Machine Learning model for predicting agricultural commodities prices

A Report

Submitted in partial fulfillment of the Requirements for the completion of

THEME BASED PROJECT

BACHELOR OF ENGINEERING
IN
INFORMATION TECHNOLOGY

By

AKHIL 1602-21-737-005
NAGASAI 1602-21-737-036
SREERAM 1602-21-737-054

Under the guidance of
Ms.C.SIREESHA
ASSISTANT PROFESSOR

Department of Information Technology

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE.

(Affiliated to Osmania University and Approved by AICTE)

Ibrahim Bagh, Hyderabad-31

2024

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University and Approved by AICTE)

Ibrahim Bagh, Hyderabad-31

Department of Information Technology

DECLARATION BY CANDIDATES

We, **AKHIL, NAGASAI, SREERAM,** bearing hall ticket number, 1602-21-737-005, 1602-21-737-036, 1602-21-737-054, hereby declare that the project report entitled **"Machine Learning model forpredicting agricultural commodities prices"** under the guidance of Mrs.C.Sireesha, Asst Professor, Department of Information Technology,Vasavi College of Engineering, Hyderabad, is submitted in partial fulfillment of the requirement for the completion of Theme-based project,VI emester, Bachelor of Engineering in Information Technology. This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other institutes.

AKHIL 1602-21-737-005
NAGASAI 1602-21-737-036
SREERAM 1602-21-737-054

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University and Approved by AICTE)

Ibrahim Bagh, Hyderabad-31

Department of Information Technology

BONAFIDE CERTIFICATE

This is to certify that the project entitled "**Machine Learning model for predicting agricultural commodities prices**" being submitted by AKHIL, NAGASAI, SREERAM, bearing hall ticket number 1602-21-737-005, 1602-21-737-036, 1602-21-737-054, in partial fulfillment of the requirements for the completion of Theme-based project of Bachelor of Engineering in Information Technology is a record of bonafide work carried out by them under my guidance.

Mrs.C.Sireesha Asst Professor Internal Guide	**External Examiner**	**Dr. K. Ram Mohan Rao** **Professor, HOD IT**

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the project would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

It is with immense pleasure that we would like to take the opportunity to express our humble gratitude to **MRS. C. SIREESHA, Asst Professor, Information Technology** under whom we executed this project. Her constant guidance and willingness to share their vast knowledge made us understand this project and its manifestations in great depth and helped us to complete the assigned tasks.

We are very much thankful to **Dr. K. Ram Mohan Rao, Professor and HOD, Information Technology,** for his kind support and for providing the necessary facilities to carry out the work.

We wish to convey our special thanks to Dr. S. V. Ramana**, Principal** of **Vasavi College of Engineering** for giving the required information in doing our project work. Not to forget, we thank all other faculty and non-teaching staff, and our friends who had directly or indirectly helped and supported me in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities. Finally, we wish to convey our gratitude to our family who fostered all the facilities that we need.

ABSTRACT

Predicting and understanding agricultural commodity pricesis inherently challenging due to the various influencingfactors and market volatility. The uncertainty in price predictions affects resource allocation decisions and can lead to inefficiencies in the agricultural supply chain.

The inaccurate price predictions have a direct impact on farmers, distributors, and consumers, affecting their livelihoods and economic activities. The machine learning model for predicting market prices of agricultural commodities will be developed using historical data and market trends. This model will utilize various algorithms such as regression and time series analysis to forecast future prices accurately.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Overview:

Developing a machine learning model to predict the market prices of agricultural commodities using historical data and Random Forest Regression involves a systematic approach with a specific technology stack. The process begins with defining the problem and specifying objectives, such as predicting the price of a particular commodity like wheat or corn over a given period. Data collection focuses on gathering historical price data, ensuring it is comprehensive and relevant to the prediction task.

The collected data undergoes preprocessing using *Pandas* to handle missing values, normalize scales, and engineer useful features. Exploratory Data Analysis (EDA) is conducted with *Matplotlib* to visualize patterns and relationships within the historical data. *Scikit-learn* is utilized to implement the Random Forest Regression algorithm, which is suitable for capturing complex interactions and non-linear relationships in the data. The model is trained on the prepared data, with hyperparameters fine-tuned through cross-validation to optimize performance.

Model evaluation is performed using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) to ensure accuracy and reliability. Upon achieving satisfactory performance, the model is deployed using *Flask, which creates a web application that allows users to interact with the model through a user-friendly interface developed with **HTML* and *CSS*.

Continuous monitoring is essential to maintain the model's accuracy, involving periodic retraining with new historical data, tracking performance, and updating the model as necessary. Regular reporting and insight generation provide stakeholders with valuable information, including trend analysis and actionable predictions. This technology stack ensures a robust and scalable implementation of the commodity price prediction model based solely on historical data using Random Forest Regression and Flask for deployment.

1.2 Problem Statement:

The project aims to develop a machine learning model predicting agricultural commodity prices. It utilizies Random Forest Regression for its ability to handle complex data interactions. Challenges include ensuring data quality and effective feature engineering. The model will be trained, evaluated, and deployed using Flask for user interaction. Continuous monitoring and maintenance are crucial for accuracy. The goal is to provide valuable insights for stakeholders in commodity trading.

1.3 Motivation

Reducing the risks associated with price fluctuations, enhancing stability and sustainability in the industry.

Decision Support for Farmers and Traders Accurate predictions of market prices can provide valuable insights to farmers and traders, helping them make informed decisions about crop selection, timing of planting and harvesting, and trading strategies. This can ultimately lead to improved profitability and reduced risk in agricultural operations.

Accurate price predictions can inform policy makers and government agencies about market trends and dynamics, enabling them to formulate effective policies and interventions to support agricultural markets and ensure food security.

2. LITERATURE SURVEY

The prediction of agricultural commodity prices has evolved significantly, transitioning from traditional statistical methods to advanced Machine Learning techniques. Each phase has contributed uniquely to the field, addressing different challenges and enhancing predictive accuracy.

Traditional Methods in Agricultural Price Prediction: Initially, time series analysis and regression models were the primary tools for forecasting. Box, Jenkins, and Reinsel's Time Series Analysis: Forecasting and Control (1970) introduced ARIMA models for analyzing historical price data to predict future trends. Gujarati and Porter's Basic Econometrics (2009) provided a comprehensive overview of regression techniques, essential for identifying predictors from historical data.

Expert analysis also played a critical role. Mankiw's Principles of Economics (2014) highlighted how experts leveraged domain knowledge to interpret trends and assess qualitative factors like geopolitical events and economic indicators. Goodwin and Schnepf (2000) examined how expert analysis evaluated price risks in commodity markets, emphasizing qualitative assessments.

Government reports and market intelligence were fundamental. The USDA and FAO regularly published detailed reports on crop production, market trends, and price forecasts, such as the USDA's monthly crop reports and the FAO Food Price Index, providing insights into global supply and demand dynamics.

Challenges of Traditional Methods: Despite their contributions, traditional methods faced significant challenges. Statistical models like ARIMA struggled with complex, nonlinear patterns and large datasets, as noted by Chatfield in Time-Series Forecasting (2000). Makridakis and Hibon (2000) further revealed performance issues in real-world applications.

Expert analysis was fraught with cognitive biases, as discussed by Kahneman in Thinking, Fast and Slow (2011). Tetlock (2005) critically examined expert forecasts, showing they often failed to outperform basic statistical models due to inherent biases. Scalability and transparency were problematic. Kitchin (2014) discussed the difficulties of scaling traditional methods to handle big data, while Pasquale's The Black Box Society (2015) explored transparency issues in algorithmic decision-making, highlighting the need for clearer, more interpretable models.

Transition to Machine Learning Techniques:

Machine Learning techniques addressed these limitations, significantly enhancing economic forecasting. Varian (2014) introduced the potential of big data and Machine Learning in economics, demonstrating their ability to improve prediction accuracy and manage complex datasets. James et al.'s *An Introduction to Statistical Learning* (2013) provided foundational knowledge of these techniques for economic applications.

Random Forest Regression emerged as a key advancement. Breiman's seminal paper on Random Forests (2001) highlighted their ability to capture complex, nonlinear relationships and improve predictive performance through an ensemble of decision trees. Cutler et al. (2007) demonstrated their robustness in handling diverse ecological data, underscoring their versatility for agricultural price prediction.

Applications in Agricultural Markets and Advantages of Machine Learning:

Machine Learning's application in agricultural markets has shown promising results. Dutta, Das, and Ghose (2018) explored how Machine Learning enhances agricultural data analysis and price prediction by improving predictive accuracy and decision-making. Kang, Özdoğan, and Zhu (2020) demonstrated the scalability and interpretability of Machine Learning models in predicting crop yields, directly correlating with commodity prices.

Enhanced accuracy and efficiency were significant advantages. Choi and Varian (2012) showed how real-time data sources like Google Trends could enhance economic forecasts by providing timely market indicators. Wang et al. (2016) highlighted the efficiency gains from big data analytics in supply chain management and price predictions, illustrating how Machine Learning streamlines operations.

Automation and scalability were also crucial. LeCun, Bengio, and Hinton (2015) discussed deep learning techniques' potential for automating complex prediction tasks, making them suitable for large-scale agricultural data. Domingos (2012) emphasized Machine Learning's capability to automate prediction tasks, enhancing efficiency and scalability.

Finally, **transparency and reproducibility** in Machine Learning models are essential for building trust among stakeholders. Doshi-Velez and Kim (2017) stressed the importance of interpretability, while Lipton (2018) explored model interpretability, advocating for clear, understandable models in price prediction applications.

In conclusion, the shift from traditional statistical methods to advanced Machine Learning techniques has significantly enhanced the accuracy, scalability efficiency, and transparency of agricultural commodity price predictions. This evolution supports informed decision-making, streamlines market operations, and drives innovation in agricultural economics and commodity trading.

3. EXISTING SYSTEM

The current system for predicting agricultural commodity prices relies on traditional statistical models, expert analysis, government reports, market intelligence firms, and commodity exchanges. Traditional statistical methods like time series analysis and regression use historical data to identify trends, while expert analysis incorporates qualitative factors such as weather and geopolitical events. Government and market intelligence reports provide valuable insights, and commodity exchanges facilitate price discovery through futures and options markets.

Challenges include the inability of traditional methods to capture complex, nonlinear patterns and handle large data volumes. Expert analysis can be subjective and slow to adapt to market changes. The system's reliance on manual processes limits scalability and transparency, affecting stakeholder trust.

Integrating Machine Learning techniques, such as Random Forest Regression, can address these challenges by identifying complex patterns in large datasets, enhancing prediction accuracy, and automating processes for faster decision-making. Machine Learning models improve scalability, transparency, and reproducibility, fostering trust and promoting innovation in agricultural markets.

4. PROPOSED SOLUTION

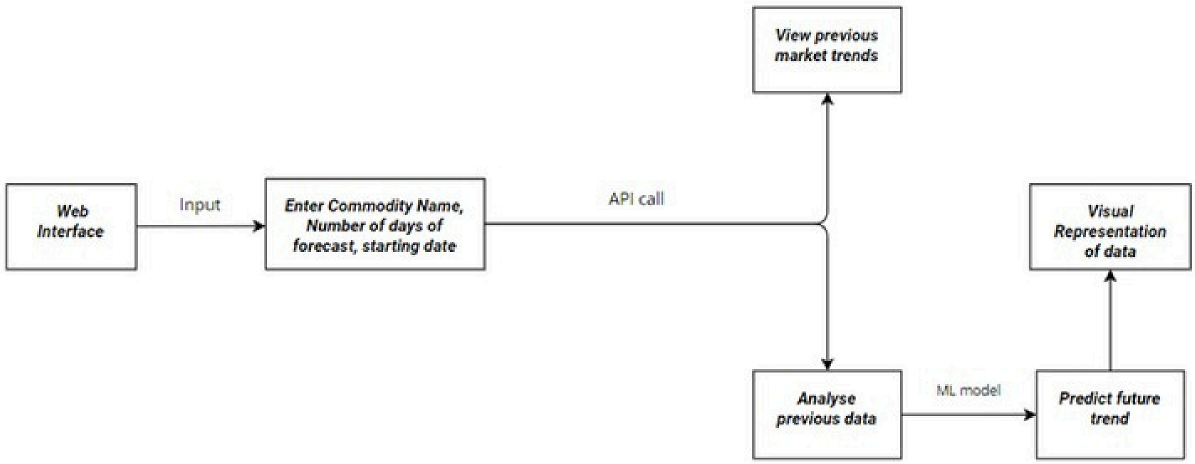
The proposed system for predicting agricultural commodity prices leverages advanced Machine Learning, particularly Random Forest Regression, to enhance forecasting accuracy and efficiency. This system analyzes extensive historical and real-time data, capturing complex market patterns to provide precise and reliable price predictions.

Key components include data preprocessing and feature engineering to clean data, handle outliers, and extract relevant features such as seasonal trends and economic indicators. Automation of the prediction process and real-time data integration reduce reliance on manual analysis, speeding up decision-making. Scalability and transparency are emphasized to handle large data volumes and provide clear, reproducible results, fostering trust among stakeholders.

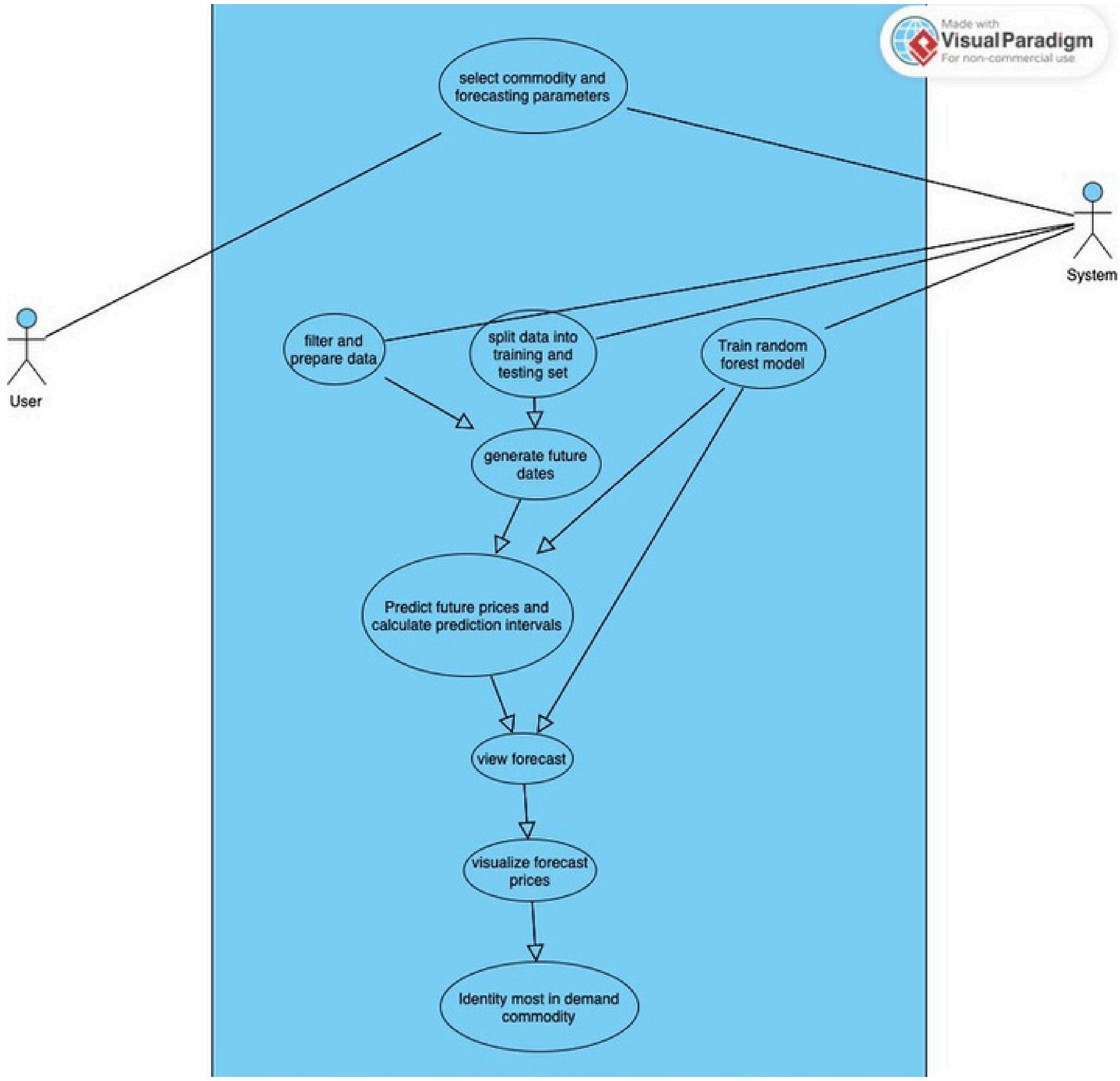
Overall, the system aims to support informed decision-making, streamline market operations, and drive innovation in agricultural economics and commodity trading, contributing to sustainable growth in the sector.

4.1. System Design:

4.1.1. Architectural Design:



4.1.2. Use-Case Diagrams:



4.1.2.1 Use-Case Descriptions:

Use case ID: UC01

Name: User Input and Submit form

Actors: User

Description: Users selects commodity and inputs forecasting parameters

User	Peer
1. User enter the commodity name	
2. User specifies forecast period, prediction target	
3. User inputs starting date and submit the form	

Table 4.1.2.1

Use case ID: UC02

Name: System compiles and forecast results

Actors: Peer

Description: Predicted prices and intervals are calculated ad determines most in demand commodity

User	Peer
	1. System train the model using random forest regressor
	2. System calculates the margin of error for the 95% confidence interval
	3. System creates dataframe by future dates,predicted prices and send result to web interface

Table 4.1.2.2

4.2 Functional Modules

4.2.1. Pseudo Code:

```
def index():
    return render_template('index.html')

@app.route('/select_prediction', methods=['POST'])
def select_prediction():
    prediction_type = request.form['prediction_type']
    if prediction_type == 'prices':
        return render_template('predict_prices.html')
    elif prediction_type == 'volumes':
        return render_template('volumes_traded.html')
```

- Based on what we select between predicting prices or predicting the volumes traded, the corresponding html file will be rendered.

```python
commodity = request.form['commodity']
forecast_period = int(request.form['forecast_period'])
start_date = request.form['start_date']
target_variable = request.form['target_variable']

# Filter data for selected commodity
df_commodity = df[df['commodity'] == commodity]
df_commodity['date'] = pd.to_datetime(df_commodity['date'])
df_commodity.sort_values(by='date', inplace=True)
df_commodity.reset_index(drop=True, inplace=True)

df_commodity['Day'] = df_commodity['date'].dt.day
df_commodity['Month'] = df_commodity['date'].dt.month
df_commodity['Year'] = df_commodity['date'].dt.year


model = train_model(df_commodity, target_variable)

# Generate future dates for forecasting
last_date = datetime.strptime(start_date, '%Y-%m-%d')
future_dates = pd.date_range(start=last_date, periods=forecast_period, freq='D')

future_df = pd.DataFrame({
    'Day': future_dates.day,
    'Month': future_dates.month,
    'Year': future_dates.year
})

predicted_prices = model.predict(future_df)

# Calculate prediction intervals
std_dev = np.std([tree.predict(future_df) for tree in model.estimators_], axis=0)
margin_of_error = 1.96 * std_dev  # 95% confidence interval
lower_prediction_interval = predicted_prices - margin_of_error
```

- Takes commodity(Coffee, Cocoa, Sugar, Cotton, Orange Juice, Random Length Number), forecast period, start date and target variable(open, high, close, low) as inputs from the form and runs the random forest regressor on commodity as independent variable and target variable as dependent. Also the data is further broke down into day month and year and the data is filtered only for the given commodity before training. The outputs are also calculated in a data frame with columns as date(ranging from start date and goes on until no of forecasting days), predicted price and lower prediction and upper prediction intervals.

```python
def visualize():
    commodity = request.form['commodity']
    forecast_df = pd.read_html(request.form['forecast_df'], index_col=0)[0]

    plt.figure(figsize=(8, 6))

    plt.plot(forecast_df['Date'], forecast_df['Predicted_Price'], color='green', label='Predicted Price')

    plt.fill_between(forecast_df['Date'], forecast_df['Lower_Prediction_Interval'], forecast_df['Upper_Prediction_Interval'], color='blue

    plt.xlabel('Date')
    plt.ylabel('Price')
    plt.title(f'Forecasted Prices with Prediction Intervals for {commodity}')
    plt.legend()

    plt.xticks(rotation=45)

    # Save the plot to a BytesIO object
    img = BytesIO()
    plt.tight_layout()
    plt.savefig(img, format='png')
    img.seek(0)
    plt.clf()

    # Return the plot as an image file
```

- Visualizes the obtained data frame of predicted prices for the given dates. X variable is date and Y variable is predicted price. Plotted using matplotlib and prediction intervals are denoted using a shade of lighter colors.

```python
plt.xlabel('Commodity')
plt.ylabel('Total Volume Traded')
plt.title('Total Volume Traded for Each Commodity')
plt.xticks(rotation=45, ha='right')

img = BytesIO()
plt.tight_layout()
plt.savefig(img, format='png')
img.seek(0)

# Clear the plot
plt.clf()

# Return the plot as an image file
return send_file(img, mimetype='image/png')
```

- This is a bar graph showing all commodities and their respective volumes traded in a given date range.

```python
start_date = request.form['start_date']
df['date'] = pd.to_datetime(df['date'])

prediction = 'volume'

total_volume_traded = {}

# Loop through each commodity
for commodity in df['commodity'].unique():

    df_commodity = df[df['commodity'] == commodity]
    df_commodity.sort_values(by='date', inplace=True)
    df_commodity.reset_index(drop=True, inplace=True)

    future_dates = pd.date_range(start=start_date, periods=forecast_period, freq='D')

    df_commodity['Day'] = df_commodity['date'].dt.day
    df_commodity['Month'] = df_commodity['date'].dt.month
    df_commodity['Year'] = df_commodity['date'].dt.year

    X = df_commodity[['Day', 'Month', 'Year']]
    y = df_commodity[prediction]

    model2 = train_model(df_commodity, prediction)

    future_df = pd.DataFrame({
        'Day': future_dates.day,
        'Month': future_dates.month,
        'Year': future_dates.year
    })

    predicted_volume_traded = model2.predict(future_df[['Day', 'Month', 'Year']])
    # Sum predicted volume traded for the commodity
    total_volume_traded[commodity] = np.sum(predicted_volume_traded)
```

- Loops through every commodity and an another RFR model is trained with date as input variable and volume traded as output variable. for each iteration through commodities, values stored in a dictionary with commodity as key and it's total volume traded within the given date range.

5. EXPERIMENTAL SETUP & IMPLEMENTATION

5.1 System Specifications

5.1.1 Hardware Requirements

Recommended:

Any modern desktop or laptop computer.

Processor: Intel Core i5, equivalent or higher.

RAM: 8 GB RAM or higher.

Storage: SSD for faster performance and ample free disk space.

5.1.2 Software Requirements

Recommended:

Operating System: Windows 11, macOS 14.4.1, Ubuntu 24.04 LTS

Browser: Google Chrome, Mozilla Firefox, Safari

Backend: Python 3.12, Flask 3.0.3, Scikit learn

Frontend: HTML,CSS

5.2 Methodology:

The following methodology outlines the steps and algorithm for forecasting agricultural commodity prices using a machine learning model (Random Forest Regressor) and a Flask-based web interface.

Step 1: **Data Collection and Preparation**

Load Dataset: Load the dataset containing historical agricultural commodity prices from a CSV file (all_agricultural_products_data.csv).

Data Cleaning: Convert the 'date' column to datetime format. Ensure that the dataset is free of missing or erroneous values.

Feature Engineering: Extract relevant features from the 'date' column, such as 'Day', 'Month', and 'Year'.

Step 2: **Model Training**

Commodity Selection: Allow the user to select the commodity of interest via the web interface.

Data Filtering: Filter the dataset to include only the selected commodity. Sort the filtered data by date and reset the index.

Split Data into Features and Target: Define the features (X) as 'Day', 'Month', and 'Year'. Define the target variable (y) based on user input (e.g., 'price', 'volume').

Train-Test Split: Split the data into training and testing sets using an 80-20 split.

Model Training: Initialize and train a Random Forest Regressor model using the training data.

Step 3: **Forecasting**

Forecast Period: Allow the user to specify the forecast period (number of days into the future).

Generate Future Dates:Based on the forecast period and the starting date input by the user, generate future dates for forecasting.

Feature Engineering for Future Dates: Extract 'Day', 'Month', and 'Year' from the generated future dates.

Predict Future Prices: Use the trained Random Forest model to predict future prices for the generated dates.

Calculate Prediction Intervals: Compute the standard deviation of the predictions from all trees in the forest. Calculate the margin of error for a 95% confidence interval. Determine the lower and upper prediction intervals.

Step 4: **Results Presentation**

Prepare Forecast DataFrame: Combine future dates, predicted prices, and prediction intervals into a DataFrame. Visualize Forecast: Plot the forecasted prices and prediction intervals using Matplotlib.

Step 5: **Web Interface**

Setup Flask Application: Create a Flask web application with routes for the home page (/) and the forecast results (/forecast). Home Page: Design a form to collect user inputs: commodity, forecast period, prediction target, and starting date. Forecast Page: Display the visualizations in image format.

6. RESULTS

The web application allows users to input parameters such as commodity name, forecast period, target variable (price or volume), and starting date. It then generates predictions for commodity prices or analyzes trading volumes based on the selected parameters.

The project utilized Flask, a lightweight web framework for Python, to create a web application. Pandas library was used to manipulate and analyze the commodity data stored in a CSV file. scikit-learn was used to train a machine learning model (Random Forest Regressor) to predict commodity prices.

The web application empowers users to input specific parameters for commodity analysis, such as commodity name, forecast period, target variable (price or volume), and starting date. It then provides predictions for commodity prices or conducts analyses on trading volumes based on the selected parameters. The project utilized Flask, a lightweight web framework for Python, to develop the web application. Pandas library was employed to manipulate and analyze the commodity data stored in a CSV file. scikit-learn was utilized to train a machine learning model, specifically the Random Forest Regressor, for predicting commodity prices. This project showcases how Flask can be utilized to construct a web application that integrates machine learning models for predictive analysis. By combining Flask with libraries like Pandas and scikit-learn, users can create dynamic and engaging data-driven applications for diverse purposes, such as financial forecasting and analysis.

6.1. Screenshots :

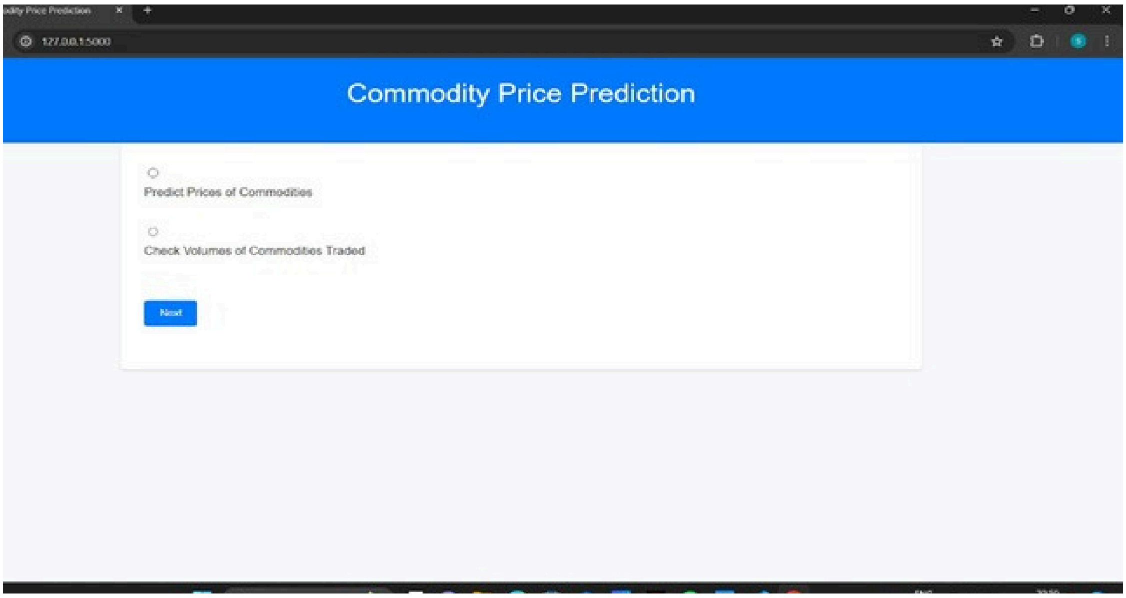


Fig 6.1.1 Page to select whether to predict price or predict volumes traded

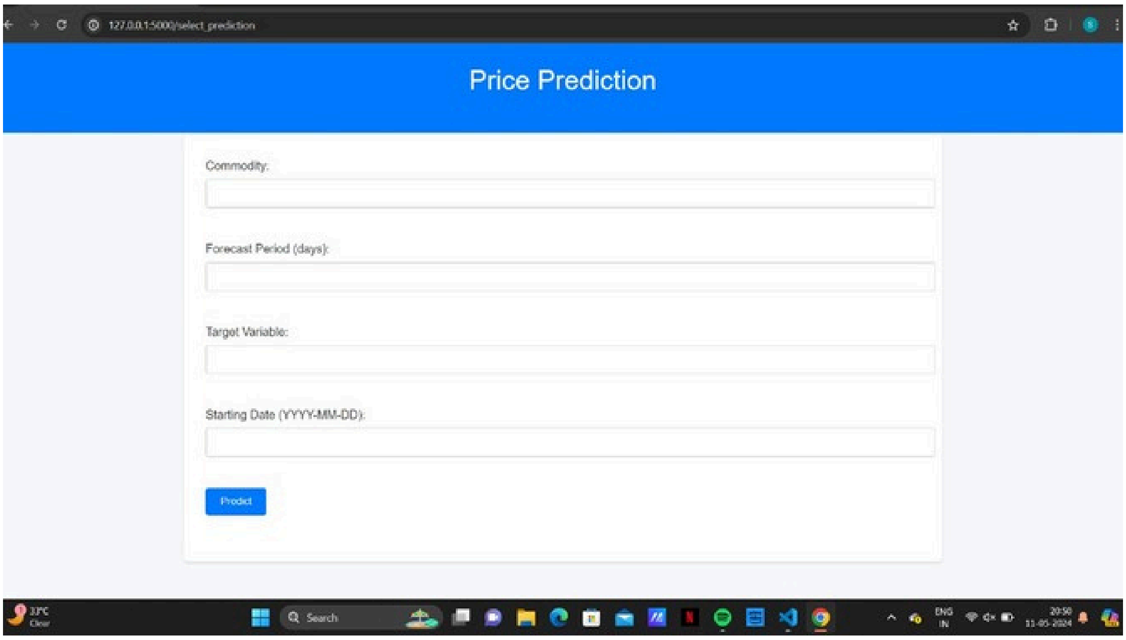


Fig 6.1.2 Page for price prediction of a commodity

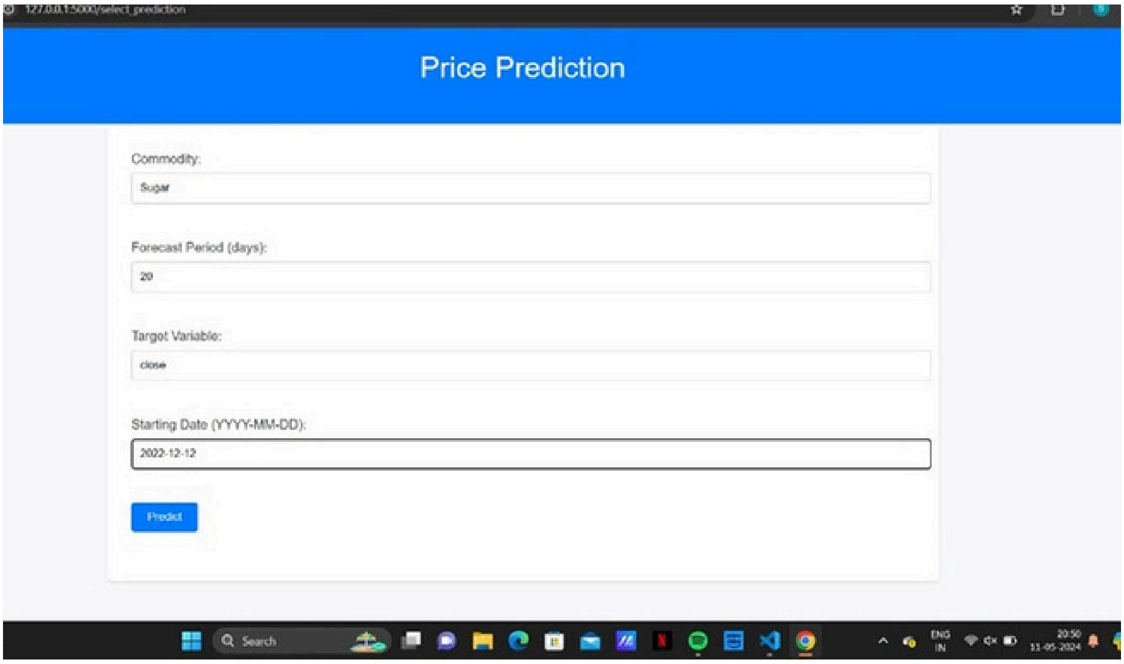
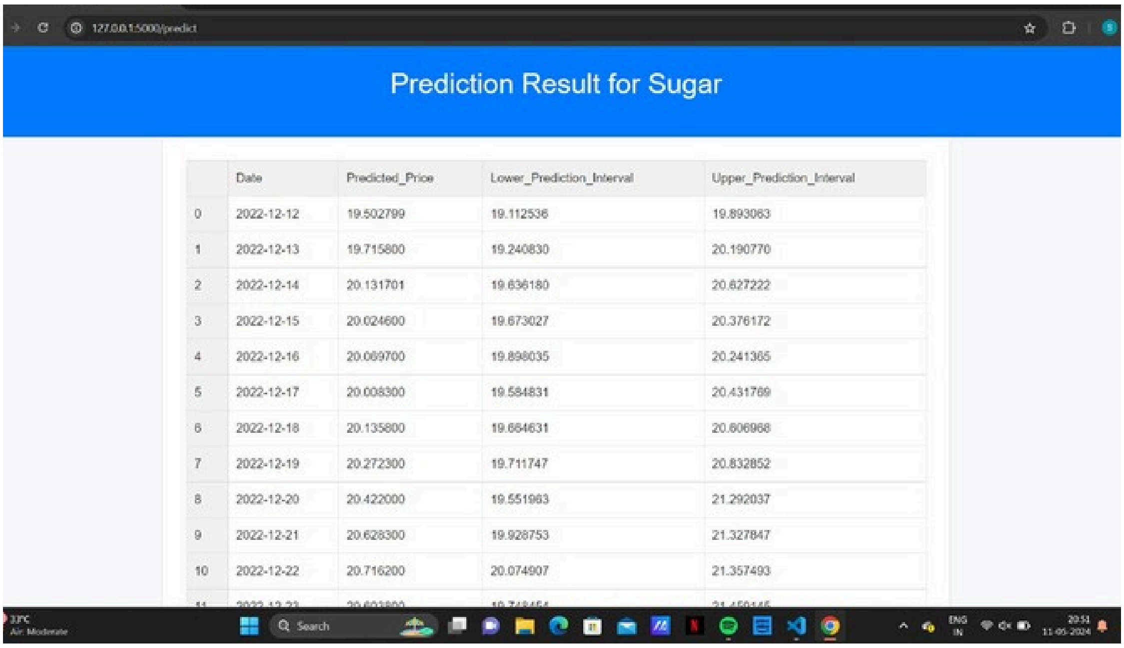
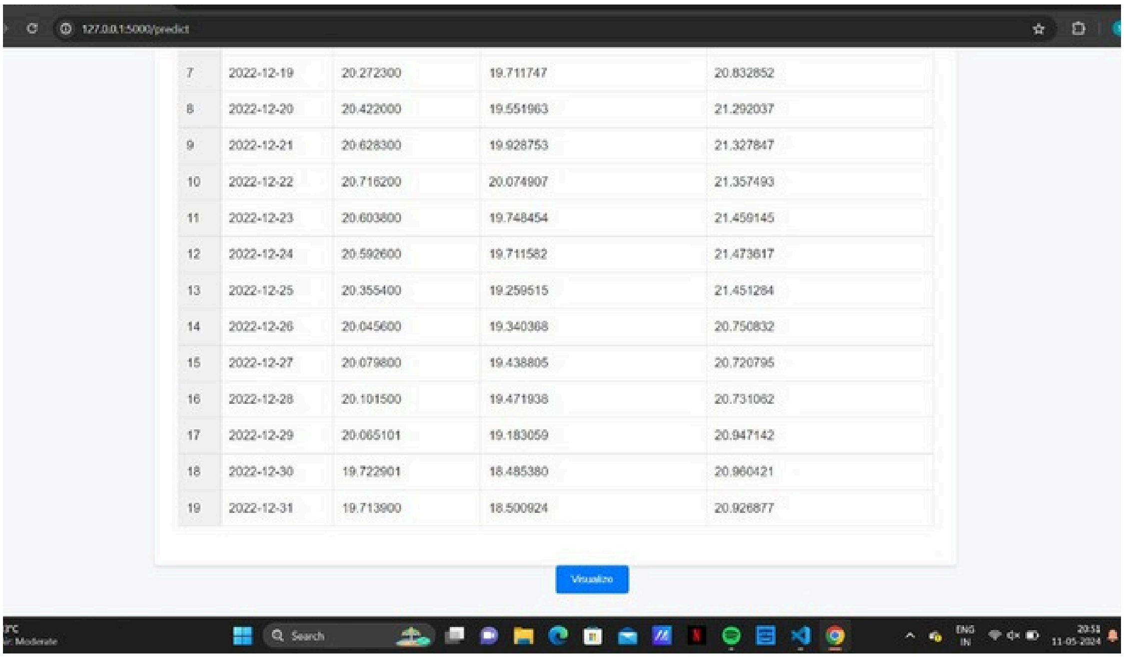


Fig 6.1.3 Example entries



	Date	Predicted_Price	Lower_Prediction_Interval	Upper_Prediction_Interval
0	2022-12-12	19.502799	19.112536	19.893063
1	2022-12-13	19.715800	19.240830	20.190770
2	2022-12-14	20.131701	19.636180	20.627222
3	2022-12-15	20.024600	19.673027	20.376172
4	2022-12-16	20.069700	19.898035	20.241365
5	2022-12-17	20.008300	19.584831	20.431769
6	2022-12-18	20.135800	19.664631	20.606968
7	2022-12-19	20.272300	19.711747	20.832852
8	2022-12-20	20.422000	19.551963	21.292037
9	2022-12-21	20.628300	19.928753	21.327847
10	2022-12-22	20.716200	20.074907	21.357493

Fig 6.1.4 Prediction output for the given input



7	2022-12-19	20.272300	19.711747	20.832852
8	2022-12-20	20.422000	19.551963	21.292037
9	2022-12-21	20.628300	19.928753	21.327847
10	2022-12-22	20.716200	20.074907	21.357493
11	2022-12-23	20.603800	19.748454	21.459145
12	2022-12-24	20.592600	19.711582	21.473617
13	2022-12-25	20.355400	19.259515	21.451284
14	2022-12-26	20.045600	19.340368	20.750832
15	2022-12-27	20.079800	19.438805	20.720795
16	2022-12-28	20.101500	19.471938	20.731062
17	2022-12-29	20.065101	19.183059	20.947142
18	2022-12-30	19.722901	18.485380	20.960421
19	2022-12-31	19.713900	18.500924	20.926877

Fig 6.1.5 Prediction output for the given input

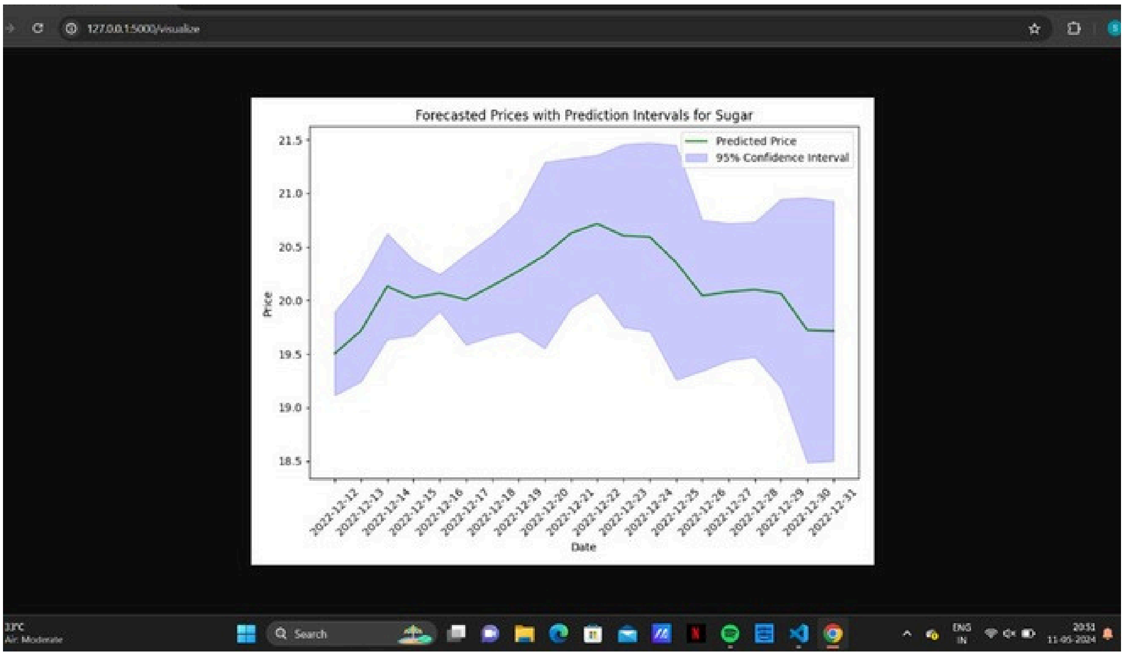


Fig 6.1.6 Visual representation of the predicted output

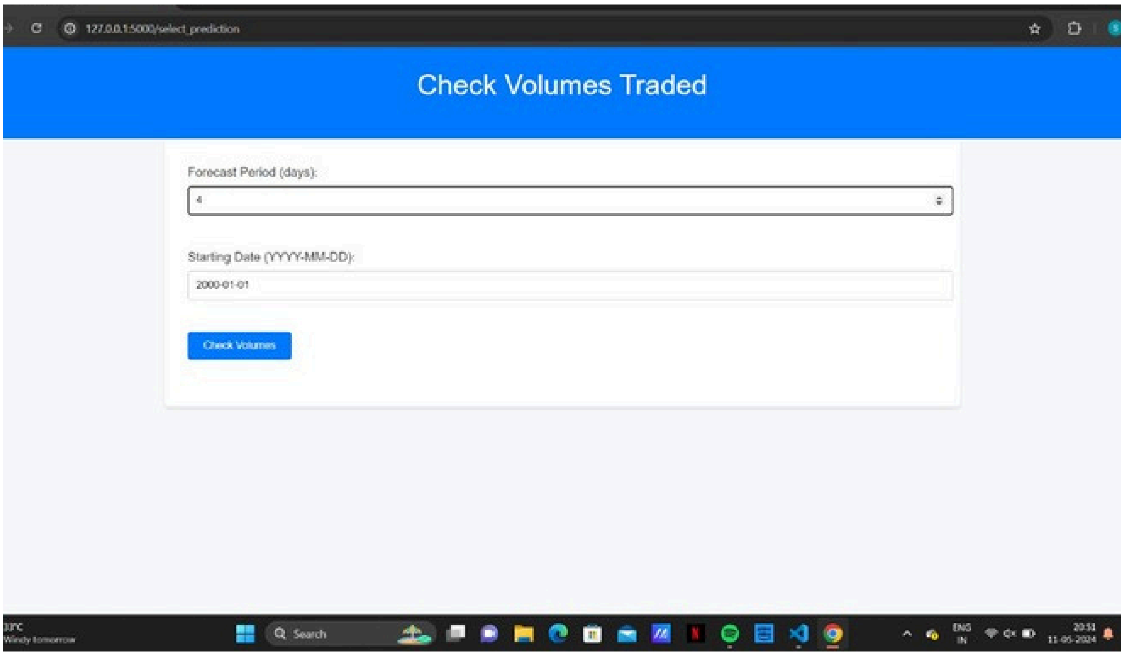


Fig 6.1.7   Page for total volumes traded for a given period

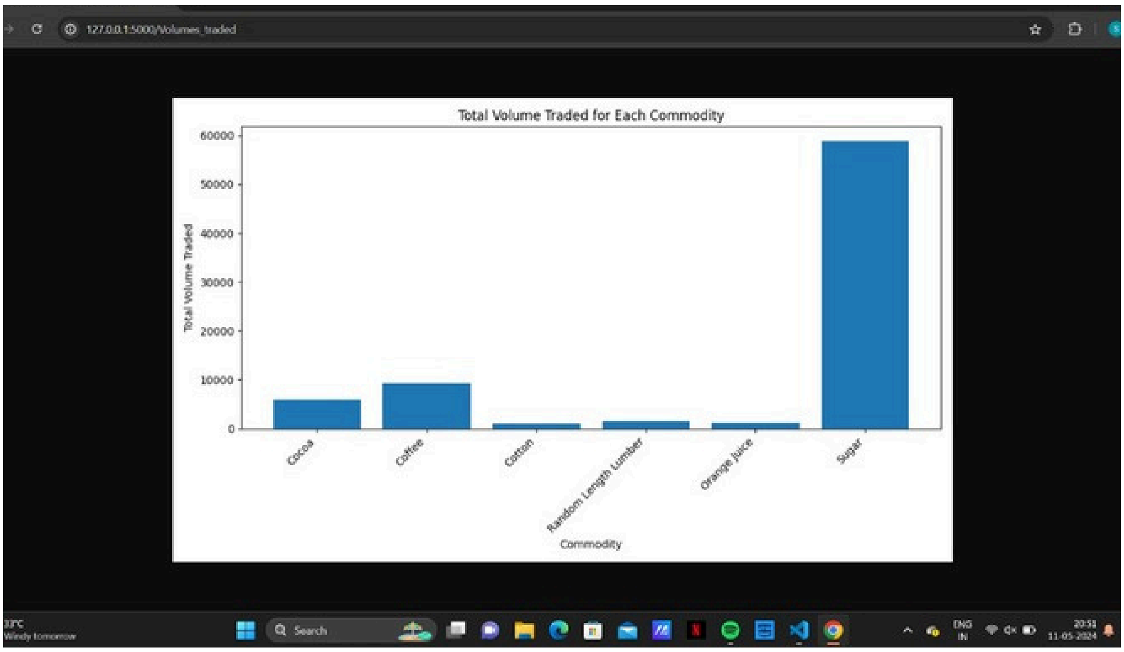


Fig 6.1.8   Bar graph plot showing total volumes traded for each commodity

7. CONCLUSION & FUTURE SCOPE

In conclusion, the proposed Machine Learning-based system for predicting agricultural commodity prices offers a significant advancement over traditional forecasting methods. By employing techniques such as Random Forest Regression, the system can capture complex market relationships and nonlinear patterns, enhancing the accuracy and reliability of price predictions. The integration of robust data preprocessing, feature engineering, and real-time data processing ensures high-quality inputs and timely insights, streamlining decision-making for stakeholders. Prioritizing automation, scalability, and transparency, this innovative approach promises to foster trust, support sustainable growth, and drive development in the agricultural sector, ultimately revolutionizing agricultural economics and commodity trading.

FUTURE SCOPE

- User Authentication and Authorization: Implement user authentication and authorization to secure the application, allowing registered users to save their predictions, customize settings, and access personalized features.

- Real-Time Data Integration: Integrate APIs or data streaming services to fetch real-time commodity prices and trading volumes, enabling users to make more timely and accurate predictions . Enhanced Visualization: Improve data visualization by incorporating interactive charts and graphs using libraries like Plotly or D3.js. This would provide users with a richer and more intuitive way to explore and analyze commodity data.

- Advanced Machine Learning Models: Experiment with more advanced machine learning algorithms and techniques to improve prediction accuracy, such as deep learning models or ensemble methods.

8. REFERENCES

Flask Documentation: https://flask.palletsprojects.com/en/2.0.x/

Pandas Documentation: https://pandas.pydata.org/docs/

scikit-learn Documentation: https://scikit learn.org/stable/documentation

Random Forest Algorithm for statistical learning.
https://journals.sagepub.com/doi/full/10.1177/1536867X20909688

An Introduction to Statistical Learning (James et al., 2013)
https://www.statlearning.com/

Towards a Rigorous Science of Interpretable Machine Learning (Doshi-Velez and Kim, 2017)
https://arxiv.org/abs/1702.08608