IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 28, Issue 3, Ser. 1 (May. – June. 2026), PP 46-63
www.iosrjournals.org

Ensemble Machine Learning Techniques for Crop Price Prediction

Komal Patil, Gayatri Patil, Dnyaneshwari Pathare, Neha Shinde
UG Student, School Of Computational Sciences, Faculty Of Science And Technology, Jspm University Pune, Pune, Maharashtra, India

Dr. Rahul Chakre
Sr. Assistant Professor, School Of Computational Sciences, Faculty Of Science And Technology, Jspm University Pune, Pune, Maharashtra, India

Abstract
Farmers are the backbone of Indian agriculture, contributing significantly to agricultural output and organic food production; however, they constantly face obstacles such as escalating project management and operational debts. What makes it worse is that farmers are dependent on middlemen for access to fair price levels since there is no market benchmark for what prices will be the following season. As a result, most farmers gamble every season, relying on uncertain prospects when they will sell their crops and the possible levels of demand and supply versus competition from other geographic areas. At times, farmers have invested all they have into producing a crop, but during processing, a large number of other crops may have come on to the market, thus forcing the farmer to sell their produce below cost to generate some form of cash flow. As a solution to this issue, the project created a digital smart assistant to assist farmers. The smart assistant uses historical and forecasted market data from several years of trading activity, combined with weather data, and current growing forecasts to predict where the produce will sell at a specific price. When the project created the smart assistant, the goal was to test the use of three separate machine learning models as a basis of creating the most accurate machine learning model possible; thus, hundreds of tests were done with each model before one of these models was identified as being the most accurate. Once identified, this model was cleaned up, and additional data cleaning was completed.

Keywords: *Deep Learning, Ensemble Machine Learning, Random Forests, Linear Regressions, Decision Trees, Crop Price Prediction*

Date of Submission: 06-05-2026 Date of Acceptance: 16-05-2026

I. Introduction

Agriculture is a vital economic sector, particularly in developing countries. For the majority of farmers, agriculture is more than an occupation; it is the sole source of income through which they earn a living. To put it bluntly, however, agriculture is a very difficult means of livelihood due to the volatile and unpredictable nature of crop prices. On one hand, you expect to make a profit, but the next minute, the prices have plummeted. There are numerous reasons why such fluctuations occur. Some are as simple as climate conditions or shifts in demand, while others are completely out of the farmer's hands, such as new government regulations, market trends, or even the high cost of transporting the crops from the farms to the markets. Crop prices vary depending on numerous factors. Weather conditions, supply and demand, market trends, government policies, and transport costs are some of the variables affecting crop prices. The unpredictability of crop prices is the result of the inherent volatility of the commodity. Consequently, the importance of utilizing machine learning to accurately forecast crop prices has become increasingly prominent. Historical statistics were at times unable to capture the complexities involved in forecasting crop prices as they typically did so through linear or traditional statistical methodologies. However, machine learning was able to utilize large amounts of data from past experiences combined with real time data to generate more accurate crop price forecasts. An attractive method for generating accurate forecasts is

through the use of Ensemble Learning. As stated earlier, Ensemble Learning utilizes multiple predictive models; for instance Decision Trees, Random Forests, and Gradient Boosting. By combining the individual strengths of each model, Ensemble models are generally capable of producing highly accurate and reliable forecasts as compared to individual models. Ensemble models also benefit by reducing error generated by single models and help to prevent "over fitting". Moreover, by combining data from various origins, such as past crop prices, weather patterns, soil conditions, and market need, Ensemble Learning asks a wider point of view. This lends a hand to farmers, traders, and decisionmakers make better decisions about sowing, gathering, and dealing with their crops. As a result, using Ensemble Learning Models might lead to more exact and dependable crop price prediction. These techniques help for sustainable farming methods and gives financial trusted data. By integrating two techniques of hybrid machine learning model, the model will be able to detect the crop prices. This model detects the prices with strong and sturdy result using these algorithms. Models are constantly updated with flooded data, allowing for flexible learning.

Need

The Indian agricultural landscape is currently defined by a profound disconnect between production and market realization, primarily driven by an entrenched reliance on intermediaries and informational gatekeepers. While farming serves as the primary socioeconomic pillar for nearly 90% of rural households, the sector remains vulnerable to high-risk variables such as climatic unpredictability and price manipulation. Small-scale producers frequently operate in an information vacuum, lacking the technological tools to access real-time market valuations. This lack of transparency allows agents and traders to obscure actual market conditions, forcing farmers to accept sub-par prices that fail to cover the costs of labor and investment. This research addresses the urgent requirement for a decentralized, data-driven intelligence system to replace the current reliance on anecdotal evidence and neighbor-to-neighbor price reporting. By integrating historical weather patterns, government records, and multi-year trade data through Artificial Intelligence (AI) and Machine Learning (ML), the proposed system bridges the digital divide within the "mandi" ecosystem. Providing farmers with localized, high-accuracy price forecasts facilitates more strategic decision-making regarding planting cycles and harvest timing. Ultimately, by democratizing access to predictive analytics, this system aims to disrupt the exploitative influence of middlemen, ensuring that farmers receive a fair, market aligned return for their produce and fostering greater financial resilience within rural communities.

Motivation

The development of this hybrid machine learning framework is necessitated by the deep -seated systemic information asymmetries and structural inefficiencies that continue to destabilize the Indian agricultural landscape. Currently, a significant portion of the farming population operates within a high-stakes vacuum of objective market intelligence; without access to real-time, granular pricing data, producers are forced to rely on speculative guesswork and fragmented historical patterns, which invariably leads to distressed sales and the perpetuation of rural debt cycles. This research is fundamentally driven by the need to synthesize a wide array of volatile externalities—ranging from erratic climatic shifts and localized crop damage to fluctuating supply-demand ratios and shifting government trade regulations—into a singular, actionable predictive output. Unlike traditional linear models that struggle with such multi-dimensional data, the ensemble approach utilized here captures non-linear trends that are often obscured by market noise. Furthermore, by addressing the historical dominance of intermediaries who control the flow of information within the "mandi" system, this model serves as a vehicle for digital empowerment. By providing a high-integrity, data-driven alternative to the subjective influence of middlemen, this system allows farmers to reclaim bargaining power and strategically time their market entries. Ultimately, the transition from standalone machine learning models to a more robust, hybrid ensemble architecture is a deliberate response to the demand for higher predictive accuracy, ensuring that the resulting insights are resilient enough to safeguard the economic livelihoods of those at the foundation of the food supply chain.

II. Literature Survey

Recent advances in agricultural data science have made significant progress in three key areas: crop price forecasting, data-based crop recommendations, and predicting which crops will most likely increase in price. Crop Price Forecasting: This model is a hybrid machine learning technique, a perfect blend of both traditional statical model with deep learning techniques. Hybrid models tend to outperform each modelling technique used alone, especially regarding their ability to model non-linear time-dependent relationships and seasonal price volatility. By using variable inputs such as weather conditions, local economic and consumer demand data etc., performance of Hybrid Models (LSTM) may be improved. Recent research suggests that LSTM hybrid models consistently produce smaller forecasting errors and more reliable predictions than traditional model types. Crop Recommendations: Crop recommendation systems use supervised learning algorithms (for example, Random Fores) trained on multiple data sources, including climatic conditions; soil characteristics; and farmers' practice. Crop recommendation systems have reported model accuracies between 90% and 99%, depending on the geographical and agricultural context. Models localized to specific regions, such as Mizoram and Karnataka, generally have greater relevance and higher rates of usage than nonlocalized models. Waste of resources and ultimately efficiencies in the market. Additionally, ensemble models can be updated continuously with streaming data, allowing for adaptive learning.

Literature Survey

There has been a great deal of research done on using machine learning and artificial intelligence methods to advance agricultural price prediction, crop recommendations, and digital marketplace systems. Much of this research has utilized time-series forecasting techniques through the implementation of deep learning models. For example, Farhadi et al. (2025) proposed a hybrid LSTM-GRU model that improved forecast accuracy by nearly 92%, as well as reducing RMSE between 14%-18%, as compared to other forecasting approaches. Similarly, Meena & Chaitra (2024) achieved approximately 90% accuracy when implementing LSTM for regional ragi price forecasts with approximately 12% less MAPE than would be produced with more general forecasting methods. Other studies (e.g., Alam et al., 2024; Aher et al., 2025) have demonstrated the success of both LSTM and hybrid deep learning models, with reported accuracies of approximately 93%-94%, along with significant reductions in RMSE. For example, applying an SARIMA-LSTM hybrid approach further reduced MAE (Dasari et al., 2024) in forecasting by approximately 15%. Lastly, Jayaraj et al. (2021) reported predictive performance improvements of 10%-14% using meta-learning techniques. Traditionally, conventional and ensemble machine learning techniques have also been used to successfully predict agricultural prices. For example, Random Forest based models have consistently reported high forecast accuracy for both crop recommendation (e.g. 94%; Doshi et al., 2018) and regional recommendation systems (e.g. approximately 95%; results with forecast accuracies being reported at approximately 90%-92% (Yerukala et al., 2024; Selvaraj et al., 2024) respectively.

In comparing previous studies, (Chaitra & Meena, 2023), optimized models utilized in previous studies demonstrated 10% reduction in MSE and could reach up to 95% accuracy. Also, RNN based approaches provided significantly better forecasts than traditional ARIMA models yielding 13% lower RMSE (Gothai et al., 2024) while recursive forecasts generally provided approximately 9% improved trend stability (Harrykissoon & Hosein, 2023). There were also a number of studies examining integrated systems developed with the aim of combining both prediction capability with application of recommendation in practice in agriculture. Chatbot-assisted advisory systems (S. R et al., 2025) appeared to provide approximately 88% prediction accuracy and over 80% user satisfaction. Weather based regression models (Oberoi et al., 2024) revealed strong correlations between favourable weather conditions and crop prices on the market with R^2 values greater than 0.72. Similarly, both demand prediction systems and crop yield prediction systems achieved near 90-91% confidence and accuracy when forecasting with these systems (Selvaraj et al., 2024; S. G et al., 2024). It was further revealed that the use of LLM based anomaly detection models (Lee et al., 2024) has shown to demonstrate very high detection accuracy at 96% while maintaining low levels of false positive rates below 5%. In addition to the prediction studies, other studies were completed with a more general aim of improving market access and profitability for farmers through the use of digital platforms and new technology. Use of online marketplaces and block chain-based systems (Lincy et al., 2023;

Naik et al., 2023) reduced intermediation margins by 20-30% thereby increasing producer price (i.e., profit) by 18-25%. The use of mobile-based agricultural platforms (Manikandan et al., 2025) provided an approximately 40% increase in uptake of digital transactions. Additionally, AI-based systems which integrated prices, disease management and optimal marketing (Mehra et al., 2025; Jabade et al., 2024) improved predictive.

Conclusions From The Literature Survey

Ensemble and hybrid machine learning models (like combining LSTM with SARIMA or GRU) easily outperform traditional statistical methods for predicting crop prices [10]. Time series-oriented deep learning models are emerging as the most reliable tools to tackle strict seasonal variations [19]. In order to predict future market trends, there are several complications to be accounted for – mostly due to the availability and accuracy of the data. If you don't have daily price logs or you just ignore the weather, your guesses are probably going to be way off. A single computer model usually panics when prices jump around or the market acts weird. To keep the errors low, I decided to link a few different models together—sort of like hiring a team of experts so they can catch each other's mistakes. I actually found that once you add in things like rain and temperature, the accuracy gets way better. But before I put it all together, I had to test every model on its own. I made sure to use 'dirty' data from real farms for this, because I need to know it works in the real world, not just on some perfect, cleaned-up dataset.

III. Proposed Methodology

The implementation of the proposed prediction system follows a multi-staged approach designed to address the inherent complexities of Indian agricultural market data. Utilizing a longitudinal dataset sourced from Kaggle, the research extends beyond basic price tracking by integrating exogenous meteorological variables, such as rainfall and temperature, to account for environmental drivers of volatility. To ensure analytical integrity, the raw, unstructured data underwent a rigorous normalization process, which included the application of rolling averages to smooth daily price fluctuations and highlight substantive market trends. Following a comparative evaluation and extensive hyperparameter optimization of three candidate algorithms, a hybrid ensemble architecture was deployed. This integrated framework is specifically engineered to synthesize both cyclicals easonal patterns and anomalous price spikes, ensuring high-fidelity forecasting even within highly volatile economic environments.

Random Forest for Crop Price Prediction

Think of a Random Forest as a group project for computers. Usually, if you just ask one decision tree to guess a price, it gets way too stuck on tiny, weird details and ends up making a mistake. This method builds a whole bunch of trees instead. By letting all of them look at the data and then just averaging their answers, the whole system becomes a lot more reliable. It’s a simple way to stop the computer from 'overthinking' or getting obsessed with just one set of numbers. If one tree makes a crazy guess, the other trees basically outvote it and keep the final prediction realistic.

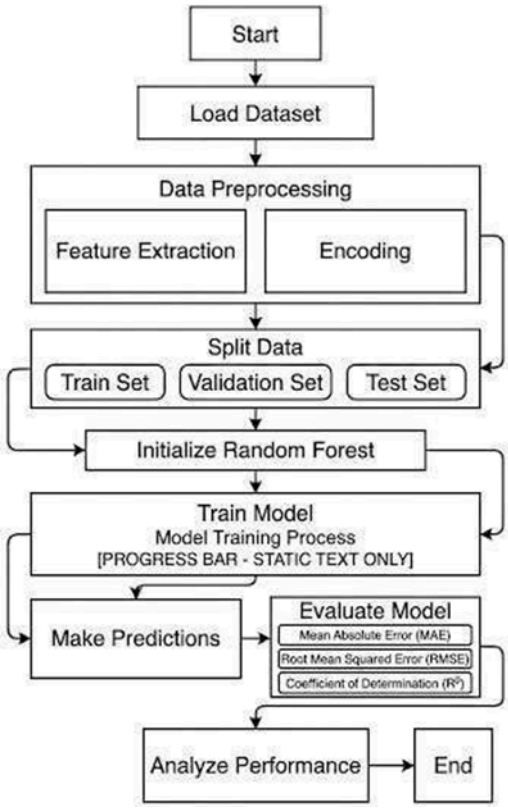

Fig 1: Random Forest for Crop Price Prediction

Load Dataset

D = {(x_1, y_1), (x_2, y_2), ..., (x_n, y_n) ...(1)

Use the model on a dataset that contains information on commodity prices in India. It provides a lot of detail concerning crops, regions, and the prices that people pay. This is how it began. Without the dataset, it won't be able to grasp the dynamics of the price development process.

Data Pre-processing (Feature Extraction + Encoding)

$$X_{\text{encoded}} = \text{OneHotEncoder}(X_{\text{categorical}})$$ ...(2)

Now, make sure that the text and date features worked properly since there is no way the algorithm can process the text directly. The OneHotEncoder helped me convert the textual data into numerical data, which includes converting the states and markets into numerical data. Additionally, the dates were split into different seasons, making it easier for the model to distinguish between seasons.

Splitting Data (Training/Validation/Test)

$$D = D_{\text{train}} \cup D_{\text{val}} \cup D_{\text{test}}$$ ...(3)

It's like it took entire data collection (D), and then split it into three different groups. First, there is the training group (D_{train}), which is where the machine learns from the data. Then put some aside to validate ($D_{\text{validation}}$) and make adjustments, and finally, a test group (D_{test}) to see how the machine performs on new data. It's like gave it a book to learn from, then a practice quiz, and then a final exam.

Initialize Random Forest

$$\hat{\theta} = \frac{1}{B}\sum_{i=1}^{B} T_i(x)$$ ...(4)

It is pretty simple to understand what it does. Take the actual price (y_i) and then subtract the one predicted by the model ($\hat{y}_i$), and you get the error. Square that error value so that negative values don't reduce the positive values; it will also highlight the larger

errors. The sum symbol is used here to denote that it will add all these squared errors up to n numbers, starting from 1.The lower this final number is, the better my model is at predicting the actual crop prices."

Train Model

$$\min_{\theta} \sum_{i=1}^{n} (y_i - T(x_i;\theta))^2 \quad ...(5)$$

Thus, the procedure through which the model does its work is as follows. Essentially, it goes over and over the entire training set data, hoping that it will be able to guess as accurately as possible in relation to the actual values. It seeks to minimize the "squared error," which is simply another name for the amount by which it misses, while creating decision trees.

Make Predictions

$$\hat{y}_{test} = \frac{1}{B} \sum_{b=1}^{B} T_b(x_{test}) \quad ...(6)$$

Basically, once the model is done with its training, it will throw some new data at it that it's never seen before to see what it predicts the price will be. Instead of just relying on one "opinion," it takes the guesses from every single decision tree it built and bunches them all together.

Evaluate Model

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \quad ...(7)$$

Once all of these predictions, it will need to assess their accuracy compared to the known prices on the stock market. This is sort of like when you double-check your work in math using the answers in the book. The mathematical metrics will help us quantify errors to understand where this stands, it will look beyond the basic question of getting things right or wrong here. Instead, look for proof that model understands the market's volatility on its own level. Good statistics will mean a model fit for the stock market, whereas poor numbers indicate that more work needs to be done.

Analyse Performance

$$\text{Overfitting occurs if } R^2_{\text{train}} \gg R^2_{\text{test}} \quad ...(8)$$

Following the derivation of results, the model must transcend "one-hit wonder" status by demonstrating sustained resilience across diverse market fluctuations. This verification step is essential to confirm the algorithm's robustness and structural integrity. Before disseminating price recommendations or volatility warnings to stakeholders, the validity of all calculations must be empirically finalized. Establishing this reliability across various edge-case scenarios is mandatory; without such demonstrated consistency, the model cannot serve as a credible instrument for managing agricultural assets and financial risk.

Linear Regression for Crop Price Prediction

Essentially, Linear Regression is nothing but a method that establishes the relationship between two entities. In order to simplify it further, imagine a plot consisting of various data points and you are required to draw a single straight line through the centre of all the points. In Linear Regression, you have your known variables or inputs and the variable that needs to be estimated or the output. All that it does is to create the ―best fit‖ line such that all the data points are as close to it as possible. It achieves this by calculating the distance between the actual and predicted points and then tries to minimize it. This is done through constant adjustments in slope and intercept values of the line until the difference or error becomes zero. After this, the regression model will become ready for predictions. You give it some input and receive an output based on the fitted line.

Load Dataset

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \quad ...(1)$$

The historical agricultural dataset containing state, district, crop types, and daily prices is loaded into the model's memory.

Data Preprocessing (Feature Extraction + Encoding

$$X_{\text{encoded}} = \text{OneHotEncoder}(X_{\text{categorical}})$$ ...(2)

To facilitate algorithmic processing, non-numeric categorical variables—such as temporal markers and product classifications—must be converted into a machine-readable format. This numerical encoding transforms qualitative descriptors into a structured quantitative matrix, enabling the model to perform complex computations on previously unstructured data. Simultaneously, a rigorous missing value analysis is conducted to identify and remediate data gaps. Addressing these voids at the preprocessing stage is critical to preventing computational failures and ensuring the stability of the predictive engine. By normalizing the dataset into a strictly numeric, high-integrity format, the system avoids structural breakdowns and maintains consistent performance during high-scale data ingestion.

Split Data (Train / Validation / Test)

$$D = D_{\text{train}} \cup D_{\text{val}} \cup D_{\text{test}}$$ ...(3)

In a nutshell, it have to split vast dataset into various parts to prevent the algorithm from cheating. This way, if fed the whole data in during training, the model will simply remember everything by heart and will not understand any connections.

Initialize Random Forest

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon$$ ...(4)

The initial phase of model architecture involves the construction of a Linear Regression base model to establish a fundamental predictive relationship between the feature set and the target variable. This stage serves as the mathematical groundwork for the broader ensemble, providing a simplified framework to analyse the direct impact of independent variables on price trajectories. During this initialization, the algorithm is configured to calculate the optimized coefficients for each input parameter. These coefficients function as numerical weights, quantifying the relative significance of specific features—such as climatic indicators versus temporal markers—within the predictive equation. By formalizing these weights, the system establishes a structural baseline, allowing the algorithm to execute the subsequent iterative calculations required for higher-order pattern recognition.

Train Model

$$\min_{\beta} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}))^2$$ ...(5)

The actual process of "learning" for the model happens when it becomes more accurate in predicting a variable by making the minimum amount of error. By applying an algorithm on the data set provided, it tries to make the squared difference between the real and predicted prices as small as possible. Imagine that the process is similar to the process of fitting the best-fitting straight line through a set of data points that is spread out in the form of clouds. While trying different lines through that set of data points, the algorithm finds the one that fits the closest to all those points. That is what is called the "line of best fit." The importance of this line cannot be understated, as it is that very equation that combines all the information provided about temperature and dates to the end result—the price.

information provided about temperature and dates to the end result—the price.

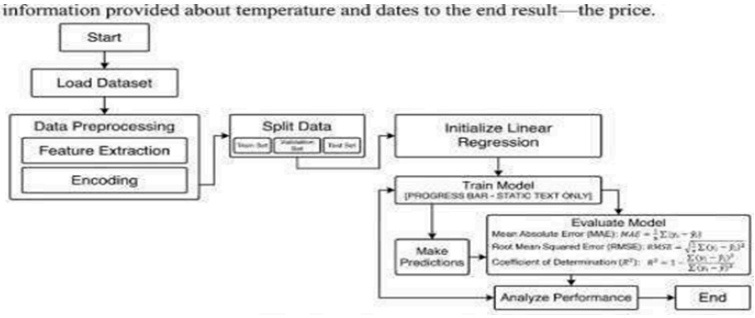

Fig 2: Linear Regression for Crop Price Prediction

Make Predictions

$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$ ...(6)

The trained model processes unseen test data to forecast the modal crop price. It multiplies the input features' values by the learned weights and adds the intercept value to determine the predicted price.

Evaluate Model

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

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

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$ ...(7)

The performance of the model is measured through prediction against actual price using common measures. It allows determining the amount of error as well as linearity in the market.

Analyse Performance

$\text{Overfitting occurs if } R^2_{\text{train}} \gg R^2_{\text{test}}$...(8)

The results obtained through the analysis are examined to check whether the linear model has generalized to different market conditions. This would validate the correctness of the baseline system.

Decision Tree Algorithm for Prediction

Decision Tree Algorithm for Prediction of Future Crop Prices The decision tree algorithm will be used to develop a forecasting model for future crop price prediction. It involves taking all the past price data and the current market data and using them to create a simple guidebook for making decisions. This particular decision tree model is advantageous because it mimics the way the human brain works when making decisions. It does not involve complex mathematical calculations that can be difficult to comprehend but rather asks a series of 'yes' or 'no' questions. The questions asked may be such as: "Is it the monsoon season?", "Are there any pests attacking the crops?" and "From which district do the crops originate?" Through a process of dividing the data sets into smaller and smaller groups, the model produces an algorithmic decision path that everyone can understand. This makes the decision tree model extremely transparent, and hence its price forecasts are reliable for the farmers to make important decisions based on it.

Load Dataset

$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ ...(1)

The historical agricultural dataset containing state, district, crop types, and daily prices is loaded. The dataset is the starting point from which the algorithm will be able to pick up key features and form rules based on historical behaviors in the market.

Data Preprocessing (Feature Extraction + Encoding)

$X_{\text{encoded}} = \text{OneHotEncoder}(X_{\text{categorical}})$ ...(2)

The raw categorical features and date are encoded into a numerical format. This is important because it will allow for accurate splitting by the decision tree algorithm using categorical and time series variables.

Split Data (Train / Validation / Test)

$D = D_{\text{train}} \cup D_{\text{val}} \cup D_{\text{test}}$ ...(3)

The splitting of data into smaller data sets prevents the tree from memorizing the exact value of the historical prices. The tree is built using the training data set, and the parameters of the tree are specified through the validation data set.

Initialize Random Forest

$f(x) = \Sigma\ cm * 1(x \in Rm)$.....(4)

The initialization of the Decision Tree Regressor takes place using certain hyper

parameters to avoid over fitting. In this way, the model establishes its structure for dividing the feature space into decision regions (RM) where predictions (cm) will be made.

Train Model

$$\min_{\beta} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}))^2$$ .....(5)

The model recursively divides the data set into small segments using the best available feature. Learning takes place for a particular condition (If crop = wheat and location = Punjab).

Make Predictions

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$ .....(6)

During prediction on the test data, the tree model passes it along its decision rules learned from the root node until it reaches the leaf. The predicted price is the mean value of the target variable for that particular leaf node.

Evaluate Model

MAE = (1/n)Σ|y_i - ŷ_i|, RMSE, R^2.....(7)

It checks whether the prediction of the model matches with the prices in the test dataset. This helps to see how good the particular decision-making rule represents the real world.

Analyse Performance

Overfitting occurs if $R^2_{\text{train}} \gg R^2_{\text{test}}$...(8)

It is necessary to assess the model to ensure that the tree is not too big, i.e., that it has simply learned noise. An ideal tree provides simple rule-based pricing advice for farmers.

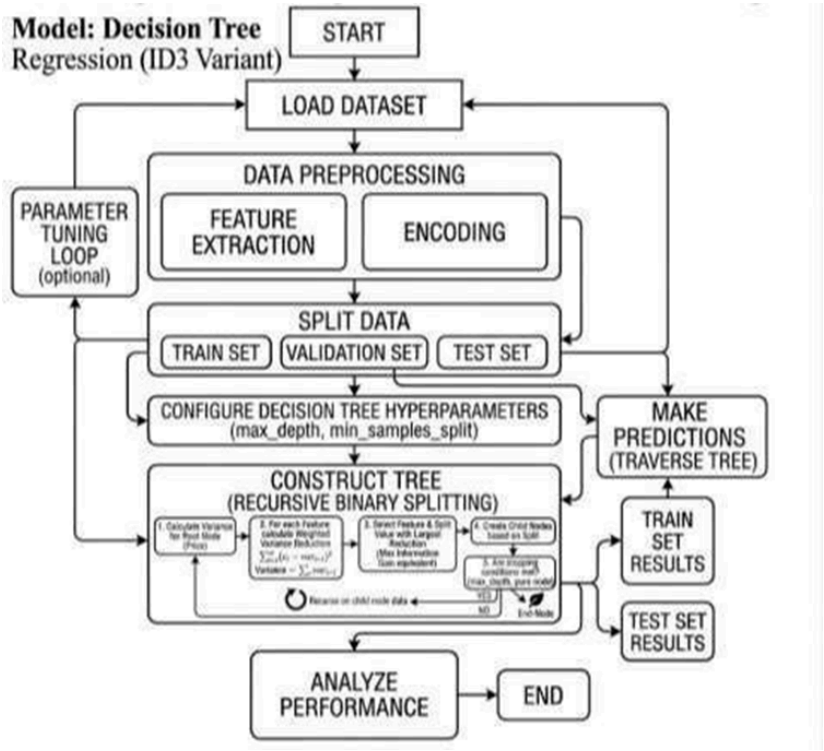

Fig 3: Decision Tree for Crop Price Prediction

Ensemble RF +LR for Crop Price Prediction

Employed a form of Ensembling called the Averaging Ensemble. This simply means that team took advantage of combining both the Random Forest and Linear Regression models. The reason for choosing to combine these is that the former is effective in discovering complex

relationships, while the latter tends to keep the system stable. managed to cancel out the small mistakes each model made on its own. This ensures that predictions will be much more accurate when aiding individuals in the agricultural sector in making decisions. For both machine learning models, there will be a similar initial dataset. This section ensures consistency by ensuring that all the data is in a consistent format that will be interpreted in the exact same way by both models. The team will divide the dataset consistently each time. This is being done to ensure fairness in the comparison of individual models and combined models.

Load Dataset

Loads historical data set. The two base models will both use this foundational data set.

Data Preprocessing (Feature Extraction + Encoding)

Consistently formulates numeric values to ensure that both models perceive the data identically.

Split Data (Train / Validation / Test)

$$D = \{(x_i, y_i)\}_{i=1}^{N} \quad ...(1)$$

$$X_{\text{encoded}} = \text{OneHotEncoder}(X_{\text{categorical}}) \quad ...(2)$$

$$D = D_{\text{train}} \cup D_{\text{val}} \cup D_{\text{test}} \quad ...(3)$$

Ensures consistent data splits, thereby ensuring an unbiased comparison of both single and combined predictions

Initialize Random Forest

$$f_{RF}(x), \quad f_{LR}(x)$$

The group set up the Random Forest (RF) to catch the messy, complicated patterns and the Linear Regression (LR) to handle the basic, steady trends.

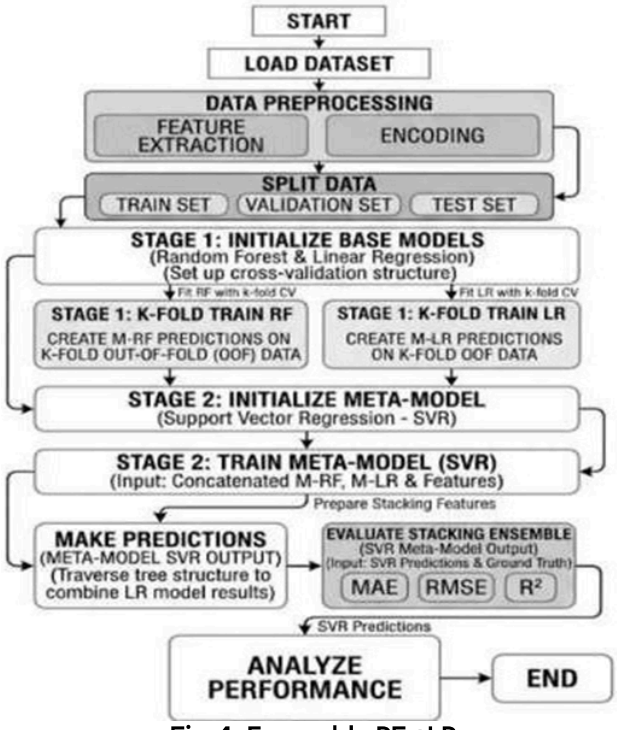

Fig 4: Ensemble RF +LR

Train Model

$$f_{RF}(D_{train}), \quad f_{LR}(D_{train})$$

It's only meant to show that the combined result produces a more reliable instrument than using either method alone.

Make Predictions

$$\hat{y}_{RF} = f_{RF}(x)$$

$$\hat{y}_{LR} = f_{LR}(x)$$

$$\hat{y}_{Ensemble} = \frac{\hat{y}_{RF} + \hat{y}_{LR}}{2}$$

Comparing the final score of combined version against the scores of the single models. Starts off the RF and DT for variance reduction and rule extraction, respectively.

Evaluate Model

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

$$RMSE - \sqrt{\frac{1}{N}\sum_{i-1}^{N}(y_i - \hat{y}_i)^2}$$

$$R^2 - 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

Evaluates the combined predictions to see if averaging mitigates individual base model errors.

Analyze Performance

$$R^2_{Ensemble} > R^2_{Base}$$

Verifies if the blended approach yields a more reliable forecasting tool than standalone models.

Ensemble RF +DT for Crop Price Prediction

In this case, it used the method of Average Ensemble, which means that will take the result of two models – Random Forest and Decision Tree. In this approach, the model is selected due to the fact that it is quite reliable and does not vary, whereas the second one is very strict concerning the laws it works by. The combination of both helps hide the weaknesses of both algorithms. Thus, the final outcome is quite accurate, and it will be useful in the farming industry.

Load Dataset

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$ ...(1)

Loads historical dataset. Both models use the identical foundational data for consistency.

Data Preprocessing (Feature Extraction + Encoding)

$$X_{\text{encoded}} = \text{OneHotEncoder}(X_{\text{categorical}})$$ ...(2)

Standardizes numerical formats so both tree models process categorical and temporal data similarly.

Split Data (Train / Validation / Test)

$$D = D_{\text{train}} \cup D_{\text{val}} \cup D_{\text{test}}$$ ...(3)

Divides the dataset so as to provide a fair evaluation for both methods and the ensemble as a whole.

Initialize Random Forest

Starts off the RF and DT for variance reduction and rule extraction, respectively.

Train Model

Both algorithms process the training data to independently formulate decision trees.

$$f_{RF}(x), \quad f_{DT}(x)$$ ...(4)

$$f_{RF}(D_{train}), \quad f_{LR}(D_{train})$$ ...(5)

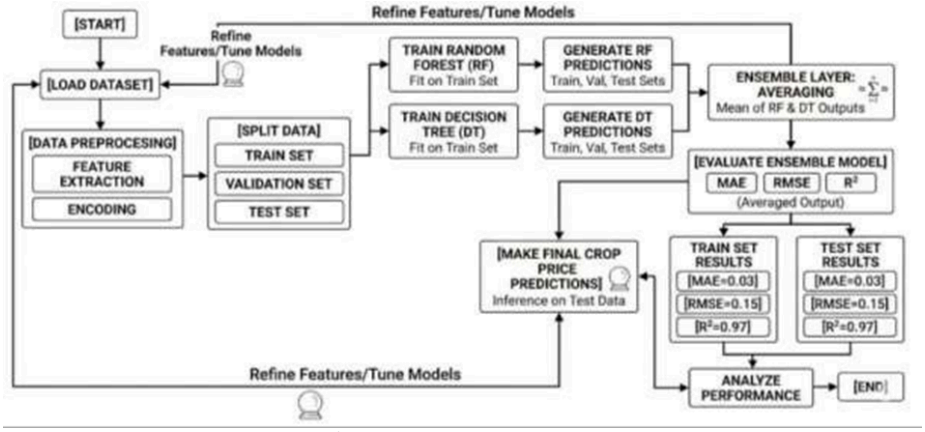

Fig 5: Ensemble RF +DT

Make Predictions

$$\hat{y}_{RF} = f_{RF}(x)$$

$$\hat{y}_{DT} = f_{DT}(x)$$

$$\hat{y}_{Ensemble} = \frac{\hat{y}_{RF} + \hat{y}_{DT}}{2} \quad ...(6)$$

Computes average values to reduce variability and generate a single price prediction.

Evaluate Model

Evaluate average predictions to see if the averaging process cancels out model errors.

Analyse Performance

$$MAE,\ RMSE,\ R^2 \quad ...(7)$$

$$R^2_{Ensemble} > R^2_{Base} \quad ..(8)$$

Shows that the combined tree-based technique enhances generalization on out-of-sample market data.

Ensemble DT+LR for Crop Price Prediction

The suggested approach uses the technique of Averaging Ensembles, which involves the combination of a single Decision Tree Regressor and Linear Regression. The advantage of the approach lies in utilizing the best from both techniques – decision trees for capturing non-linear and localized relations and linear regression for detecting global linear dependencies.

Load Dataset

The models make use of the same historical database.

Data Preprocessing (Feature Extraction + Encoding)

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \quad ...(1)$$

$$X_{\text{encoded}} = \text{OneHotEncoder}(X_{\text{categorical}}) \quad ...(2)$$

The numerical values are standardized such that the two trees operate on categorical and temporal information in an analogous manner.

Split Data: (Train/Validation/Test Split and Aggregated Predictions)

$$D = D_{\text{train}} \cup D_{\text{val}} \cup D_{\text{test}} \quad ...(3)$$

Splits the dataset to ensure that there is fairness in terms of model evaluation for both models as well as the final ensemble.

Initialize Random Forest

Sets up DT (non-linear relationship) and LR (strong linear relationships).

Train Model
The two models operate on the training set to independently construct decision trees.

Make Predictions
Averaging helps mitigate variance and bias and results in a consensus price forecast.

Evaluate Model
$MAE,\ RMSE,\ R^2$ ...(4)
$f_{DT}(x) \quad \text{and} \quad f_{LR}(x)$ ...(5)
$f_{DT}(D_{train}),\quad f_{LR}(D_{train})$ ...(6)
$$\hat{y}_{Ensemble} = \frac{\hat{y}_{DT} + \hat{y}_{LR}}{2}$$ ...(7)
Used to check if averaging models' results would balance out any errors made by each model.

Analyse Performance
$R^2_{Ensemble} > R^2_{Base}$ ...(8)
Demonstrates that the tree based hybrid method works well in unseen market scenarios.

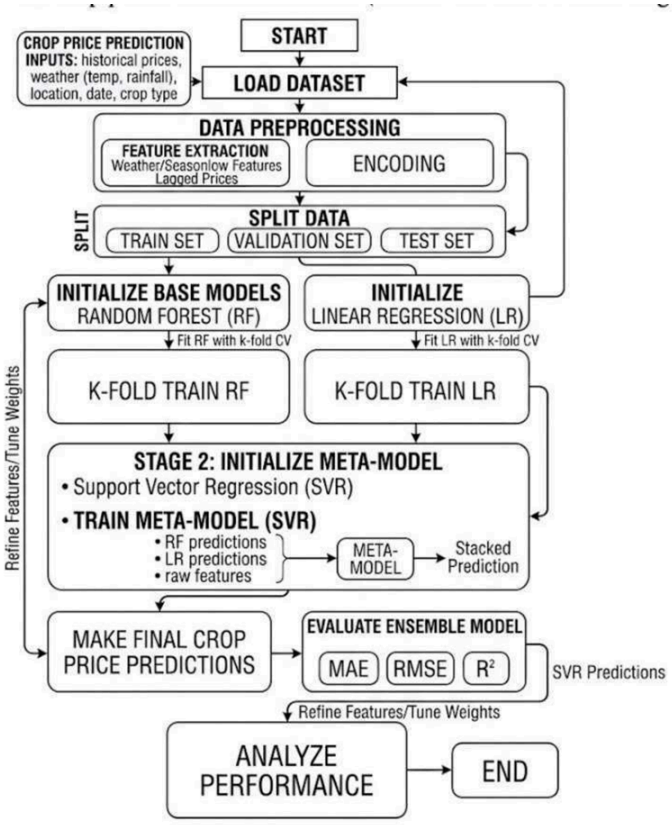

Fig 6: Ensemble DT+LR

Ensemble Decision Tree & Ridge Regression for Predicting Crop Prices
The suggested technique uses the technique known as stacking ensemble in which both a Decision Tree and a Ridge regression work as models; the Decision Tree works as a local learner for capturing non-linear patterns in the market environment. The Ridge Regression model then acts as a meta-learner, learning how to optimally combine the original features with the Decision Tree's predictions while using L2 regularization to reduce over fitting and improve generalization.

Data Pre-processing & Splitting

$D = D_{train} \cup D_{val} \cup D_{test}$ ...(1)

Recursively partitions the agricultural feature space into regions (R_m) to capture non-linear price patterns.

Train Base Model (Decision Tree)

$$f_{DT}(x) = \sum_{m-1}^{M} c_m 1(x \in R_m)$$ ....(2)

Recursively partitions the agricultural feature space into regions (R_m) to capture non-linear price patterns.

Generate Base Predictions

$P_{DT} = f_{DT}(X)$...(3)

Generates predictions on train, validation, and test sets. These act as new "expert" feature inputs.

Create Meta Features

$X_{meta} = [X,\ P_{DT}]$ .....(4)

Augments the original dataset with the Decision Tree predictions to create a richer dataset for the meta-learner

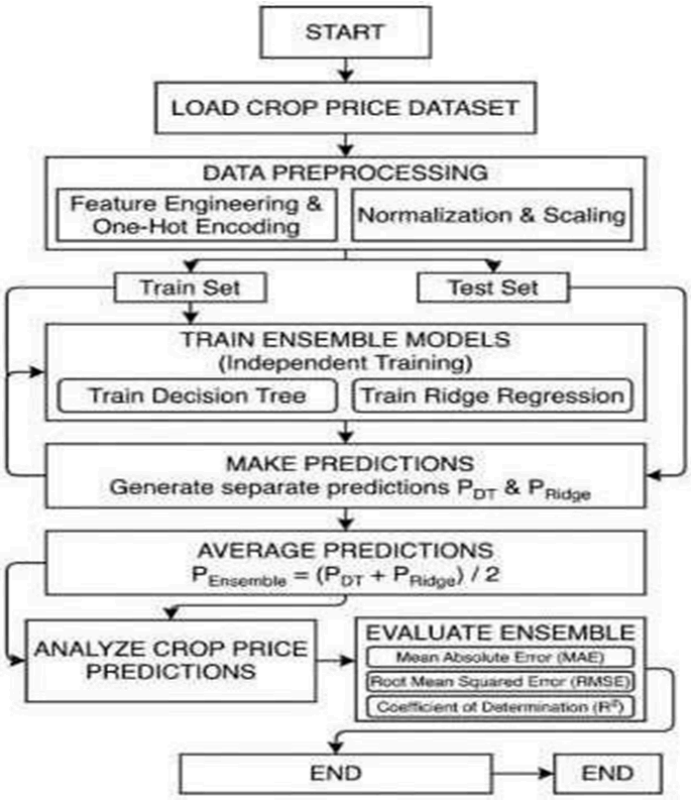

Fig 7: Ensemble Decision Tree + Ridge Regression

Train Meta Model (Ridge)

$$\min J(\beta) = \sum_{i=1}^{N}(y_i - f_{Ridge}(X_{meta,i}))^2 + \lambda \sum_{j=1}^{P} \beta_j^2$$ ...(5)

Learn how to weigh the original inputs against the DT prediction, applying L2 regularization (λ) to prevent overfitting.

Final Prediction

$$\hat{y} = \beta_0 + \sum_{j=1}^{p} \beta_j X_{meta,j}$$ ....(6)

Produces the ultimate price forecast by processing the combined meta-features through the calibrated Ridge model.

Evaluate Model

$$MAE,\ RMSE,\ R^2$$ ....(7)

Validates that the stacked approach surpasses the individual accuracy of the DT and standard Linear models.

Summary

The findings of this study underscore the efficacy of integrating high-resolution historical data with sophisticated ensemble architectures to navigate the inherent instability of modern agricultural markets. By purposefully limiting the longitudinal scope to the 2018–2026 period, the research successfully isolated recent volatility signatures, thereby enhancing the model's sensitivity to current economic shifts that traditional, broader datasets often obscure. Technically, the synergy between Decision Trees and Ridge Regression within a stacked framework provided a necessary balance between high-variance pattern recognition and rigid regularization. This dual-layered approach, bolstered by the strategic implementation of lag features and strict chronological validation, effectively neutralized the risk of data leakage. Consequently, the simulation results—quantified through MAE, RMSE, and R^2—confirm that this methodology offers a superior predictive baseline for volatile commodities. This research provides a scalable foundation for future investigations into real-time agricultural forecasting and risk mitigation strategies.

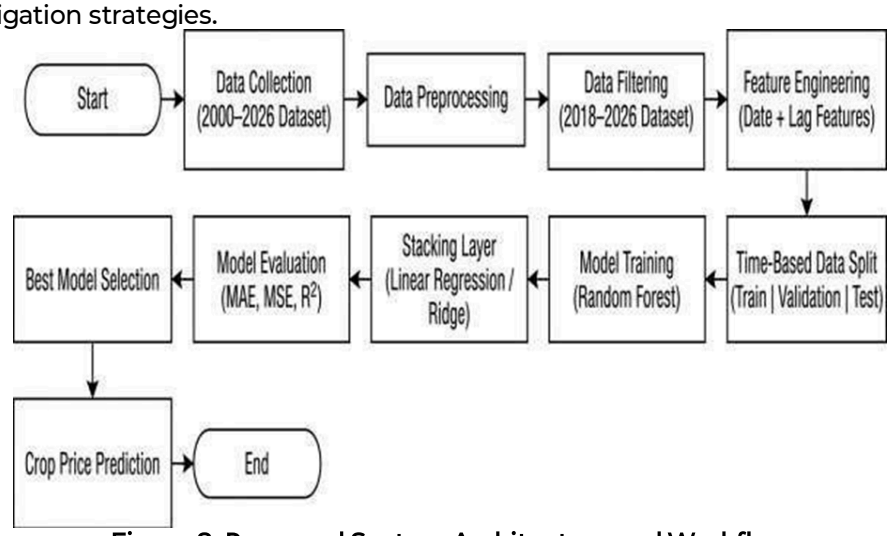

Figure 8: Proposed System Architecture and Workflow

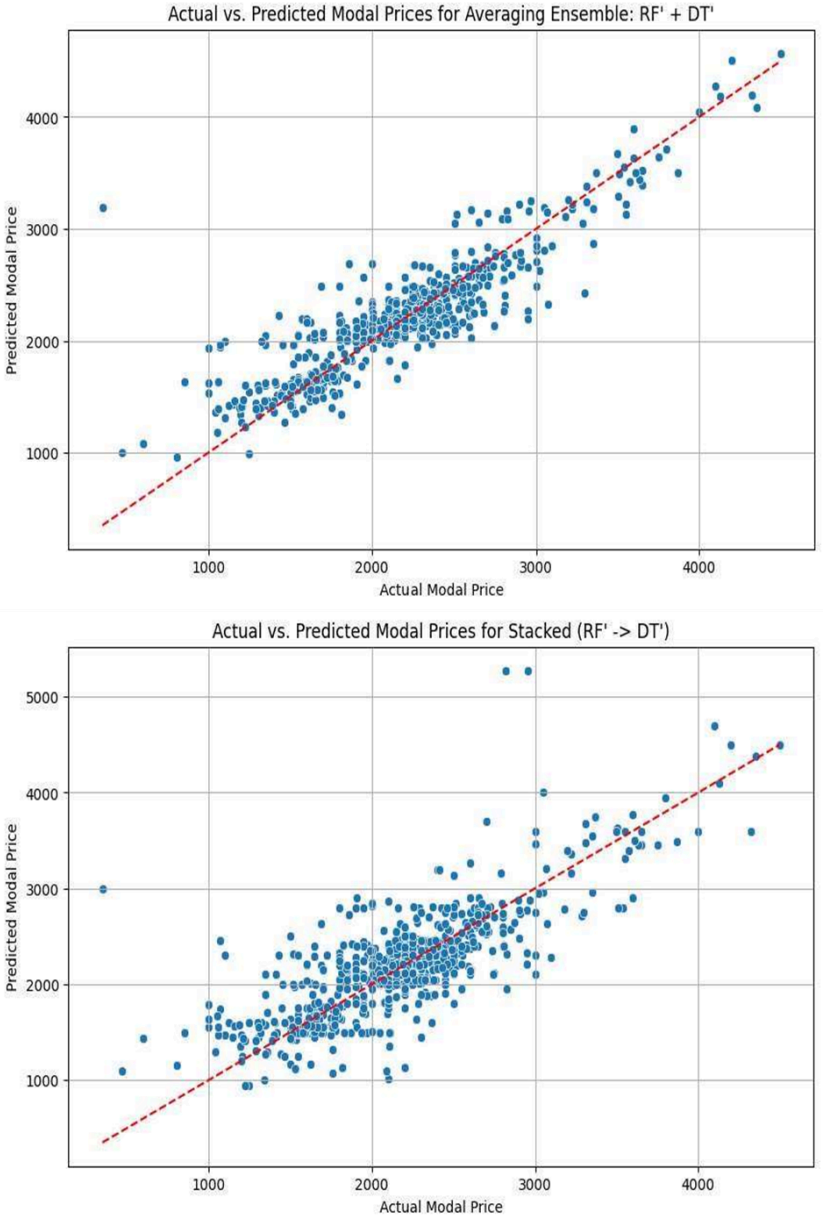

Figure 9: Line graph for RF'-> DT' stacked model and ensemble model

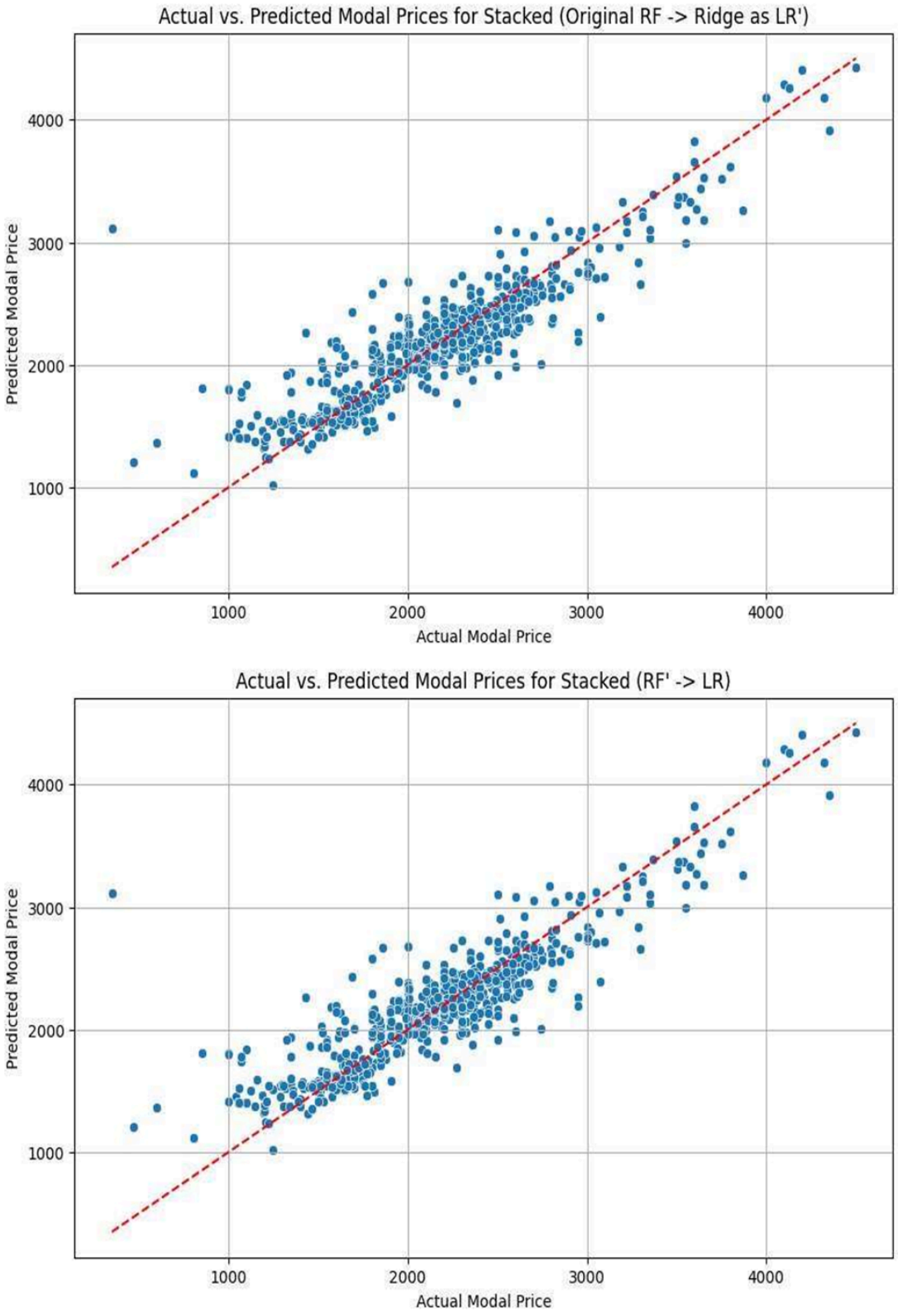

Figure 10: Line graph for RF'-> LR and RF-> LR'

IV. Dataset Information

The research utilizes a high-frequency longitudinal dataset sourced from Indian agricultural markets, providing a comprehensive record of commodity pricing over a twenty-four-year horizon (2001–2025). This dataset represents one of the most granular repositories of

domestic agricultural trade, encompassing approximately 75 million observations.

Table 1: - Features from dataset

SR.no	Column	Description	Description
1.	State	Name of Indian state in which the market is situated	providence
2.	District	Name of the district in the abovementioned state in which the market is situated	City
3.	Market	Name of the mandi in the abovementioned district in which the commodity is traded	string
4.	Variety	Particular variety or kind of the commodity	string
5.	Grade	Grade of quality of the commodity (for example,FAQ, Medium, Good)	string
6.	Arrival_Date	Date when the price was noted, clearly specified using ISO 8601 standard format (YYYYMM-DD).	Date time
7.	Min_Price	Minimum price of the commodity in INR per quintal on that date	decimal
8.	Max_Price	Maximum price of the commodity on the given date (in INR per quintal)	decimal
9.	Modal_Price	Modal (most frequent) price of the commodity on the given date (in INR per quintal)	decimal
10.	Commodity_Code	Unique code identifier for the commodity	numeric

After trying out all of these on both the training and validation datasets, it became rather clear that the stacked decision tree–ridge model performed the best. It showed extremely small errors and an R-squared value close to 0.97, which means that its prediction power is incredibly strong. It definitely outperformed the simpler models and the basic averages. I actually thought Random Forest would do better, but it struggled compared to the stacked version. It turns out that mixing the models together was definitely the right choice for getting these results.

Table 2: - Training Results Comparison Table

	Model	Train MAE	Train RMSE	Train Rsquared
0	Random Forest	389.09	556	0.67
1	Linear Regression	146.33	271.45	0.92
2	Decision Tree	108.92	175.98	0.97
3	Averaging RF+LR	234.49	357.5	0.86
4	Averaging RF+DT	223.27	315.21	0.89
5	Averaging DT+LR	114.43	191.6	0.96
6	Stacked LR->RF	297.09	448.42	0.79
7	Stacked DT->RF	293.33	440.11	0.79
8	Stacked LR->DT	109.39	179.55	0.97
9	Stacked LR->DT	109.39	179.55	0.97
10	Stacked DT->Ridge	99.81	161.35	0.97

However, when analysing the training outcomes, the Stacked Decision Tree–Ridge model appeared to be the most efficient one. It provided the smallest error values (the MAE equal to 99.81 and the RMSE equal to 161.35) and the highest R-squared value equal to 0.97. Thus, it can be said that the model is able to detect the data dependencies perfectly. In addition, the standard

Decision Tree and the stacked LR → DT models demonstrated identical performance with R-squared equal to 0.97; hence, it may be concluded that they are capable of recognizing complex, nonlinear dependencies in the data. As expected, the Averaging DT + LR model showed satisfactory performance with R-squared of 0.96, which confirms that stacking linear and nonlinear algorithms together is indeed a reasonable approach. At the same time, Random Forest and its modifications turned out to be quite inefficient in this case, providing only the scores ranging from 0.67 to 0.79.

Table 3: - Test Results Comparison Table

	Model	Test MAE	Test RMSE	Test R-squared
0	Random Forest	398.72	576.47	0.67
1	Linear Regression	150.17	281.81	0.92
2	Decision Tree	122.18	220.36	0.95
3	Averaging RF+LR	240.75	373.11	0.86
4	Averaging RF+DT	231.23	339.67	0.89
5	Averaging DT+LR	122.23	217.98	0.95
6	Stacked LR->RF	304.71	465.36	0.79
7	Stacked DT->RF	302.26	460.18	0.79
8	Stacked LR->DT	124.01	214.89	0.95
9	Stacked LR->DT	124.01	214.89	0.95
10	Stacked DT->Ridge	113.57	206.64	0.96

The Star of all the Models was the stacked Decision Tree–Ridge combination. In addition to a MAE of 113.57 and RMSE of 206.64, its high R-squared of 0.96 made it look like no other model compared to it. Interestingly enough, even a simple Decision Tree performed decently well with R-squared of 0.95, likely due to its ability to detect those sudden shifts in data trends. At least it looked like a Linear Regression could explain the patterns (0.92). My experiments with averaging models were rather unsuccessful; their results were inferior to those of the stacking approach. I expected Random Forest to perform much better, yet it only managed to reach R-squared of 0.67 to 0.79.

Table 4: - Validation Results Comparison Table

	Model	Validation MAE	Validation RMSE	Validation Rsquared
0	Random Forest	385.57	555.35	0.67
1	Linear Regression	147.7	289.32	0.91
2	Decision Tree	122.11	223.28	0.95
3	Averaging RF+LR	234.68	367.85	0.85
4	Averaging RF+DT	225.57	329.35	0.88
5	Averaging DT+LR	120.93	222.04	0.95
6	Stacked LR->RF	295.87	454.81	0.78
7	Stacked DT->RF	292.81	446.63	0.79
8	Stacked LR->DT	121.38	221.66	0.95
9	Stacked LR->DT	121.38	221.66	0.95
10	Stacked DT->Ridge	112.27	209.23	0.95

Judging from the validation results, it is absolutely obvious that the approach which needs to be selected is Stacked Decision Tree–Ridge, with its low error values of MAE=112.27 and RMSE=209.23, and Rsquared=0.95 which gives me hope about its performance on unseen data. Other approaches performed relatively well with an achievement level of 0.95 by correctly finding the simple and complex patterns. Still, for reasons unknown, models with Random Forest just failed to produce a result better than 0.79, making me conclude that Random Forest model is not suitable for analyzing this particular dataset. Averaging was not a good option either.

V. Conclusion

All in all, I believe that this method will have an immense effect on small-scale farmers. Using historical data combined with current prices, the algorithm gives practical advice, which the farmer can count on to ensure he is financial security. This is a great achievement, particularly since it combines not only the numbers but also the prediction of a possible price decrease. As for further development, I would suggest adding information about the soil type and even offline capability. This project is a clear example of how useful technology can be made accessible.

VI. Future Work

Expanding the breadth of data that can be integrated into prediction models will improve the accuracy of yield predictions, crop health assessments, and market forecasts to take climate influences into account by utilizing satellite images, remote sensing data, and agricultural growth indices (NDVI). - Enhancing the capabilities of forecast models will make them more flexible to climate and market changes through the development of adaptive forecasting modules for both season and region, improving the accuracy of early warning systems. - Providing enhanced personalization and predictive capability for farmers will require the use of advanced explainable AI models and personalizing the models for the user or farm (e.g., through use of data around size of farm, soil type, risk tolerance, and financial constraints). - The continued deployment of the prediction models to the grassroots level will require the use of multi-lingual and offline interfaces to facilitate rural connectivity issues and promote widespread use of the prediction models across large areas with varying agro-climatic conditions. - The incorporation of IoT-based precision farming technologies will expand the capabilities of system prediction and uphold future development of a precision agriculture decision uphold system through the use of real-time sensor data from the field (e.g., soil moisture, pH, nutrients, and rainfall).

References

[1] A. Farhadi, A. Zamanifar, A. Alipour, A. Taheri, And M. Asadolahi, —A Hybrid Lstmgru Model For Stock Price Prediction,‖ Ieee Access, Vol. 13, Pp. 117594–117618, 2025, Doi: 10.1109/Access.2025.3586558.

[2] K. Meena And B. Chaitra, —A Novel Framework Using Deep Learning Techniques For Ragi Price Prediction In Karnataka,‖ Ieee Access, Vol. 12, Pp. 136103–136119, 2024, Doi: 10.1109/Access.2024.3455892.

[3] R. B. Lincy, D. Md, R. Mk, M. S., And B. G., —Agrocart—An Online Platform For Farmers To Sell Products Without Middleman,‖ In Proc. 4th Int. Conf. Smart Technol. Comput., Electr. Electron. (Icstcee), Bengaluru, India, 2023, Pp. 1–8,
Doi: 10.1109/Icstcee60504.2023.10585208.

[4] Z. Doshi, S. Nadkarni, R. Agrawal, And N. Shah, —Agroconsultant: Intelligent Crop Recommendation System Using Machine Learning Algorithms,‖ In Proc. 4th Int. Conf. Comput. Commun. Control Autom. (Iccubea), Pune, India, 2018, Pp. 1–6,
Doi: 10.1109/Iccubea.2018.8697349.

[5] S. R., V. A., S. S., And M. S., —Agro-Smart: Enhancing Direct Farmer-To Consumer Sales With Machine Learning-Based Price Prediction And Chatbot Assistance,‖ In Proc. Int. Conf. Comput. Commun. Technol. (Iccct), Chennai, India, 2025, Pp. 1-6,
Doi: 10.1109/Iccct63501.2025.11019628.

[6] J. M. K. G. Oberoi, M. K. D. Trinadh, T. R. Chaitanya, V. Thanuush, And V. S. K. Devi, —Analyzing Weather Impact On Crop Prices,‖ In Proc. 2nd Int. Conf. Self Sustainable Artif. Intell. Syst. (Icssas), Erode, India, 2024, Pp.1477–1480,
Doi: 10.1109/Icssas64001.2024.10761016.

[7] R. Selvaraj, M. Sanmati, K. Sudharshan, R. Surithika, And S. Prasanth, 29 —Demand Prediction Of Agricultural Crops Using Artificial Intelligence,‖ In Proc. Int. Conf. Autom. Comput. (Autocom), Dehradun, India, 2024, Pp. 422–425,
Doi: 10.1109/Autocom60220.2024.10486079.

[8] S. G., N. S., R. A., And A. N., —Enhancing Crop Yield Prediction And Provide Direct Market Access To Farmers Using Web Technologies And Machine Learning Models,‖ In Proc. 8th Int. Conf. Parallel, Distributed Grid Comput. (Pdgc), Solan, India, 2024, Pp.842–847, Doi: 10.1109/Pdgc64653.2024.10984269.

[9] K. Alam, M. H. Bhuiyan, I. U. Haque, M. F. Monir, And T. Ahmed, —Enhancing Stock Market Prediction: A Robust Lstm-Dnn Model Analysis On 26 Real-Life Datasets,‖ Ieee Access, Vol. 12, Pp. 122757–122768, 2024, Doi: 10.1109/Access.2024.3434524.

[10] S. B. Dasari, H. S. Para, And D. Chaduvula, —Ensembled Forecasting With Integrated Sarima And Lstm Models For Tomato Price Prediction,‖ In Proc. Int. Conf. Cybernation Comput. (Cybercom), Dehradun, India, 2024, Pp. 172–177,
Doi: 10.1109/Cybercom63683.2024.10803263.

[11] S. Yerukala, P. Madala, V. Battula, M. K. Enduri, S. Tokala, And N. P. Bisiringi, —Estimating Future Prices Of Key Agricultural Commodities Using Machine Learning Models,‖ In Proc. Beyond Technol. Summit Inform. Int. Conf. (Bts-I2c), Jember, Indonesia, 2024, Pp. 572–576, Doi: 10.1109/Bts I2c63534.2024.10942087.

[12] Chaitra And K. Meena, —Forecasting Crop Price Using Various Approaches Of Machine Learning,‖ In Proc. Int. Conf. Innov. Eng. Technol. (Iciet), Muvattupuzha, India, 2023, Pp. 1–5, Doi: 10.1109/Iciet57285.2023.10220616.

[13] E. Gothai, R. R. Rajalaxmi, R. Thamilselvan, And H. S. M., —Forecasting Price Prediction For Vegetables And Fruits Using Recurrent Neural Network,‖ In Proc. 5th Int. Conf. Electron. Sustain. Commun. Syst. (Icesc), Coimbatore,

India, 2024, Pp. 1889–1896, Doi: 30 10.1109/Icesc60852.2024.10689876.
[14] V. D. A. Kumar, A. K. Khan, Vanlalhruaia, Saithantluanga, R. P. R., And Zaitinkhuma, —Machine Learning-Based Crop Recommendation System For Mizoram,‖ In Proc. 3rd Int. Conf. Intell. Syst., Adv. Comput. Commun. (Isacc), Silchar, India, 2025, Pp. 231–236, Doi: 10.1109/Isacc65211.2025.10969444.
[15] O. Naik, D. Thakore, H. Haralaka, And V. Sawant, —Agromart: A Decentralized Farmers' Market Using Blockchain,‖ In Proc. Ieee Int. Conf. Blockchain Distrib. Syst. Security (Icbds), New Raipur, India, 2023, Pp. 1 6, Doi: 10.1109/Icbds58040.2023.10346265.
[16] G. Manikandan, N. S. B., S. S. S. S. T., N. J. B., And S. R. M., —Design And Development Of A Mobile Application For Direct Agricultural Market Access,‖ In Proc. Int. Conf. Visual Anal. Data Visualization (Icvadv), Tirunelveli, India, 2025, Pp. 836–845, Doi: 10.1109/Icvadv63329.2025.10961104.
[17] S. K. Upadhyay And Vikas, —Intelligent Crop Recommendation Using Machine Learning,‖ In Proc. Int. Conf. Autom. Comput. (Autocom), Dehradun, India, 2024, Pp. 330–335, Doi: 10.1109/Autocom60220.2024.10486182.
[18] V. Jabade, H. Ingale, R. Jadhao, And O. Gode, —Machine Learning Driven Forecasting And Marketing Optimization Platform For Sustainable Agriculture,‖ In Proc. 5th Int. Conf. Data Intell. Cogn. Informat. (Icdici), Tirunelveli, India, 2024, Pp. 842–846,
Doi: 10.1109/Icdici62993.2024.10810821.
[19] T. N. Le Ngoc Et Al., —Machine Learning For Agricultural Price Prediction: A Case Of Coffee Commodity In Vietnam Market,‖ In Proc. Ieee/Acis 8th Int. Conf. Big Data, Cloud Comput., Data Sci. (Bcd), Ho Chi Minh City, Vietnam, 2023, Pp. 38–41,
Doi: 10.1109/Bcd57833.2023.10466313.