

# Exploratory Data Analysis on

AGRICULTURAL PRODUCTION

by

## Group 2



Shreshtha Modi  
ID: 202411015  
Course: MTech(ICT)



Khushi Prajapati  
ID: 202201062  
Course: BTech(ICT)



Jinay Vora  
ID: 202201473  
Course: BTech(ICT)

Course Code: IT 462  
Semester: Autumn 2024

---

Under the guidance of

**Dr. Gopinath Panda**



Dhirubhai Ambani Institute of Information and Communication Technology

December 2, 2024

# ACKNOWLEDGMENT

We write this letter to express our heartfelt gratitude to, Prof. Gopinath Panda for your guidance and support throughout the "**Agricultural Production**" project. Your invaluable assistance has played a pivotal role in shaping the successful completion of this endeavour.

We greatly benefitted from your informative presentations and hands-on teaching methods, which have improved my grasp of data analysis techniques. The positive feedback and support throughout the project improved our analysis abilities and motivated me to explore the topic further.

The clear explanations and willingness to address doubts gave us a solid foundation, enabling us to approach this project. We are fortunate to have been guided by an experienced and supportive mentor.

Once again, thank you sir for your unwavering guidance and belief in our abilities. Your mentorship has been invaluable, and we are truly grateful for the opportunity to work with you.

Sincerely,

[Shreshtha (202411015), Khushi (202201062), Jinay (202201473)]

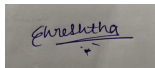
# DECLARATION

We, [Shreshtha, Khushi, Jinay] declare that the EDA project work presented in this report is our original work and has not been submitted for any other academic degree. All the sources cited in this report have been appropriately referenced.

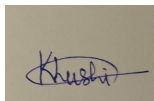
We acknowledge that the data used in this project is obtained from the "**data.gov.in**" site. We also declare that we have adhered to the terms and conditions mentioned on the website for using the dataset. We confirm that the dataset used in this project is true and accurate to the best of our knowledge.

We affirm that this project was conducted independently, with up to 20% overlap with external sources for references and standard methods. Guidance was provided solely by our mentor, Prof. Gopinath Panda and there is no conflict of interest in this EDA project.

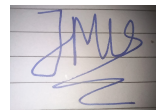
As a result, we sign the declaration statement and confirm the submission of this report on 2nd December 2024.



Shreshtha Modi  
ID: 202411015  
Course: MTech(ICT)



Khushi Prajapati  
ID: 202201062  
Course: BTech(ICT)



Jinay Vora  
ID: 202201473  
Course: BTech(ICT)

# CERTIFICATE

This is to certify that Group 2 comprising Shreshtha Modi, Khushi Prajapati, and Jinay Vora have completed an exploratory data analysis (EDA) project on **"Agricultural Production"**, which was obtained from data.gov.in.

The EDA project presented by Group 2 is their original work. It has been completed under the guidance of the course instructor, Prof. Gopinath Panda, who has provided support and guidance throughout the project. The project is based on a thorough analysis of the **"Variety-wise Daily Market Prices Data of Commodity"** dataset, and the results presented in the report are based on the data obtained from the dataset.

This certificate is issued to recognize the successful completion of the EDA project on the ....., which demonstrates the analytical skills and knowledge of the students of Group 2 in the field of data analysis.

Signed,  
Dr. Gopinath Panda,  
IT 462 Course Instructor  
Dhirubhai Ambani Institute of Information and Communication Technology  
Gandhinagar, Gujarat, INDIA.

December 2, 2024

# Contents

<b>List of Figures</b>	<b>5</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Your Project idea . . . . .	1
1.2 Data Collection . . . . .	1
1.3 Dataset Description . . . . .	1
1.4 Packages required . . . . .	3
<b>2 Data Cleaning</b>	<b>4</b>
2.1 Missing data analysis . . . . .	4
2.1.1 Filtering and Categorization . . . . .	4
2.1.2 Encoding categorical Variables . . . . .	4
2.1.3 Outlier detection and Removal . . . . .	4
2.1.4 Date Column transformation . . . . .	5
2.1.5 Variance and Correlation Analysis . . . . .	5
2.2 Imputation . . . . .	6
<b>3 Visualization</b>	<b>7</b>
<b>4 Feature Engineering</b>	<b>14</b>
4.1 Feature extraction . . . . .	14
4.1.1 Date Handling . . . . .	14
4.1.2 Categorical Encoding . . . . .	14
4.1.3 Outlier Detection and Treatment . . . . .	14
4.2 Feature selection . . . . .	16
4.3 Feature Dropping . . . . .	17
<b>5 Model fitting</b>	<b>19</b>
5.1 Regression . . . . .	19
5.1.1 Detection of Overfitting . . . . .	19
5.1.2 Visualizations . . . . .	20
5.2 ML algorithms . . . . .	20
<b>6 Conclusion &amp; future scope</b>	<b>24</b>
6.1 Findings/observations . . . . .	24
6.1.1 Model Performance: . . . . .	24
6.1.2 Comparative Model Evaluation: . . . . .	24

6.1.3 Data Insights: . . . . . 24

6.2 Challenges . . . . . 25

6.3 Future plan . . . . . 25

# List of Figures

1.1	dataset . . . . .	2
1.2	grouping dataset . . . . .	2
1.3	. . . . .	3
2.1	outlier code . . . . .	5
2.2	correlation matrix . . . . .	6
3.1	. . . . .	8
3.2	. . . . .	9
3.3	. . . . .	9
3.4	. . . . .	10
3.5	. . . . .	10
3.6	. . . . .	11
3.7	. . . . .	11
3.8	. . . . .	12
3.9	. . . . .	13
4.1	Before Removing Outliers . . . . .	15
4.2	After Removing Outliers . . . . .	16
4.3	. . . . .	17
4.4	Scree Plot for the Principal Components . . . . .	17
5.1	Multiple Linear Regression . . . . .	21
5.2	Decision Tree Regressor . . . . .	21
5.3	Random Forest Regressor . . . . .	22
5.4	Gradient Boosting Regressor . . . . .	22
5.5	XGBoost Regressor . . . . .	23
5.6	LightGBM Regressor . . . . .	23

# List of Tables

3.1	Categorization of Commodities - Part 1 . . . . .	7
3.2	List of Commodities . . . . .	8



## **Abstract**

Stakeholders in competitive agricultural commodity markets need to understand the dynamics of agricultural commodity pricing. In this study, we perform a detailed exploratory analysis of agricultural data on an attempt to uncover patterns and relationships which influence the Modal Price. Preprocessing of the dataset was used to enhance usability (filter low representation commodities, categorize items, and integrate geospatial information). Through advanced visualizations (network graphs that link categories and commodities, density maps illustrating supply distribution, etc.) key insights were drawn including seasonal trends in modal prices.

Results suggested a strong modality in price that varies across categories and markets. High supply density was geospatially mapped, while commodity trends were studied across seasons to understand their demand-supply dynamics.

Extracting actionable insights from patterns in price distributions and supply behaviour represents the first step in the study that lays the foundation for predictable modelling. These findings can also serve as a basis for strategic decisions such as price forecasting and prioritizing the market. Future work involves taking this framework to real-time predictive systems and integrating external factors like weather and logistics.

# Chapter 1. Introduction

## 1.1 Your Project idea

The goal of this project is to design an integrated framework for understanding and predicting the Modal Price of agricultural commodities using advanced exploratory data analysis, visualization techniques, and machine learning. This will help categorize commodities according to their characteristics and analyze supply patterns across different geographical regions. Identify Pricing and Seasonality Trend of Supply. Use data visualization-including geospatial maps and temporal trends-to derive actionable insights for stakeholders. Ground it on knowledge of the dominant drivers of commodity prices and their distributions.

## 1.2 Data Collection

The dataset was collected to analyse the modal price of each agricultural commodity. The data used in this project has been collected from "**data.gov.in**".The dataset is generated through the AGMARKNET portal "**agmarknet.gov.in**".The data has been filtered to Gujarat state and Ahmedabad district. The dataset taken represents data from year 2001-2024.

## 1.3 Dataset Description

The dataset represents the prices of various commodities for Gujarat state.It has the wholesale maximum prices, minimum prices and modal price on daily basis.Below is the detailed description of the dataset.

**General description:**

- **Columns:** 11
- **Rows:** 320,323

**Column description**

- **State:** State where the market is located.
- **District:** Particular district in the state.
- **Market:** The specific market where data was recorded.
- **Commodity:** List of all the agricultural commodity.

- **Variety:** Specific variety of the commodity.
- **Grade:** Quality of the commodity.
- **Arrival date:** Date when the commodity arrives the market.
- **Max price:** Maximum price recorded for the commodity.
- **Min price:** Minimum price recorded for the commodity.
- **Modal price:** Most frequently recorded price for the commodity.
- **Commodity code:** Unique numeric identifier for the commodity.

	State	District	Market	Commodity	Variety	Grade	Arrival Date	Min Price	Max Price	Modal Price	Commodity Code	Category
0	Gujarat	Ahmedabad	Ahmedabad	Bitter gourd	Other	FAQ	2006-01-03	600.0	1300.0	1100.0	81	Gourds
1	Gujarat	Ahmedabad	Ahmedabad	Bitter gourd	Other	FAQ	2006-01-05	1100.0	1400.0	1300.0	81	Gourds
2	Gujarat	Ahmedabad	Ahmedabad	Bitter gourd	Other	FAQ	2006-01-16	1500.0	1800.0	1800.0	81	Gourds
3	Gujarat	Ahmedabad	Ahmedabad	Bitter gourd	Other	FAQ	2006-01-19	700.0	900.0	800.0	81	Gourds
4	Gujarat	Ahmedabad	Ahmedabad	Bitter gourd	Other	FAQ	2006-01-24	500.0	1400.0	1000.0	81	Gourds

Figure 1.1: dataset

Most of the fields of the dataset are categorical (string \object type) except:

- **Min.Price,Max.Price,andModal.Price:** Float values
- **Commodity.Code:**Integer Values

```
data_grouped['Category'].unique()

array(['Gourds', 'Leafy Greens', 'Pods/Beans', 'Uncategorized',
       'Other Vegetables', 'Alliums', 'Root Vegetables', 'Tubers',
       'Spices', 'Chillies and Peppers', 'Fruits',
       'Other Agricultural Products', 'Grains', 'Pulses', 'Oilseeds',
       'Seeds'], dtype=object)

data_grouped['Grade'].unique()

array(['FAQ', 'Small', 'Medium', 'Large'], dtype=object)

data_grouped['Variety'].unique()

array(['Other', 'Kufri Giriraj', 'Capsicum', 'Bold', 'Coriander', 'Chips',
       'Desi', 'Jalander', 'Local', 'Nasik', 'Hapus(Alphaso)', 'Keshar',
       'Deshi', 'Milbar', 'White', 'Basumathi', 'I.R. 8', 'Masuri',
       'Castor seed', 'Assam Comilla', 'Shanker 6 (B) 30mm Fine',
       'V-797 22mm Fine', 'Black', 'Shanker 4 31mm Fine', 'Lokwan Gujrat',
       'G. R. 11', 'Gram Raw(Chholia)', 'Brinjal', 'Seethaphal',
       'Cucumbar', 'Sapota', 'Onion Green', 'Water Melon',
       'Banana - Green', 'Mint(Pudina)', 'Arhar (Whole)', 'Moath (W)',
       'Big 100 Kg', '777 New Ind', 'Lokwan', 'Moath Dal', 'Kalyan',
       'Mustard', 'Cummin Seed(Jeera)', 'H.D.', '147 Average',
       'Arhar Dal(Tur)', 'Sharbati', 'Paddy', 'Dehradun', 'Dara',
       'A-51-9 24mm. Fine'], dtype=object)
```

Figure 1.2: grouping dataset

```
data_cleaned['Commodity'].value_counts()

Commodity
Potato      14175
Onion       9700
Wheat       8756
Tomato      7855
Brinjal     7597
...
Bajra(Pearl Millet/Cumbu)    658
Seetapal                     652
Mustard                      606
Sesamum(Sesame,Gingelly,Til) 566
Jowar(Sorghum)               514
Name: count, Length: 75, dtype: int64
```

Figure 1.3:

## 1.4 Packages required

- **numpy:** Provide many functions for working with arrays, matrices and linear algebra.
- **pandas:** Read data from CSV \ excelfiles, manipulate data and generate insights from it. It is also used to clean, filter and visualize data.
- **matplotlib:** Used to create static, animated and interactive visualizations.
- **seaborn:** Provide a high-level interface for creating statistical graphics.
- **scikit-learn:** For machine learning and statistical modelling tasks, such as classification, regression, and clustering.
- **networkx:** Used to model data in the form of graphs.
- **plotly:** Used to make interactive graphs such as scatter plots, bar charts, box plots, etc.

# Chapter 2. Data Cleaning

Data cleaning is a crucial step of data pre-processing, aiming to ensure that the dataset is free from errors, inconsistency, and missing values. This step is essential to obtain reliable and accurate results in subsequent analysis. For the dataset used, it will be useful to ensure the integrity of commodity prices, market information and dates.

## 2.1 Missing data analysis

- The dataset was evaluated for missing values using `data.isnull().sum()`. As there are no missing data points present imputation is not needed.
- Identifying and removing the duplicate records using `df.drop_duplicates()`.

### 2.1.1 Filtering and Categorization

- The dataset contains numerous unique commodities with very low frequency. To streamline the analysis, commodities with less than 500 entries were removed.
- The remaining commodities are grouped into various categories based on a predefined mapping directory, to enable better aggregation and interpretation of results.

### 2.1.2 Encoding categorical Variables

- `LabelEncoder()` is applied to all object-type columns to convert them into a numeric representation.

### 2.1.3 Outlier detection and Removal

- Using `seaborn.boxplot()` all the outliers are identified in numeric columns.
- Using the IQR method outliers are removed to ensure data is free from extreme values

```
1 def iqr_clipping(df, columns, multiplier=1.5):
2     df_clipped = df.copy()
3     for col in columns:
4         Q1 = df[col].quantile(0.25)
5         Q3 = df[col].quantile(0.75)
6         IQR = Q3 - Q1
7         lower_bound = Q1 - multiplier * IQR
8         upper_bound = Q3 + multiplier * IQR
9         df_clipped[col] = df[col].clip(lower=lower_bound, upper
            =upper_bound)
10    return df_clipped
11
12 outlier_cols = ['Market', 'Variety', 'Grade', 'Min_Price', 'Max_Price'
13                , 'Modal_Price', 'Commodity_Code']
14 df_clipped = iqr_clipping(df, outlier_cols, multiplier=1.5)
```

Figure 2.1: outlier code

### 2.1.4 Date Column transformation

- The "Arrival\_Date" column was split into "Arrival\_Day", "Arrival\_Month", "Arrival\_Year", temporarily to improve analysis. The original "Arrival\_Date" column is later dropped post-transformation as it became redundant.

### 2.1.5 Variance and Correlation Analysis

- We check for the Variance in the data. The columns "District", "State", "Variety" and "Grade" had 0 variance and 0 correlation, meaning their value for all the rows was the same. So we dropped these columns.
- We also dropped the "Arrival\_Date" column as it had been split into three columns and also its datatype is not computable.

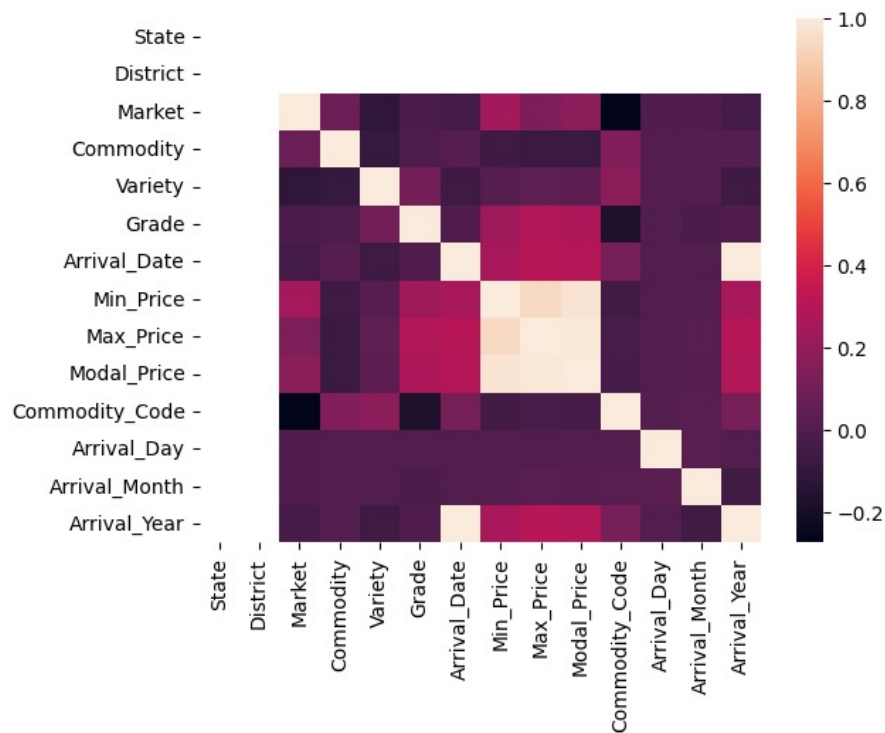


Figure 2.2: correlation matrix

## 2.2 Imputation

- The dataset contains no missing values so imputation techniques like mean substitution, median replacement or predictive modelling are unnecessary.

## Chapter 3. Visualization

Given the dataset, we had agricultural produce of various kinds, i.e., seeds like sunflower seeds, vegetable oils, fruits like pumpkin, leafy greens like green onions, vegetables, etc. We decided to categorize them into various categories as follows:

Table 3.1: Categorization of Commodities - Part 1

Category	Commodities
Leafy Greens	Coriander (Leaves), Methi (Leaves), Amaranthus, Spinach, Mint (Pudina), Leafy Vegetable, Onion Green
Root Vegetables	Carrot, Radish, Sweet Potato, Beetroot, Yam (Ratalu), Elephant Yam (Suran), Turnip
Gourds	Bitter Gourd, Bottle Gourd, Sponge Gourd, Little Gourd (Kundru), Snakeguard, Pointed Gourd (Parval), Round Gourd (Tinda), Pumpkin
Pods/Beans	Surat Beans (Papadi), French Beans (Frasbean), Cluster Beans, Peas Wet, Cowpea (Veg), Guar, Pigeon Pea (Arhar Fali)
Alliums	Onion, Onion Green
Tubers	Potato, Colacasia
Chillies and Peppers	Chilly Capsicum, Green Chilli, Capsicum

The value for the agricultural produce is given as follows:



Table 3.2: List of Commodities

Commodity 1	Commodity 2	Commodity 3
Other	Kufri Giriraj	Capsicum
Bold	Coriander	Chips
Desi	Jalander	Local
Nasik	Hapus (Alphaso)	Keshar
Deshi	Milbar	White
Basumathi	I.R. 8	Masuri
Castor Seed	Assam Comilla	Shanker 6 (B) 30mm Fine
V-797 22mm Fine	Black	Shanker 4 31mm Fine
Lokwan Gujrat	G. R. 11	Gram Raw (Chholia)
Brinjal	Seethaphal	Cucumbar
Sapota	Onion Green	Water Melon
Banana - Green	Mint (Pudina)	Arhar (Whole)
Moath (W)	Big 100 Kg	777 New Ind
Lokwan	Moath Dal	Kalyan
Mustard	Cummin Seed (Jeera)	H.D.
147 Average	Arhar Dal (Tur)	Sharbati
Paddy	Dehradun	Dara
A-51-9 24mm Fine		

Before beginning any sort of data visualization, setting expectations with the data and working with the data is very crucial. Almost all classical and deep learning techniques require that the underlying distribution of the data is normal or approximately normal.

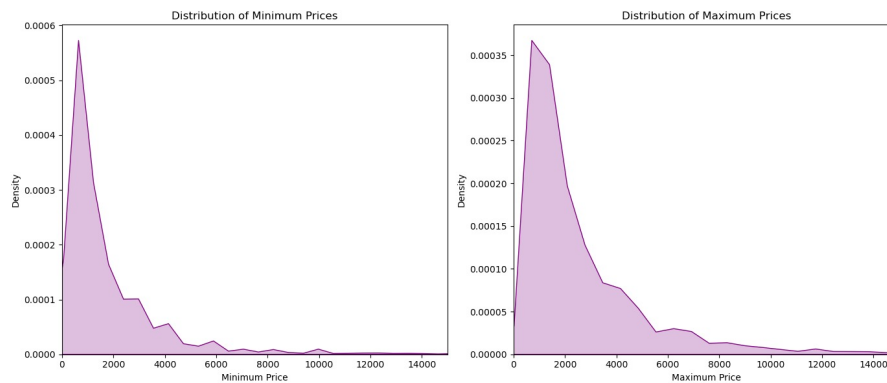


Figure 3.1:

Both the minimum and maximum prices follow a positively skewed normal distribution which means that there are no extreme transformations needed to deal with the data. We can simply shift the scale and the skewness will be dealt with.

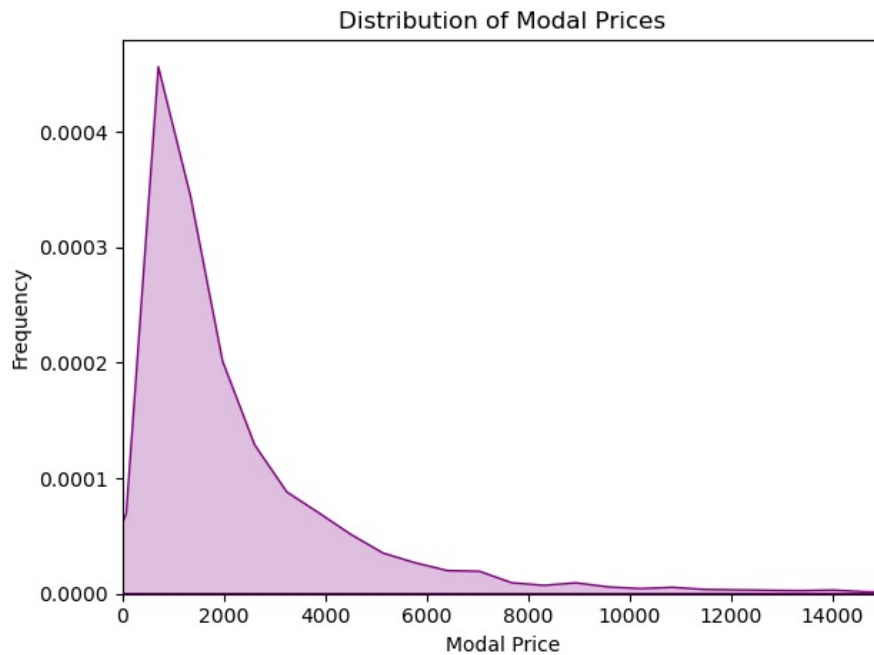


Figure 3.2:

The modal price variable has a right-skewed distribution; hence, it needs some normalization. Due to its inherent nature of extreme values, transformations like log transformations also work well.

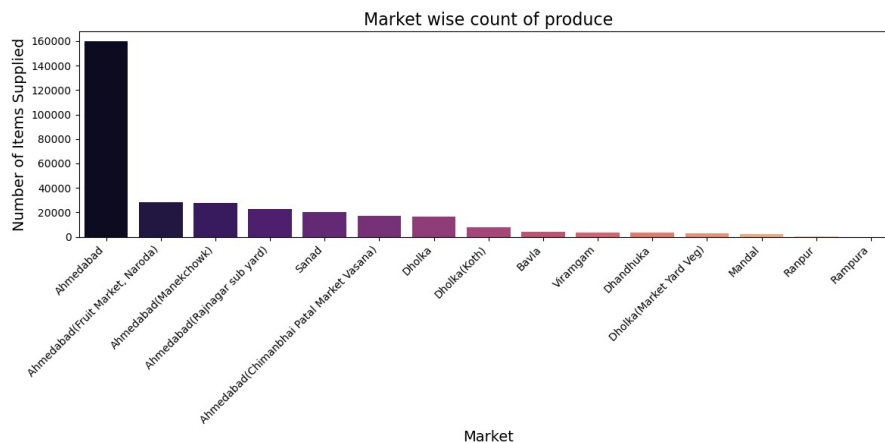


Figure 3.3:

The above plot shows the market-wise supply for agricultural products and unsurprisingly, the main APMC market is responsible for supplying most of the agricultural produce to the city of Ahmedabad, whereas nearby markets supply things in trace quantities which suggests proximity being an important factor when it comes to supplying fresh produce.

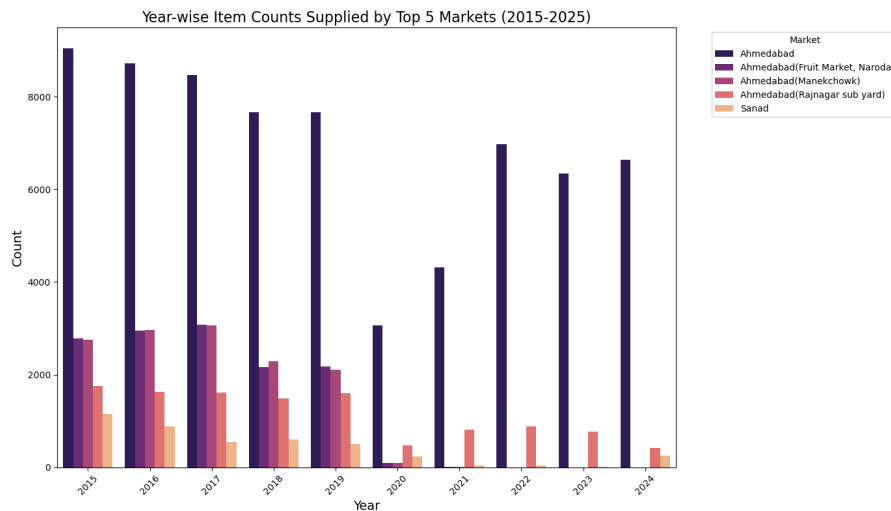


Figure 3.4:

The above graph shows the supply of top 5 markets for a 10 year period and we can see a steady decline in the supplies for some markets and the net overall supply across all the markets as well

Category Distribution

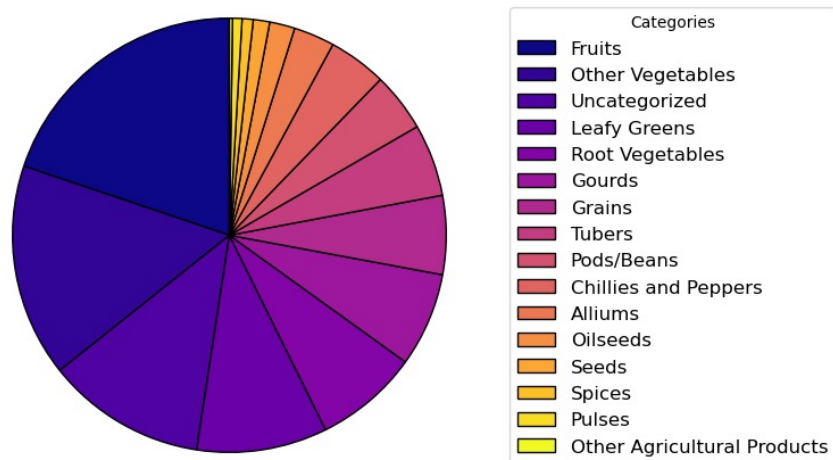


Figure 3.5:

The above figure shows the distribution of counts of agricultural produce across all the markets for all the years, and as you can see, fruits, leafy greens, and root vegetables make up the majority of the supply from all the markets. Which makes sense since we need them the most for cooking and survival

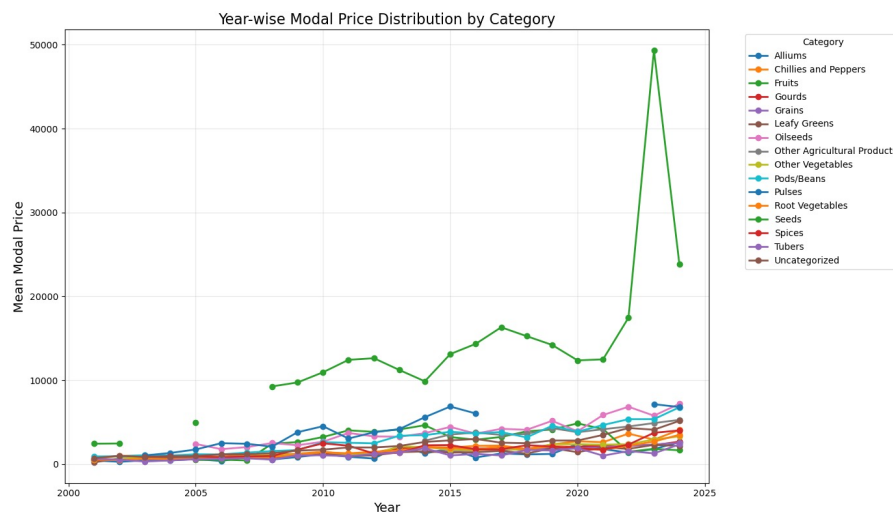


Figure 3.6:

The below graph shows variations for prices across various categories mentioned above for every five years. Since there are a lot of categories, it would make sense to

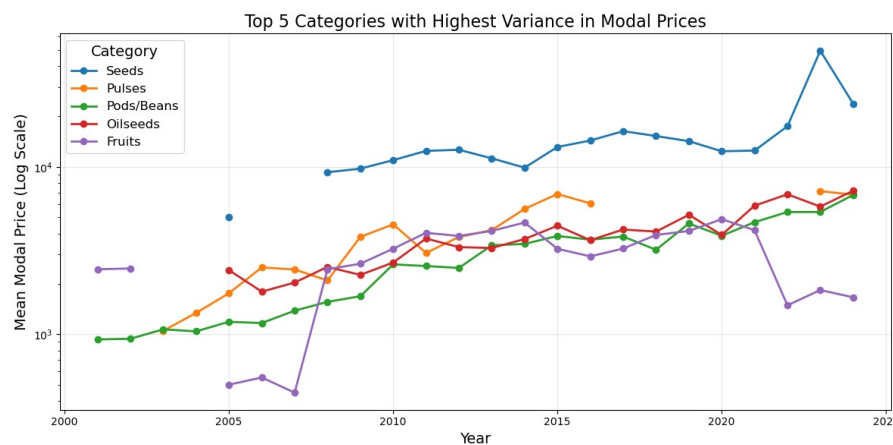


Figure 3.7:

As it can be seen from the above graph, seeds, pulses, fruits, and oilseeds are the produce that have had the most fluctuations in prices across a 25-year period, which we can attribute to the cultural and political trends over the years

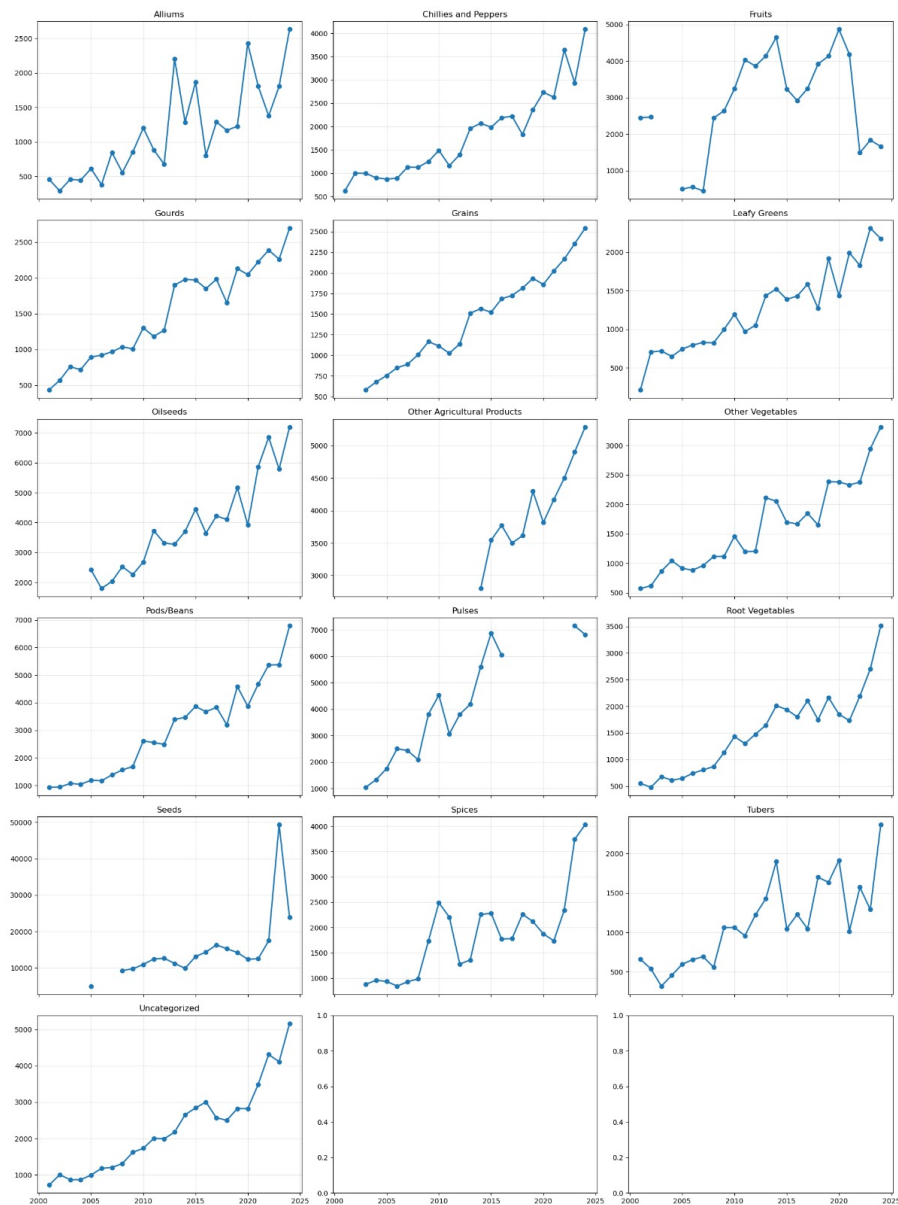


Figure 3.8:

One of the biggest questions that we have had with our data was how the prices fluctuate year-wise and if there is a correlation between different seasons of the year(s) and the prices of the crops

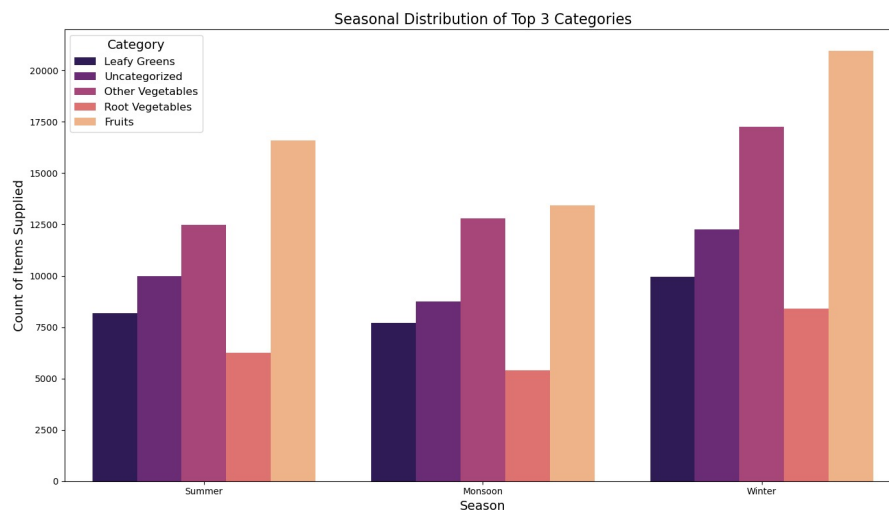


Figure 3.9:

The above graph shows that the supply of vegetables increases in winters as compared to the other seasons. In general, there seems to be a positive correlation between the supply of the produce from market and winter season, whereas no such trend can be interpreted for monsoon or summer

# Chapter 4. Feature Engineering

Feature engineering is an important data-preprocessing step for predictive modelling and thus an important part of improving the performance of a model by transformation and enrichment of a dataset. In this research, numerous feature engineering techniques have been adopted to refine the data with meaningful features for machine learning models.

## 4.1 Feature extraction

### 4.1.1 Date Handling

- It has a column, 'Arrival\_Date', which is a string so converted to a date-time type for further computation. Another feature that has been developed involves three: Arrival\_Day, Arrival\_Month, and Arrival\_Year. The three include some temporal patterns in the data set, seasonal and yearly that tend to drive prices of commodities.
- Academic Insight: Time-related features work pretty well to reproduce trend and seasonality, which occurs quite frequently to be highly significant in most time-series models, such as commodity price prediction.

### 4.1.2 Categorical Encoding

- The categorical variables in our dataset were encoded in the number format by using label encoding for Variety, Grade, State, and 'District'. In label encoding, the categorical gets translated into a unique integer; however, it retains intrinsic relationships between.
- Academic Insight: Encoding categorical data as numbers ensures that the data will work with machine learning algorithms that cannot process non-numeric inputs.

### 4.1.3 Outlier Detection and Treatment

- To handle the outliers, boxplots for the quantitative variables were created to represent their distribution. The IQR method was implemented on the key columns, which are Market, Variety, Grade, Min\_Price, Max\_Price, Modal\_Price, and Commodity\_Code, to eliminate outliers.
- The missing values in the columns with outliers were filled by the calculated values within the following limits:

$$Lower\ Bound = Q1 - (1.5 * IQR)$$

$$Upper\ Bound = Q3 + (1.5 * IQR)$$

- Academic Insight: the management of the outliers guarantees extreme values will not skew the model, thereby leading to training instability and suboptimal prediction.

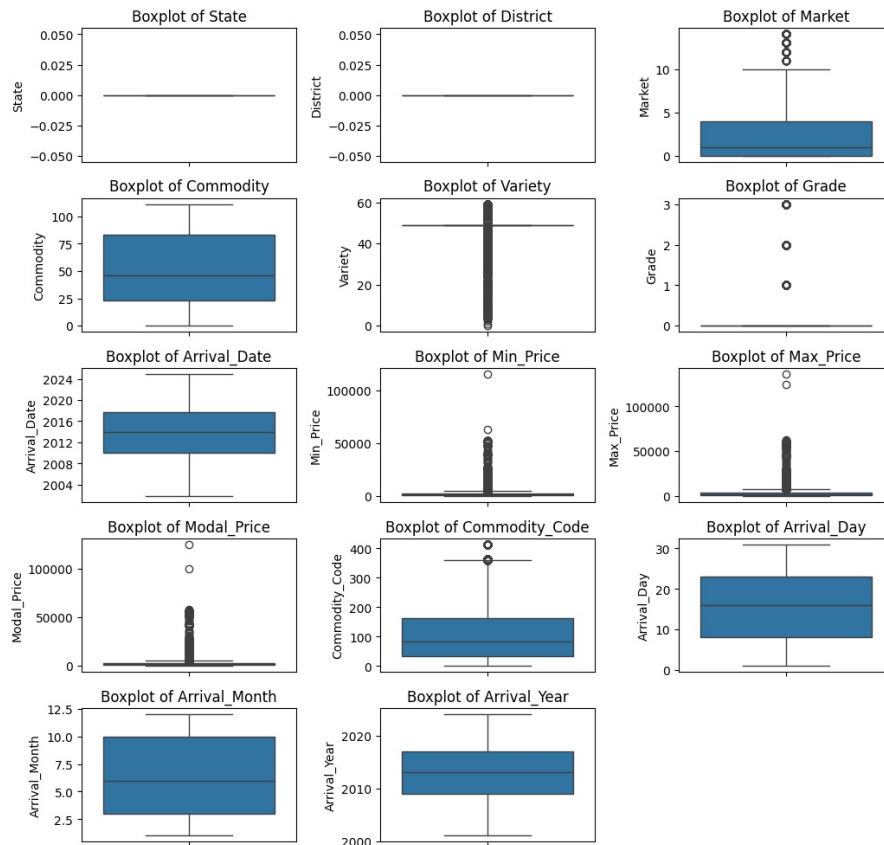


Figure 4.1: Before Removing Outliers



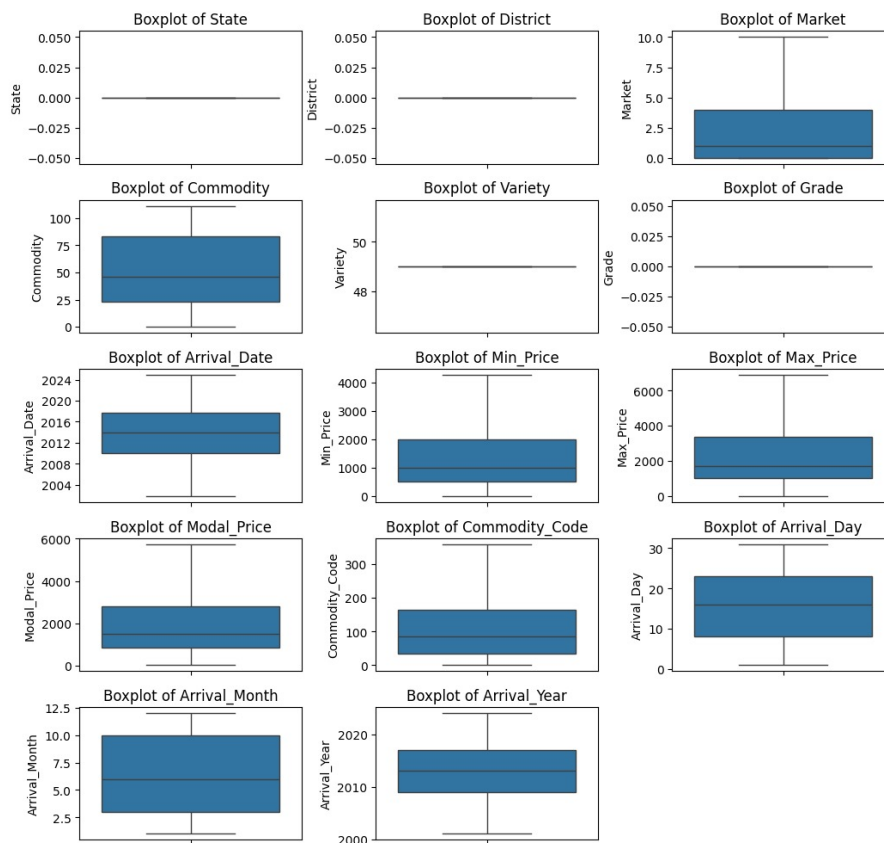


Figure 4.2: After Removing Outliers

## 4.2 Feature selection

The process was much variance- and correlation-based; features selected through variance are:

- **Variance-based Analysis:** It mainly identifies features that vary more and which consequently contributed much to the model more often with significant changes.
- **Correlation analysis** was carried out on individual features to establish the validity of their predictions on the target variable, ('Modal\_Price',
- **Technical Insight:** Features chosen with high variance and those highly correlated eliminate noise in this process and only guarantee the predictors that are most influential to be in the dataset.
- **Applying PCA for Dimensionality Reduction** The third refinement of the dataset was on redundancy removal using PCA.
  1. All the features standardized by StandardScaler as PCA is sensitive to feature magnitude and scales the data uniformly.
  2. The data is transformed into a set of principal components using PCA, the number of components to be taken to retain 90
  3. That reduced the dimensionality datasets so highly but retained all information.

```

1 scaler = StandardScaler()
2 df_scaled = scaler.fit_transform(df_edited)
3
4 pca = PCA()
5 pca_result = pca.fit_transform(df_scaled)
6
7 pca_df = pd.DataFrame(pca_result, columns=[f'PC{i+1}' for i in range
8                               (pca_result.shape[1])])
9
10 explained_variance_df = pd.DataFrame({
11     'Principal Component': [f'PC{i+1}' for i in range(len(pca
12     .explained_variance_ratio_))],
13     'Explained Variance Ratio': pca.explained_variance_ratio_
14 })
15
16 cumulative_variance = np.cumsum(pca.explained_variance_ratio_)
17 threshold = 0.90
18 n_components = np.argmax(cumulative_variance >= threshold) + 1

```

Figure 4.3:

- Academic Insight: It does not only reduce computational complexity but also has the added benefit of overfitting risk minimization by excluding irrelevant or redundant features.

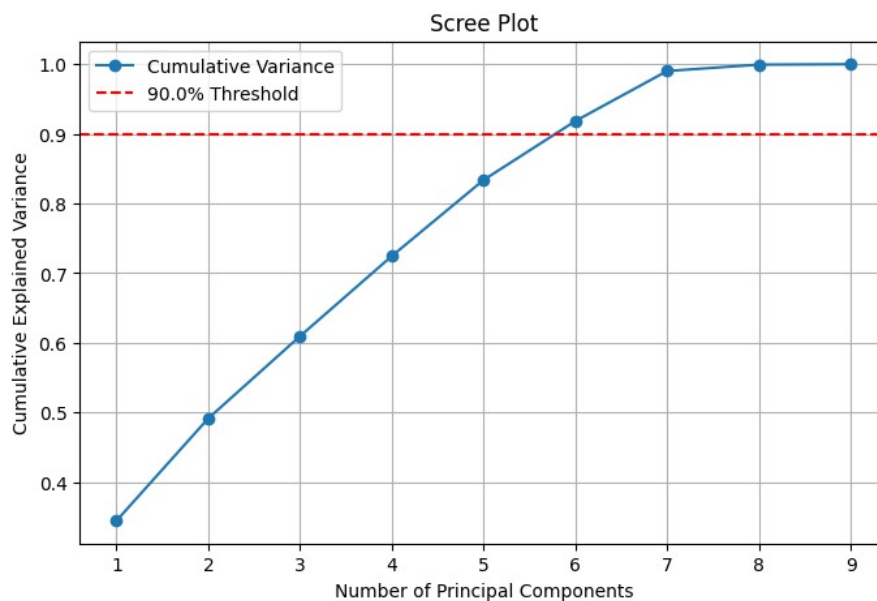


Figure 4.4: Scree Plot for the Principal Components

### 4.3 Feature Dropping

- Some columns like Arrival\_Date, Variety, Grade, State, and District have been dropped off the dataframe after feature extraction and transformation. The 'Arrival\_Date' column has been split up into 3 other columns, and it is not computable as a datatype. The other columns had 0

variance and 0 correlation about the target column "Modal\_Price". Hence, they were holding the same value for all the rows.

- Academic Insight Elimination of non-essential features helps clean the dataset, and focus the model on important predictors, making it more efficient and easier to interpret.

# Chapter 5. Model fitting

After cleaning the data and exiting the features the 6 ML models could predict the target variable known as 'Modal\_Price'. These regression algorithms indicated their predictive competency for which their performance was assessed.

- Training and Testing Split The proportion used was 80:20. The "Features\_train" and the "Labels\_Train" set were used for training the models. "Features\_Test" was used to predict the answers, which were stored in "pred". "Labels\_test" was compared to "pred" to obtain the evaluation metrics.
- Academic Insight: Fragmenting of the data set helps in the validation of the model on an unutilized data set, which measures the validity and usefulness of the model.

## 5.1 Regression

In order to compare the performance of these models, the following metrics were used for assessment:

- Mean Squared Error (MSE): It finds out the mean of the square of the difference between the predicted value and the actual value.
- R<sup>2</sup> Score: It determines the proportion of variance in the dependent variable that can be explained by the independent variable.
- Cross-Validation: To assess the generalization ability, 10-fold cross-validation was used to reduce variability due to data splitting.
- Residual Analysis: It was used to visualize the distribution of the difference between the predicted value and the actual value, also known as Residue.

### 5.1.1 Detection of Overfitting

To check whether the model is overfitting, the following conditions were used:

If Training

$$R > (TestingR + 0.05)$$

or if Training

$$R > 0.9$$

and Testing

$$R < 0.8$$

, then it is considered as overfitting.

- Academic Insight: Overfitting is the main issue that affects the practicality of the model. So, it must be identified and controlled.

### 5.1.2 Visualizations

For each model, two key plots were generated:

- Actual vs. Predicted Values: It plots a comparison between "Labels\_Test" and "Pred". Ideally, it should be a linear slope.
- Residuals Analysis: It plots a comparison between the Residue and "Pred". Residuals, ideally should be zero, as there should not be any difference between the actual value and the predicted value.

### Results and Comparisons

- Basic Linear Regression: It was a baseline model. It worked very well, even though the data was non-linear. It is still not very reliable in these types of datasets. took 1 second to run.
- Decision Tree: It achieved 100% training accuracy, meaning it was kind of overfitted. But it still achieved impressive testing accuracy. It took 42 seconds to run.
- Random Forest: It worked very well in terms of both accuracy and robustness. It achieved very high  $R^2$  scores with no overfitting. But, it took 44 minutes to run.
- Gradient Boosting: It achieved high accuracy. And the cross-validation scores were also very consistent. But it took 14 minutes to run.
- XGBoost: It worked really well in terms of accuracy and speed. It took 6.7 seconds to run.
- LightGBM: It achieved performance similar to XGBoost. It took 6.6 seconds to run.

## 5.2 ML algorithms

6 ML Models have been used. Each of the models comes with an added advantage, consequently performing an analysis of all aspects of exploitable predictability.

### 1. Multiple Linear Regression

Here, a linear line is used to estimate the level of dependency between features and the target variable. It is the simplest regression model.

- Academic Insight: Linear regression has a constraint of assuming only linear relationships. So, it might not work well with non-linear data.

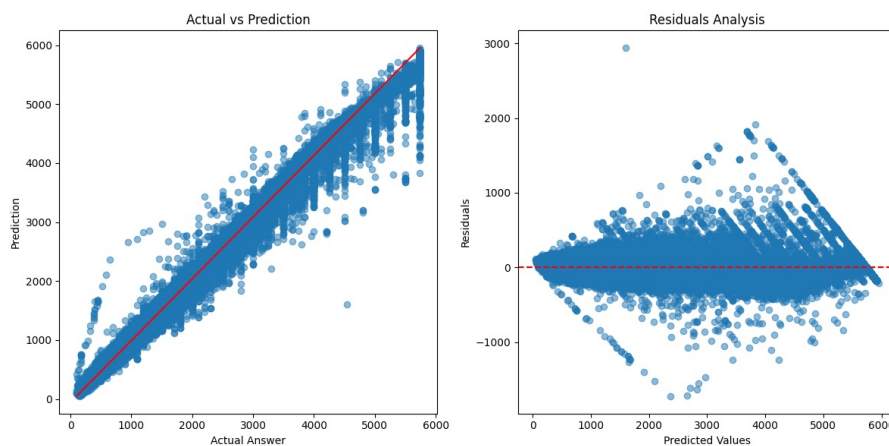


Figure 5.1: Multiple Linear Regression

## 2. Decision Tree Regressor

It identifies interactions and dependencies in the features that are not linear. It takes a tree-like form, splitting the data at nodes according to the values of the features.

- Academic Insight: While decision trees are interpretable and flexible, they overfit easily.

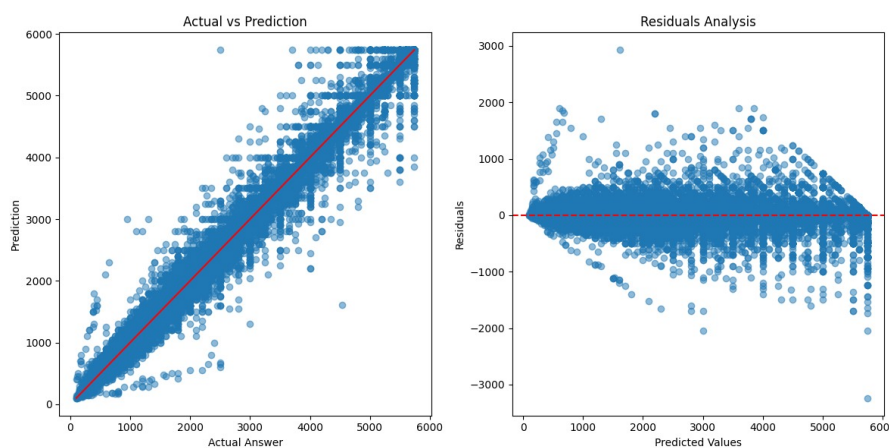


Figure 5.2: Decision Tree Regressor

## 3. Random Forest Regressor

It reduces the defects of the decision trees regressor. Here, the results of several decision trees are combined to increase the strength and minimize overfitting.

- Academic Insight: It is a good estimator of the nonlinear relationships and, with lower sensitivity to the noise and outliers, it does make a great choice in case of real data. But, it takes a very long time to compute the answers.

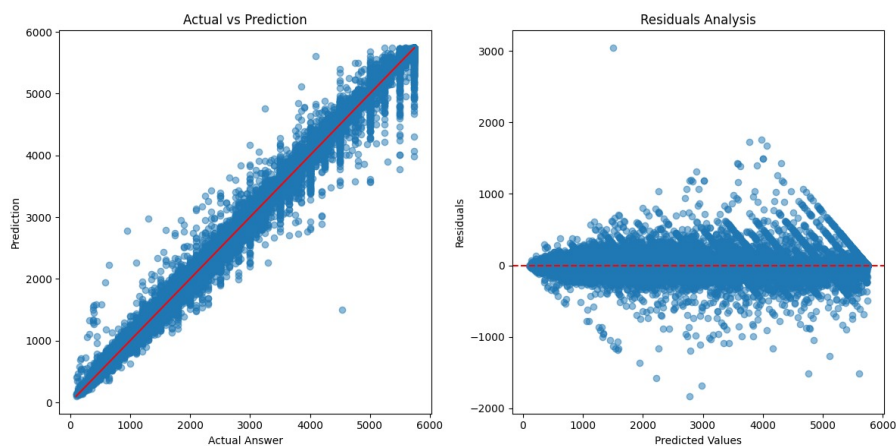


Figure 5.3: Random Forest Regressor

#### 4. Gradient Boosting Regressor

It constructs trees sequentially. It is aimed at the difficult instances of previous trees. Often, such an iterative process yields very good accuracy.

- Academic Insight: Gradient Boosting works great for complex patterns but tends to require hyperparameter tuning to find the balance between performance and computational efficiency.

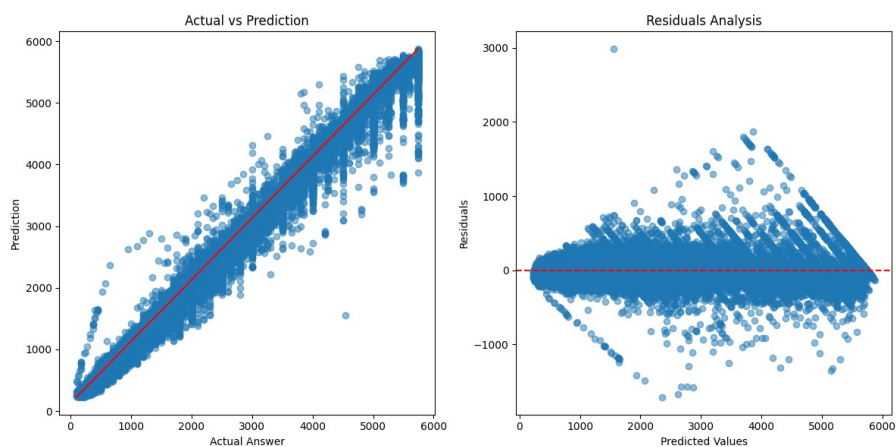


Figure 5.4: Gradient Boosting Regressor

#### 5. XGBoost Regressor

It is an optimized version of gradient boosting that is known for speed and accuracy. Its ability to do parallel processing is useful for large datasets.

- Academic Insight: XGBoost is known for scalability and ability to handle non-linear relationships well, often outperforming traditional models when working on structured data.

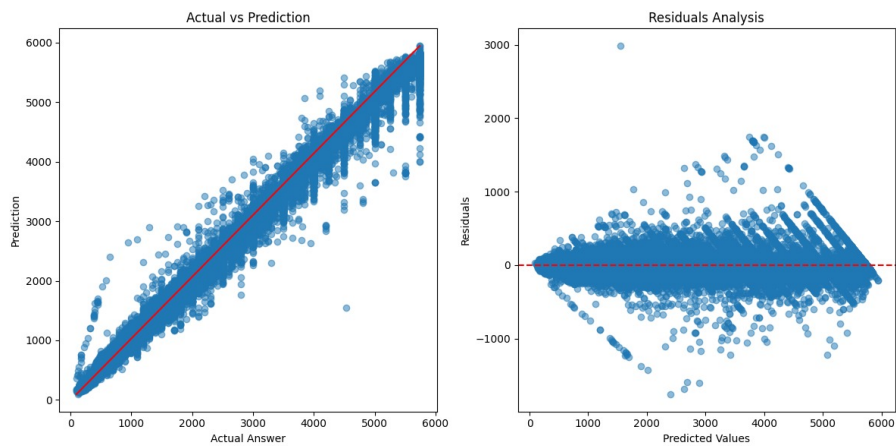


Figure 5.5: XGBoost Regressor

#### 6. LightGBM Regressor

LightGBM is a gradient-boosting algorithm which is unique and has been used to leverage histogram-based efficiency. It is designed to be effective with large datasets and reduces training time significantly.

- Academic Insight: LightGBM controls the computational cost and gives high accuracy, making it ideal for large data sets.

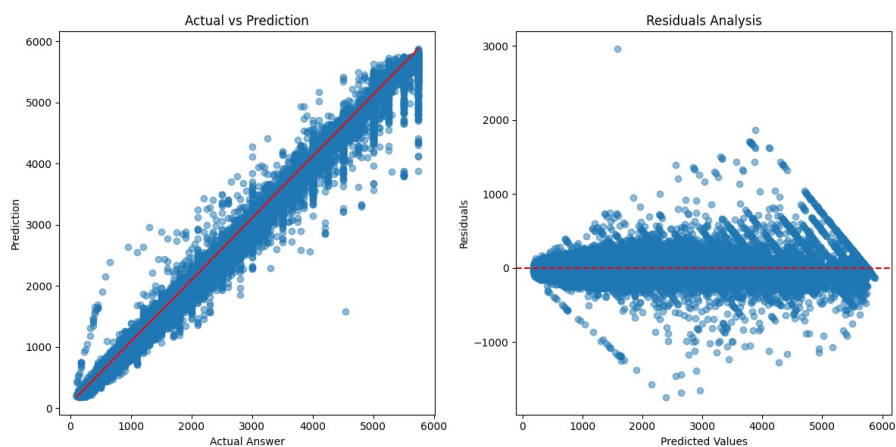


Figure 5.6: LightGBM Regressor



# Chapter 6. Conclusion & future scope

The current study has adequately analyzed the dataset with numerous predictive models, mainly XGBoost regression, that generated a tremendous frame to understand and predict the prices of commodities. Advanced machine learning methods coupled with pre-processing of data have been applied for extracting precious information from sets of data and resulting in an acceptable performance of models.

## 6.1 Findings/observations

### 6.1.1 Model Performance:

- Models tried: Linear Regression, Decision Tree Regressor, Random Forest Regressor, Gradient Boost Regressor, Light Gradient Boosting Machine (LightGBM), and XGBoost Regressor.
- Among all the models tested above, XGBoost has the best predictive power as per the metrics. It shows stable performance on both the training set and testing set with no overfitting signs. Random Forest and Gradient Boosting are almost at par stating how strong these ensemble-based approaches are to result in stable and accurate predictions of Feature Engineering and PCA.
- Feature extraction and dimensionality reduction by PCA retained more than 90% of variance in the data using fewer components. Hence, this proved to be a very effective technique that retained computational efficiency with the preserved quality of prediction.
- Important preprocessing steps like the removal of outliers, standardizing data, and correlation analysis have helped in the development of robust models and actionable insights.

### 6.1.2 Comparative Model Evaluation:

- Linear Regression and Decision Tree models are not that robust on noise and outliers, thereby relatively low performance compared to the ensemble-based methods.
- LightGBM was robust on the smaller dataset but outperformed where it is comparable to tree-based ensemble models in forming complex patterns.

### 6.1.3 Data Insights:

- All steps of preprocessing, including outlier removal and scaling, are important in making all the models more reliable.

- Correlation analysis found that it was the most significant predictor variables, thus adding more value to the stakeholders for strategic commodity price forecasting.

## 6.2 Challenges

## 6.3 Future plan

- **External Variables:** The future study may include additional external variables of weather conditions, transport cost, changes in policies, and trends of consumers' demand. Such an inclusion of variables will bring deeper contextual insights that, in turn, help enhance the predictability potential of the models.
- **Advanced Time Series Analysis:** Models like ARIMA, LSTM, and Prophet, which are particularly tailored for time-series data, will be used to capture the temporal behavior-including seasonality and long-term trends.
- **Hyperparameter Optimization:** The hyperparameters of models such as XGBoost, Random Forest, or LightGBM are fine-tuned for an exponential improvement through methodologies such as GridSearchCV, RandomizedSearchCV, or Bayesian optimization.
- **Ensemble Modeling:** The hybrid ensemble methods introduce the strength of tree-based methods, SVR, and neural networks into models, which may improve generalization with minimal bias and variance and thus result in better predictions.
- **Real-Time Applications:** The creation of a real-time prediction system that is continually being updated with data streaming may endow stakeholders with timely and well-informed decisions under ever-changing market conditions.
- Adding SHAP (SHapley Additive exPlanations) and LIME or Local Interpretable Model-agnostic Explanations to the AI model can improve the explanatory power, thus enabling stakeholders to receive something actionable and believe in their results.
- **Scalability and Deployment:** Models deployed in real production settings through integration into web-based platforms or dashboards ensure scalability and ease of access to end users.
- Establish periodic and continuous model monitoring and improvement through the development of continuous updates for model performance over time and new data introduced for further retraining.

# Group Contribution

## Shreshtha Modi

Data collection, EDA, model fitting, Slides

## Khushi Prajapati

Data collection, EDA, model fitting, Slides

## Jinay Vora

Data collection, EDA, model fitting, Slides

## Short Bio

1. **Shreshtha Modi** is a first-year Masters in Machine Learning student interested in open source and using machine learning for social good. She loves learning about new technologies and is interested in diving deeper and understanding the mathematics behind machine learning

2. **Khushi Prajapati** is a 3rd year Btech ICT student interested in understanding how algorithms that run the world around us work. She is keen

to explore algorithms related to machine learning and data science

3. **Jinay Vora** is a 3rd Year Btech ICT student at DAIICT interested in exploring recent trends in Artificial intelligence and computer science. He loves to practice programming questions and understanding the reasoning behind why things work

# References

- [1] Classroom Discussions and Notes
- [2] data.gov.in  
URL: [variety-wise-daily-market-prices-data-commodity](#)
- [3] Chen, Tianqi and Guestrin, Carlos *XGBoost: A Scalable Tree Boosting System*.  
URL: [XGBoost](#)  
Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.
- [4] ibef.org  
URL: [ibef.org](#)
- [5] isec.ac.in  
URL: [isec.ac.in](#)
- [6] sciencedirect.com  
URL: [sciencedirect.com](#)
- [7] Pandas Library  
URL: [pandas](#)
- [8] Supervised Learning  
URL: [scikit-learn](#)