Step-by-Step Theory for Predicting Crop Prices

1. Introduction:

The primary objective of this project is to predict the minimum, maximum, and modal prices of crops for the next 10 days using a non-linear regression model, specifically **XGBoost.** The dataset Total_Crops.csv comprises various attributes, including **mandiid, cropid, cropname, mandiname, arrivalquantity, maximumprice, minimumprice, modalprice**, and **date.**

2. Data Loading:

- **Purpose**: The first step is to load the dataset into a pandas Data Frame to make it suitable for processing and analysis.
- **Process**: Utilize the `pandas` library to read the CSV file into a Data Frame, which allows for easy data manipulation and analysis.

3. User Input for Filtering:

- **Purpose**: Filter the dataset to focus on specific market **(**`mandiid`) and crop (`cropid`) combinations, as specified by the user. This ensures the model is trained on relevant data.
- **Process**: Prompt the user to input the desired `mandiid` and `cropid` values. Filter the Data Frame to retain only the rows that match these criteria.

4. Drop Unnecessary Columns:

- **Purpose**: Remove columns that are not needed for the prediction model to reduce complexity and improve performance.
- **Process**: Drop the `arrivalquantity` column from the filtered Data Frame as it is not required for predicting prices.

5. Date Conversion and Formatting:

- **Purpose**: Convert the `date` column to a standard datetime format and reformat it to `day/month/year`. This ensures consistency in date representation and facilitates feature extraction.
- **Process**: Use the `pandas` library to convert the `date` column to datetime format and reformat it to the desired format. Extract additional date-related features (year, month, day) from the date for use as predictors.

6. Creating Lag Features:

- **Purpose**: Generate lagged features to capture the time-series dependencies in the data. Lag features represent the values of a variable at previous time points.
- **Process**: Create lagged variables for the past 7 days of **`modalprice`**, **`minimumprice`**, and **`maximumprice`**. This involves shifting the original price columns by 1 to 7 days and adding these shifted values as new columns in the DataFrame. Rows with missing values created by lagging are dropped.

7. Encoding Categorical Variables:

- **Purpose**: Convert categorical variables into numerical format, which is required by most machine learning algorithms.
- **Process**: Apply label encoding to the **`mandiname`** and **`cropname`** columns, transforming categorical values into unique integer labels.

8. Defining Features and Targets:

- **Purpose**: Specify the input features and target variables for the model. The features include both original and engineered variables, while the targets are the prices we aim to predict.
- **Process**: Define a list of features, including date components, lagged price features, and encoded categorical variables. The target variables are the **`minimumprice`**, **`maximumprice`**, and **`modalprice`**.

9. Train-Test Split:

- **Purpose**: Split the dataset into training and testing sets to evaluate the model's performance on unseen data.
- **Process**: Use **`train_test_split`** from the **`sklearn`** library to divide the data into training and testing sets. Typically, 80% of the data is used for training, and 20% is reserved for testing.

10. Model Training:

- **Purpose**: Train a multi-output regressor to predict multiple target variables simultaneously. **XGBoos**t is chosen for its efficiency and ability to handle non-linear relationships.
- **Process**: Utilize the **`MultiOutputRegressor`** wrapper with **`XGBRegressor`** to train a model on the training data. This involves fitting the model to the input features and target variables.

11. Model Evaluation:

- **Purpose**: Evaluate the model's performance using appropriate metrics to understand its accuracy and effectiveness.
- **Process**: Predict the prices on the test set and compute the **Mean Absolute Error (MAE)** to quantify the prediction error**. MAE** provides an average of the absolute errors between predicted and actual values.

12. Future Predictions:

- **Purpose**: Predict future crop prices for the next 10 days based on the trained model. This step involves iteratively updating the lag features with the model's predictions.
- **Process**: Start with the last known values from the filtered data and use the model to predict the next day's prices. Update the lag features with these predicted values and repeat the process for 10 days to generate future predictions.

13. Visualization:

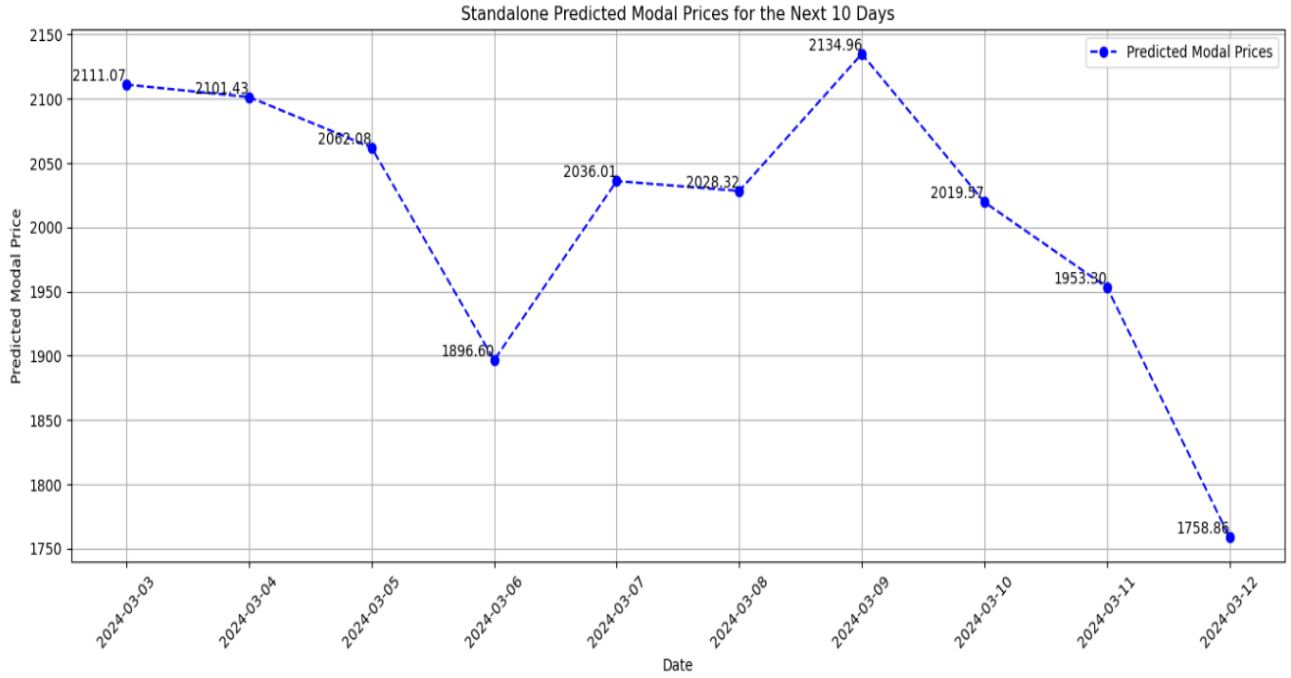
- **Purpose**: Create visual representations of the historical and predicted prices to provide a clear and intuitive understanding of the trends and future forecasts.
- **Process**:
    - **Historical and Predicted Modal Prices**: Plot both historical and predicted `modalprice` values on the same graph. Use markers to indicate data points and add labels to each predicted data point for clarity.
    - **Standalone Predicted Modal Prices**: Plot only the predicted `modalprice` values for the next 10 days, with labels for each data point.
    - **Historical and Predicted Minimum Prices**: Similarly, plot both historical and predicted `minimumprice` values with labels for predicted points.
    - **Standalone Predicted Minimum Prices**: Plot only the predicted `minimumprice` values for the next 10 days, with labels.
    - **Historical and Predicted Maximum Prices**: Plot both historical and predicted `maximumprice` values with labels for predicted points.
    - **Standalone Predicted Maximum Prices**: Plot only the predicted `maximumprice` values for the next 10 days, with labels.

Summary:

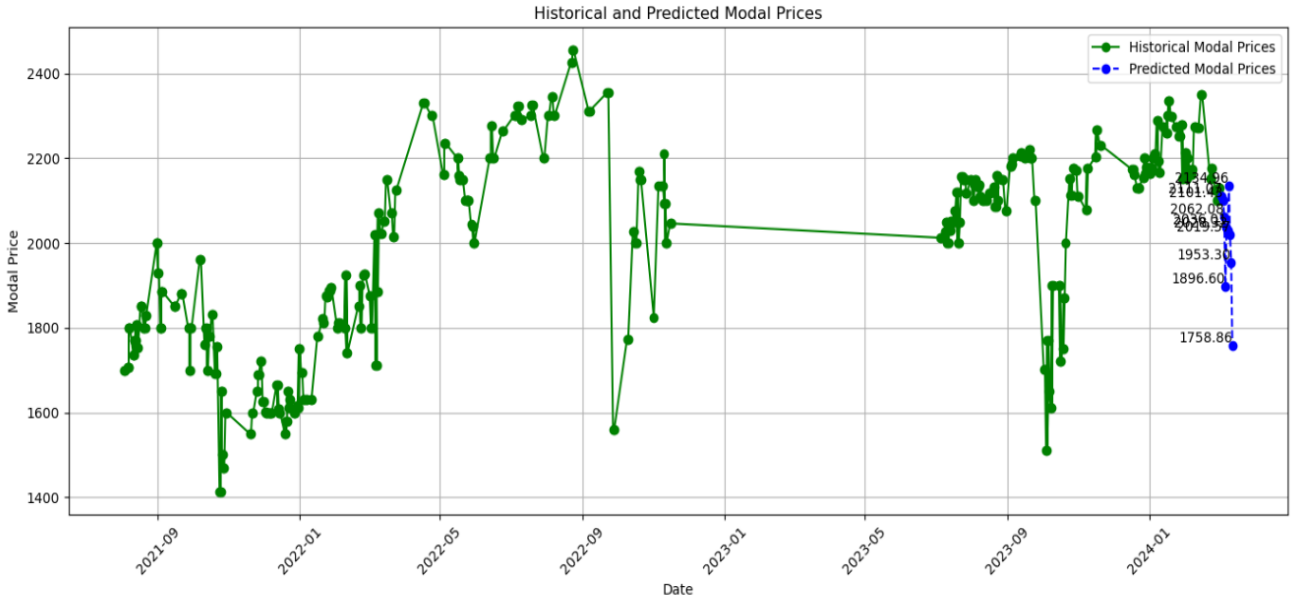
This step-by-step process provides a comprehensive approach to predicting crop prices, leveraging historical data, and using advanced machine learning techniques. The key steps involve data preprocessing, feature engineering, model training, evaluation, and visualization. This method ensures that the predictions are both accurate and interpretable, providing valuable insights into future crop price trends.

Output:

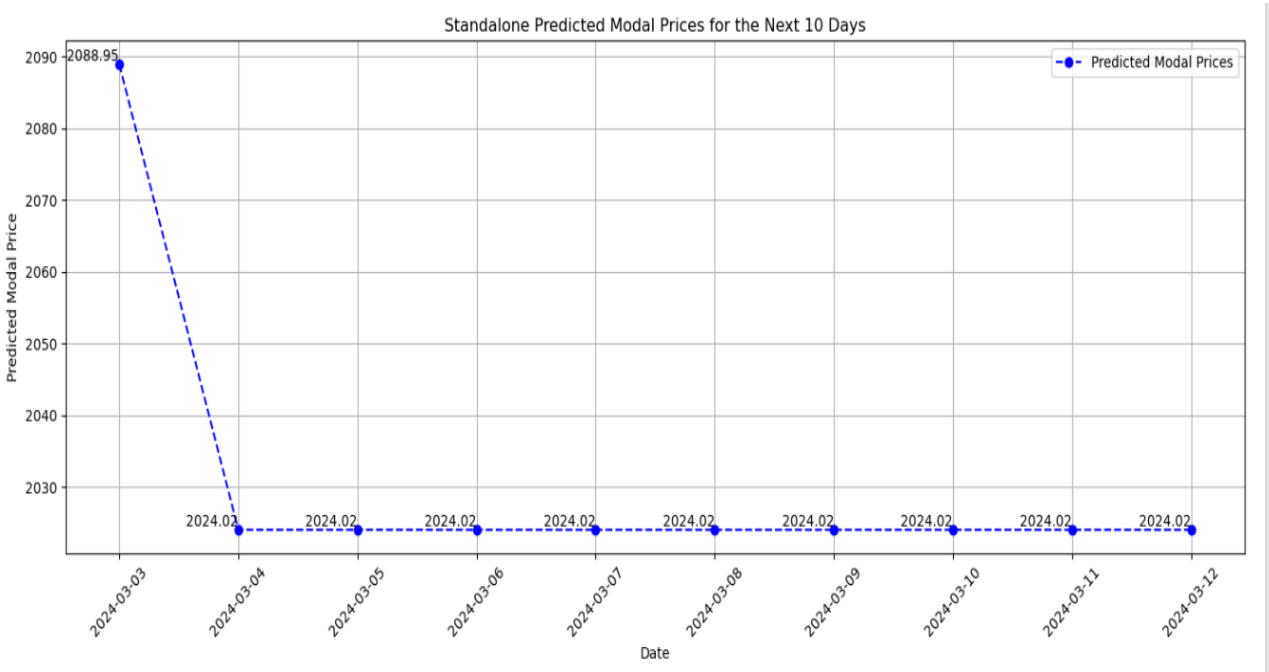
1. XGBoost Predicted values



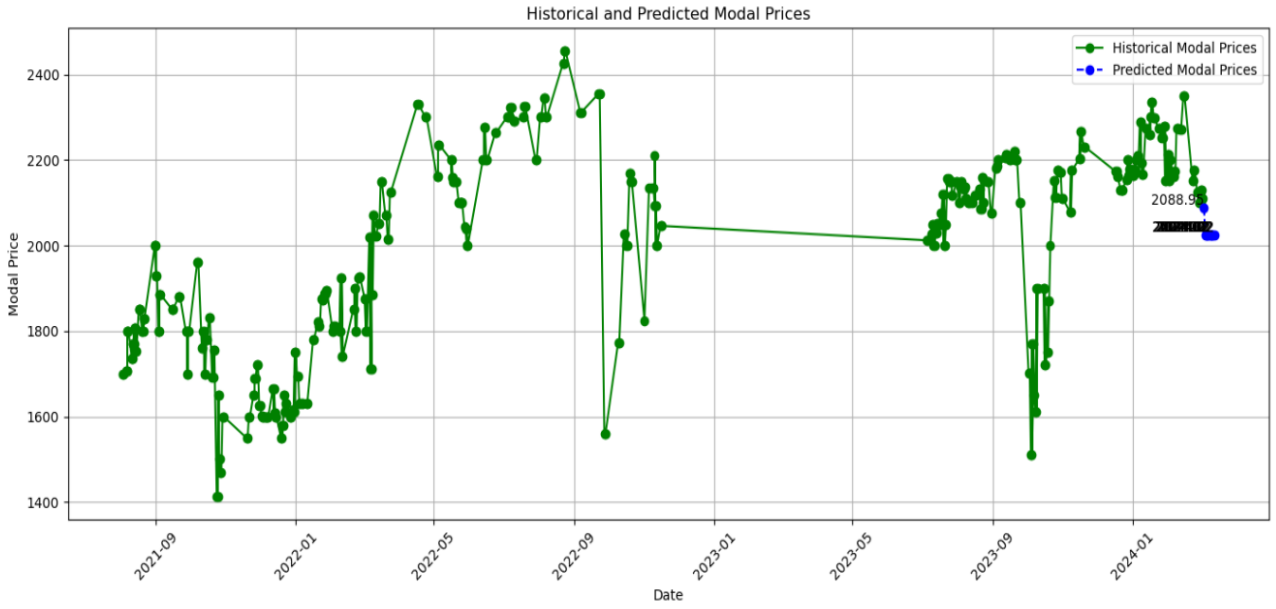
2. XGBoost Historical+Predicted values Graph



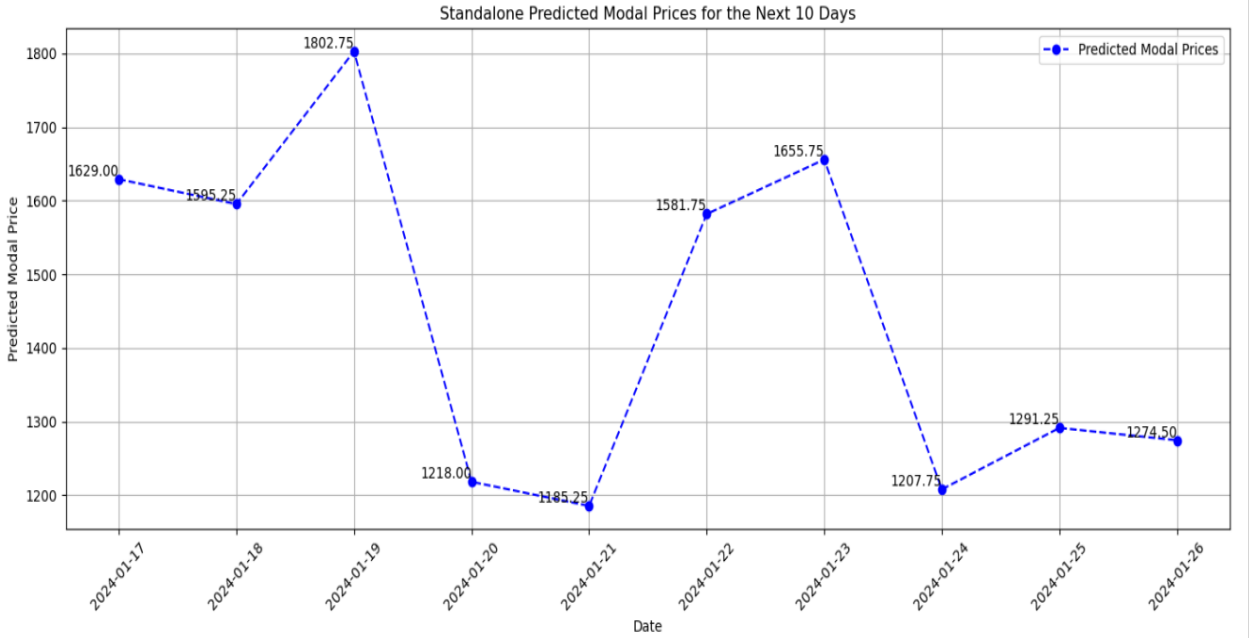
3. SVR- Predicted values



4. SVR- Historical+Predicted values Graph



5. RandomForestRegressor Predicted values



6. RandomForestRegressor Historical+Predicted values Graph