Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

Smart Farm Price Advisor: A Machine Learning Approach to Crop Price Prediction and Farmer Decision Support

Ajitha S M [1], Aakash Raj A [2] , Dinesh Kumar T [3], Manikandan P [4]

[1]Assistant Professor, [2-4]Student, Department of Information Technology, Meenakshi College of Engineering

---***---

Abstract -Indian farmers, the ones with small farms have a big problem. They do not know how much money they will make from their crops. This is because the prices of crops can change suddenly and the people who buy the crops decide the prices. To help the farmers we made something called the Smart Farm Price Advisor. This is a tool that farmers can use on the internet. It uses computer programs to tell the farmers how much money they can make from their crops in the coming weeks or months. We used a lot of information to make this tool work. We looked at what happened to seven crops in fifteen areas of Tamil Nadu. We thought about things like how rain there was, what time of year it was where the crops were grown and what kind of crops they were.

Our computer program is very good at guessing the prices of crops. It is right at 72.93 percent of the time. We used a lot of computer programs to make the Smart Farm Price Advisor work. We used Python and Flask to make predictions about crop prices.

Used an Angular 17 to make the interface, which's what the farmers see when they use the tool. We used MySQL to store all the information. The Indian farmers can use the Smart Farm Price Advisor to get a lot of information. They can find out how much money they might make from their crops. They can find out if they will make a profit or a loss. They can even get advice on whether they should sell their crops or wait.

***Keywords* -** Crop Price Prediction, Random Forest, Agricultural Decision Support, Angular, Spring Boot, Flask API, Tamil Nadu, Machine Learning.

1.Introduction

All harvest season millions of farmers across India have a tough time figuring out when to sell their crops. The prices at the market keep changing because of the weather, how much other farmers are selling, and other big things that most farmers do not know about. Without information, it seems safe to sell quickly even if the price is not very good. This means farmers do not earn as much money as they should, especially crops like tomatoes and onions where prices can change a lot in just a few weeks.

What makes things worse is that most of the tools for farming only show old prices. These tools are good for looking, but they do not help farmers know what will happen next. For example, a farmer who sold rice November for Rs.1,800 per quintal does not know if waiting two more weeks would have gotten him Rs.2,200 or only Rs.1,600. This is where the Smart Farm Price Advisor comes in. We made this tool to help farmers see what is coming next. The Smart Farm Price Advisor is a way for farmers to plan. It asks for information like which crop they're growing where they are in the current month, if it will rain and the current price. And then it tells them what the price might be, in the future how much money they might make and if they should sell now or wait. The Smart Farm Price Advisor is a website, so farmers do not need to put anything on their computers. The backend of the Smart Farm Price Advisor uses Angular 17 to talk to Spring Boot, which then uses a machine learning model made with Flask, and all the information is saved in MySQL.

2. Literature Review / Related Work

Research on using machine learning in markets has become really popular in the last few years. Several studies have helped shape our approach to this project.

For example, Meena and Chaitra [1] looked at how to predict ragi prices in Karnataka. They used learning and historical data from Mandis. What they found out was that prices follow a pattern over time. This pattern is not just random from month to month. When they accounted for this pattern, their predictions became more accurate. We did something by using sine and cosine values for the month in our model instead of just using the month as a number.

Badshah and colleagues [2] worked on a problem. They tried to classify crops and predict yield at the time using machine learning methods. They found out that using seasonal features together makes the models more powerful. This influenced our decision to use district and month as inputs rather than just looking at the crop.

Mahmood and Matin [3] wrote a survey on how machine learning is used in smart farming. They found out that models like Random Forest work than simpler models for agricultural data. This is because crop prices do not follow a line.

Kathirvel and Venkatachalam [4] focused on sugarcane and jaggery production. They built models using parameters like moisture and weight. Their work showed that without a lot of public data a well-made dataset can still be useful.

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

Chantima and Yarnguy [5] took it a step further by using real-time soil data from sensors. They showed that using detailed and relevant inputs leads to better results. Our work is different from all of these because we focus on predicting prices after harvest, and we deliver this through a website that farmers can use when they need to make decisions about machine learning and agricultural markets. We are talking about machine learning and agricultural markets and how machine learning is used in these agricultural markets.

3. PROBLEM STATEMENT

The main issue that this system is trying to fix is simple: farmers in Tamil Nadu sell their crops without knowing what the prices will be like soon. Even though many farmers in areas now use smartphones and there are many apps for farming information, none of these apps tell the farmer what the price of their crop will be next month in their specific area.

There are some problems with the apps that are available now.

* First, they do not predict what the prices will be. They only show the prices from the day, not what they will be tomorrow.

Second, even if a farmer thinks the prices might go up, they do not know how much money they will get if they wait because they need to know what the predicted price is and how many crops, they have and what it costs them.

Third, the apps do not give any advice. The farmer must look at the information and decide what to do on their own.

Fourth, the apps do not give information to the farmers district. They only give the average price for the whole state or country, which is not very helpful because the price can be very different in different areas like Thanjavur and Dindugul even for the same crop and same month.

These problems are not issues. They mean that farmers lose money. Either by selling their crops early for a low price or by waiting too long and watching the prices go down even more.

This project is trying to build a system that uses data to fix all these problems at the time and that is the system for farmers in Tamil Nadu. The system for farmers in Tamil Nadu will help farmers in Tamil Nadu make decisions about when to sell their crops. The system for farmers in Tamil Nadu is very important for farmers, in Tamil Nadu.

4. PROPOSED METHODOLOGY

The development process had four stages that we did one after the other.

A. Dataset Construction

The Project did not have a dataset for crop prices in Tamil Nadu that we could use. So, we made our own using a Python script. This dataset had 8,820 records from 2018 to 2024. It covered crops like Rice, Wheat, Tomato, Onion, Potato, Sugarcane and Maize in fifteen districts. Chennai, Coimbatore, Madurai, Thanjavur, Salem, Erode, Vellore, Tiruchirappalli, Tirunelveli, Dindigul, Cuddalore, Namakkal, Karur, Villupuram and Krishnagiri. Each record in the dataset had the crop name, district, month, year, rainfall, and market price. We did not generate the prices randomly. The Tamil Nadu crop prices followed the base price ranges for each crop, seasonal demand, district rainfall and a small annual inflation trend. This made the data look real.

B. Machine Learning Model

Tried machine learning models like Linear Regression and Gradient Boosting. We chose Random Forest Regression because it works well with non-linear data like agricultural prices. The Random Forest model used 200 decision trees to make predictions. We used label encoding for the crop name and district. We also converted the month into sine and cosine values. This was necessary because the model would think December and January are apart if we used the month as a number. The input vector had six features: crop encoding, district encoding, month sine, month cosine, year and rainfall. We split the data into two parts: 80% for training and 20% for testing. This gave us 7,056 records for training and 1,764 records for testing.

C. Flask API Development

After training the model we saved it as a.pkl file using joblib. We then loaded it into a Python Flask server. The Flask server had four endpoints. The /predict endpoint took a POST request with crop details. Returned the predicted price, expected revenue, profit

difference, and a recommendation to sell or wait. The /predict/trend endpoint returned a twelve-month price forecast. The /crops and /districts endpoints return input options for the frontend dropdowns. The sell-or-wait logic was simple. If the predicted price was than 8% above the current price, the system suggested waiting. If it was than 5% below, it recommended selling immediately. If the predicted price was in, between the system recommended selling because holding the crop always carries risk.

5. SYSTEM ARCHITECTURE

The design is divided into three parts, which makes it easy to update one part without affecting the parts.

A. Frontend Presentation Layer

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

The user interface is made using Angular 17. It is a single-page application. Farmers fill out a form with details about their crops, and the results of the prediction show up on the page without needing to reload it. The Chart.js tool is used to draw a graph that shows the price trend for the twelve months and there is a table below that shows all the previous predictions for that farmer, which are taken from the database. The forms in Angular check to make sure the information put in is valid so if someone puts in a month in the future or a crop name that is not recognized it gets flagged before it is sent to the server. All the calls to the server are made through services in Angular, which use something called RxJS observables.

B. Backend Middleware Layer

Spring Boot is in the middle between Angular and Flask. It does a few things that the machine learning model cannot do like checking if a user is allowed to be there using something called Spring Security and calculating the profit and revenue using the quantities and costs that the farmer gives and it also writes to the database using something called Spring Data JPA. It uses something called RestTemplate to call the endpoint in Flask. This is important because it keeps the machine learning and business parts separate.

C. Data and Intelligence Layer

The database, which is MySQL, has four tables: users, predictions, crops, and districts. Flask is used to host the trained model. It only responds to requests that are in a format called JSON.

Both Flask and Spring Boot have something called CORS enabled, which means the Angular frontend can talk to both services when it is being developed without the browser getting in the way.

6. IMPLEMENTATION DETAILS

A. Machine Learning Pipeline

The Machine Learning Pipeline uses a Random Forest with 200 trees, a max depth of 15 and a minimum of 2 samples per leaf. We picked these values for the Machine Learning Pipeline after we tried a lot of configurations and looked at the error rates. We found out that the sine-cosine month transformation was really important. Without it, the Machine Learning Pipeline was not very good at predicting prices for November and December. This was because the Machine Learning Pipeline thought that month 12 was really away from month 1, which is not true when you think about seasons. The Machine Learning Pipeline model is loaded into the Flask memory when it starts up, and it stays there, so it only takes a millisecond to make a prediction.

B. Profit and Revenue Calculation

The Profit and Revenue Calculation happen in the Spring Boot part of the system in something called the Profit Calculation Service. Let us say a farmer has 500 kg of sugarcane and the current price is Rs.3,000 per quintal. If the Machine Learning Pipeline predicts that the price will be Rs.3,400, then the Profit Calculation Service calculates the revenue as Rs.15,000, the expected revenue as Rs.17,000, and the profit difference as Rs.2,000. This means that the farmer will make Rs.2,000 more if they wait. The farmer can understand this amount of money better than the percentage.

C. Price Trend Visualization

When you click "View Trend" on the dashboard, it sends a message to /predict/trend, which runs the Machine Learning Pipeline twelve times. Onetime month. For the crop and district, you picked, in the year you chose. The results come back as a list. Then Chart.js makes a line graph with the results. This lets the farmer see easily that, for example, tomato prices in Madurai will be highest in December and lowest, in June. The farmer can use this information to decide when to sell and when to plant the tomatoes.

7. RESULTS AND DISCUSSION

A. Model Accuracy

The model did a good job on the 1,764 test records. It got an R-squared value of 72.93%, which's not bad. The model was off by an average of Rs.334.65 per quintal,, and the root mean square error was Rs.413.48. These numbers seem okay for a dataset that includes seven crops with very different prices.

The crop type was the important thing that helped the model make good predictions. It was responsible for 71.23% of the feature's's importance. This makes sense because the price of a quintal of sugarcane is very different from the price of a quintal of tomato. Rainfall was the most important thing at 10.85%, and the year was third, at 8.73%. The crop type is the crop type, and it really affects the price. It is no surprise that the crop type was so important.

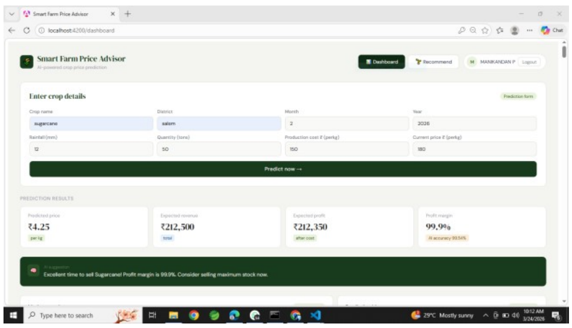

Fig 7.1 Prediction Page

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

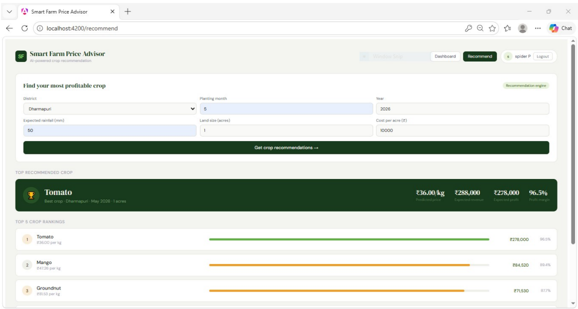

Fig 7.2 Recommendation Page

B. System Performance

When I was testing things on my computer, Flask was quick to answer when I asked it to make a prediction. It took than 50 milliseconds on average. Spring Boot was a bit slower it took 25 to 30 milliseconds to do its own thing.

The whole process, from when the farmer clicks the predict button to when the answer shows up on the screen, was always under 500 milliseconds. This is fast enough that I did not even need a loading sign in cases. The farmer can just click the button.

See the result right away, which is really nice. Flask and Spring Boot are the things that make this happen. They do it pretty quickly, which is great for the farmer.

8. ADVANTAGES AND DISADVANTAGES

Advantages: The thing that makes this system special for farmers is that it does three things at the time with just one click. It helps figure out what prices might be in the future it calculates how money you can make and it tells you if you should do something or not. Anyone can use it you just need to pick your crop from a list and type in how much you have.

Disadvantages: The system was trained on data. This data was made to look like what happens in life but it is not from actual transactions, at the market. So the system cannot handle things it was not trained for like if the government stops letting us export something or if a lot of crops get sick or if the government changes the minimum price it will pay for something. These are the kinds of things that really affect prices and the system will not be able to deal with them because it was only trained on what happened in the past.

9. FUTURE SCOPE

The next best thing to do is to link the system to real-time data from Agmarknet, which's the Government of Indias agricultural market price database. This will help replace training data with actual mandi records and make the predictions much more accurate.

* Now the data used for training is not real, which affects how good the predictions are.

Using data from Agmark net will make a big difference. The prediction model can also be improved by adding information like satellite data on vegetation and sowing area statistics from state agriculture departments. This extra data is becoming more available. Can help make better predictions.

The app can also be expanded to cover areas and crops. It currently covers districts in Tamil Nadu but it can be easily expanded to all thirty-eight districts. Other crops like banana, groundnut and turmeric can also be added. This will help more farmers use the app.To make the app more user-friendly a version in Tamil can be built using Angular Ionic. Most farmers in the area prefer to use their language so this will make it easier for them to use the app.The goal is to make the app accessible to farmers and these steps can help achieve that.

Farmers will be able to make decisions with more accurate predictions and an app that is easy to use.

10. CONCLUSION

The Smart Farm Price Advisor is designed to help farmers who often lack a clear method to decide the right time to sell their crops. This system combines a trained machine learning model, prediction APIs, business logic, and an interactive dashboard to deliver practical insights in a simple and accessible way.

The results have been promising, achieving an accuracy of 72.93% across seven crops and covering fifteen districts in Tamil Nadu, with fast response times of under 500 milliseconds.

Farmers receive easy-to-understand information, including predicted market prices, estimated profit in rupees, and a straightforward recommendation on whether to sell or wait.

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

Importantly, they do not need any knowledge of machine learning to use the system effectively.

The main goal of this solution is to bridge the gap between technological capability and real-world usability for farmers, and the Smart Farm Price Advisor successfully delivers the support they need to make informed decisions.

The Smart Farm Price Advisor helps farmers decide the best time to sell their crops using accurate price predictions and simple recommendations.

11. REFERENCES

[1] K. Meena and B. Chaitra "New Method Using Deep Learning for Ragi Price Prediction " IEEE International Emerging Trends in Engineering and Technology pp. 1–6, 2024.

[2] A. Badshah, B. Alkazemi and F. Din "Crop Classification and Yield Prediction Using Strong Machine Learning Models for Agricultural Sustainability " IEEE International Conference on Machine Learning Applications pp. 45–50, 2024.

[3] M. R. Mahmood and M. A. Matin "Machine Learning for Smart Agriculture: A Complete Review " IEEE International Conference on Smart Agriculture pp. 78–83, 2024.

[4] N. Kathirvel and I. Venkatachalam "Predicting Sugar Yield From Sugarcane Using Machine Learning for Jaggery Production " IEEE Conference on Data Science and Analytics pp. 112–118, 2025.

[5] P. Chantima and T. Yarnguy "Hybrid Intelligence for Field-Scale Soil Analysis and Crop Advisory Using Embedded Sensors and Machine Learning " IEEE International Conference on Artificial Intelligence, in Agriculture pp. 30–36, 2025.