 2.

Table 2: Feature set for machine learning model

Feature	Description
Source Market	Local markets in towns near Bangalore. For this paper Ramanagara, Chikkaballapur and Channapatana were considered as source cities. These cities/towns were selected as the source cities based on three major factors:- a. They are within the radius of 50 – 60 kilometers from the destination market. Being in the vicinity of the destination market gives the farmers a realistic chance to reach the destination market in time before the markets close for the day. b. These cities/towns have satisfactory transportation facilities (like Trucks, Vans etc.). c. These cities/towns are connected to the destination market with proper asphalted roads.
Destination Market	Bangalore's Binny Mills was chosen as the destination market as it is one of the largest markets for agricultural produce in South India.
Farm Produce	Perishables auctioned daily and are available throughout the year. Cabbage, Cauliflower, Capsicum, and Ladies' finger have been chosen for this analysis. These vegetables were chosen because they are consumed in most households across the country.
Source City Average Price	The daily average auction price of the vegetable in the source city/town markets.
Destination Market Average price	The daily average auction price of the vegetable in the destination market.
Transportation Model	The daily auctions in a market are usually conducted within a specific time window. Hence the farmers have to consider multiple factors like the travel start time, expected vehicular traffic on the highway, traffic within Bangalore city till the market doorstep, and auction closing time. The model incorporates these factors to come up with a transportation model that is input to the machine learning model. The transportation model is an important feature. A substantial delay in transportation of the farm produce could lead to delayed arrival of produce which in turn could potentially lead to unsold inventory for the farmer.
Transportation Cost	Transporting the goods from the source city to the destination market involves multiple aspects like the distance between cities, rent quoted for the transporting vehicle, fuel expense, oil expense, driver cost, road transport tax, transporter commission, highway toll, etc. The model factors all these parameters as transportation cost. Transportation cost is an important feature that helps determine whether to sell the produce in the destination market.

Data Cleansing

Data samples with empty cells for the prices of perishables were purged from the dataset. Statistical estimates based on monthly average price or based on the previous day or next day's price could have been used to augment the missing data. But that would involve possessing a detailed knowledge of daily demand-supply dynamics across India. This is a difficult proposition. Hence, these data samples

were dropped from the dataset. Similarly, samples with abnormally high or low prices were purged as well. Such samples could have gotten generated due to errors during the process of data entry.

Performance Metrics

The five classification models were compared against each other using the following performance metrics.

Confusion matrix

Confusion Matrix is a 2x2 matrix that categorizes the classification results into four different buckets, as shown in Table 3.

Table 3: Confusion Matrix

	Predicted Negative	**Predicted Positive**
Actual Negative	True Negative	False Positive
Actual Positive	False Negative	True Positive

- True Negative (TN): The predicted class from the model and the actual (or expected) class match. The result is a negative class.
- False Positive (FP): The model classifies the samples as part of the Positive class, whereas they belong to the Negative Class.
- False Negative (FN): The model classifies the samples as part of the Negative class whereas they belong to the Positive Class.
- True Positive (TP): The predicted class from the model and the actual (or expected) class match. The result is a positive class.

Precision

Precision attempts to find out the proportion of the identification that was correct (Equation 5).

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (5)$$

Recall

Recall attempts to find out the proportion of the actual positives that were identified correctly (Equation 6).

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (6)$$

Accuracy

Accuracy attempts to find out the proportion of the samples that were identified correctly (Equation 7).

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total\ Samples} \quad (7)$$

F1 score

F1 Score is the harmonic mean between precision and recall. It conveys the balance between the two values (Equation 8).

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (8)$$

RESULTS

Machine learning models were developed and tested using Scikit-Learn (Pedregosa et al., 2011) on a Windows 10 based system comprising of an 11th generation Intel® CoreTM i5-1135G7 @ 2.40GHz processor running with an 8MB Cache, 8 GB RAM, and Intel® IrisTM Xe Graphics.

The dataset was divided into train and test sets with a 70:30 split. Models were trained using the training set on five different classification techniques, namely, Logistic Regression, K-Nearest Neighbor, Support Vector Machine, Random Forest, and Gradient Boosting Classifier. Performance analysis was conducted by measuring and computing metrics like precision, recall, accuracy, and F1-Score.

Logistic regression model

Logistic regression-based models were developed with the default hyperparameters as well as with the tuned hyperparameters. Hyperparameter tuning was achieved using RandomSearchCV. The values used to create the models are listed in Table 4.

Table 4: Hyperparameter values for logistic regression models

Parameter	Default Values	Tuned Values
C: loguniform(1e-5, 100)	1	0.006
Penalty: none, l1, l2, elasticnet	l2	l2
Solver: newton-cg, lbfgs, liblinear	lbfgs	Liblinear

Table 5 lists the confusion matrix generated by both models.

Table 5: Confusion matrix for logistic regression models

	Using Default Parameter Values		Using Tuned Parameter Values	
	Predicted Negative	**Predicted Positive**	**Predicted Negative**	**Predicted Positive**
Actual Negative	2706	284	2741	249
Actual Positive	342	922	371	893

The performance metrics thus computed for these models are listed in Table 6:

Table 6: Performance comparison between the models with default and tuned hyperparameters

Using Default Parameter Values	Using Tuned Parameter Values
• Precision : 76 % • Recall : 72 % • Accuracy : 85 % • F1 Score : 74 %	• Precision : 78 % • Recall : 70 % • Accuracy : 85 % • F1 Score : 74 %

K-nearest neighbor model

K-Nearest Neighbor based models were developed with the default hyperparameters as well as with the tuned hyperparameters. Hyperparameter tuning was achieved using RandomSearchCV. The values used to create the models are listed in Table 7.

Table 7: Hyperparameter values for k-nearest neighbor models

Parameter	Default Value	Tuned Value
n_neighbors: 1, 2, …., 31	5	11
Weights: uniform, distance	Uniform	Distance

Table 8 lists the confusion matrix generated by both models.

Table 8: Confusion matrix for k-nearest neighbor models

	Using Default Parameter Values		Using Tuned Parameter Values	
	Predicted Negative	**Predicted Positive**	**Predicted Negative**	**Predicted Positive**
Actual Negative	2909	81	2916	74
Actual Positive	95	1169	97	1167

The performance metrics thus computed for these models are listed in Table 9.

Table 9: Performance comparison between the models with default and tuned hyperparameters

Using Default Parameter Values	Using Tuned Parameter Values
• Precision : 93 % • Recall : 92 % • Accuracy : 95 % • F1 Score : 92.5 %	• Precision : 94 % • Recall : 92 % • Accuracy : 95 % • F1 Score : 93 %

Support vector machine model

Support Vector Machine (SVM) based models were developed with the default hyperparameters as well as with the tuned hyperparameters. Hyperparameter tuning was achieved using RandomSearchCV. The values used to create the models are listed in Table 10.

Table 10: Hyperparameter values for support vector machine models

Parameter	Default Value	Tuned Value
C: .01, .1, 1, 5, 10, 100	1.0	5.0
Kernel: linear, RBF	RBF	RBF
Gamma: .01, .1, 1, 5, 10, 100	1	1
Random State: 0, 1	0	0

Table 11 lists the confusion matrix generated by both models.

The performance metrics computed for the SVM models are listed in Table 12:

Table 11: Confusion matrix for support vector machine models

	Using Default Parameter Values		Using Tuned Parameter Values	
	Predicted Negative	**Predicted Positive**	**Predicted Negative**	**Predicted Positive**
Actual Negative	2911	79	2950	40
Actual Positive	127	1137	31	1233

Table 12: Performance comparison between the models with default and tuned hyperparameters

Using Default Parameter Values	Using Tuned Parameter Values
• Precision : 93 % • Recall : 89 % • Accuracy : 95 % • F1 Score : 91 %	• Precision : 97 % • Recall : 97 % • Accuracy : 98 % • F1 Score : 97 %

Random forest classifier model

Random forest classifier based models were developed with the default hyperparameters as well as with the tuned hyperparameters. Hyperparameter tuning was achieved using RandomSearchCV. The values used to create the models are listed in Table 13:

Table 13: Hyperparameter values for random forest classifier models

Parameter	Default Value	Tuned Value
n_estimators: 100, 200, 300, 400, 500	100	400
max_depth: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100	default=None	50
min_samples_leaf: 1, 2, 4	1	1

Table 14 lists the confusion matrix generated by both models.

Table 14: Confusion matrix for random forest classifier models

	Using Default Parameter Values		Using Tuned Parameter Values	
	Predicted Negative	**Predicted Positive**	**Predicted Negative**	**Predicted Positive**
Actual Negative	2965	25	2966	24
Actual Positive	18	1246	17	1247

The performance metrics thus computed for these models are listed in Table 15.

Table 15: Performance Comparison between the models with default and tuned hyperparameters

Using Default Parameter Values	Using Tuned Parameter Values
• Precision : 98 % • Recall : 98 % • Accuracy : 98 % • F1 Score : 98 %	• Precision : 98 % • Recall : 98 % • Accuracy : 98 % • F1 Score : 98 %

Gradient boosting classifier model

Gradient boosting classifier based models were developed with the default hyperparameters as well as with the tuned hyperparameters. Hyperparameter tuning was achieved using RandomSearchCV. The values used to create the models are listed in Table 16.

Table 16: Hyperparameter values for gradient boosting classifier models

Parameter	Default Value	Tuned Value
learning_rate: 0.1, 0.3, 0.5, 0.7	0.1	0.7
n_estimators: 100, 200, 300, 400, 500	100	400
Subsample: 0.4, 0.6, 0.8, 1.0	1.0	0.8
max_depth: 2, 3	3	3
min_samples_leaf: 1, 10, 50, 100	1	100

Table 17 lists the confusion matrix generated by both models.

Table 17: Confusion matrix for gradient boosting classifier models

	Using Default Parameter Values		Using Tuned Parameter Values	
	Predicted Negative	**Predicted Positive**	**Predicted Negative**	**Predicted Positive**
Actual Negative	2936	54	2988	2
Actual Positive	38	1226	4	1260

The performance metrics thus computed for these models are listed in Table 18.

Table 18: Performance Comparison between the models with default and tuned hyperparameters

Using Default Parameter Values	Using Tuned Parameter Values
• Precision : 96 % • Recall : 96 % • Accuracy : 97 % • F1 Score : 96 %	• Precision : 99 % • Recall : 99 % • Accuracy : 99 % • F1 Score : 99 %

Analysis and Discussion

Table 19 lists the performance of all the models. The table captures the performance based on the tuned hyper-parameters.

Table 19: Performance Comparison between all the models with tuned hyperparameters

Performance Metric	Logistic Regression	K-Nearest Neighbor	Support Vector Machine	Random Forest	Gradient Boosting
Precision	78%	94%	97%	98%	99%
Recall	70%	92%	97%	98%	99%
Accuracy	85%	95%	98%	98%	99%
F1-Score	74%	93%	97%	98%	99%

Logistic regression classifier performance analysis

Among the five models, logistic regression had the lowest performance score. Logistic regression is a widely used algorithm for binary classification. It is fast and does not consume large amounts of computational resources. Many researchers use logistic regression as a benchmark to evaluate more complex models like ensemble models.

Logistic regression, however, performed poorly on the agricultural dataset. Even after tuning the hyperparameters, the model could not improve its performance.

Multicollinearity

One possible reason for poor performance in logistic regression-based models is Multicollinearity ("Logistic Regression," 2021). If the predictors (or the features) are strongly correlated, then there are chances that the model will not converge and hence may perform poorly. A correlation heatmap plotted to analyze the collinearity among the predictors is shown in Figure 8. The correlation matrix for the agricultural data does not indicate a strong correlation among the predictors. Hence multicollinearity is not a factor that could have influenced the performance of the model.

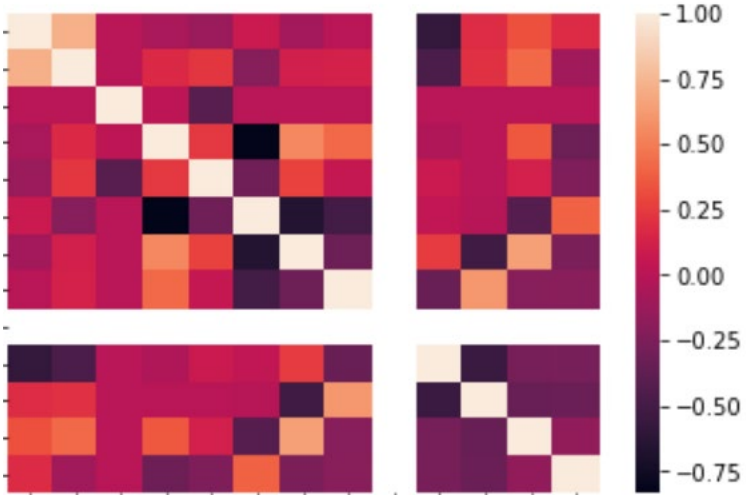

Figure 8: Correlation heat map for the predictors (i.e., features)

P-value

The other aspect that can influence the outcome of logistic regression is called the P-value (Minitab Blog, 2013). P-value tests the impact that a predictor or feature has on the overall outcome of the

model. A low predictor value, for example < 0.05, indicates that the feature has a meaningful impact on the capability of the regression model. A higher P-value indicates that the feature has a lesser impact on the prediction capability of the regression model.

When the "P-Value" was computed for the logistic regression model, it was noted that three features, Transportation Cost, Source City, and Farm Produce had a relatively higher "P-Value". Each of these features with a higher "P" value was removed from the regression model one at a time to check the performance of the model. Even after removing these predictors from the regression model, the performance of the model did not witness any noticeable improvements.

The possible optimizations to the linear regression model did not yield any improvements. Hence logistic regression model may not be suitable for the current problem.

K-nearest neighbor classifier performance analysis

K-Nearest Neighbor (KNN) employs instance-based learning and does not have a training step associated with it. It constantly evolves as the new training samples get introduced to the system. Compared to logistic regression, the KNN model had a much better performance for the agricultural dataset. However, KNN can suffer from the curse of dimensionality as described below.

Principal component analysis

The performance of KNN can get impacted if the number of features or dimensions increases ("Curse of Dimensionality," 2021). One way to mitigate the performance impact due to feature explosion is to reduce the dimensionality of the dataset. Principal Component Analysis (PCA) is an unsupervised statistical technique used to reduce the dimension of a dataset. PCA is a multi-step process wherein the input data gets normalized. Subsequently, the covariance matrix for the data gets generated. The covariance matrix acts as an input to generate the Eigenvalues and Eigenvectors. The Eigenvalues get sorted in decreasing order. The Eigenvectors corresponding to the maximum Eigenvalues get chosen. These Eigenvectors eventually form the Principal Components of the transformed dataset.

In order to reduce the dimensionality of the dataset, PCA was used with different values of variance. For example, a variance of 95% would mean that only those features would be chosen that would capture 95% of the original variance in the data. Table 20 describes the performance of KNN for the different values of variance.

Table 20: Performance Comparison between all KNN models with PCA

	Without PCA	**With PCA**			
Variance	**Not Applicable**	**Variance 90%**	**Variance 95%**	**Variance 98%**	**Variance 99%**
Number of features used by the model	13	6	7	8	9
Precision	94%	85%	86%	93.39%	93.38%
Recall	92%	86%	87%	92.87%	92.80%
Accuracy	95%	92%	92%	96.19%	96.19%
F1-Score	93%	94%	86%	93.09%	93.05%

The observations in Table 20 show that there is merit in reducing the dimensionality of the model. With variance set at 98%, there was an improvement in recall and accuracy of the model. At the same time, there was a reduction in dimensionality by about 40%. It is a general notion that PCA would be effective only on datasets with a lot of features. However, the observations in Table 20 prove that even in datasets with fewer features, the performance of KNN can witness an improvement with a noticeable reduction in the dimension.

Support vector machine classifier performance analysis

The tuned SVM model appears to be a promising candidate for the farm produce dataset. Table 12 shows that there was a marked improvement in the performance of the tuned SVM model when compared with the default SVM model. The main difference between the two models was the hyperparameter C.

Hyperparameter "C" is used to determine the tradeoff between the smoothness of the hyperplane versus the accuracy of classification ("Support Vector Machines", n.d.). The lower value of "C" will generate a hyperplane with a wider margin between the support vectors. But the model will generate lower classification accuracy. Higher values for "C" will result in a narrower margin between the support vectors, but the model will achieve higher classification accuracy. The results of the SVM model created for the dataset corroborate this fact. The default model had a lower value for C, resulting in lower accuracy. However, the tuned model has a higher value for C which resulted in higher accuracy. A scatter plot can explain the need for a higher value for C. Figure 9 shows the scatter plot drawn between two features of the dataset. On observing the features in a 2-dimensional plane, the Euclidean distance between the data points appears narrower. Hence a smaller margin between the support vectors (i.e., a large value for "C") seems justified.

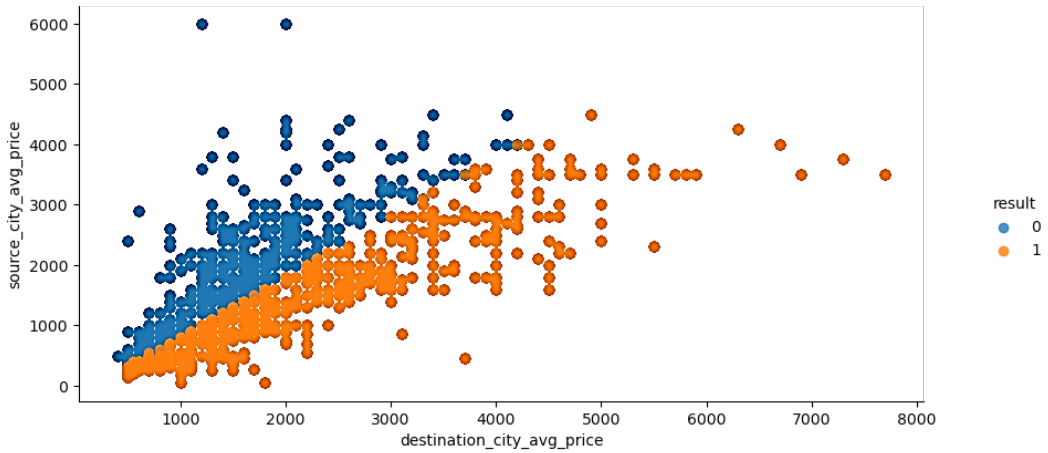

Figure 9: Scatter plot of two features with respect to the result

Random forest classifier performance analysis

Random forest is an ensemble learning method wherein multiple decision trees are used to generate the classification outcome. These decision trees learn via bagging and random subspace method. Bagging involves training the trees using a subset of observations from the dataset. This subset is also called bootstrap samples.

One of the aspects of a Random Forest model is to decide on the number of estimators, i.e., the number of decision trees used by the classifier. As noted in Table 13, the model with the default parameters uses 100 decision trees, whereas the tuned model uses 400 decision trees. And yet, both models exhibit similar performance. The reason for this observation can be understood using out-of-bag error.

Out-of-bag (OOB) error

OOB Error ("Out-of-bag error," 2021) is a method used to measure the prediction error of the random forest classifier. OOB Error shows the error in prediction introduced by not having a particular sample in the subset (i.e., the sample being out-of-bag). When the number of estimators in the forest is fewer in number, there is a chance that none of the trees within the forest may have seen a particular sample during the training phase. If such a sample gets introduced into the test set, the model may misclassify this sample during the testing phase.

Figure 10 shows the analysis of OOB error for the number of estimators for the agriculture dataset. The graph indicates that the OOB error drops and remains stable when the forest has thirty or more trees. Hence, in the case of random forest, the tuning of hyperparameters by introducing more decision trees does not yield any noticeable improvements to the performance metrics.

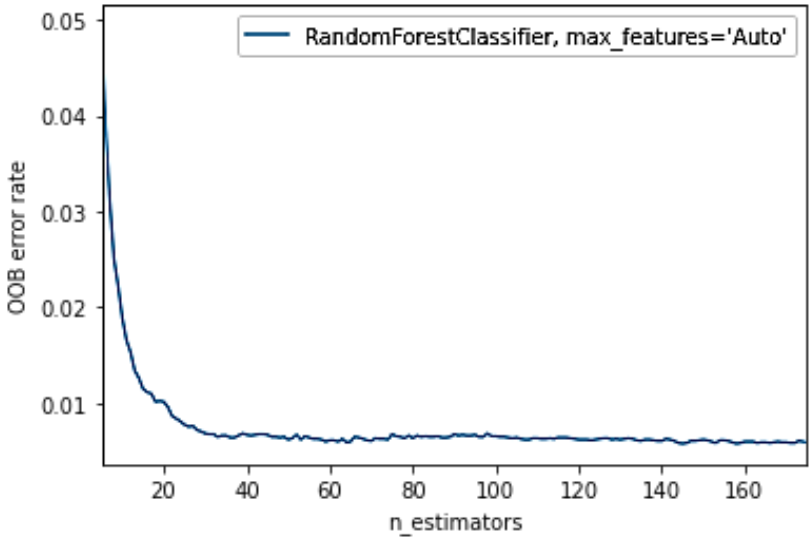

Figure 10: OOB Error v/s number of estimators

Gradient boosting classifier performance analysis

Gradient boosting classifier turned out to be the most optimal classifier among the five, with a performance measure of 99% for all the four-performance metrics. The improved performance stems from the fact that the model generated fewer false positives and false negatives.

The hyperparameter that has a material impact on the performance of a gradient boosting classifier model is the number of estimators (n_estimators). It determines the number of boosting stages applied by the model which is akin to the number of trees in the model. Learning_rate determines the impact of each stage or each tree on the outcome. In the case of the agricultural dataset, both n_estimators and learning_rate got tuned to a higher value. An AUC-ROC analysis below explains the reason why these parameters got tuned to a large value.

AUC-ROC

Receiver Operating Characteristic (ROC) Area Under Curve (AUC) is a performance measurement used for classification problems. ROC is a probability curve, and AUC determines the degree of separability. True Positive Rate (TPR) and False Positive Rate (FPR) are used to determine the AUC. A higher value of AUC indicates that the model can classify the samples accurately.

Figure 11 shows that when the learning rate was set to the default value, i.e., 0.1, AUC was observed to be 0.975. It improved to close to 1 when the learning rate changed to 0.7 during the tuning process. Similarly, when the number of estimators was set to the default value, AUC was 0.97. It improved to close to 1 when the number of estimators got increased to 400. Hence gradient boosting

classifier performs better at higher values of learning rates and with more estimators. One concern while setting a higher value for the learning rate is the problem of overfitting. But Figure 11 shows that both the training set and test set do not diverge even at higher values of learning rates. Hence there is no overfitting seen even at these higher values.

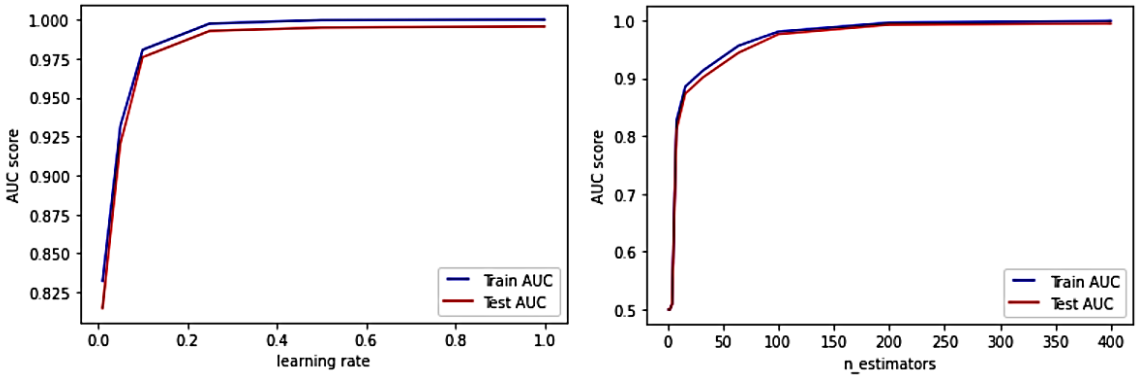

Figure 11: AUC-ROC for different learning rates and number of estimators

The default model and the tuned model differed for Subsample and minimum samples per leaf features. However, as shown in Figure 12, a change in the value of these parameters does not materially impact the AUC.

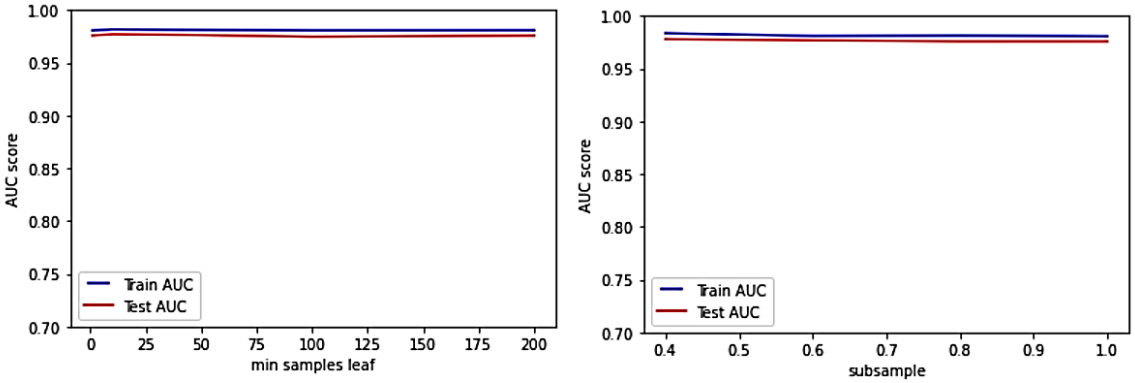

Figure 12: AUC-ROC for varying number of samples at leaf and the subsample

DEEP LEARNING MODELS

As a part of the current research, deep learning models based on Artificial Neural Networks have been developed to help farmers predict the appropriate markets for the farm produce. The dataset, sample selection procedure, and the feature set were reused from the traditional machine learning models as described under the section "Machine Learning Models".

NETWORK LAYERS AND NEURONS

The proposed Deep Neural Network (DNN) models consist of an input layer, two hidden layers, and an output layer. The input layer consists of 12 inputs. The two hidden layers contain 24 and 8 neurons. The output layer consists of a single neuron. The number of layers in a DNN and the number of neurons within each layer are determined based on iterative experiments. The network structure for the current research was determined heuristically.

Activation Functions

Each neuron generates an output based on the activation function. The intermediate layers were designed to use "ReLU" activation. The output neuron uses a sigmoid activation. ReLU or "Rectified Linear Unit" is a ramp function that is defined only for the positive part of the function as shown in Figure 13 and Equation 9.

$$f(x) = x^+ = \max(0, x) \tag{9}$$

Figure 13: ReLU activation function

Sigmoid or Logistic function is the "S" shaped curve that has the function as shown in Equation 10.

$$f(x) = \frac{L}{1 + e^{-k(x - x_0)}} \tag{10}$$

Where x_0 represents the midpoint value for the function and L represents the maximum value for the curve. The parameter "k" determines the growth rate or the steepness of the curve. The graph for the sigmoid function is shown in Figure 14.

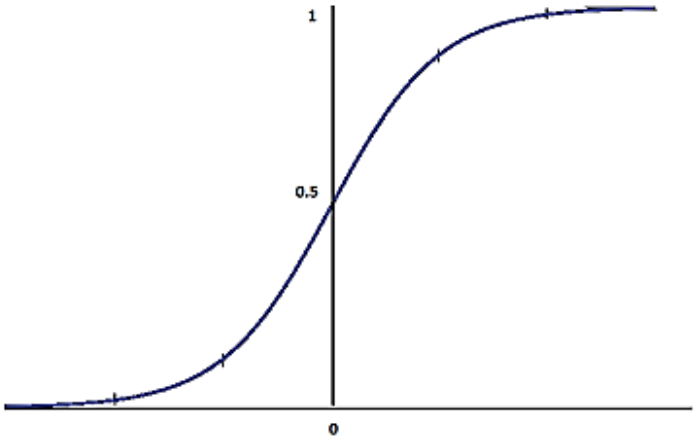

Figure 14: Sigmoid activation function

Loss Function

A loss function in a neural network determines the variation of estimated values when compared to the true values. Binary cross-entropy or the log loss function was used to measure the divergence of the estimated values from the true values. If y_i is the true value and $\hat{y}_i$ represents the estimated value then the binary cross-entropy is computed as shown in Equation 11.

$$J = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \tag{11}$$

OPTIMIZERS

An optimizer adjusts the weights and learning rates in a neural network. This operation minimizes the loss function. For this research work, Adam, Stochastic Gradient Descent (SGD), and AdaGrad optimizers were used to reduce the loss in the neural network.

Gradient descent is a way to minimize an objective function J(θ). Weight gradients are computed using the loss function, data, and weights. The weights are then adjusted using the computed gradients. Stochastic gradient descent is a form of gradient descent wherein, a subset of the training samples are used to compute the gradients as shown in Equation 12.

$$\theta = \theta - \eta . \nabla_{\theta} J(\theta; x^{(i)}; y^{(i)}) \quad (12)$$

Where, η, $x^{(i)}$ and $y^{(i)}$ represent the learning rate, i^{th} training sample and i^{th} label respectively.

Adam (Kingma & Ba, 2014) is a form of stochastic gradient descent technique that improvises on top of AdaGrad and RMSProp Optimizers. Adam is one of the most widely used optimizers for ANN-based models.

RESULTS

Three ANN models were developed using three different optimizers namely SGD, AdaGrad, and Adam. The network parameters for the three models are shown in Table 21.

Table 21: Parameters for the ANN model

Parameter	**AdaGrad based model**	**Stochastic Gradient Descent (SGD) based Model**	**Adam based Model**
Number of hidden layers	2	2	2
Number of neurons in hidden layer 1	24	24	24
Number of neurons in hidden layer 2	8	8	8
Activation function in hidden layers	ReLU	ReLU	ReLU
Activation function in the output layer	Sigmoid	Sigmoid	Sigmoid.
Loss function	Binary CrossEntropy	Binary CrossEntropy	Binary CrossEntropy
Optimizer	Adagrad	SGD	Adam
Epochs	100	100	100

AdaGrad based DNN model

The confusion matrix for AdaGrad based model is shown in Table 22.

Table 22: Confusion matrix for AdaGrad based DNN Model

	Predicted Negative	**Predicted Positive**
Actual Negative	2936	54
Actual Positive	670	594

The performance metric for AdaGrad is shown in Table 23.

Table 23: Performance metric for AdaGrad

Using Default Parameter Values		
• Precision	:	92 %
• Recall	:	47 %
• Accuracy	:	83 %
• F1 Score	:	62 %

From the performance metric, it is clear that the "recall" in the case of the Adagrad optimizer is suboptimal. 53% of the positive samples were predicted as negative samples (false negatives).

Stochastic gradient descent based DNN model

The confusion matrix for SGD based model is shown in Table 24.

Table 24: Confusion matrix for SGD based DNN Model

	Predicted Negative	**Predicted Positive**
Actual Negative	2904	86
Actual Positive	63	1201

The performance metric for SGD is shown in Table 25.

Table 25: Performance metric for SGD

Using Default Parameter Values		
• Precision	:	93 %
• Recall	:	95 %
• Accuracy	:	97 %
• F1 Score	:	94 %

Adam based DNN model

The confusion matrix for Adam based model is shown in Table 26.

Table 26: Confusion matrix for Adam based DNN Model

	Predicted Negative	**Predicted Positive**
Actual Negative	2968	22
Actual Positive	7	1257

The performance metric for Adam is shown in Table 27.

From the metric for the three deep learning models, it is clear that Adam based DNN model has a better performance compared to the other two models.

Table 27: Performance metric for Adam

Using Default Parameter Values		
• Precision	:	99 %
• Recall	:	99 %
• Accuracy	:	99 %
• F1 Score	:	99 %

Compared to the ARIMA based model for Onion price prediction (Darekar et al., 2016) that has an accuracy of 94%, and compared to the ARIMA model for coriander price prediction (Verma et al., 2016) that has an accuracy of 90% and compared even with the ARIMA model for Bengal-gram price prediction (Divya et al., 2017) that had an accuracy of 91%, the proposed gradient boosting model has better accuracy. When the proposed DNN model is compared with ANN models like the Backpropagation and Genetic algorithm-based ANN models (Subhasree & Priya, 2016), that was able to achieve an accuracy of 79% and 89% respectively, the proposed Adam based deep neural network model was able to perform better with an accuracy of 99%. The machine learning model that was developed as a part of the study commissioned by the government of Madhya Pradesh (AIGGPA, 2020) was able to achieve an accuracy of 95% for a Random Forest-based classifier, whereas the proposed gradient-based classification model achieves a higher accuracy of 99%.

Hence the proposed models in this paper are better than existing models. They are more effective in helping the farmers to obtain a better price for their crops.

CONCLUSION

This paper addresses two research questions on the impact of Covid-19 on Agriculture in India. Firstly, the research evaluates whether there was an impact on the pricing of agricultural produce due to Covid-19. Subsequently, the paper addresses the second question on whether a model can be developed that helps the farmers sell their produce in the correct market. This paper analyzed the auction prices for agricultural commodities (perishables in particular) in the local markets in India for the first half of the calendar year 2020. This was the period when the country was going through the first wave of the Covid-19 pandemic. The paper also proposes data models to help the farmer sell their produce at the right price in the appropriate market.

The statistical analysis of the data for the agricultural produce pricing indicates that certain farm produce (like French Beans) saw an increase in price in many parts of India during the lockdown. Supply constraints would have contributed to the rise in prices. Some vegetables (like Ladies' Finger) experienced a drop in price during the lockdown due to a lack of demand for the supply coming into the market. Some vegetables (like Cabbage) did not see a major change in prices during the lockdown compared to the prices before the lockdown.

Five machine learning models, Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Random Forest, and Gradient Boosting were developed to assist the farmers in selling their produce at an appropriate price in the correct market. These five models displayed satisfactory performance across different performance metrics like precision, recall, accuracy, and F1-score. Among the five models, Logistic Regression has the worst performance for the performance metrics with precision, recall, accuracy, and F1-Scores of 78%, 70%, 85%, and 74% respectively. Gradient boosting classifier exhibited the most promising results with precision, recall, accuracy, and F1-Scores of 99%. Using the gradient boosting model, farmers can now choose the most suitable market and earn a better price for their produce. India has a strongly regulated market with a homogeneous population; hence the modeling of prices is easier for a country like India.

Three deep neural network models were developed that were based on three different optimizers, namely, Adagrad, SGD, and Adam. Among the three models, Adagrad based model had a very poor recall rate of 47% that impacted the overall performance of the model. Adam based DNN model showed promising results with precision, recall, accuracy, and F1-Scores of 99%. Hence Adam based deep neural network model can be used to predict the correct market for the farmers.

Future research work can be carried out by expanding the scope of the classification models by adding more cities and towns and tuning the hyper-parameters appropriately to come up with a model that can be applied to all markets in India. Future research work shall also explore parameter optimizations to improve the performance metric of Adagrad based DNN model.

REFERENCES

Agrawal, R. (2020). Identification of minimum support price using linear and logistic regression. *International Journal of Advanced Science and Technology*, *29*(5), 2429-2435. http://sersc.org/journals/index.php/IJAST/article/view/11129

Agricultural Produce Market Committee. (2021, May 8). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Agricultural_produce_market_committee&oldid=1022075423

Atal Bihari Vajpayee Institute of Good Governance and Policy Analysis (AIGGPA). (2020, March). *Crop price prediction using machine learning in Madhya Pradesh: A pilot study*. http://www.aiggpa.mp.gov.in/uploads/project/Crop_price_predictions_using_machine_learning_in_MP-_A_pilot_study.pdf

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992, July). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop of Computational Learning Theory* (pp. 144–152). Pittsburgh Pennsylvania: ACM. https://doi.org/10.1145/130385.130401

Box, G., & Jenkins, G. (1976). *Time series analysis – Forecasting & control*. John Wiley & Sons

Breiman, L. (1999, October). Random forests. *Machine Learning*, *45*(1), 5–32. http://doi.org/10.1023/A:1010933404324

Cariappa, A. G. A., Acharya, K. K., Adhav, C. A., Sendhil, R., & Ramasundaram, P. (2021). Impact of COVID-19 on the Indian agricultural system: A 10-point strategy for post-pandemic recovery. *Outlook on Agriculture*, *50*(1), 26-33. https://doi.org/10.1177/00307270211989060

Chaudhari, D. J., & Tingre, A. S. (2014). Use of ARIMA modeling for forecasting green gram prices for Maharashtra. *Journal of Food Legumes*, *27*(2), 136-139. https://www.indianjournals.com/ijor.aspx?target=ijor:jfl&volume=27&issue=2&article=012

Choudhary, K., Jha, G. K., Das, P., & Chaturvedi, K. K. (2019). Forecasting potato price using ensemble artificial neural networks. *Indian Journal of Extension Education*, *55*(1), 73-77. https://krishi.icar.gov.in/jspui/bitstream/123456789/44873/1/Forecasting%20Potato%20Price%20using%20Ensemble%20Artificial%20Neural%20Networks.pdf

Co, C. H, & Boosarawongse, R. (2007). Forecasting Thailand's rice export: Statistical techniques vs. artificial neural networks. *Computers and Industrial Engineering*, *53*(4), 610-627. https://doi.org/10.1016/j.cie.2007.06.005

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297, https://doi.org/10.1007/BF00994018

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21-27. https://doi.org/10.1109/TIT.1967.1053964

Curse of Dimensionality. (2021, May 22). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Curse_of_dimensionality&oldid=1024543902

Darekar, A. S., Pokharkar, V. G., & Datarkar, S. B. (2016). Onion price forecasting in Kolhapur market of Western Maharashtra using ARIMA technique. *International Journal of Information Research and Review*, *3*(12), 3364-3368. https://www.ijirr.com/onion-price-forecasting-kolhapur-market-western-maharashtra-using-arima-technique

Darekar, A., & Reddy, A. A. (2017). Forecasting of common paddy prices in India. *Journal of Rice Research*, *10*(1), 71-75. https://doi.org/10.2139/ssrn.3064080

Das, K., & Mohanty, B. (2021, March 01). *The impact of COVID-19 on smallholder farmers in India and the way forward*. International Growth Centre (IGC). https://www.theigc.org/blog/the-impact-of-covid-19-on-smallholder-farmers-in-india-and-the-way-forward/

Deshmukh, S. S., & Paramasivam, R. (2016). Forecasting of milk production in India with ARIMA and VAR time series models. *Asian Journal of Dairy and Food Research*, *35*(1), 17-22. https://doi.org/10.18805/ajdfr.v35i1.9246

Divya, K., Rajeswari, S., Devi I. B., & Sumathi, P. (2017). Forecasting monthly prices of Bengal gram in selected markets of Andhra Pradesh. *International Journal of Research in Agricultural Sciences*, *4*(4), 213-218. https://ijras.com/administrator/components/com_jresearch/files/publications/IJRAS_595_FINAL.pdf

Fix, E., & Hodges, J. L., Jr. (1951, February 01). *Discriminatory analysis – Nonparametric discrimination: Consistency properties*. Technical Report. Defense Technical Information Center (DTIC). https://apps.dtic.mil/sti/pdfs/ADA800276.pdf

Freund, Y., & Schapire, R. E. (1996, July). Experiments with a new boosting algorithm. In *Proceedings of International Conference on Machine Learning* (pp. 148–156). https://cseweb.ucsd.edu/~yfreund/papers/boostingexperiments.pdf

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189-1232. https://doi.org/10.1214/aos/1013203451

Government of India. (1928). *Royal commission on agriculture in India: Report*. Bombay: Government of India Press. https://indianculture.gov.in/royal-commission-agriculture-india-report

Government of India (1955). *The Essential Commodities Act, 1955*. https://legislative.gov.in/sites/default/files/A1955.-10.pdf

Government of India. (2020a). *Variety wise daily market price*. https://data.gov.in/resources-from-web-service/3670701

Government of India. (2020b, September 27). *The Farmers' Produce Trade and Commerce (Promotion and Facilitation) Act, 2020*. New Delhi: Ministry of Law & Justice. https://egazette.nic.in/WriteReadData/2020/222039.pdf

Government of India. (2020c, September 27). *Farmers (Empowerment and Protection) Agreement on Price Assurance and Farm Services Act, 2020*. New Delhi: Ministry of Law & Justice. https://egazette.nic.in/WriteReadData/2020/222040.pdf

Government of India. (2020d). *Essential Commodities (Amendment) Act, 2020*. https://consumeraffairs.nic.in/sites/default/files/file-uploads/acts-and-rules/EC%28Amendment%29%20Act2020.pdf

Government of India. (2021). Agriculture & food management. In *Economic survey 2020-21, volume 2* (pp. 230-260). https://www.indiabudget.gov.in/economicsurvey/doc/vol2chapter/echap07_vol2.pdf

Jha, G. K., & Sinha, K. (2013). Agricultural price forecasting using neural network model: An innovative information delivery system. *Agricultural Economics Research Review*, *26*(2), 229-239. https://agecon-search.umn.edu/bitstream/162150/2/8-GK-Jha.pdf

Jha, G. K., & Sinha, K. (2014). Time-delay neural networks for time series prediction: An application to the monthly wholesale price of oilseeds in India. *Neural Computing and Applications*, *24*(3), 563-571. https://doi.org/10.1007/s00521-012-1264-z

Kearns, M. (1988, December). *Thoughts on hypothesis boosting*. Machine Learning Class Project. https://www.cis.upenn.edu/~mkearns/papers/boostnote.pdf

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations* (pp. 1-15). https://arxiv.org/abs/1412.6980v8

Kumar, A., Padhee, A. K., & Kumar, S. (2020). How Indian agriculture should change after COVID-19. *Food Security*, *12*(4), 837–840. https://doi.org/10.1007/s12571-020-01063-6

Logistic Regression. (2021, May 22). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Logistic_regression&oldid=1024548320

Maggo, D. (2020, June 02). *Impact of COVID-19 on small holder farmers – Insights from India*. World Business Council for Sustainable Development (WBCSD). https://www.wbcsd.org/Overview/News-Insights/WBCSD-insights/Impact-of-COVID-19-on-smallholder-farmers-in-India

Minitab Blog. (2013, July 01). *How to interpret regression analysis results: P-values and coefficients*. https://blog.minitab.com/en/adventures-in-statistics-2/how-to-interpret-regression-analysis-results-p-values-and-coefficients

Out-of-bag Error. (2021, May 15). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Out-of-bag_error&oldid=1023317101

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*, 2825– 2830. https://jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, *6*(3), 21-45. https://doi.org/10.1109/mcas.2006.1688199

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, *33*(1), 1–39. https://doi.org/10.1007/s10462-009-9124-7

Sain, V., Kumar, R., & Kundu, K. K. (2020). Price analysis and forecasting of basmati rice crop in Karnal district of Haryana. *Economic Affairs*, *65*(1), 107-115. https://doi.org/10.30954/0424-2513.1.2020.14

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, *5*(2), 197–227. https://doi.org/10.1023/A:1022648800760

Subhasree, M., & Priya, C. A. (2016). Forecasting vegetable price using time series data. *International Journal of Advanced Research*, *3*, 535-541. https://www.academia.edu/download/43376940/004.pdf

Support Vector Machines (n.d.). In *Scikit-Learn*. https://scikit-learn.org/stable/modules/svm.html

Tripathi, A. K. (2012). Agricultural price policy, output, and farm profitability – Examining linkages during post-reform period in India. *Asian Journal of Agriculture and Development*, *10*(1), 91-111. https://doi.org/10.22004/ag.econ.199109

Verma, V. K., Kumar, P., Singh, S. P., & Singh, H. (2016). Use of ARIMA modeling in forecasting coriander prices for Rajasthan. *International Journal Seed Spices*, *6*(2), 40-45. http://isss.ind.in/pdf/2016/v2/8.pdf

Wang, F. (2016). *Forecasting agricultural commodity prices through supervised learning*. http://cs229.stanford.edu/proj2016/report/Wang-Forecasting%20Agricultural%20Commodity%20Prices%20through%20Supervised%20Learning-Report.pdf

Zelingher, R., Makowski, D., & Brunelle, T. (2021). Assessing the sensitivity of global maize price to regional productions using statistical and machine learning methods. *Frontiers in Sustainable Food Systems*, *5*, 171. https://doi.org/10.3389/fsufs.2021.655206

AUTHOR

Dr. Niharika P. Kumar is currently an Associate Professor in the department of Information Science and Engineering, R V Institute of Technology and Management. She has completed her Ph.D. in the field of Broadband wireless communication from Visvesvaraya Technological University, Belagavi, India. She has 17 years of experience in Academics and Research & Development. Her research interest lies in the field of Computer and Wireless Networks, Data Science, Machine Learning and Blockchain. She has published numerous papers in International Journals, International Conferences, and published a book chapter in the domain of Wireless Communications and Networks.