PGPDSE FT Aug-2023

Capstone Project

Final Report

Group-5

Project title:

CROPWISE: Smart Crop Decisions for Profitable Farming

Team members

Harishwar T G Akash L Noor Fathima M D Zikrullah Khan

Summary of Problem Statement, Data, and Findings

Problem Statement

The objective was to develop a predictive model for crop production and pricing using a comprehensive dataset that includes various factors such as temperature, humidity, rainfall, and soil conditions (nitrogen, phosphorus, potassium, pH levels). Additionally, the goal was to forecast crop prices for the years 2022 and 2023.

The project "CROPWISE: Smart Crop Decisions for Profitable Farming" focuses on recommending the best crop to grow and predicting crop prices based on soil and climatic conditions. By utilizing machine learning, this project aims to optimize agricultural production and enhance farming efficiency.

Data

Dataset 1: Optimizing Agriculture Production.csv

The dataset includes variables such as temperature, humidity, rainfall, nitrogen content (N), phosphorus content (P), potassium content (K), and pH of the soil. It also includes historical crop prices for the years 2022 and 2023.

Variable name	Data type	Description
Nitrogen	int	Nitrogen (N), phosphorus (P) and potassium (K) are important essential nutrients for plant growth and development.
Phosphorus	Int	
Potassium	int	
Temperature	Float	The optimum soil temperature for seed germination ranges between 68 and 86°F (20-30°C)
Humidity	Float	Humidity is the amount of water vapor in the air
pH	float	pHs of less than 7 indicate acidity, whereas a pH of greater than 7 indicates a base
Rainfall	Float	provides the water needed for plants to uptake nutrients and transport them to the leaves and stems
label	Category	It denotes crops, such as Rice, maize, chickpea, etc.

Dataset 2: all crop 2022.csv and all crop 2023.csv

This dataset contains monthly price data for various crops over a period spanning from January 2022 to December 2023. It provides detailed price information for each month, allowing for an analysis of trends and patterns over time. The data shows fluctuations in prices, reflecting seasonal variations, market dynamics, and potentially the impact of external factors such as weather conditions, market demand, and supply chain disruptions. By examining this dataset, one can gain insights into the economic conditions affecting agricultural commodities and make informed predictions about future price movements.

```
df_2022.head()
```

	label	2022-01	2022-02	2022-03	2022-04	2022-05	2022-06	2022-07	2022-08	2022-09	2022-10	2022-11	2022-12
1	Paddy ET	1816.0	1806.0	1802.0	1785.0	1748.0	1727.0	1730.0	1753.0	1774.0	1886.0	1910.0	1910.0
2	rice	3568.0	3541.0	3551.0	3568.0	3538.0	3573.0	3564.0	3570.0	3590.0	3619.0	3637.0	3650.0
3	Bajra TERR	1658.0	1642.0	1647.0	1642.0	1654.0	1672.0	1716.0	1793.0	1797.0	1842.0	1945.0	1992.0
4	Barley	1728.0	1764.0	1704.0	1791.0	1735.0	1826.0	1849.0	1923.0	1983.0	2053.0	2095.0	2185.0
5	muskmelon	2461.0	2525.0	2528.0	2505.0	2520.0	2474.0	2363.0	2367.0	2313.0	2212.0	2258.0	2355.0

```
df_2023.head()
```

	label	2023-01	2023-02	2023-03	2023-04	2023-05	2023-06	2023-07	2023-08	2023-09	2023-10	2023-11	2023-12
1	Paddy ET	1881.0	1887.0	1888.0	1878.0	1869.0	1895.0	1902.0	1954.0	1994.0	2045.0	2093.0	2151.0
2	Rice	3675.0	3673.0	3680.0	3699.0	3695.0	3729.0	3800.0	3862.0	3965.0	3996.0	4011.0	4001.0
3	Bajra TERR	2020.0	2050.0	2179.0	2282.0	2274.0	2303.0	2348.0	2449.0	2278.0	2345.0	2465.0	2537.0
4	Barley	2194.0	2228.0	2267.0	2552.0	2592.0	2611.0	2600.0	2685.0	2618.0	2639.0	2674.0	2668.0
5	muskmelon	2430.0	2456.0	2503.0	2610.0	2630.0	2659.0	2770.0	2811.0	2801.0	2835.0	3126.0	3238.0

Findings

- Outliers were identified and plotted to understand their impact on the data, but they were not removed to preserve the dataset's integrity.
- Various statistical tests and visualizations were performed to comprehend the distribution and relationships between variables.
- Machine learning models were developed to predict crop types and prices, focusing on improving accuracy and robustness.
- Price forecasting was performed using ARIMA and SARIMAX models, with suitable models identified for each crop, and forecasts were made up to July 2024.

Overview of the Final Process

Methodology

1. **Data Pre-processing:** Handling missing values, renaming columns, and removing duplicates.
2. **Exploratory Data Analysis (EDA):** Visualizing data distributions and relationships.
3. **Outlier Analysis:** Identifying and plotting outliers.
4. **Feature Engineering:** Creating new features based on domain knowledge.
5. **Modeling:** Developing and fine-tuning machine learning models such as Decision Trees, Random Forest, XGBoost, and multiclass classification models for crop prediction.
6. **Evaluation:** Using metrics like accuracy, recall, precision, and F1 score to evaluate model performance for crop prediction and RMSE for price forecasting.
7. **Comparison:** Benchmarking against initial models and improving upon them.

Features of the Data

- Soil conditions (N, P, K, pH)
- Environmental factors (temperature, humidity, rainfall)
- Historical crop prices for 2022 and 2023

Algorithms Used

- Decision Trees
- Random Forest
- XGBoost
- Multiclass Classification Models for crop prediction
- ARIMA and SARIMAX for price forecasting

Step-by-Step Walkthrough of the Solution

Step 1: Data Collection and Cleaning

- Loaded datasets for crop production and prices.
- Removed null values and irrelevant columns.
- Renamed columns for better readability.

```
RangeIndex: 2200 entries, 0 to 2199
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   N            2200 non-null   int64
 1   P            2200 non-null   int64
 2   K            2200 non-null   int64
 3   temperature  2200 non-null   float64
 4   humidity     2200 non-null   float64
 5   ph           2200 non-null   float64
 6   rainfall     2200 non-null   float64
 7   label        2200 non-null   category
dtypes: category(1), float64(4), int64(3)
memory usage: 123.3 KB
```

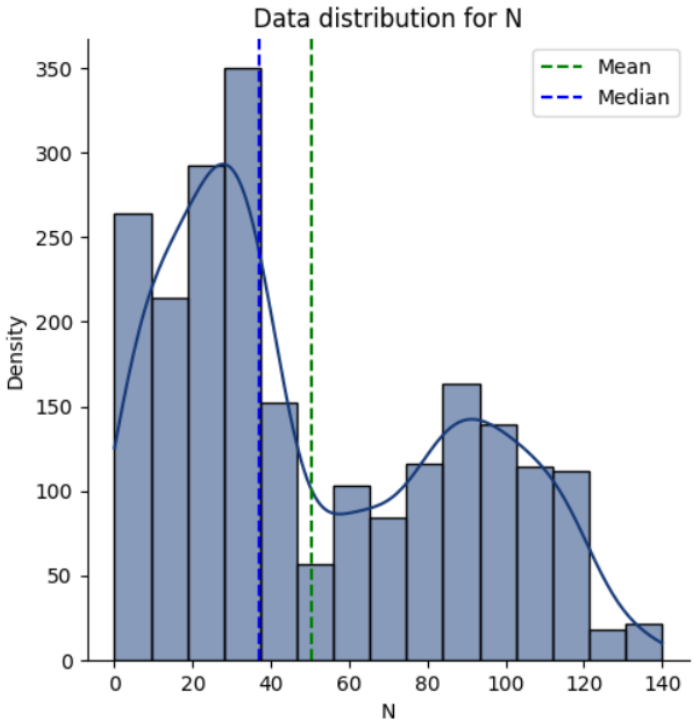
```
df_crop.isnull().sum()
```

```
N              0
P              0
K              0
temperature    0
humidity       0
ph             0
rainfall       0
label          0
dtype: int64
```

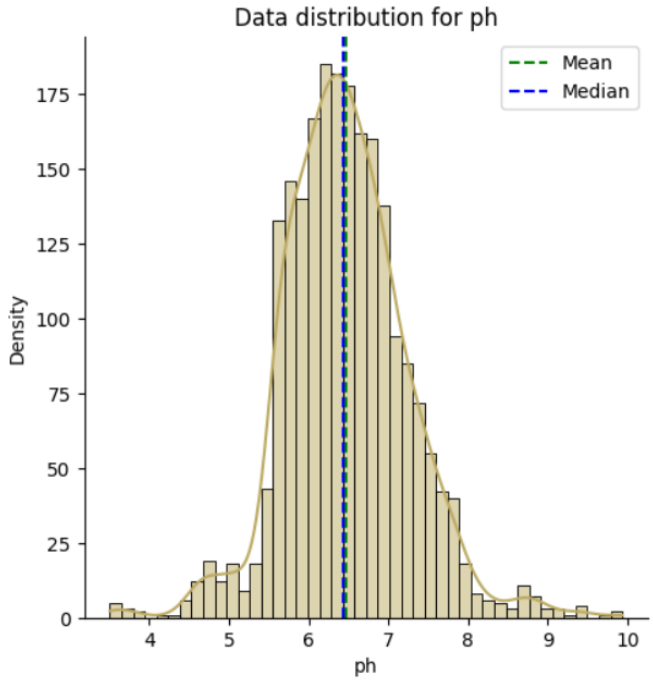
Step 2: Exploratory Data Analysis (EDA)

- Visualized the distribution of each feature.

Here is an example of visualization result of a crop

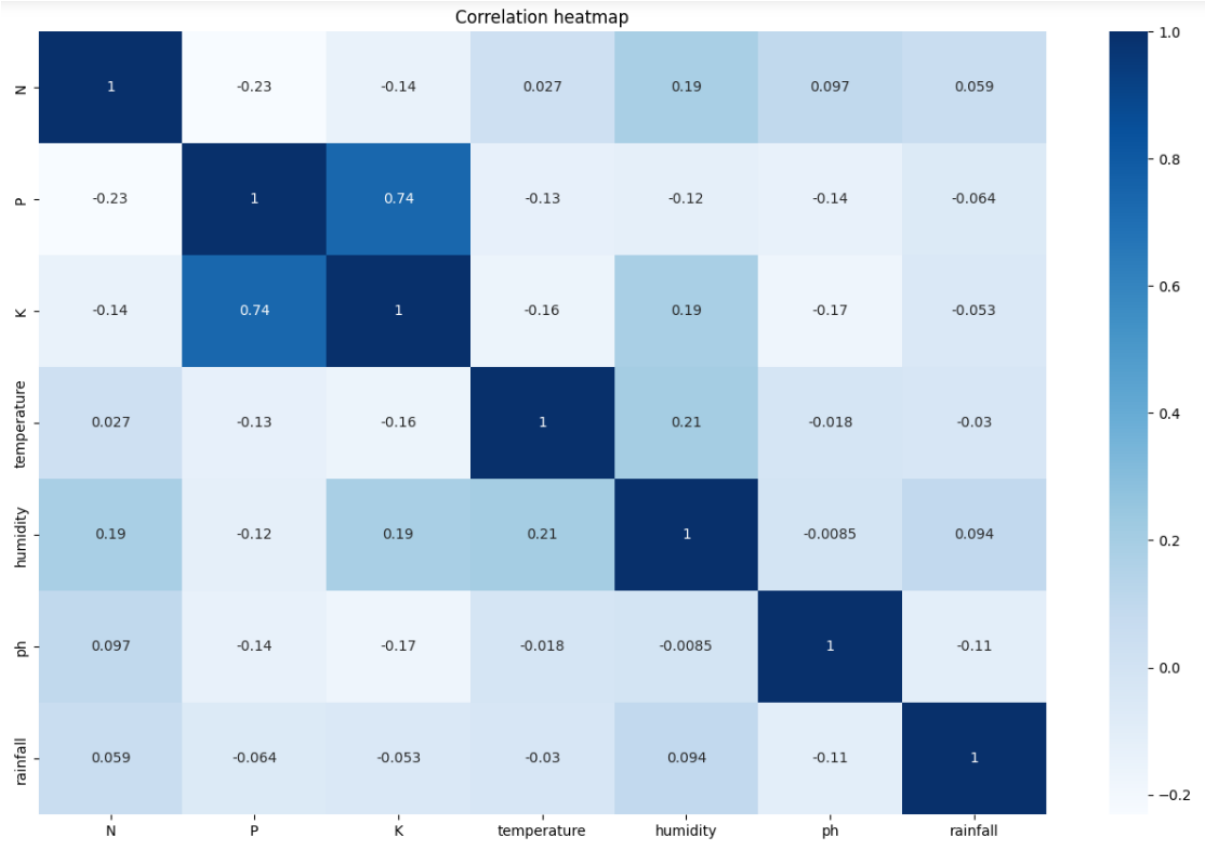


The histogram shows how the Nitrogen column is skewed throughout the dataset.

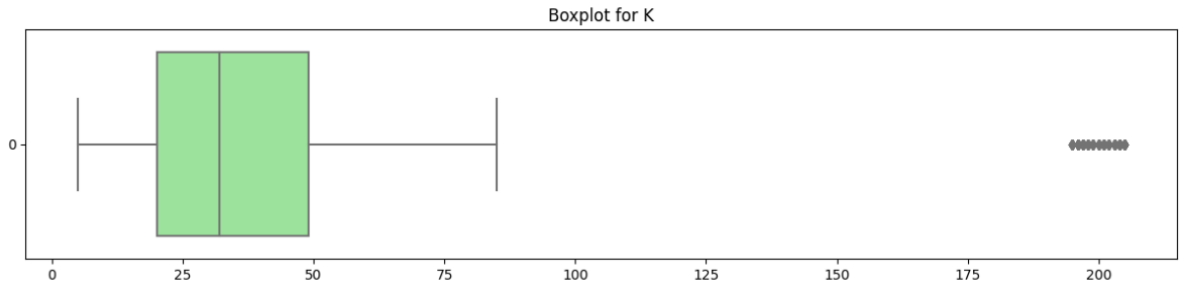


Here we can observe that very few rows have extreme acidic or basic pH values. (Refer notebook/file for output of other columns.

- Analyzed the relationship between soil conditions and crop types.



- Checked for outliers and plotted them.
  - By creating boxplots for each feature column in the DataFrame **df_crop** by iterating over the columns, generating a boxplot for each feature to visualize its distribution.



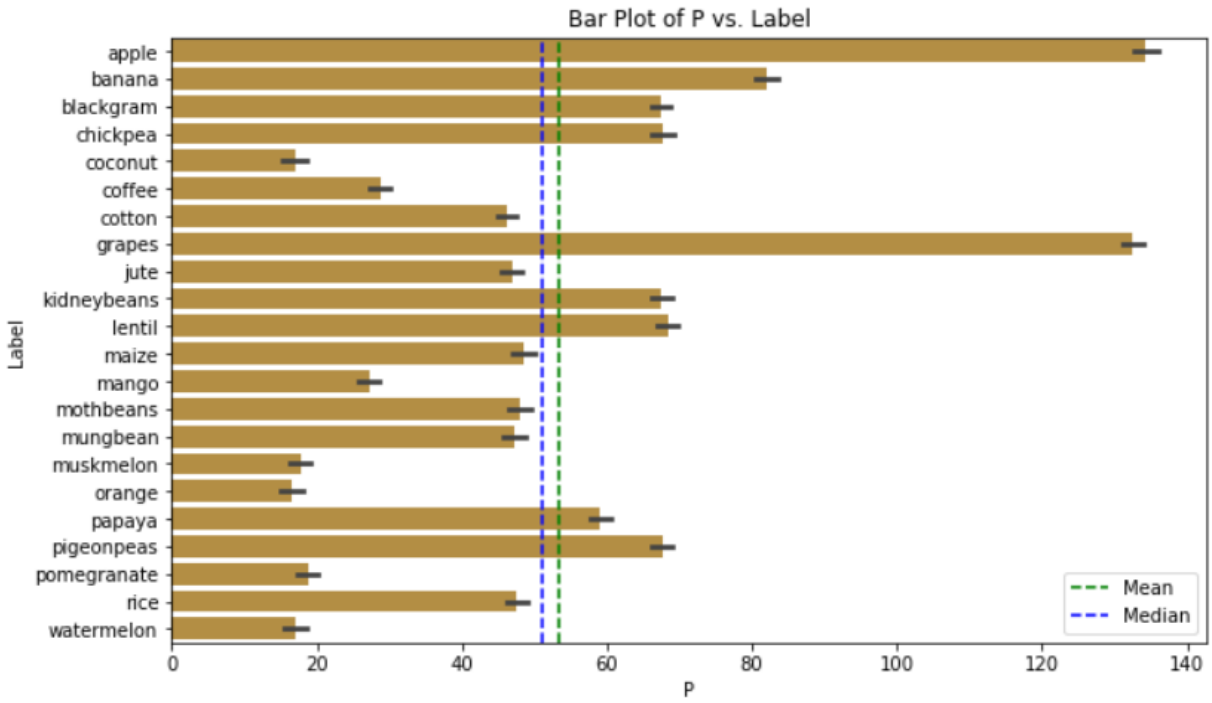
-

Figure shows the boxplot containing outliers for Potassium (K)

  - The plots display quartiles, median, and potential outliers, providing insights into the spread and variability of the dataset.
  - **In this scenario, each data point represents unique environmental conditions from diverse geographical regions. Treating outliers could potentially compromise the dataset's integrity, as these extremes contain valuable insights for predicting crops across various landscapes. Thus, retaining the data in its entirety ensures the**

preservation of agricultural diversity, enhancing the reliability of predictive models for crop recommendation.

- Analyse relationship between attributes and Target variable



  - **Here is an example of P(Phosphorus), we can observe that Apple and Grapes need high Phosphorus in general to grow them as the mean value is indeed higher than the values for other crops. (**Refer notebook/file for output of all other columns**)**

Step 3: Feature Engineering

- Standardized features to improve model performance.
- Whether any transformations required
- New feature 'Season' is created based on climate condition to help under the dataset based on the Domain.

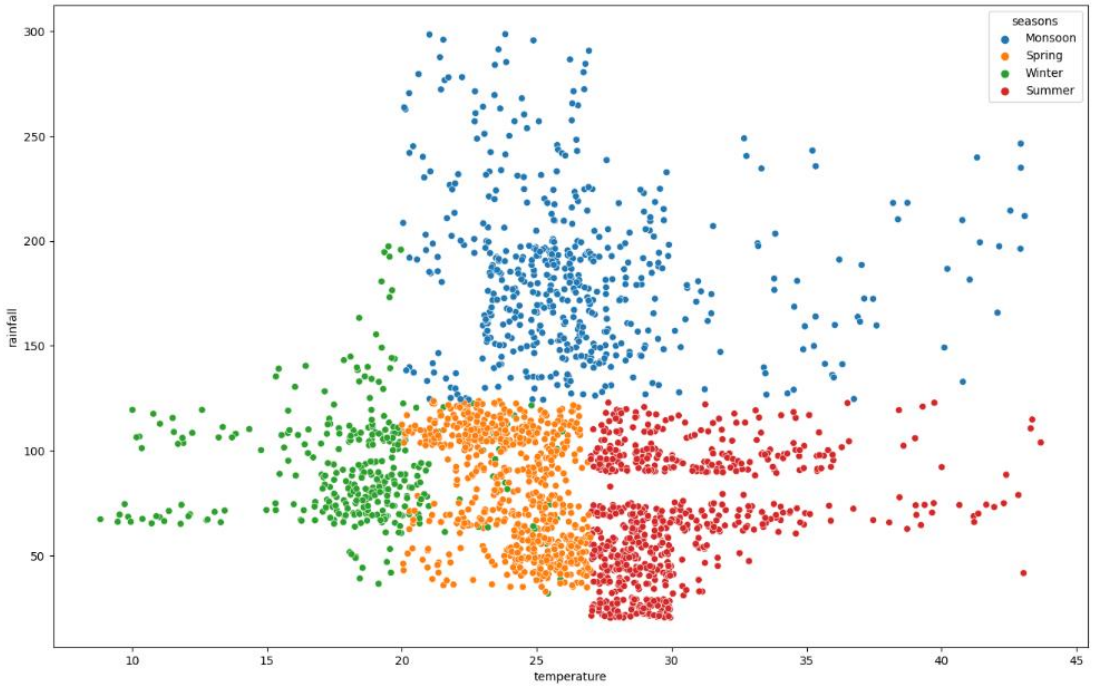
In [44]:
```
def find_season(row):
    if row['rainfall'] > 124 and row['temperature'] > 20:
        return 'Monsoon'
    elif row['temperature'] >= 27 and row['rainfall'] <= 124:
        return 'Summer'
    elif (row['temperature'] < 26 and row['humidity'] <= 50) or (row['temperature'] < 20):
        return 'Winter'
    else:
        return 'Spring'

# Apply the function to the DataFrame
df_temp['seasons'] = df_temp.apply(find_season, axis=1)
```

There are 3 major seasons for crop cultivation in India.

- **Monsoon Season**: These crops are also known as the Kharif season, Kharif crops are sown at the beginning of the monsoon season, typically from May to October. These crops require a lot of water and hot conditions to grow well.
- **Winter Season**: These crops are also known as Rabi season.Rabi crops are planted after the monsoon season, around October or mid-November, and are harvested in April or May.
- **Summer Season**: These crops are also known as Zaid season.Zaid crops are grown between the Rabi and Kharif seasons, typically from March to June. These crops require warm, dry weather and mature early.

In [45]:
```
plt.figure(figsize=(16,10))
sns.scatterplot(data=df_temp , x='temperature',y='rainfall' , hue='seasons')
plt.show()
```



Feature Selection

- We have used hypothesis testing to find significance/importance of feature for predicting target.
- The ANOVA test– f_oneway is used here to find p_value and test the significance with 95% confidence level.
- All features highly contribute to predict the crop.

Hypothesis testing

In [49]:
```
numerical_vars = ["N", "P", "K", "temperature","humidity", "ph","rainfall"]

for var in numerical_vars:
    f_statistic, p_value = f_oneway(*[group[var] for label, group in df_crop.groupby('label')])
    if p_value > 0.05:
        print(f"As the p-value-{p_value} is greater than alpha (0.05), {var} is less significant to predicting label.\n")
    else:
        print(f"As the p-value-{p_value} is less than alpha (0.05), {var} is highly significant to predicting label.\n")
```

```
As the p-value-0.0 is less than alpha (0.05), N is highly significant to predicting label.

As the p-value-0.0 is less than alpha (0.05), P is highly significant to predicting label.

As the p-value-0.0 is less than alpha (0.05), K is highly significant to predicting label.

As the p-value-4.019323818173197e-305 is less than alpha (0.05), temperature is highly significant to predicting label.

As the p-value-0.0 is less than alpha (0.05), humidity is highly significant to predicting label.

As the p-value-6.4931618988390225e-199 is less than alpha (0.05), ph is highly significant to predicting label.

As the p-value-0.0 is less than alpha (0.05), rainfall is highly significant to predicting label.
```

Step 4: Model Development

- Split data into training and testing sets.

Train Test Split

```
scaler=StandardScaler()
X=df_crop.drop('label',axis=1)
y=df_crop['label']
# X=scaler.fit_transform(X)
X_train, X_test, y_train, y_test=train_test_split(X,y,test_size=0.30,random_state=40)
# with open('scaler.pkl', 'wb') as scaler_file:
#     pickle.dump(scaler, scaler_file)
```

- Developed initial models using multiclass classification algorithms.

  Below we can see the Accuracy score, Recall score, Precision score and F1 score for all the multiclass algorithms (Refer ipynb file for the code)

Logistic Regression

```
accuracy_score: 0.9651515151515152
recall_score: 0.9609078636108669
precision_score: 0.9648625993767059
f1_score: 0.9619442953565263
```

SGD Classifier

```
accuracy_score: 0.9833333333333333
recall_score: 0.7032815923098835
precision_score: 0.7696594944768901
f1_score: 0.690144602791304
```

Naïve Bayes

```
accuracy_score: 0.9939393939393939
recall_score: 0.993006993006993
precision_score: 0.9948051948051948
f1_score: 0.9934573002754821
```

Decision Tree

```
accuracy_score: 0.9318181818181818
recall_score: 0.9296593829686155
precision_score: 0.9459310422600308
f1_score: 0.9242112490449521
```

Random Forest Classifier

```
accuracy_score: 0.9954545454545455
recall_score: 0.9950372208436725
precision_score: 0.9953409090909091
f1_score: 0.995161701044054
```

Tuned Decision Tree

```
accuracy_score: 0.9863636363636363
recall_score: 0.9862027686980248
precision_score: 0.9855854920169438
f1_score: 0.9857098789755575
```

Tuned Random Forest

```
accuracy_score: 0.9984848484848485
recall_score: 0.9982517482517482
precision_score: 0.9985795454545454
f1_score: 0.9983872336813513
```

Bagging using Tuned RF

```
accuracy_score: 0.996969696969697
recall_score: 0.9965682465682465
precision_score: 0.996896043771043
f1_score: 0.996671967815142
```

Adaboost

```
accuracy_score: 0.9954545454545455
recall_score: 0.9950372208436725
precision_score: 0.9953409090909091
f1_score: 0.995161701044054
```

Gradient boost

```
accuracy_score: 0.9893939393939394
recall_score: 0.988597683151763
precision_score: 0.9894499837335203
f1_score: 0.9889048272895544
```

XGBoost

```
accuracy_score: 0.996969696969697
recall_score: 0.9963636363636365
precision_score: 0.9972078508663874
f1_score: 0.9966872604451219
```

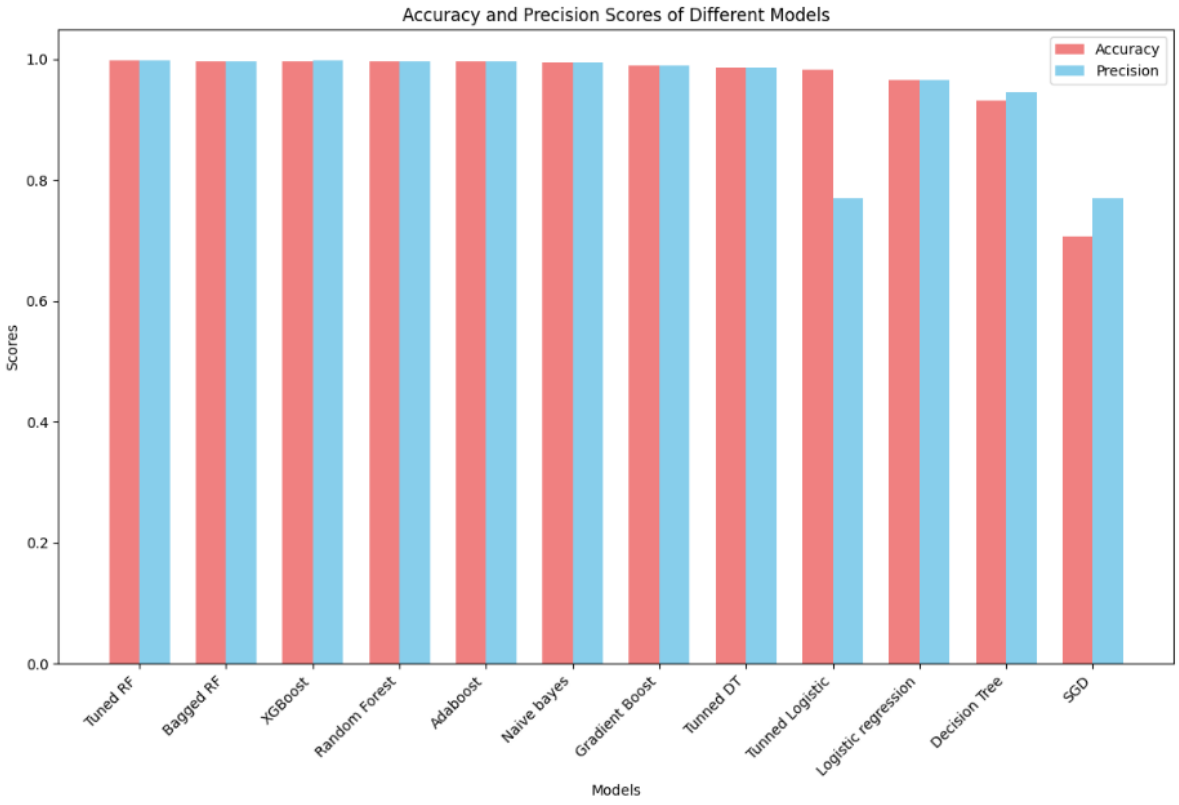
- Enhanced models with Decision Trees and ensemble methods (Random Forest, XGBoost).

Step 5: Model Evaluation

- Evaluated models using accuracy, recall, precision, and F1 score.

	Model	Accuracy	Recall	Precision	F1_score
7	Tuned RF	0.998	0.998	0.999	0.998
8	Bagged RF	0.997	0.997	0.997	0.997
11	XGBoost	0.997	0.996	0.997	0.997
5	Random Forest	0.995	0.995	0.995	0.995
9	Adaboost	0.995	0.995	0.995	0.995
3	Naive bayes	0.994	0.993	0.995	0.993
10	Gradient Boost	0.989	0.989	0.989	0.989
6	Tunned DT	0.986	0.986	0.986	0.986
2	Tunned Logistic	0.983	0.703	0.770	0.690
0	Logistic regression	0.965	0.961	0.965	0.962
4	Decision Tree	0.932	0.930	0.946	0.924
1	SGD	0.706	0.703	0.770	0.690

- Conducted cross-validation to ensure robustness.

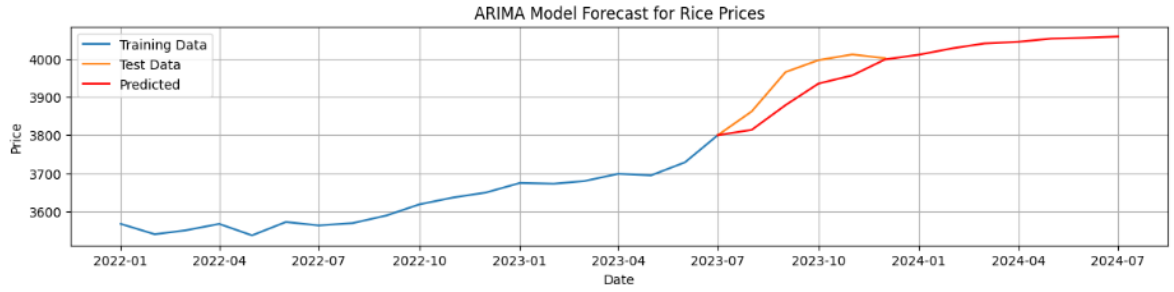


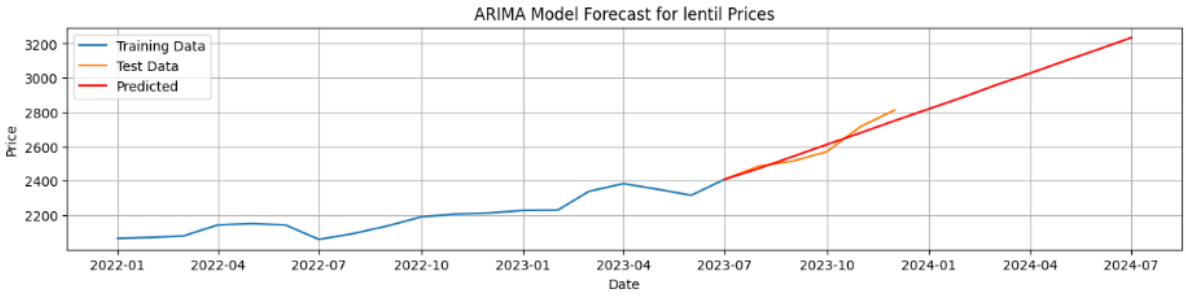
Step 6: Model Tuning

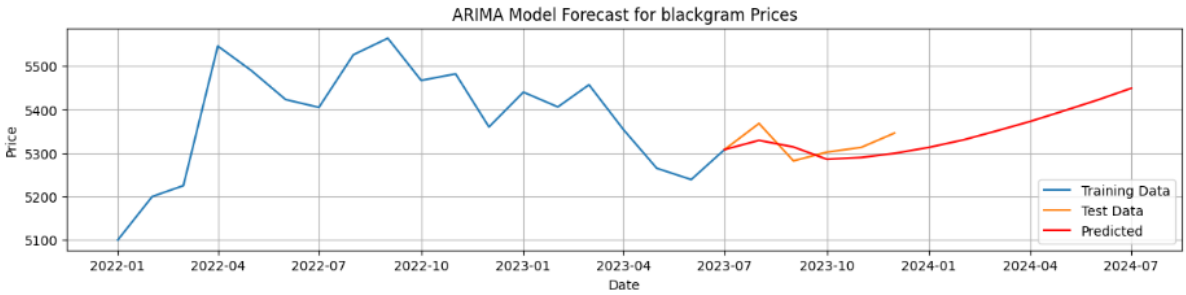
- Fine-tuned hyperparameters using GridSearchCV.
- Improved model performance by adjusting parameters and ensemble methods.
- **Finalized a tuned Random Forest model as the best model.**

Step 7: Price Forecasting

- ARIMA and SARIMAX models were implemented for price forecasting.
- Permutation and combination techniques were used to find the best order for each crop, ensuring optimal model performance.
- The models were evaluated using RMSE to measure their accuracy.
- Suitable crops for ARIMA and SARIMAX models were identified, and price forecasts were made until July 2024.







Model Evaluation

Final Model

In this Crop Recommendation and Price Prediction project, the final model employed was a tuned Random Forest model.

The primary objective was to maximize accuracy, recall, precision, and F1 score for crop prediction to ensure reliable and robust predictions. Additionally, efforts were made to minimize the Root Mean Square Error (RMSE) for price forecasting, enhancing the precision of future price estimates. Outliers were identified and plotted to understand their influence on the data, though they were not removed to maintain the dataset's integrity. Various statistical tests and visualizations were performed to analyse the distribution and relationships between variables.

Price forecasting was carried out using ARIMA and SARIMAX models, with suitable models identified for each crop, generating reliable forecasts up to July 2024. This comprehensive approach provides valuable insights for stakeholders, aiding in strategic planning and decision-making in the agricultural sector.

Prominent Parameters

- Several prominent parameters were carefully optimized to enhance the performance of the final tuned Random Forest model.
- These parameters included the ***number of trees***, which determines the number of decision trees in the forest, ***the learning rate***, which controls the contribution of each tree to the final prediction, the ***maximum depth for trees***, which limits the depth of each individual tree to prevent overfitting, and the minimum samples split, which specifies the minimum number of samples required to split an internal node.

Evaluation Metrics

The evaluation metrics were carefully chosen to ensure comprehensive assessment of the model's performance.

For crop prediction, the key metrics included **accuracy, recall, precision, and F1 score**, each providing a different perspective on the model's ability to correctly identify crop types.

For price forecasting, the **Root Mean Square Error (RMSE)** was used to measure the differences between predicted and actual prices, aiming to minimize this value for more accurate forecasts. These evaluation metrics collectively ensured that both the classification and regression aspects of the project were rigorously evaluated for optimal performance.

Model Performance

- The final tuned Random Forest model performed better than the initial models in terms of accuracy, recall, precision, and F1 score for predicting crop types. The low RMSE showed that the price forecasting was very accurate. These improvements ensured that the model provided reliable predictions for both crop types and prices, giving useful information for people involved in agriculture.

Comparison to Benchmark

Benchmark

- In CropWise, initial models, such as basic classification models, served as the benchmark.
- These benchmarks were based on simple models without hyperparameter tuning, providing a starting point for comparison with the final tuned Random Forest model. The improvements in accuracy, recall, precision, F1 score, and RMSE demonstrated the effectiveness of the advanced model over these initial benchmarks.

Improvement

- In the Crop Recommendation and Price Prediction project, the final ensemble model significantly outperformed the benchmark in terms of accuracy, recall, precision, and F1 score for crop prediction.
- These improvements were achieved by finding the best hyperparameters, which enhanced the model's performance and reliability. This optimization ensured more accurate and robust predictions compared to the initial basic models.

Visualizations

Key Visualizations

- **Distribution Plots:** These showed how data is spread out for each feature, giving us a clear picture of their ranges.
- **Boxplots:** These helped spot outliers in the data, highlighting any unusual values that might affect our analysis.
- **Heatmaps:** These displayed correlations between different variables, helping us see how they relate to each other.
- **Model Performance Plots:** These graphs allowed us to compare the actual values of crop types and prices with the values predicted by our models, showing us how well our predictions matched reality.

These visualizations were essential for exploring the data, understanding relationships between variables, identifying anomalies, and evaluating how accurately our models predicted crop types and prices.

Implications

Impact

- Enhanced prediction accuracy from our models can empower farmers and stakeholders by providing reliable insights into crop production and pricing. This enables them to make more informed decisions, such as when to plant, harvest, or sell their crops, optimizing their operations and maximizing profitability.
- Moreover, integrating these models into agricultural advisory systems can offer real-time recommendations based on current market conditions and predictive trends.
- This capability not only improves decision-making efficiency but also supports sustainable agricultural practices by aligning production with market demand and economic realities.

Recommendations

- Utilize the model for strategic planning and resource allocation.
- Regularly update the model with new data to maintain accuracy.

Limitations

Model Limitations

Unseen Data Variability: Model performance may differ when applied to new, unseen data.

Data Quality Constraints: Model accuracy is limited by the quality and quantity of available historical data.

Seasonal and External Factors: The models may not fully account for seasonal variations and external factors like policy changes, impacting prediction accuracy.

Enhancements

- **Diverse Dataset Integration:** Enhance model robustness by incorporating diverse datasets, which can provide a broader perspective and improve accuracy.
- **Advanced Techniques Implementation:** Utilize advanced techniques such as deep learning to potentially achieve higher performance and more nuanced predictions.
- **Incorporation of External Factors:** Improve model relevance and reliability by including external factors such as policy changes and economic trends, ensuring more comprehensive and adaptive forecasting capabilities.

Closing Reflections

Learnings

- **Data Preprocessing and Feature Engineering:** Effective preprocessing and feature engineering are critical for enhancing model performance and accuracy.
- **Ensemble Methods:** Ensemble methods offer robust solutions for predictive modelling by combining multiple models to improve predictions.
- **Continuous Evaluation and Tuning:** Ongoing evaluation and fine-tuning are essential to ensure that models maintain high accuracy and relevance over time.

Future Work

- **Explore Additional Features and External Datasets:** Incorporate more diverse features and external datasets to enrich model inputs and improve prediction capabilities.
- **Experiment with Deep Learning Models:** Investigate the use of deep learning models to leverage complex patterns and potentially enhance predictive performance.
- **Develop Real-Time Prediction System:** Build a real-time prediction system for continuous updates, providing stakeholders with timely and accurate insights for decision-making in dynamic agricultural environments.