AGRICULTURE COMMODITIES PRICE PREDICTION AND FORECASTING

ABSTRACT

Recent days interaction between computer and human is gaining more popularity or momentum, especially in the area of speech recognition. There are many speech recognition systems or applications got developed such as, Amazon Alexa, Cortana, Siri etc. To provide the human like responses, Natural Language Processing techniques such as Natural Language Toolkit for Python can be used for analyzing speech, and responses. In our country, INDIA, agriculture is backbone of economy and major contributor for GDP. However, farmers often, do not get sufficient support or required information in the regional languages. Prediction analysis for farmers in agriculture is not only for crop growing but is essential to develop Crop recommendation system based on price forecasting for agricultural commodities in addition to providing useful advisories for the farmers of any state. Currently, to protect the farmers from price crash or control the inflation, the governments (Central and State) predicting the price for agricultural commodities using short-term arrivals and historical data. However, these methods are not giving enough recommendations for the farmers to decide the storage/sales options with evidence-based explanations. This project implements machine learning algorithms such as multi linear regression, Random Forest and Decision Tree regressor. To achieve the commodity price, by analyze the r2 score. The highest r2 score, the best model could be selected.

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Technological advancements have fundamentally altered how humans interact with computers, particularly in the field of speech recognition. Applications like Amazon Alexa, Microsoft Cortana, and Apple Siri demonstrate how speech recognition technology can provide consumers with realistic, human-like responses. These systems use Natural Language Processing (NLP) techniques, such as the Natural Language Toolkit (NLTK) for Python, to analyze voice and support meaningful communications between people and gadgets. In a country like India, where agriculture is the backbone of the economy and makes a considerable contribution to GDP, technology advancements have the potential to play a critical role in assisting farmers. Farmers, on the other hand, frequently struggle to obtain critical information and recommendations in their own language, limiting their ability to make informed decisions. Such difficulties underscore the need for technology-driven solutions to close the information gap and empower farmers to improve their agricultural methods. Prediction analysis in agriculture goes beyond crop cultivation to include crop recommendation systems based on price predictions for agricultural commodities. These systems serve as vital advisories for farmers in many states, assisting them in making key crop selection, storage, and sales decisions. Traditionally, government agencies (both central and state) used short-term data on arrivals and historical trends to forecast agricultural commodity prices. Although these methods offer some insights, they frequently fall short of providing full suggestions that allow farmers to make evidence-based decisions about when

and how to store or sell their product. The purpose of this study is to review existing research on agricultural prediction models and highlight the strengths and limits of various techniques. By assessing the current landscape, the study hopes to find areas for improvement and future research directions to increase the performance of crop recommendation systems and price forecasting tools.

1.2 CHALLENGES

The difficulties you've mentioned represent an exciting opportunity to use technology to empower farmers and enhance agricultural practices. Here's a breakdown of possible ways and considerations for tackling these issues:

Language Accessibility: Creating speech recognition and natural language processing systems in regional languages can considerably improve accessibility for farmers who are not fluent in English. To effectively interpret and reply to requests in several languages, robust language models must be trained.

Price Forecasting: Advanced predictive analytics and machine learning algorithms can help improve the accuracy of agricultural commodity price forecasts. Market demand, supply chain dynamics, weather patterns, and government regulations can all be factored into predictions to improve their accuracy.

Crop Recommendation System: A crop recommendation system must consider a variety of criteria, including soil quality, climate conditions, market demand, and historical production data. Machine learning techniques can be used to create individualized suggestions for farmers based on their unique conditions and aims.

Advisory Services: Providing timely and relevant advisories to farmers necessitates real-time data collection and processing. Weather predictions, pest and disease outbreak alerts, and market trends can be integrated into advisory systems to assist farmers in making informed crop management, pest control, irrigation scheduling, and marketing decisions.

Evidence-Based Explanations: To increase openness and trust in pricing forecasting and advising systems, recommendations must be supported by evidence. This entails presenting not only the expected results, but also the underlying data, assumptions, and reasoning for the suggestions. Farmers can use interactive visualization tools to better understand and analyze information.

Government Support: Collaboration among government agencies, academic institutions, and technology businesses is critical for creating and implementing creative solutions to help farmers. Governments may play an important role in sponsoring research, giving access to data and infrastructure, and enacting regulations to encourage the use of technology in agriculture.

Continuous Improvement: The creation of agricultural technology solutions is a continual process that necessitates constant monitoring, review, and improvement. Obtaining feedback from farmers, agricultural experts, and stakeholders is critical for finding areas for improvement and iterating on existing systems to better meet end-user demands.

By addressing these issues through interdisciplinary collaboration and innovation, we can harness the power of technology to alter agriculture and enhance farmers' livelihoods in India and elsewhere.

1.3 MOTIVATION OF THE WORK

The motivation behind this work stems from the recognition of the critical role that agriculture plays in India's economy and the challenges faced by farmers in accessing timely and relevant information, particularly in their regional languages. Despite the significant contributions of agriculture to the GDP, many farmers struggle to make informed decisions due to the lack of support and information available to them.

- The increasing popularity and advancements in speech recognition and natural language processing technologies offer a promising avenue for addressing these challenges. By leveraging these technologies, it becomes possible to develop innovative solutions such as crop recommendation systems and advisory services tailored to the specific needs of farmers.
- The primary motivation of this study is to bridge the gap between technological advancements and the agricultural sector by exploring the potential of predictive analysis and machine learning in providing actionable insights to farmers. By integrating price forecasting models with evidence-based explanations, farmers can make more informed decisions regarding storage, sales, and other aspects of agricultural management.
- Furthermore, by conducting a comprehensive review of existing research in this area, the study aims to identify the strengths and weaknesses of different models and propose avenues for improvement. This includes exploring the use of advanced analytics techniques, real-time data integration, and user-centric design

principles to enhance the effectiveness and usability of agricultural prediction and advisory systems.

- Ultimately, the goal of this work is to empower farmers with the tools and information they need to improve productivity, mitigate risks, and enhance their livelihoods. By leveraging the latest advancements in technology and data analytics, we can contribute to the sustainable development of India's agricultural sector and ensure the well-being of its farmers.

1.4 PROBLEM STATEMENT

In India, agriculture plays a pivotal role in driving economic growth and ensuring food security. Yet, farmers encounter barriers due to the absence of regional language support, hindering their access to crucial information and decision-making tools. Limited data sources used in current price forecasting methods further exacerbate the problem, resulting in recommendations that lack accuracy and relevance. As a result, farmers struggle to make informed decisions regarding crop management, market timing, and risk mitigation strategies, ultimately affecting their livelihoods and the overall agricultural productivity of the nation. Farmers' access to information and advisory services is limited due to a lack of regional language assistance, especially in rural areas with low literacy rates. This language barrier maintains inequities in information access and widens the digital divide, preventing technology breakthroughs from reaching vulnerable farming areas. Furthermore, relying on limited data sources for price forecasting produces poor recommendations since these methods fail to capture the dynamic character of agricultural markets and the various factors impacting price swings. Addressing these

issues necessitates complete solutions that stress linguistic inclusion and data-driven insights targeted to farmers' unique requirements and environments. By improving language support and utilizing modern analytics techniques, we can provide Indian farmers with the tools and knowledge they require to make informed decisions and increase agricultural output and sustainability.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING METHOD

Feihu Sun et al., on agricultural price prediction has garnered attention due to its importance in sustainable agricultural development. Traditional methods like time series analysis and econometric models have been complemented by intelligent forecasting methods such as machine learning and deep learning techniques. Moreover, combination models that integrate various approaches have shown promise in enhancing prediction accuracy. Emerging trends involve blending structured data (e.g., historical prices) with unstructured data (e.g., news and social media) for comprehensive insights. Researchers face challenges in balancing forecast accuracy and trend precision while exploring optimal model combinations. This literature review underscores the potential of hybrid models and the importance of integrating diverse data sources to improve agricultural price forecasting.

Nhat-Quang Tran., application of machine learning algorithms in agricultural price prediction has become increasingly popular due to their potential to enhance prediction accuracy and adaptability. This review explores recent research on machine learning techniques for forecasting agricultural prices. The importance of agriculture, particularly in developing countries, and the impact of crop price volatility highlight the necessity of improved prediction methods. Various machine learning approaches, such as decision trees, support vector machines, and neural networks, have been investigated for their effectiveness. While these algorithms offer significant

promise, challenges remain regarding data quality, model interpretability, and scalability. Further research is needed to optimize these techniques and overcome limitations for more robust and precise agricultural price forecasting.

Zhiyuan Chen., research on automated agricultural commodity price prediction systems utilizing novel machine learning techniques focuses on improving prediction accuracy and addressing challenges in forecasting historical data. Recent studies have shifted from traditional statistical methods to advanced machine learning approaches due to large datasets and the complexity of price fluctuations. Popular algorithms such as ARIMA, SVR, Prophet, XGBoost, and LSTM have been extensively compared using historical data from Malaysia. Findings suggest that the LSTM model, with its ability to handle nonlinearity and long-term dependencies, performs best with an average mean square error of 0.304. While machine learning strategies show promise, careful selection of data and optimization of model parameters remain critical for effective predictions.

Arushi Singh., research on modern agricultural advances has focused on using machine learning algorithms for crop prediction to help farmers make informed decisions on crop cultivation based on climatic conditions and soil nutrients. Popular algorithms such as K-Nearest Neighbor (KNN), Decision Tree, and Random Forest Classifier have been compared in recent studies to evaluate their effectiveness in crop prediction. Different criteria like Gini and Entropy have been used for these evaluations. Results indicate that Random Forest Classifier outperforms the other models, providing the highest accuracy in predictions. This approach assists farmers in selecting appropriate

crops, improving productivity, and adapting to environmental challenges while promoting sustainable agriculture.

Banupriya N., Recent research on crop yield prediction in India has shifted focus from complex environmental and agricultural factors to simpler, more accessible data points. This approach aims to facilitate the direct application of predictions by farmers without requiring in-depth understanding of underlying technology. By utilizing basic factors such as state, district, crop type, and season (e.g., Kharif, Rabi), researchers can efficiently gather and analyze data from the Indian Government Repository. Advanced regression techniques like Random Forest, Gradient Boosting, and Decision Tree algorithms have been explored to predict yield, while ensemble algorithms are employed to enhance accuracy and minimize errors. This streamlined approach aids farmers in making informed decisions for improved productivity and sustainability.

CHAPTER 3

SYSTEM ENVIRONMENT

3.1 HARDWARE SPECIFICATION

- System : Intel i5 processor
- Hard Disk : 500GB
- Monitor : 15’’’LED
- Input Devices : Keyboard , Mouse
- Ram : 8GB

3.2 SOFTWARE SPECIFICATION

- Operating System : Windows 11
- Coding Language : Python
- Working Platform : Google Colab

CHAPTER 4

SYSTEM ANALYSIS

4.1 EXISTING SYSTEM

- The Existing system doesn't forecast the data accurately.
- In the previous research, the dataset attributes is limited.
- The farmers will not get the proper yield or some time the farmers will not get the proper price for the commodities they grown.

4.2 EXISTING SYSTEM DRAWBACKS

- In some cases, they preferred the soil types or in some cases, they focused on the temperature.
- The proposed technique does not guarantee perfect forecasts.
- The farmers will not get the proper yield or some time the farmers will not get the proper price for the commodities they grown.

4.3 PROPOSED SYSTEM

- The proposed model is introduced to overcome all the disadvantages that arises in the existing system.
- The datasets includes all attributes like soil. temperature, historical places and etc to gain more insight on the data.
- It enhances the performance of the overall forecasting results.

4.3.1 DATA PREPROCESSING

df.isnull().sum(): this line of code provides a series where the index represents the column names of the DataFrame, and the values represent the count of null values in each column.This is useful for understanding which columns have missing data and the extent of that missing data.

df.info(): The df.info() method provides a concise summary of a pandas DataFrame (df).It includes information about the DataFrame's index and data types of each column, as well as memory usage.It also provides the number of non-null values for each column, helping you understand the data's completeness.This method is useful for quickly understanding the structure of your data, including the types of data each column contains and how much data is missing.

4.3.2 FEATURE ENGINEERING FOR NUMERICAL COLUMNS

When you apply MinMaxScaler from sklearn.preprocessing to normalize the numerical columns in your DataFrame **(**df_num**)** and then assign the transformed data to a new DataFrame **(**df_num_mn**),** you will end up with a DataFrame where the numerical columns have been scaled to a range of 0 to 1.

Fit and Transform: By applying the fit_transform method, the MinMaxScaler calculates the minimum and maximum values for each column in the input DataFrame (df_num) and scales the data to a range between 0 and 1.

New DataFrame: The result of this transformation is a new DataFrame (df_num_mn) with the same column names as df_num but with the values scaled.

4.3.3 FEATURE ENGINEERING FOR CATEGORICAL COLUMNS

When you use LabelEncoder from sklearn.preprocessing to encode categorical columns in your DataFrame (df_cat), the function transforms each categorical column into a numerical format by assigning integer labels to each unique value in the column. This process is known as label encoding.

Label Encoding: For each specified categorical column (e.g., 'APMC', 'Commodity', 'Month', 'district_name', 'state_name'), LabelEncoder assigns a unique integer value to each distinct category.

Encoded Columns: The encoded columns replace the original categorical columns with their respective integer labels.

4.3.4 MODEL SELECTION:

Linear Regression Model/OLS Model Overview:

Ordinary Least Squares (OLS) is a linear regression technique used to estimate the relationship between a dependent variable and one or more independent variables. The primary goal is to find the line (or hyperplane in higher dimensions) that minimizes the sum of the squared differences between the observed and predicted values. OLS is widely employed in statistical modeling, econometrics, and machine learning.

Decision Tree Regression

Decision Tree Regression is a supervised machine learning algorithm used for predicting continuous outcomes. Unlike decision trees in classification, which predict discrete class labels, decision tree regression predicts a numeric target variable. The algorithm works by recursively partitioning the dataset into subsets based on feature conditions, ultimately producing a tree structure where each leaf node corresponds to a predicted numerical value.

Random Forest Regression

Random Forest Regression is an ensemble learning technique that extends the concept of Random Forests, originally designed for classification problems, to regression tasks. It is a powerful and flexible algorithm that leverages the strength of multiple decision trees to make more accurate and robust predictions for continuous outcomes.

Key Features and Concepts:

Ensemble of Decision Trees: Random Forest Regression is built on an ensemble of decision trees. Multiple decision trees are constructed independently, and their predictions are averaged to obtain a final result.

Bagging (Bootstrap Aggregating): Each tree in the Random Forest is trained on a bootstrap sample (randomly selected with replacement) from the original dataset. This helps introduce diversity among the trees.

Random Feature Selection: At each node of a decision tree, a random subset of features is considered for splitting. This randomness adds further diversity to the individual trees.

Prediction Aggregation: For regression, the predictions of individual trees are averaged to produce the final output. This ensemble approach helps mitigate overfitting and improves generalization.

Handling Missing Values: Random Forests can effectively handle missing values in the dataset, reducing the need for extensive data preprocessing.

Robust to Overfitting: The ensemble nature of Random Forests tends to reduce overfitting, making them less sensitive to noise and outliers in the data.

Versatility: Random Forests can be applied to a wide range of regression tasks and are suitable for datasets with a large number of features.

4.4 PROPOSED SYSTEM ADVANTAGES

- To measure or to learn the performance of the proposed model Random Forest and Support Vector Machine techniques are applied.
- Maximum Relevance Approach issued to improve the forecast accuracy.
- Provide accurate forecasting results.
- The datasets includes all attributes like soil. temperature, historical places and etc to gain more insight on the data.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 SPECIFICATION

- The hardware requirements for the system include a hard disk with a capacity of 500GB or higher, a minimum of 4GB RAM, and a processor equivalent to or higher than an Intel Core i5. These specifications ensure sufficient storage space, memory, and processing power to support the execution of machine learning tasks, including data preprocessing, model training, and inference.
- On the software side, the system requires an operating system compatible with Windows 7, 8, or 11 (64-bit). Additionally, Python is necessary as the primary programming language for implementing machine learning algorithms and frameworks. Google colab, along with the Notebook IDE, serves as essential tools for managing Python environments, libraries, and interactive development. Google colab simplifies package management and provides a comprehensive suite of data science tools, while Notebook offers an intuitive interface for writing, executing, and documenting code, making it well-suited for developing and prototyping machine learning models. These software components collectively enable efficient development and deployment of machine learning solutions on the specified hardware platform.

5.2 SOFTWARE DESCRIPTION

PYTHON

Python is a versatile and widely-used programming language known for its simplicity and readability. It offers a vast ecosystem of libraries and tools for various applications, including web development, data analysis, machine learning, artificial intelligence, and scientific computing. Python's syntax emphasizes code readability and productivity, making it an ideal choice for beginners and experienced developers alike. With its rich set of features and extensive community support, Python continues to be a popular choice for diverse programming tasks.

PANDAS

Pandas is a powerful open-source library in Python used for data manipulation and analysis. It offers data structures and functions to efficiently handle structured data, such as tabular data, time series, and heterogeneous data. Pandas provides DataFrame objects for organizing and manipulating data in rows and columns, along with tools for reading and writing data from various file formats. It is widely used in data science, machine learning, and other domains for data preprocessing, exploration, and transformation tasks.

SECURITY AND PRIVACY:

Windows 11 incorporates several changes to its security measures, which are critical for protecting both the software and users. This includes hardware-based security solutions, safe boot processes, and enhancements to user data protection. By aligning the program with Windows 11, you may assist ensure that the application runs in a secure environment.

CODING LANGUAGE:

PYTHON

Python is a high-level, extensible programming language with dynamic typing that is well-known for its simplicity and use. It has a strong and active development community, resulting in a rich ecosystem of libraries and frameworks. Python is an ideal choice for this project because of its versatility and ability to handle a wide range of tasks, including data processing and analysis, natural language processing, and machine learning.

MACHINE LEARNING:

Python's strong support for machine learning and artificial intelligence possible. These frameworks include numerous tools and pre-trained models for developing and training powerful machine learning algorithms that can power crop recommendation and price forecasting systems.

WORKING PLATFORM: GOOGLE COLAB

Google Colab (or Google Colaboratory) is a cloud-based platform for running Python programs in a notebook environment. This platform provides the flexibility and resources needed to execute complicated activities like machine learning, data analysis, and natural language processing, eliminating the need for a strong local machine.

CLOUD-BASED INFRASTRUCTURE:

Google Colab's cloud-based architecture enables developers to work from anywhere, on any device with an internet connection. This flexibility is useful for remote teams and collaborative projects, allowing for seamless work even when team members are in different places.

REAL-TIME COLLABORATION:

Colab allows multiple users to work on the same notebook at the same time thanks to its real-time collaboration feature. This tool is especially valuable for collaborative projects, allowing team members to collaborate, exchange thoughts, and provide comments in real time.

INTEGRATION WITH GOOGLE SERVICES:

Google Colab effortlessly connects with Google Drive, providing simple access to cloud-stored data. Developers can read data straight from Drive, save work to Drive, and share notebooks with others.

PRE-INSTALLED LIBRARIES AND ENVIRONMENTS:

Colab has many popular Python libraries pre-installed, which reduces setup time and complexity. The platform's support for various Python environments and packages allows you to construct a wide range of apps without worrying about dependencies and compatibility issues.

EVALUATION METRICS

Mean Absolute Error (MAE)

In the context of machine learning, absolute error refers to the magnitude of difference between the prediction of an observation and the true value of that observation.

Mean Square Error (MSE)

Crucial metric for evaluating the performance of predictive models. It measures the average squared difference between the predicted and the actual target values within a dataset.

R2- squared

R squared (R2) value in machine learning is referred to as the coefficient of determination or the coefficient of multiple determination in case of multiple regression. R squared in regression acts as an evaluation metric to evaluate the scatter of the data points around the fitted regression line.

INTERACTIVE DEVELOPMENT:

The notebook-based workspace in Google Colab allows for interactive programming and visualization. Developers may run code cells individually and see the results right away, which is great for trying out new approaches and swiftly iterating on ideas.

CONCLUSION

In the pursuit of enhancing agricultural decision-making and supporting farmers in India, this project delved into the analysis of a comprehensive dataset encompassing market transactions, crop details, and pricing information. The primary objectives were to develop models for crop recommendation and price forecasting, addressing the critical challenges faced by farmers. This project lays the foundation for leveraging data-driven approaches to empower farmers and strengthen the agricultural sector. By combining traditional statistical models with advanced machine learning techniques, we have strived to provide valuable insights and tools that contribute to the overall well-being of the farming community in India. The journey doesn't end here; it opens avenues for ongoing research, collaboration, and innovation in the realm of agriculture and data science.

APPENDIX

CODING

```
import pandas as pd

from google.colab import files

uploaded=files.upload()

import io

df=pd.read_csv(io.BytesIO(uploaded['Agriculture_commodities_dataset.csv']))

df

df.isnull().sum()

df.info()

df.date=pd.to_datetime(df.date)

df_num=df.select_dtypes(include=['int64','float64'])

df_num

df_cat=df.select_dtypes(include=object)

df_cat

from sklearn.preprocessing import MinMaxScaler

mn=MinMaxScaler()

a=mn.fit_transform(df_num)

df_num_mn=pd.DataFrame(a,columns=df_num.columns)

df_num_mn
```

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df_cat['APMC']=le.fit_transform(df_cat['APMC'])
df_cat['Commodity']=le.fit_transform(df_cat['Commodity'])
df_cat['Month']=le.fit_transform(df_cat['Month'])
df_cat['district_name']=le.fit_transform(df_cat['district_name'])
df_cat['state_name']=le.fit_transform(df_cat['state_name'])
df_cat
df_pred=pd.concat([df_cat,df_num_mn],axis=1)
df_pred
x=df_pred.iloc[:, :9]
x
y=df_pred.iloc[:,[-1]]
y
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
x_test.sort_index(ascending=True,inplace=True)
y_test.sort_index(ascending=True,inplace=True)
x_train.sort_index(ascending=True,inplace=True)
y_train.sort_index(ascending=True,inplace=True)
```

```
import statsmodels.api as sm
MLR_model1=sm.OLS(y_train,x_train).fit()
print(MLR_model1.summary())
y_test_pred=MLR_model1.predict(x_test)
y_test_pred.count()
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test['modal_price'],y_pred=y_test_pred)
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y_test['modal_price'],y_pred=y_test_pred)
from sklearn.metrics import r2_score
r2_score(y_true=y_test['modal_price'],y_pred=y_test_pred)
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
from sklearn.tree import DecisionTreeRegressor
dtree=DecisionTreeRegressor(criterion='squared_error',max_depth=5,min_sample
s_split=2,min_samples_leaf=1)
dtree.fit(x_train,y_train)
y_pred1=dtree.predict(x_test)
y_pred1
from sklearn.metrics import mean_squared_error
```

```
mean_squared_error(y_test['modal_price'],y_pred=y_pred1)
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y_test['modal_price'],y_pred=y_pred1)
from sklearn.metrics import r2_score
r2_score(y_true=y_test['modal_price'],y_pred=y_pred1)
from sklearn.ensemble import RandomForestRegressor
classifier=RandomForestRegressor(n_estimators=500,criterion='squared_error')
classifier.fit(x_train,y_train)
y_pred2=classifier.predict(x_test)
y_pred2
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test['modal_price'],y_pred=y_pred2)
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y_test['modal_price'],y_pred=y_pred2)
from sklearn.metrics import r2_score
r2_score(y_true=y_test['modal_price'],y_pred=y_pred2)
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
plt.subplot(2, 1, 1)
```

```
sns.countplot(x='min_price', data=df, palette='viridis')
plt.title('Count Plot for Min_price')
plt.subplot(2, 1, 2)
sns.countplot(x='modal_price', data=df, palette='viridis')
plt.title('Count Plot for Modal_price')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8, 6))
sns.scatterplot(x='min_price', y='max_price', data=df, color='blue', alpha=0.7)
plt.title('Scatter Plot for Min_price vs Maximum_price')
plt.xlabel('Min_price')
plt.ylabel('Max_price')
plt.show()
cross_tab = pd.crosstab(df['APMC'], df['Commodity'])
print(cross_tab)
correlation_matrix = df.corr(numeric_only=True)
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Correlation Plot')
plt.show()
```

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
plt.figure(figsize=(10, 6))
sns.countplot(x='Month', hue='Commodity', data=df, palette='Set2')
plt.title('Bar Chart for Month and Commodity')
plt.xlabel('Month')
plt.ylabel('Count')
plt.show()
plt.figure(figsize=(10, 6))
sns.countplot(x='Commodity', data=df, palette='Set2')
plt.title('Bar Chart for Commodity')
plt.xlabel('Commodity')
plt.ylabel('Count')
plt.show()
```

SCREENSHOTS

	Year	arrivals_in_qtl	min_price	max_price	modal_price
0	2015	79	1406	1538	1463
1	2016	106	1788	1925	1875
2	2015	1253	1572	1890	1731
3	2016	387	1750	2220	1999
4	2015	3825	1600	2200	1900
...	...	...	...	...	...
62424	2016	586	5700	6367	6200
62425	2016	2	5000	5000	5000
62426	2016	46	4700	6933	6400
62427	2016	166	2583	2708	2633
62428	2016	74	2933	3200	3067

62429 rows × 5 columns

Figure 1. Dataset description

	APMC	Commodity	Month	district_name	state_name
0	Ahmednagar	Bajri	April	Ahmadnagar	Maharashtra
1	Ahmednagar	Bajri	April	Ahmadnagar	Maharashtra
2	Ahmednagar	Wheat(Husked)	April	Ahmadnagar	Maharashtra
3	Ahmednagar	Wheat(Husked)	April	Ahmadnagar	Maharashtra
4	Ahmednagar	Sorgum(Jawar)	April	Ahmadnagar	Maharashtra
...	...	...	...	...	...
62424	Shrigonda	GRAM	November	Ahmadnagar	Maharashtra
62425	Shrigonda	GREEN GRAM	November	Ahmadnagar	Maharashtra
62426	Shrigonda	BLACK GRAM	November	Ahmadnagar	Maharashtra
62427	Shrigonda	SOYBEAN	November	Ahmadnagar	Maharashtra
62428	Shrigonda	SUNFLOWER	November	Ahmadnagar	Maharashtra

62429 rows × 5 columns

Figure 2. Data-preprocessing

	Year	arrivals_in_qtl	min_price	max_price	modal_price
0	0.5	5.378372e-05	0.000446	0.000961	0.010278
1	1.0	7.240116e-05	0.000567	0.001203	0.013172
2	0.5	8.632976e-04	0.000499	0.001181	0.012161
3	1.0	2.661605e-04	0.000555	0.001387	0.014043
4	0.5	2.636781e-03	0.000507	0.001375	0.013348
...	...	...	...	...	...
62424	1.0	4.033779e-04	0.001808	0.003979	0.043556
62425	1.0	6.895349e-07	0.001586	0.003125	0.035126
62426	1.0	3.102907e-05	0.001491	0.004333	0.044962
62427	1.0	1.137733e-04	0.000819	0.001692	0.018497
62428	1.0	5.033604e-05	0.000930	0.002000	0.021546

62429 rows × 5 columns

Figure 3. Feature Engineering on numerical columns

	APMC	Commodity	Month	district_name	state_name
0	4	24	0	0	0
1	4	24	0	0	0
2	4	348	0	0	0
3	4	348	0	0	0
4	4	310	0	0	0
...	...	...	...	...	...
62424	298	114	9	0	0
62425	298	117	9	0	0
62426	298	19	9	0	0
62427	298	287	9	0	0
62428	298	296	9	0	0

62429 rows × 5 columns

Figure 4. Feature Engineering on categorical columns

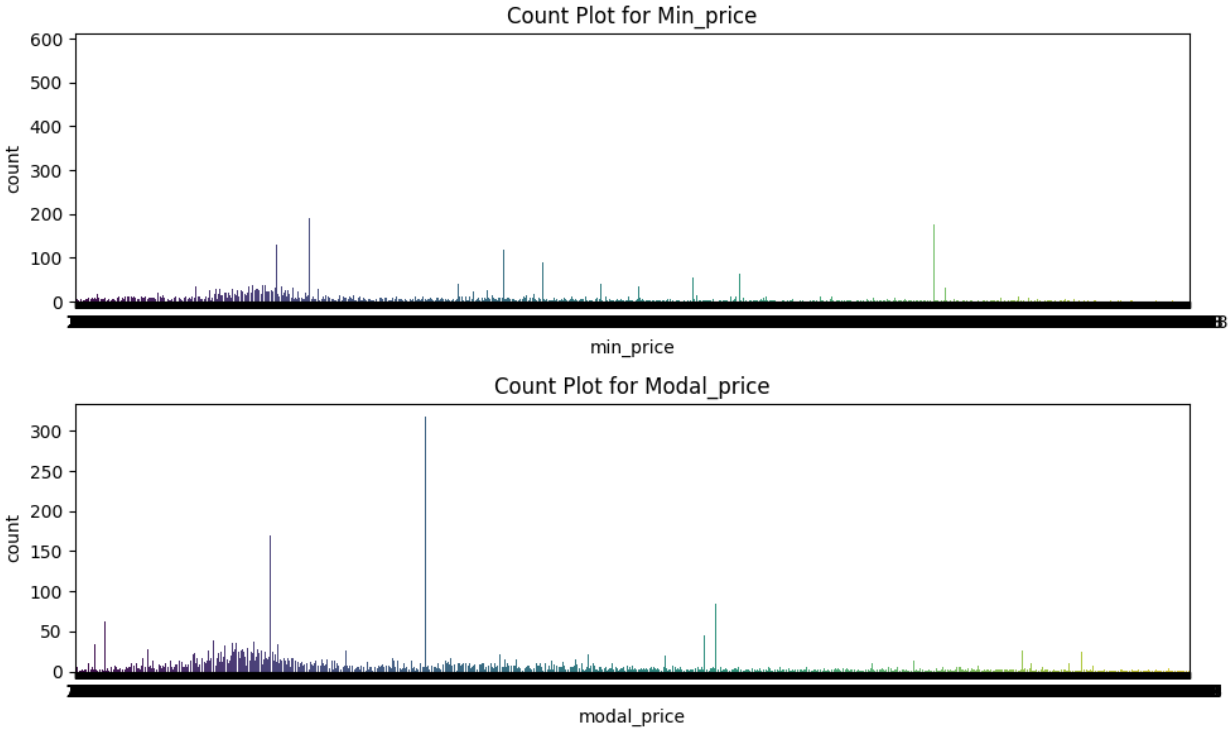


Figure 5. Visualization

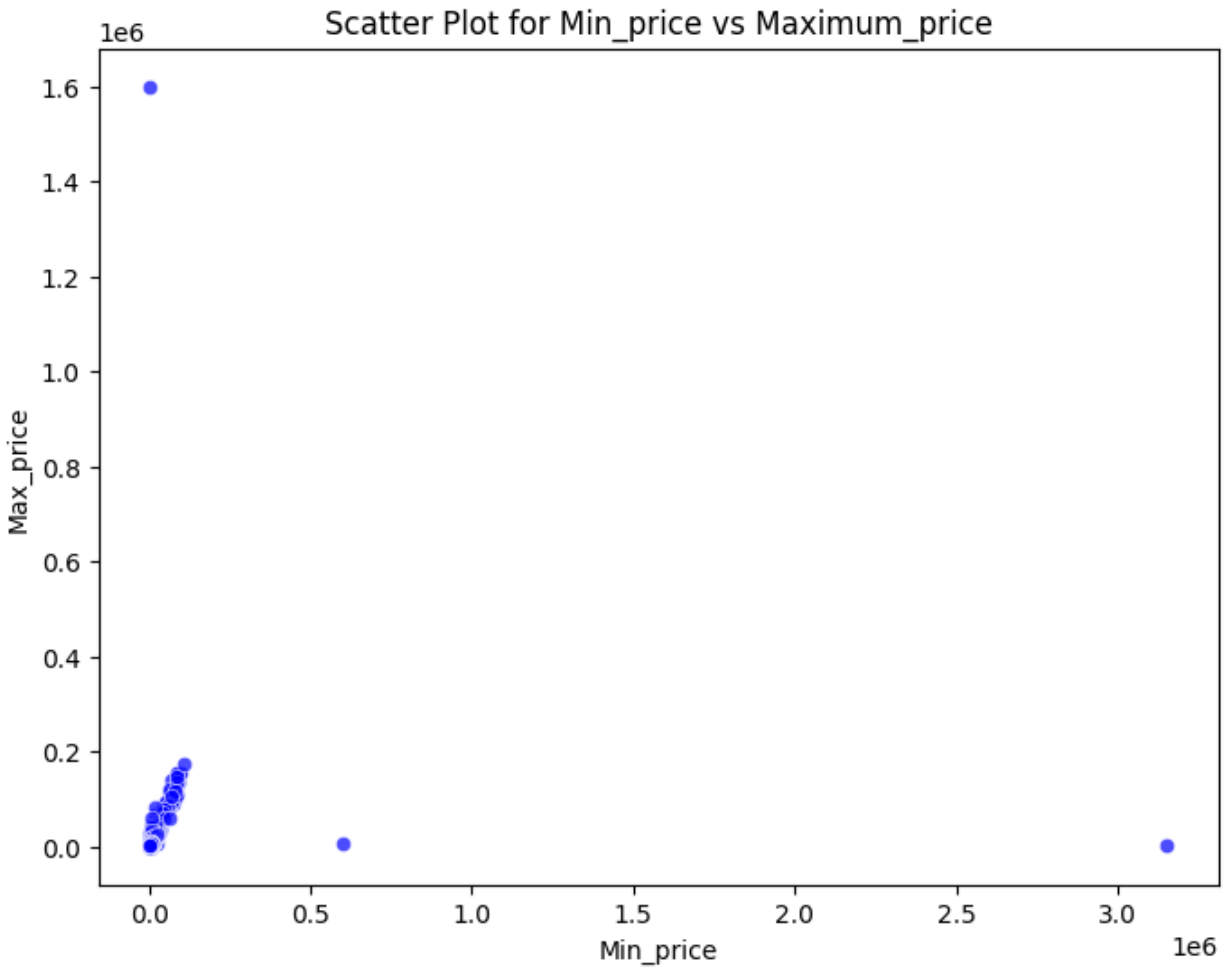


Figure 6. Scatter plot

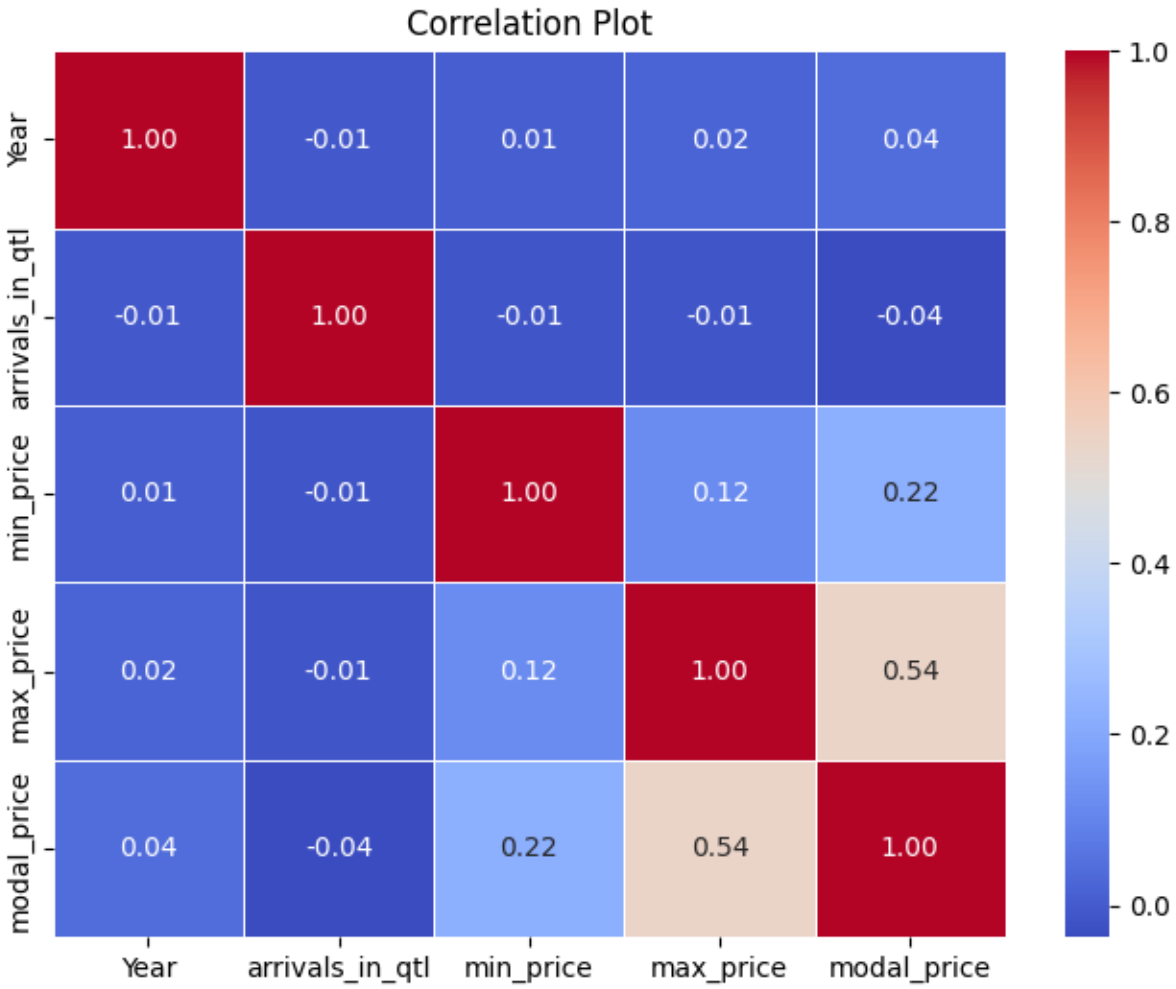
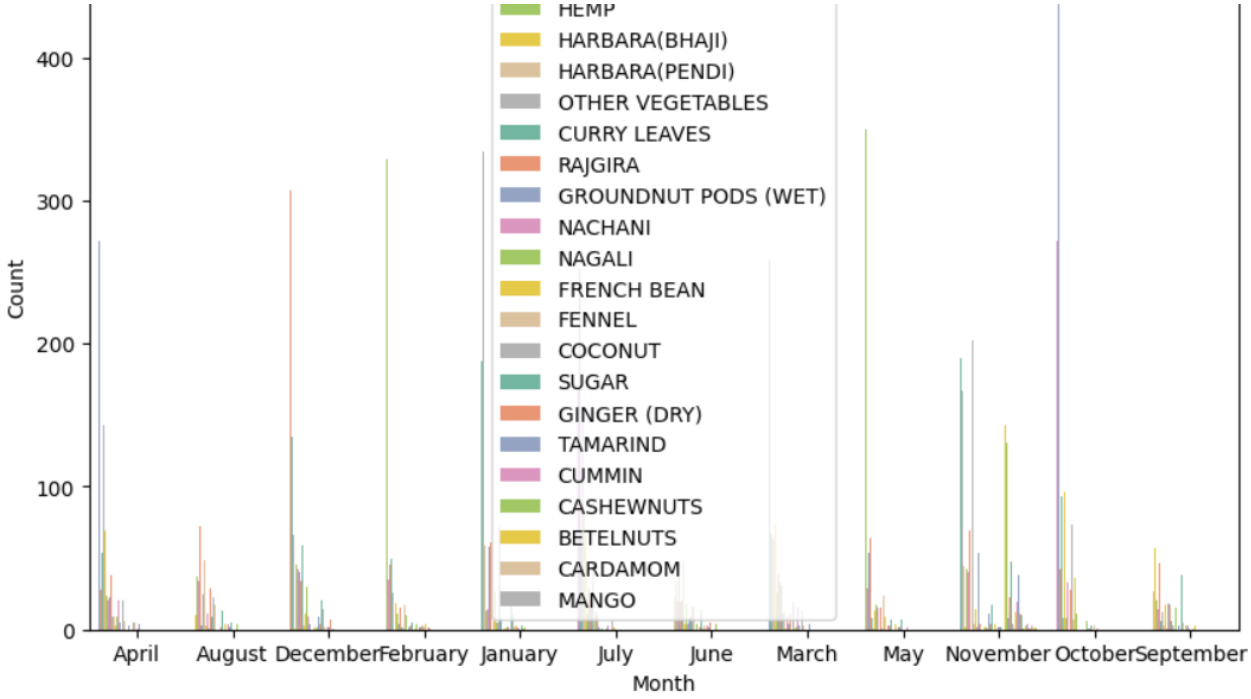
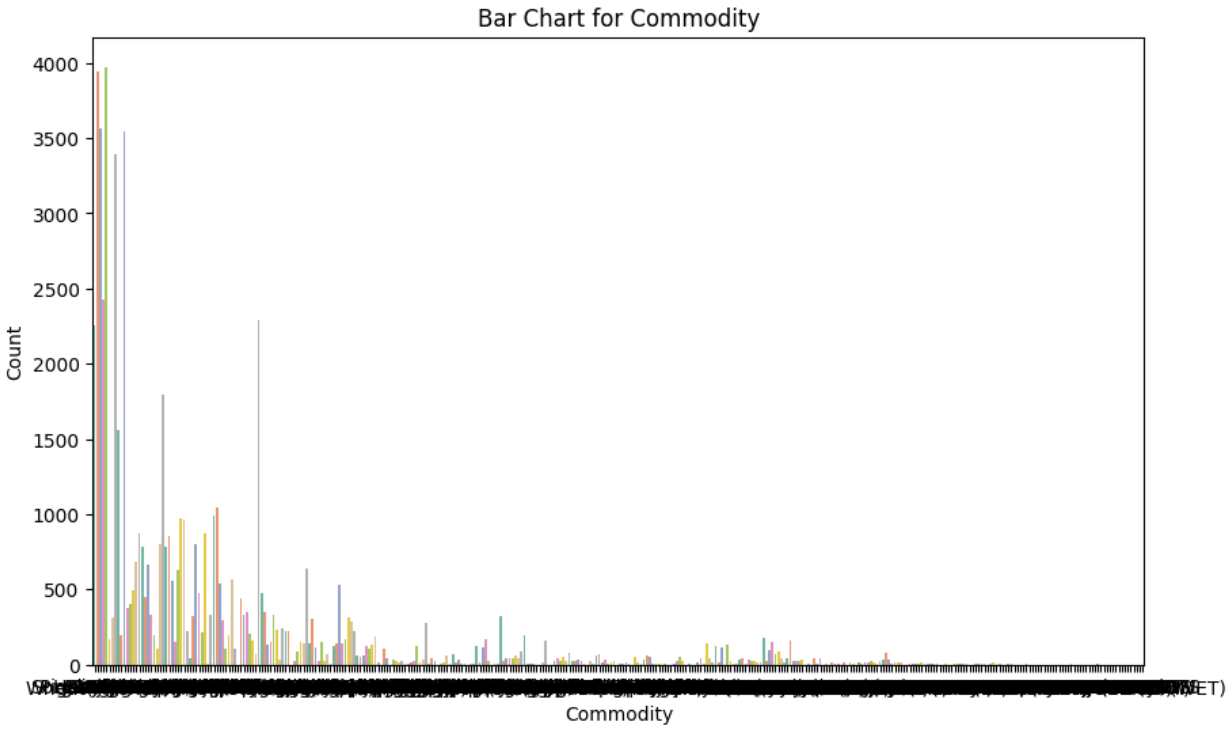


Figure 7. correlation matrix


HEMP
HARBARA(BHAJI)
HARBARA(PENDI)
OTHER VEGETABLES
CURRY LEAVES
RAJGIRA
GROUNDNUT PODS (WET)
NACHANI
NAGALI
FRENCH BEAN
FENNEL
COCONUT
SUGAR
GINGER (DRY)
TAMARIND
CUMMIN
CASHEWNUTS
BETELNUTS
CARDAMOM
MANGO
Count
0
100
200
300
400
April
August
December
February
January
July
June
March
May
November
October
September
Month


Bar Chart for Commodity
Count
4000
3500
3000
2500
2000
1500
1000
500
0
Commodity