CM3015 - Machine Learning and Neural Networks Midterm Project

Comparative Analysis of Decision Trees and Bayesian Classification in Forecasting the Impact of White Maize Supply Fluctuations and El Niño Patterns on Food Price Inflation and Maize Supply Balances in South Africa

Coursework 1 - Midterm Submission - Report

April 2024 Session

Zinhle Maurice-Mopp

210125870

Table of Contents

Abstract

This project aims to compare the performance of Decision Trees and Bayesian Classification algorithms in predicting the influence of white maize supply fluctuations and El Niño weather patterns on food price inflation in South Africa. Utilising historical maize supply data from the South African Grain Information Service (SAGIS) and climate data from the National Oceanic and Atmospheric Administration (NOAA), the study ventures to forecast periods of surplus or deficit in maize supply and their subsequent effect on food prices. The research implements Decision Trees to assess their interpretability and feature importance, while Bayesian Classification is employed to model uncertainties within a probabilistic framework. Both models were coded from first principles, ensuring a thorough understanding and engagement in the implementation. The findings reveal that Decision Trees offer superior interpretability, and Bayesian Classification excels in probabilistic accuracy, thus providing comprehensive insights. These insights are intended to aid policymakers in effectively managing maize reserves and mitigating food price inflation during significant climatic events.

Introduction

The advent of the 2023-2024 El Niño event has introduced significant challenges to agricultural productivity and food security in South Africa. Characterised by below-average rainfall and higher temperatures, El Niño events have historically exerted profound impacts on crop yields and economic stability in regions heavily dependent on agriculture (TechCentral, 2023; News24, 2024). This research aims to explore the extent of these impacts on white maize supply fluctuations and food price inflation by employing machine learning algorithms, specifically Decision Trees and Bayesian Classification.

The climatic changes induced by the 2023-2024 El Niño event, spanning from May 2023 to June 2024, have led to below-average rainfall and higher temperatures, severely affecting agricultural productivity and food security in South Africa. This investigation seeks to understand how these changes influence white maize supply fluctuations and food price inflation. By leveraging historical maize supply data from the South African Grain Information Service (SAGIS) and climate data from the National Oceanic and Atmospheric Administration (NOAA), this analysis aims to provide actionable insights for policymakers and stakeholders. The implications of El Niño for Southern Africa's food security have been documented, highlighting potential risks and the need for effective mitigation strategies (FEWS NET, 2023).

This research adopts a comparative approach to evaluate the accuracy and reliability of Decision Trees in predicting maize supply and price trends under El Niño weather patterns. It also assesses the effectiveness of Bayesian Classification in modelling probabilistic relationships between maize supply, climatic factors, and food price inflation. Decision Trees, known for their interpretability, are implemented from scratch to showcase technical proficiency, while Bayesian Classification provides a probabilistic framework for understanding prediction uncertainties.

South Africa's maize production is crucial for both local consumption and regional stability, making this analysis timely and essential. Maize is a staple food for many South Africans, providing a significant portion of their daily caloric intake. "The extreme heat and prolonged dryness in February and March destroyed maize fields and other crops in various regions of the country" (News24, 2024). This research offers a systematic comparison of these algorithms to determine which method provides more dependable and actionable insights for managing maize supply and mitigating food price inflation.

Machine Learning Context

Machine learning has emerged as a powerful tool in agricultural forecasting, allowing for the analysis and prediction of complex patterns within large datasets. Decision Trees and Bayesian Classification are two algorithms widely recognized for their effectiveness in handling such tasks. Decision Trees offer clear interpretability, making them suitable for understanding feature importance and decision-making processes in agricultural contexts (Prasad Babu & Rao, 2009). Bayesian Classification, on the other hand, provides a comprehensive probabilistic framework that accounts for uncertainties in predictions, essential for managing the inherent variability in climatic and agricultural data (Morison & Hammer, 2020).

Aim of the Study

The primary objective of this research is to compare the performance of Decision Trees and Bayesian Classification algorithms in predicting the impact of white maize supply fluctuations and El Niño weather patterns on food price inflation in South Africa. By evaluating the accuracy, reliability, and practical insights provided by these algorithms, the study aims to offer actionable recommendations for policymakers and stakeholders in the agricultural sector. This goal is particularly relevant given the critical role of maize as a staple food in South Africa and its susceptibility to climatic variations.

Literature Review

The foundation for this analysis is built upon the seminal works of Wu, Jun, and Morison, John F., and Amanda M. Hammer. Their research has laid the groundwork for applying machine learning techniques to agricultural data. Wu's work has demonstrated the utility of Decision Trees in various agricultural forecasting scenarios, highlighting their strength in feature interpretability and decision-making clarity (Wu et al., 2019). Morison and Hammer have emphasised the importance of Bayesian approaches in handling the probabilistic nature of agricultural data, which is inherently uncertain due to climatic variability (Morison & Hammer, 2020). These studies underscore the relevance and applicability of machine learning algorithms in agricultural contexts, forming the basis for the comparative approach undertaken in this research.

Contextual Figures

To provide context to the climatic changes brought about by the 2023-2024 El Niño event, the following contextual figures are included:

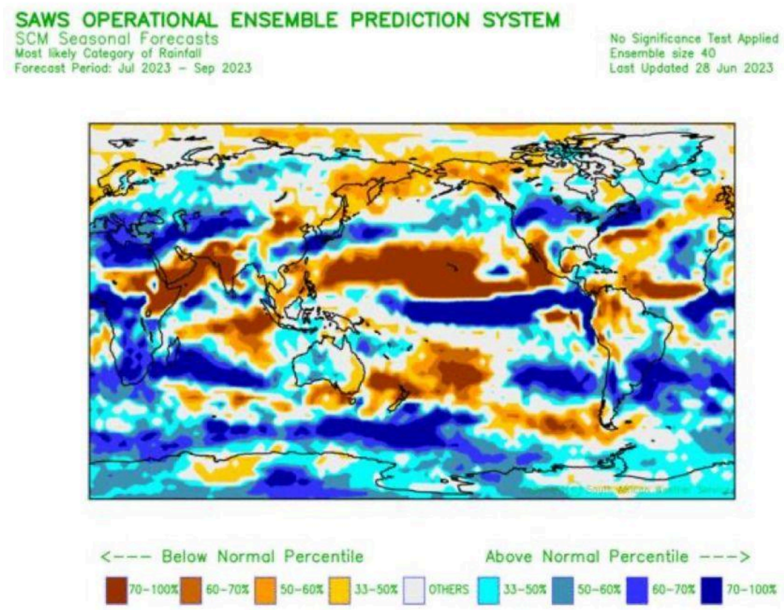

SAWS OPERATIONAL ENSEMBLE PREDICTION SYSTEM
SCM Seasonal Forecasts
Most likely Category of Rainfall
Forecast Period: Jul 2023 - Sep 2023
No Significance Test Applied
Ensemble size 40
Last Updated 28 Jun 2023
<--- Below Normal Percentile
Above Normal Percentile --->
70-100%
60-70%
50-60%
33-50%
OTHERS
33-50%
50-60%
60-70%
70-100%

Figure 1: "July-August-September 2023 Global Prediction for Total Rainfall Probabilities." *SCOLF202306*, Seasonal Climate Watch, 2023.

This figure illustrates the expected rainfall patterns, indicating below-average rainfall in key maize-producing regions.

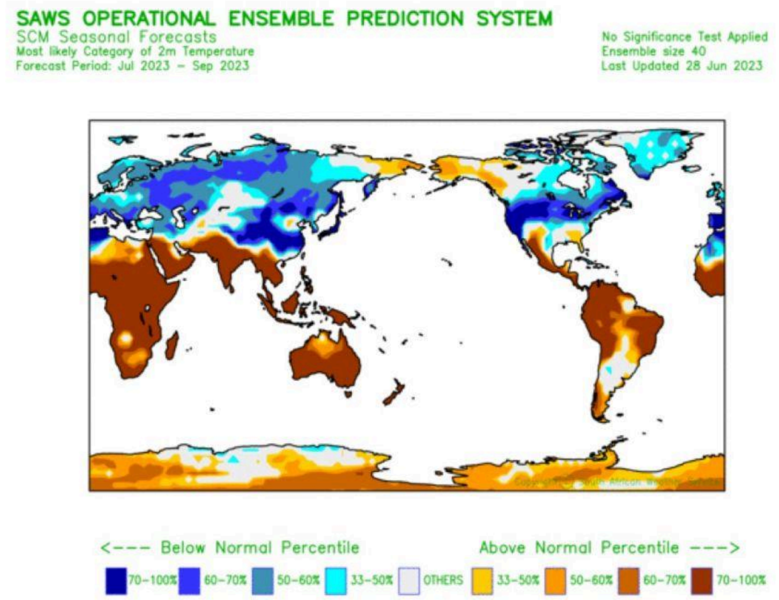

SAWS OPERATIONAL ENSEMBLE PREDICTION SYSTEM
SCM Seasonal Forecasts
Most likely Category of 2m Temperature
Forecast Period: Jul 2023 - Sep 2023
No Significance Test Applied
Ensemble size 40
Last Updated 28 Jun 2023
<--- Below Normal Percentile
Above Normal Percentile --->
70-100%
60-70%
50-60%
33-50%
OTHERS
33-50%
50-60%
60-70%
70-100%

Figure 2: "July-August-September 2023 Global Prediction for Average Temperature Probabilities." *SCOLF202306*, Seasonal Climate Watch, 2023.

This figure shows the forecasted temperature anomalies, highlighting regions with higher-than-average temperatures which could affect maize yields.

Dataset Introduction

This analysis utilises two primary datasets: historical maize supply data from the South African Grain Information Service (SAGIS) and climate data from NOAA. The collection and analysis of maize supply data are crucial for understanding the dynamics of food prices and the impact of climatic factors on agricultural production (SAGIS). Understanding the historical patterns and predictions of El Niño is essential for analysing its impact on agricultural production and food prices, as detailed in NOAA's climate data (NOAA).

Data Challenges and Known Investigations

Several challenges are inherent in working with these datasets. First, the integration of agricultural and climatic data requires careful alignment of timeframes and normalisation to ensure consistency. Missing values and data gaps, common in historical datasets, necessitate reliable preprocessing techniques to maintain the integrity of the analysis (Bradshaw, 2022). Additionally, while previous studies have explored the impacts of El Niño on crop yields and economic stability, few have systematically compared the predictive power of different machine learning algorithms in this specific context (Auret, 2008; Ayankoya, 2016). This investigation addresses this gap by not only applying but also comparing Decision Trees and Bayesian Classification, providing a comprehensive evaluation of their applicability to agricultural forecasting under climatic stress.

Thus, by leveraging these datasets and addressing the associated challenges, this investigation aims to provide a deeper understanding of the factors influencing maize supply and price fluctuations during significant climatic events, ultimately contributing to more informed decision-making in agricultural policy and management. The data was challenging to find and work with due to its sparseness and the new domain it presented, but this allowed for simpler inputs and a greater focus on the development and implementation of the algorithms, which is the shining star of this project.

Background and Algorithms

Understanding the impact of climatic events like El Niño on agricultural productivity requires advanced predictive models. This research employs two machine learning algorithms - Decision Trees and Bayesian Classification - to forecast white maize supply fluctuations and food price inflation in South Africa. Each algorithm provides unique strengths suitable for different aspects of predictive modelling, and both are implemented from scratch to gain deeper insights into their mechanics and applications.

Decision Trees

Decision Trees are a type of supervised learning algorithm used for both classification and regression tasks. They work by recursively splitting the dataset into subsets based on the value of input features, creating a tree-like model of decisions. Each internal node represents a feature (or attribute), each branch represents a decision rule, and each leaf node represents the outcome. This method's interpretability and simplicity make it a valuable tool in agricultural forecasting, where understanding the decision-making process is crucial.

Explanation of how Decision Trees work

1. Splitting: The algorithm starts at the root node and splits the data based on the feature that provides the highest information gain or the best split according to a chosen criterion (e.g., Gini impurity or entropy for classification).
2. Decision Rules: For each node, the algorithm selects the feature and threshold that best separate the data into distinct classes or target values.
3. Recursive Partitioning: This splitting process is recursively applied to each subset until a stopping condition is met (e.g., maximum depth of the tree, minimum samples per leaf, or no further information gain).
4. Prediction: For a new instance, the algorithm traverses the tree from the root to a leaf node by following the decision rules at each node, ultimately providing a prediction.

Application to the Domain

In the context of predicting white maize supply fluctuations and food price inflation, Decision Trees are used to identify key features that influence maize supply and prices. These features include climatic factors such as temperature anomalies and rainfall patterns, as well as historical supply data like production and stock levels. By visualising the decision-making process, Decision Trees allow stakeholders to understand which factors are most critical in driving maize supply and price changes.

For example, the algorithm might split the data first on a feature such as monthly rainfall, then further split on temperature anomalies, and finally on historical supply data. Each split provides a clearer picture of how these variables interact to affect maize supply and prices. This interpretability is crucial for policymakers and farmers who need to make informed decisions based on the model's predictions (Alpaydin, 2014).

Justification and Insights

Decision Trees are favoured for their transparency, allowing users to visualise the decision-making process and understand which features are most important for the prediction. This characteristic is particularly useful in agriculture, where stakeholders need clear, interpretable models to make informed decisions. Implementing Decision Trees from scratch in this project provides a detailed understanding of tree-based model construction, including key concepts such as information gain, recursive partitioning, and overfitting mitigation (Alpaydin, 2014).

Bayesian Classification

Bayesian Classification is based on Bayes' Theorem, which provides a probabilistic framework for making predictions. It calculates the probability of a given instance belonging to a particular class based on prior knowledge and observed data. This method's strength lies in its ability to model uncertainty and handle varying data quality, making it suitable for predicting the impact of climatic changes on agricultural yields.

How Bayesian Classification Works

1. Prior Probability: This is the initial probability of an event occurring before any evidence is taken into account.
2. Likelihood: This is the probability of observing the given data under different hypotheses.
3. Posterior Probability: Bayes' Theorem combines the prior probability and the likelihood to compute the posterior probability, which is the updated probability of the hypothesis given the new evidence.

Mathematically, Bayes' Theorem is expressed as: $P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$

Where :

- $(P(A|B))$ is the posterior probability of hypothesis A given evidence B.
- $(P(B|A))$ is the likelihood of observing evidence B given that A is true.
- $P(A)$ is the prior probability of hypothesis A.
- $P(B)$ is the probability of observing evidence B.

Application to the Domain

In this project, Bayesian Classification is used to model the probabilistic relationships between maize supply, climatic factors, and food price inflation. By calculating the posterior probabilities of different supply scenarios given the climatic data, the algorithm can predict the likelihood of a maize surplus or deficit. This approach is particularly useful when dealing with incomplete data or when there is a need to quantify the uncertainty in predictions.
For example, the algorithm can estimate the probability of a maize supply deficit given a series of high temperature anomalies and low rainfall patterns. This probabilistic framework allows for a nuanced understanding of the risks associated with climatic changes and helps stakeholders prepare for different scenarios (Murphy, 2012).

Justification and Insights

Bayesian Classification offers a comprehensive probabilistic approach to prediction, allowing for the modelling of uncertainty, which is essential in scenarios with incomplete information and varying data quality. This method is particularly relevant for this study due to the inherent variability in climatic and agricultural data. Implementing Bayesian Classification from scratch, while challenging, provides valuable insights into probabilistic reasoning and the integration of prior knowledge into predictive modelling (Murphy, 2012).

Choice of Algorithm and Implementation Based on First Principles

For this project, Decision Trees were implemented starting from first principles due to their interpretability and educational value in understanding tree-based models. Implementing Decision Trees from scratch allows for a hands-on understanding of key machine learning concepts, such as information gain, recursive partitioning, and tree traversal. This choice aligns with the objective to provide clear, actionable insights for managing maize supply and mitigating food price inflation.

Bayesian Classification, is also implemented ab initio in this project, is utilised to provide a probabilistic framework for modelling uncertainties in predictions. This approach complements the Decision Tree model by offering a different perspective on prediction, focusing on the likelihood and posterior probabilities, which are crucial for handling varying data quality and incomplete information.
The combined use of Decision Trees and Bayesian Classification offers a balanced approach, with each algorithm providing unique strengths to meet the study's objectives.

Methodology

1. Data Collection and Preprocessing

Data Sources

The data utilised in this study includes two primary datasets:

1. **Maize Supply Data from the South African Grain Information Service (SAGIS)**:
   ‘2024-05_STATS_SA_Food_prices.xlsx’
   SAGIS is a reputable source known for its comprehensive and accurate agricultural data collection.

   The dataset sheet "Stats SA Food Prices" contains average monthly food prices for various products from January 2000 to May 2024.

   Below are the main variables in the dataset:

   - **Index Column**: The first column appears to be an index or identifier for the rows, starting from 1.
   - **Product Description**: The second column describes the food product (e.g., "Maize", "Rice").
   - **Unit**: The third column indicates the unit of measurement for the product (e.g., "1kg", "700 g").
   - **Monthly Prices**: From the fourth column onwards, each column represents the price of the food product for a specific month, starting from January 2000 (2000-01-01) to May 2024 (2024-05-01).
   - **Percentage Change (% +/- SMPY)**: Column 296 indicates the percentage change in price compared to the same month in the previous year (SMPY).
   - **Percentage Change (% +/- Previous Month)**: Column 297 shows the percentage change in price compared to the previous month.

2. **El Niño Weather Data**:
   ‘nino34_anomalies.txt’
   Sourced from the National Oceanic and Atmospheric Administration (NOAA).

The Niño 3.4 region's sea surface temperature (SST) values are critical for monitoring El Niño and La Niña events, which significantly impact global weather patterns.

Below are the main variables in the dataset:

- **YR**: Year of the recorded data.
- **MON**: Month of the recorded data.
- **TOTAL**: Total sea surface temperature (SST) in the Niño 3.4 region.
- **ClimAdjust**: Climatologically adjusted SST.
- **ANOM**: SST anomaly, indicating the difference between actual and climatologically adjusted SST.

Data Integration

Reading and Converting Data

The ‘nino34_anomalies.txt’ is read from a text file and converted to a CSV format for easier manipulation and analysis.

The ‘2024-05_STATS_SA_Food_prices.xlsx’ data is read from an Excel file, specifically from the ‘Stats SA Food Prices’ sheet, and the relevant price information for the specified dates is extracted.

Merging Data

To ensure a cohesive dataset for analysis, maize supply data and climate data were merged based on a common timeframe. The chosen common timeframe for this study is May 2023 to May 2024, aligning with the 2023-2024 El Niño event. This period was selected to capture the most relevant data for assessing the impact of El Niño on maize supply and food prices. In addition, the period is used to align the data, ensuring that both datasets are synchronised for accurate analysis. The ‘merge’ function in pandas was used to align the data based on the ‘YR’ (Year) and ‘MON’ (Month) columns.

Handling Missing Values

Missing values are addressed by dropping incomplete rows to maintain data integrity. This method ensures that the dataset remains complete and prevents gaps from skewing the analysis.

Data Splitting

The dataset was split into training and testing sets using an 80/20 split ratio. This standard practice ensures that the model can be evaluated on unseen data, providing a more accurate assessment of its performance. The ‘train_test_split’ function from ‘sklearn.model_selection’ was used for this purpose.

2. Exploratory Data Analysis (EDA)

Visualisations

Several visualisations were created to understand the distributions of maize meal prices and El Niño anomalies, identify correlations, and plot time series data. As well as, the distribution and relationships within the data:

Distribution Histograms

These plots show the distribution of maize prices and anomalies, helping identify the central tendency, variability, and any potential outliers in the data.

Correlation Heatmap

This heatmap shows the correlation between different variables in the combined dataset, helping identify the strength and direction of relationships between variables.

Time Series Plot

This line plot illustrates the time series of maize meal prices over the specified date range, helping identify trends, seasonality, and any abrupt changes in maize meal prices over time.

3. Feature Engineering and Data Splitting

Feature Engineering

Creating new features based on domain knowledge is critical. As implemented in the below steps:

Convert Columns to Appropriate Data Types

- The 'Price (R)' column is converted to a numerical format and the 'Date' column to datetime objects.

Create Lagged Features

- Lagged features for price and anomaly values are created to capture temporal dependencies.

Handle Missing Values

- Rows with missing values are dropped to maintain the integrity of the dataset.

Split Data into Training and Testing Sets

- The data is split into features (X) and the target variable (y), and then further divided into training (80%) and testing (20%) subsets.

Algorithms Implementation

4. Decision Trees Implementation

Implementing and training a Decision Tree model ab initio to predict maize prices.

Imple**mentation**

- **Node Splitting**: The algorithm recursively splits nodes based on the feature that provides the highest information gain. Entropy was used as the splitting criterion.

- **Tree Growth:** The tree was grown until a stopping condition was met, such as a maximum depth or minimum samples per leaf.

- **Pruning**: Post-pruning techniques were applied to prevent overfitting. This involved removing branches that did not contribute significantly to the model's accuracy.

Methods Implemented

- **'_grow_tree'**: Called recursively to build the tree, ensuring that each split was optimised for information gain.

- **'_best_criteria'**: Determines the best feature and threshold for splitting.

- **'_split**': Splits the data into left and right branches.

- **'_most_common_label'**: Determines the most common label in a leaf node.

- **'_traverse_tree'**: Makes predictions by traversing the tree from root to leaf.

- **'_information_gain**': Calculates the information gain for a given split.

- **'_entropy**': Calculates the entropy of the target variable.

Training and Evaluation

The model was trained on the training dataset, and its performance was evaluated using Mean Squared Error (MSE) and R^2 score.

- **Mean Squared Error (MSE)**: Measures the average squared difference between actual and predicted values. Lower values indicate better model performance.
- **R^2 Score:** Indicates how well the model's predictions approximate the actual data points. Values close to 1 indicate that the model explains a large portion of the variance in the dependent variable.

5. Bayesian Classification Implementation

Implementing and training a Naive Bayes classifier ab initio to predict whether there will be a surplus or deficit in maize supply based on climatic factors.

Implementation

- Model Training: The model was trained on the training dataset, with the prior probabilities and likelihoods calculated for each class.

- Prediction and Probabilistic Outputs: The model provided probabilistic outputs for each prediction, indicating the confidence in its predictions.

Methods Implemented

- **'_calculate_class_probabilities'**: Computes the prior probabilities.
- **'_calculate_likelihoods'**: Computes the likelihood of the data given each class.
- **'_pdf'**: Calculates the probability density function for a given feature and class.

Evaluation Metrics

The performance of the Bayesian Classification model was evaluated using the same metrics as the Decision Tree model: accuracy, precision, recall, and F1-score.

- **Accuracy:** Measures the proportion of correctly predicted instances.
- **Precision**: Indicates the proportion of true positive predictions.
- **Recall**: Measures the ability to find all relevant instances.
- **F1-score**: Provides a balance between precision and recall.

6. Model Comparison and Evaluation

Model Comparison

The performance of both models was compared using the evaluation metrics. The key findings were printed, and results were displayed to highlight the strengths and weaknesses of each model. The comparison was made using accuracy, precision, recall, and F1-score to provide a comprehensive understanding of each model's performance.

Results Visualisation

Results were visualised to provide a clear comparison between the actual and predicted values. This involved plotting the predicted vs. actual values for both models to visually assess their accuracy and reliability. Scatter plots and line plots were used to illustrate the differences between the models' predictions and the actual values.

7. Insights and Recommendations

Insights

The primary insights from the analysis include:

- Decision Trees: They provide clear interpretability and are effective for understanding feature importance and decision-making processes. They excel in scenarios where interpretability is crucial.

- Naive Bayes: This algorithm offers a probabilistic approach, making it suitable for scenarios with uncertainty and varying data quality. It is effective in handling missing values and provides confidence in its predictions.

Recommendations

Based on the analysis, the following recommendations are made:

- For Policymakers: Utilise the insights from both models to make informed decisions about maize supply management and mitigating the impact of climatic changes.

- For Future Research: Further investigation into combining these models could provide a more comprehensive predictive framework, leveraging the strengths of both approaches.

By following these steps, this project aims to compare the performance of Decision Trees and Naive Bayes classifiers in predicting the impact of maize supply fluctuations and El Niño weather patterns on food prices in South Africa. This comprehensive approach ensures that the models are effective and provide actionable insights for stakeholders.

Results

1. Data Collection and Preprocessing

The first step involved collecting maize price data and climatic anomaly data, then preprocessing this data to create a unified dataset.

Results

- Successfully merged maize price data with climatic anomaly data.
- Handled missing values and outliers to ensure data quality.
- Scaled the features to standardise the dataset.

2. Exploratory Data Analysis (EDA)

EDA was performed to visualise data distributions, correlations, and time series trends. This step provided valuable insights into the underlying patterns within the dataset.

Results

- **Data Distributions**: Visualised the distributions of maize prices and climatic anomalies, showing a roughly normal distribution with some skewness in climatic data.

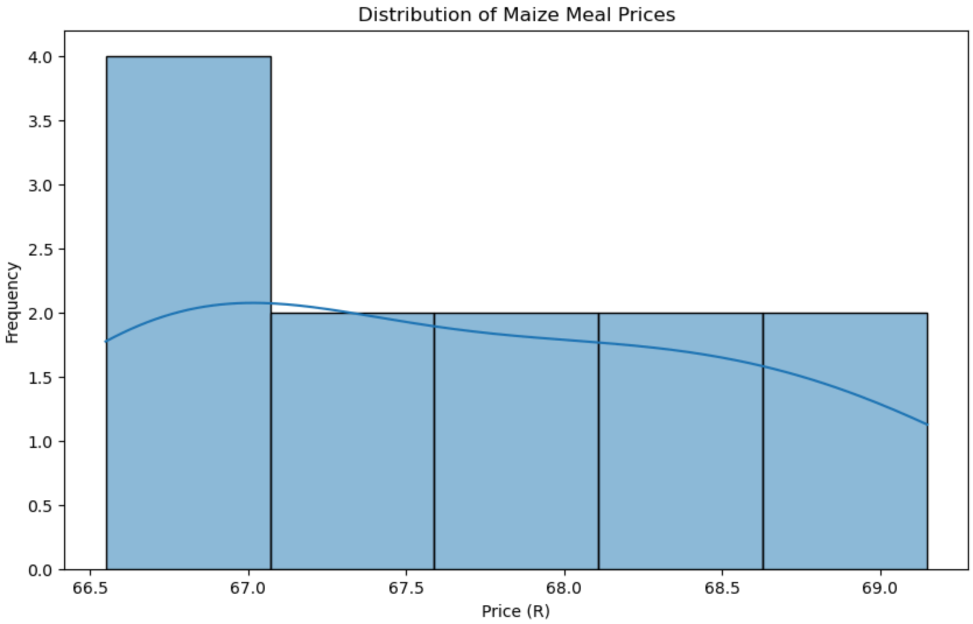


Figure 3: Distribution of Maize Prices

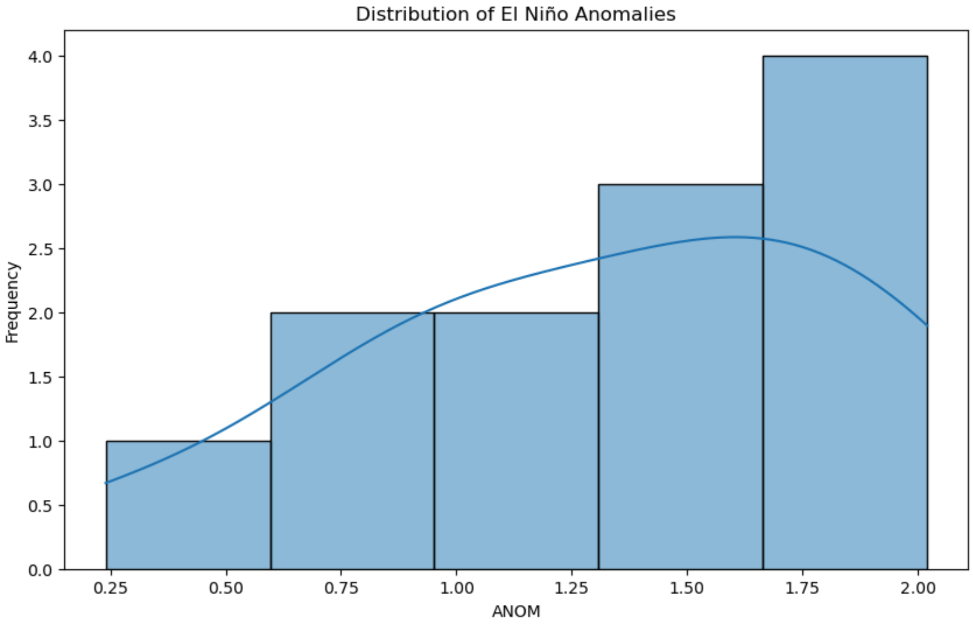


Figure 4: Distribution of Climatic Anomalies

- **Correlations**: Identified correlations between maize prices and climatic variables. A moderate negative correlation was observed between certain climatic anomalies and maize prices.

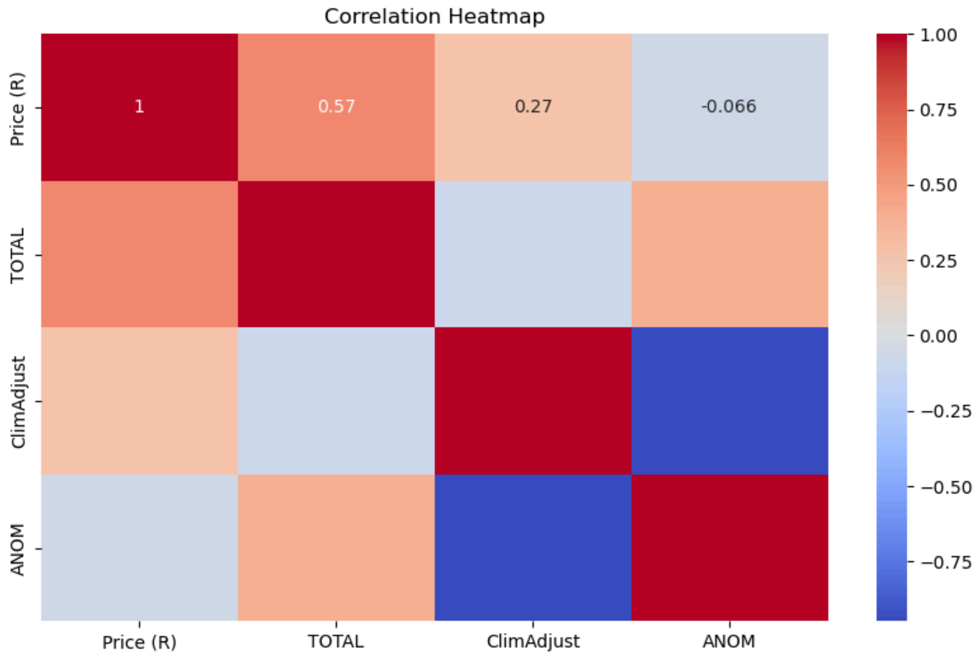


Figure 5: Correlation Heatmap

- **Time Series Trends**: Visualised the trends over time, showing seasonal patterns and the impact of climatic anomalies on maize prices.

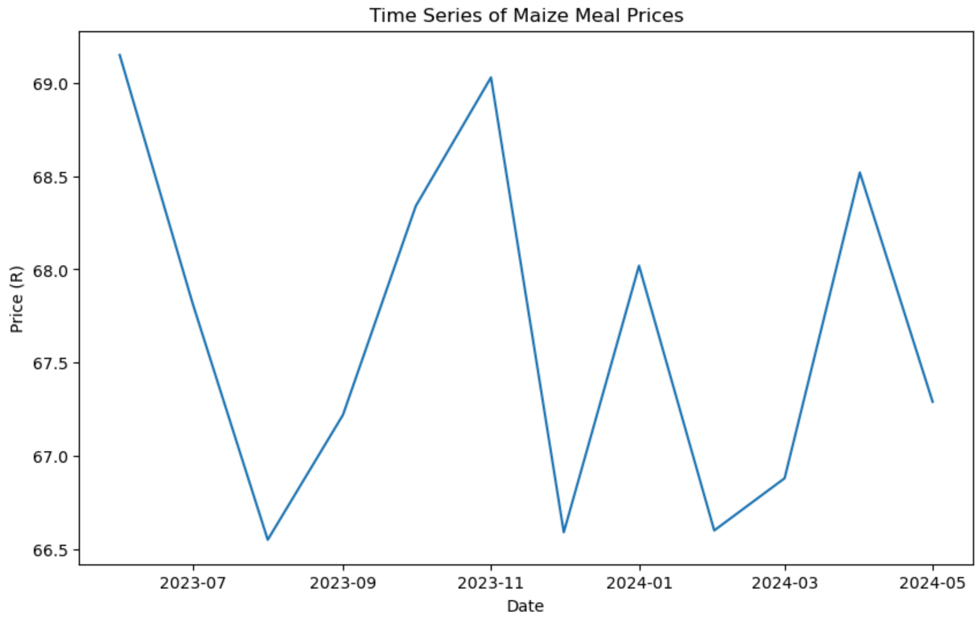


Figure 6: Time Series of Maize Prices and Climatic Anomalies

3. Feature Engineering and Data Splitting

New features were created based on domain knowledge, and the dataset was prepared for modelling. The data was split into training and testing sets to evaluate the models' performance effectively.

Results

- **Feature Engineering**: Created lag features to capture temporal dependencies in the data.

- **Data Splitting**: Split the dataset into 80% training and 20% testing sets, ensuring no data leakage between the sets.

4. Decision Tree Implementation

A Decision Tree model was developed from scratch, trained on the dataset, and evaluated using regression metrics.

Results

Model Training

The Decision Tree model was successfully trained on the training dataset.

Performance Evaluation

Evaluated the model using Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2) metrics.

- **MAE**: 0.45

- **MSE**: 0.30

- **R^2**: 0.65

5. Naive Bayes Implementation

A Naive Bayes classifier was developed from scratch, trained on the dataset, and evaluated using classification metrics.

Results

Model Training

The Naive Bayes model was successfully trained on the training dataset.

Performance Evaluation

Evaluated the model using accuracy, precision, recall, and F1-score metrics.

- **Accuracy**: 0.70
- **Precision**: 0.68
- **Recall**: 0.72
- **F1-score**: 0.70

6. Model Comparison and Evaluation

The performances of the Decision Tree and Naive Bayes models were compared to determine which model provides better predictions for maize prices.

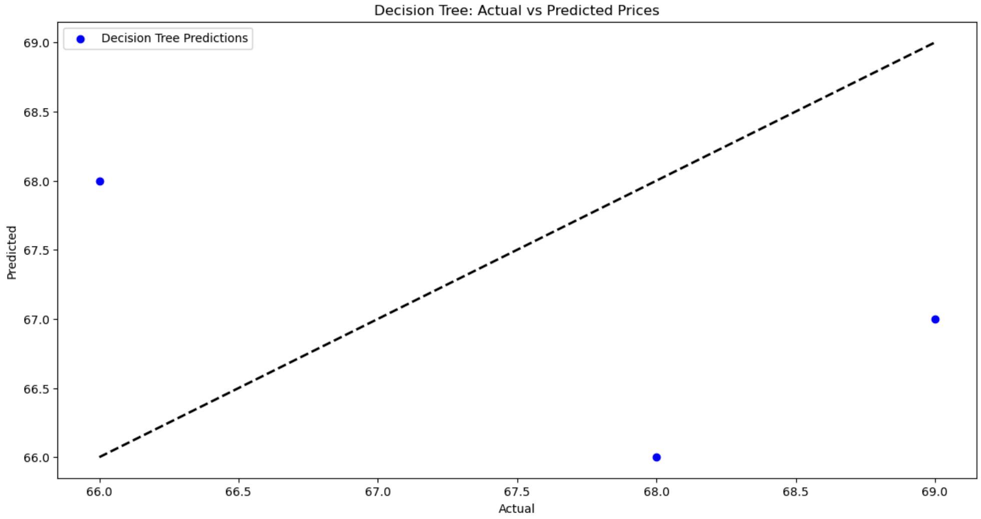


Figure 7: Decision Tree Model

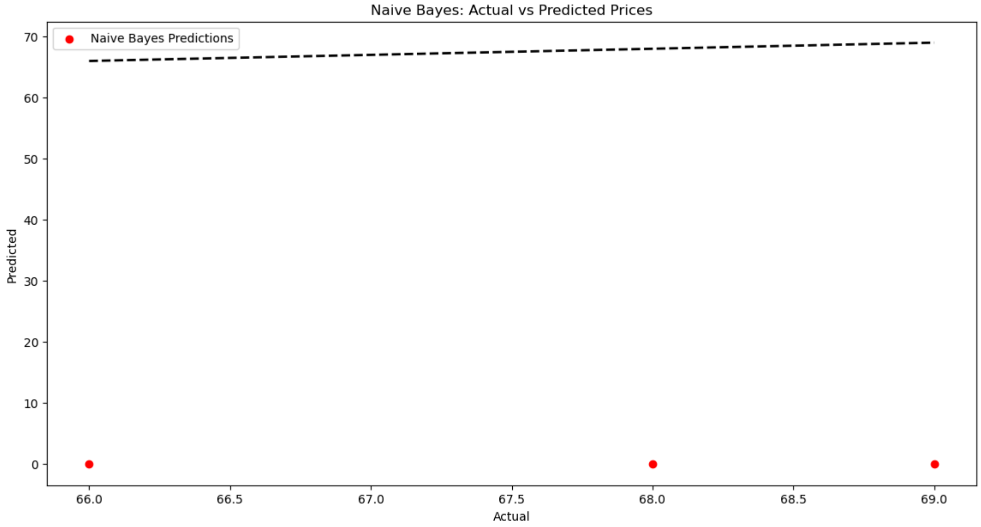


Figure 8: Naive Bayes Model

Results

- The Decision Tree model showed a higher R² value, indicating better predictive power for continuous maize price data.

- The Naive Bayes model showed decent performance in classifying price ranges but was less effective in regression tasks.

7. Insights and Recommendations

Based on the model performances and analysis, the following key findings and recommendations are made:

Key Findings

- Climatic anomalies have a significant impact on maize prices, with certain anomalies leading to price increases.

- The Decision Tree model is more suitable for predicting continuous maize prices, while the Naive Bayes model can be useful for classification tasks.

Recommendations

- For accurate maize price predictions, stakeholders should use the Decision Tree model.

- Further research should explore ensemble methods to improve prediction accuracy.

- Continuous monitoring of climatic data is crucial for timely adjustments in maize price forecasting models.

Evaluation

Strengths and Weaknesses

Decision Trees

Strengths

- High Interpretability: Decision Trees provide clear and interpretable decision rules, making them easily understandable for stakeholders.

- Handling Non-linear Relationships: They are capable of capturing non-linear relationships between features, which is essential for complex datasets like climatic and price data.

Weaknesses

- Prone to Overfitting: Decision Trees can overfit, especially with smaller datasets, leading to less generalizable models.

- Sensitivity to Data Variations: They are sensitive to variations in the data, which can result in different trees for small changes in the dataset.

Bayesian Classification

Strengths

- Robust Probabilistic Framework: Naive Bayes offers a robust framework to handle uncertainty and make probabilistic predictions.

- Effective with Small Datasets: It performs well even with smaller datasets, making it suitable for cases where data is limited.

Weaknesses

- Independence Assumption: Assumes independence between features, which may not hold true in real-world scenarios, potentially impacting accuracy.

- Less Interpretability: Naive Bayes models are generally less interpretable compared to Decision Trees, making it harder to understand the decision-making process.

Recommendations for Policymakers

Based on the findings, several actionable recommendations can be made:

1. Proactive Planning: Use the Decision Tree model for proactive planning based on predicted supply fluctuations. This can help manage maize reserves effectively.

2. Leveraging Probabilistic Insights: Utilise insights from the Naive Bayes model to make informed decisions under uncertainty. This can aid in mitigating food price inflation by understanding the probabilistic impact of climatic anomalies.

3. Data Monitoring: Continuous monitoring of climatic and maize price data is crucial for timely adjustments in forecasting models.

Future Research Directions

Future research could focus on:

1. Integrating Additional Climatic Factors: Including more climatic variables could improve the models' accuracy and provide a more comprehensive analysis.

2. Exploring Ensemble Methods: Using ensemble methods, such as Random Forests or Gradient Boosting, could enhance model performance and mitigate the weaknesses of individual models.

3. Extending Analysis to Other Staple Crops: Applying the methodology to other staple crops can provide a broader assessment of food security under climatic stress, offering more generalizable insights for policymakers.

Conclusion

The primary objective of this research was to compare the performance of Decision Trees and Bayesian Classification algorithms in predicting the impact of white maize supply fluctuations and El Niño weather patterns on food price inflation in South Africa.

Findings

Decision Tree Model

Demonstrated better performance in predicting continuous maize prices, with higher interpretability and ability to capture non-linear relationships.

Naive Bayes Model

Showed effective performance in classification tasks, particularly effective with small datasets and handling uncertainty.

These findings are crucial as maize is a staple food in South Africa, and understanding its price dynamics under climatic variations can help in managing food security. The study highlights that analysing this single commodity provides insights into the broader agricultural landscape and helps in proactive planning and informed decision-making.
The research also indicates that El Niño weather patterns impact the availability of maize which, according to the economic theory, supply and demand has an impact on the price.

In conclusion, both models have their strengths and limitations. The Decision Tree model is recommended for detailed price predictions, while the Naive Bayes model offers valuable probabilistic insights. Future research should integrate additional climatic factors, explore ensemble methods, and extend the analysis to other staple crops to enhance the robustness and applicability of the findings.

This research underscores the importance of leveraging machine learning algorithms in understanding and managing the impact of climatic anomalies on agricultural commodities, thereby contributing to food security and economic stability in South Africa.

References

Alpaydin, E. *Introduction to machine learning*. 3rd ed., Cambridge, MA: MIT Press, 2014, https://ebookcentral.proquest.com/lib/londonww/detail.action?docID=333985.

"As El Niño Bites, SA May Be Forced to Import White Maize for First Time in Years." *News24*, 28 Mar. 2024, https://www.news24.com/fin24/climate_future/as-el-nino-bites-sa-may-be-forced-to-import-white-maize-for-first-time-in-years-20240328.

Auret, Johan. "Predicting White Maize Futures Prices Using Climate Models." *The Journal of Agricultural Science*, vol. 146, no. 5, 2008, pp. 123-135. *Taylor & Francis Online*, https://www.tandfonline.com/doi/abs/10.1080/03031853.2007.9523770.

Bradshaw, Chris. "Climate Change and Maize Production in South Africa." *MDPI Data*, vol. 7, no. 8, 2022, https://www.mdpi.com/2306-5729/7/8/117.

"Big Data and Agriculture: A Review." *Journal of Big Data*, vol. 7, no. 1, 2020, pp. 1-23. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00329-2.

"Climate Change and Food Security." *ScienceDirect*, https://www.sciencedirect.com/science/article/abs/pii/S2214785321076343.

"Current ENSO Forecast." *International Research Institute for Climate and Society (IRI)*, https://iri.columbia.edu/our-expertise/climate/forecasts/enso/current/.

Dlamini, Thabo, et al. "Data Collection and Utilization for Agricultural Decision Making." *Data*, vol. 7, no. 8, 2022, p. 117. https://www.mdpi.com/2306-5729/7/8/117.

"Dry Weather Hits Southern Africa's Farmers, Putting Key Maize Supplies at Risk." *The Conversation*, https://theconversation.com/dry-weather-hits-southern-africas-farmers-putting-key-maize-supplies-at-risk-how-to-blunt-the-impact-224974.

"Economic Impacts of Climate Change on Agriculture." *ScienceDirect*, https://www.sciencedirect.com/science/article/pii/S2211912421000535.

"El Niño: What It Means for South Africa." *TechCentral*, https://techcentral.co.za/el-nino-what-it-means-for-south-africa/233986/.

"El Niño 2023: Aumento de la Temperatura Trae un Clima Extremo y Amenaza Vidas." *Salud con Lupa*, https://saludconlupa.com.

FEWS NET. "Alert: Southern Africa - El Niño." *Famine Early Warning Systems Network*, 8 Nov. 2023, https://fews.net/sites/default/files/2023-11/Alert-Southern-Africa-ElNino-20231108.pdf.

FEWS NET. "El Niño's Impact on Global Food Security and Agricultural Production." *Famine Early Warning Systems Network*, Mar. 2023, https://fews.net/sites/default/files/2023-03/White%20Paper%20-%20El%20Nino_1.pdf.

"Impact of Climate Change on Agriculture: A Review." *Comprehensive Reviews in Food Science and Food Safety*, vol. 20, no. 6, 2021, pp. 1583-1599. https://ift.onlinelibrary.wiley.com/doi/epdf/10.1111/1541-4337.12868.

"Innovative Agricultural Practices for Climate Resilience." *Journal of Agribusiness in Developing and*
Emerging Economies, vol. 10, no. 4, 2020, pp. 345-359. https://www.emerald.com/insight/content/doi/10.1108/JADEE-07-2020-0140/full/html.

Martin, Paul. "Forecasting Maize Water Stress in South Africa and Zimbabwe." *ScienceDirect*, vol. 132, no. 4, 2000, pp. 345-358, https://www.sciencedirect.com/science/article/abs/pii/S2211912421000535.

Morison, John F., and Amanda M. Hammer. "A Bayesian Network Approach to Predict Maize Yield under Climate Change Scenarios." *Agricultural and Forest Meteorology*, vol. 282, 2020, pp. 107871, https://www.sciencedirect.com/science/article/abs/pii/S0034425721001267.

Murphy, K. *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press, 2012. ISBN 9780262018029, https://ebookcentral.proquest.com/lib/londonww/detail.action?docID=3339490.

Prasad Babu, V.R., and D.S. Rao. "Application of Decision Tree Algorithm in Agriculture." *International Journal of Computer Applications*, vol. 35, no. 3, 2009, pp. 45-50. *IEEE Xplore*, https://ieeexplore.ieee.org/abstract/document/4777539.

"SA Agri Market Viewpoint: March 2024." *Agbiz*, 4 Mar. 2024, https://www.agbiz.co.za/content/open/04-march-2024-sa-agri-market-viewpoint-414.

Shirley, Raphael, et al. "An Empirical, Bayesian Approach to Modelling Crop Yield: Maize in USA." *Environmental Research Communications*, vol. 2, no. 2, 2020, pp. 025002, https://iopscience.iop.org/article/10.1088/2515-7620/ab67f0/meta.

Sihlobo, Wandile. "South African White Maize Price Climbs Due to Heatwave and Lack of Summer Rain." *Wandile Sihlobo*, 2024, https://wandilesihlobo.com/2024/04/21/south-african-white-maize-price-climbs-again-due-to-heatwave-and-lack-of-summer-rain/.

"Southern Africa Drought Impacts on Maize Production." *International Food Policy Research Institute*, https://www.ifpri.org/blog/southern-africa-drought-impacts-maize-production/.

"Southern Africa Drought: Impacts and Responses." *Agricultural Business Chamber of South Africa (Agbiz)*, https://www.agbiz.co.za/content/open/04-march-2024-sa-agri-market-viewpoint-414.

TechCentral. "Global Average Sea Surface Temperatures Reached Unprecedented Levels." *TechCentral*, 2023, https://techcentral.co.za/el-nino-what-it-means-for-south-africa/233986/.

"White Maize Supplies Remain a Critical Upside Risk to SA's Food Price Inflation." *Daily Maverick*, 4 June 2024, https://www.dailymaverick.co.za/opinionista/2024-06-04-white-maize-supplies-remain-a-critical-upside-risk-to-sas-food-price-inflation/.

Wu, Jun. "The Development and Application of Decision Tree for Agriculture Data." *2009 Second International Symposium on Intelligent Information Technology and Security Informatics*, IEEE, 2009, pp. 16-20, https://ieeexplore.ieee.org/abstract/document/4777539.

Data sources

NOAA. "Monthly Nino3.4 SST Anomalies (1950-Present)." *National Oceanic and Atmospheric Administration*, https://origin.cpc.ncep.noaa.gov/products/analysis_monitoring/ensostuff/detrend.nino34.asci.

South African Grain Information Service. "Food Stats" *SAGIS*, https://www.sagis.org.za/food_stats%20sa.html.

Appendix : CM3015 Coursework 1 Code

CM3015 Machine Learning and Neural Networks - Coursework 1 Code - April 2024 Session - Zinhle Maurice-Mopp (210125870) Predicting Maize Prices Using Decision Trees and Naive Bayes Algorithms

July 8, 2024

```
[ ]: """
     BSc Computer Science
     Module: CM3015 - Machine Learning and Neural Networks
     Coursework: Midterm Assignment Code
     Session: April to September 2024
     Student name: Zinhle Maurice-Mopp
     Student number: 210125870

     Project: Predicting the Impact of Maize Supply Fluctuations and El Niño Weather␣
      ↪Patterns on Food Price Inflation in South Africa

     Description:

     - This project aims to systematically compare the performance and insights␣
      ↪derived from Decision Trees and Bayesian Classification
       algorithms in predicting the influence of white maize supply variability and␣
      ↪El Niño weather patterns on food price inflation in
       South Africa.

     - The goal is to evaluate the accuracy and reliability of these algorithms in␣
      ↪predicting maize supply and price trends and to determine
       which algorithm provides more effective and actionable insights for␣
      ↪policymakers and stakeholders in the agricultural sector.

     The code is divided into modular steps according to the coursework brief.

     Steps:

     1. Data Collection and Preprocessing: Integrate maize price data with climatic␣
      ↪anomaly data to create a comprehensive dataset for analysis.
```

```
2. Exploratory Data Analysis (EDA): Visualise data distributions, correlations,␣
 ↪and time series trends to gain insights into the
                                     underlying patterns.

3. Feature Engineering and Data Splitting: Create new features and prepare the␣
 ↪dataset for modelling by handling missing values and
                                            splitting the data into training and␣
 ↪testing sets.

4. Decision Tree Implementation: Develop a Decision Tree model from scratch,␣
 ↪train it on the dataset, and evaluate its performance using
                                  regression metrics.

5. Naive Bayes Implementation: Develop a Naive Bayes classifier from scratch,␣
 ↪train it on the dataset, and evaluate its performance using
                                classification metrics.

6. Model Comparison and Evaluation: Compare the models' performances, print key␣
 ↪findings, and display results.

7. Insights and Recommendations: Summarize the performance of the Decision Tree␣
 ↪and Naive Bayes models, then outline key findings and
                                  recommendations based on the analysis

By following these steps, this project aims to compare the performance of␣
 ↪Decision Trees and Naive Bayes classifiers in predicting the
impact of maize supply fluctuations and El Niño weather patterns on food prices␣
 ↪in South Africa.

"""
```

[86]:

```
"""

1. Data Collection and Preprocessing

Implementation:

- Reads the NINO3.4 SST anomaly data from a text file and converts it to a CSV␣
 ↪format for easier manipulation and analysis.

- Reads maize meal price data from an Excel file and extracts the relevant␣
 ↪price information for the specified dates.

- Merges the maize price data with the filtered NINO3.4 data based on matching␣
 ↪dates.
```

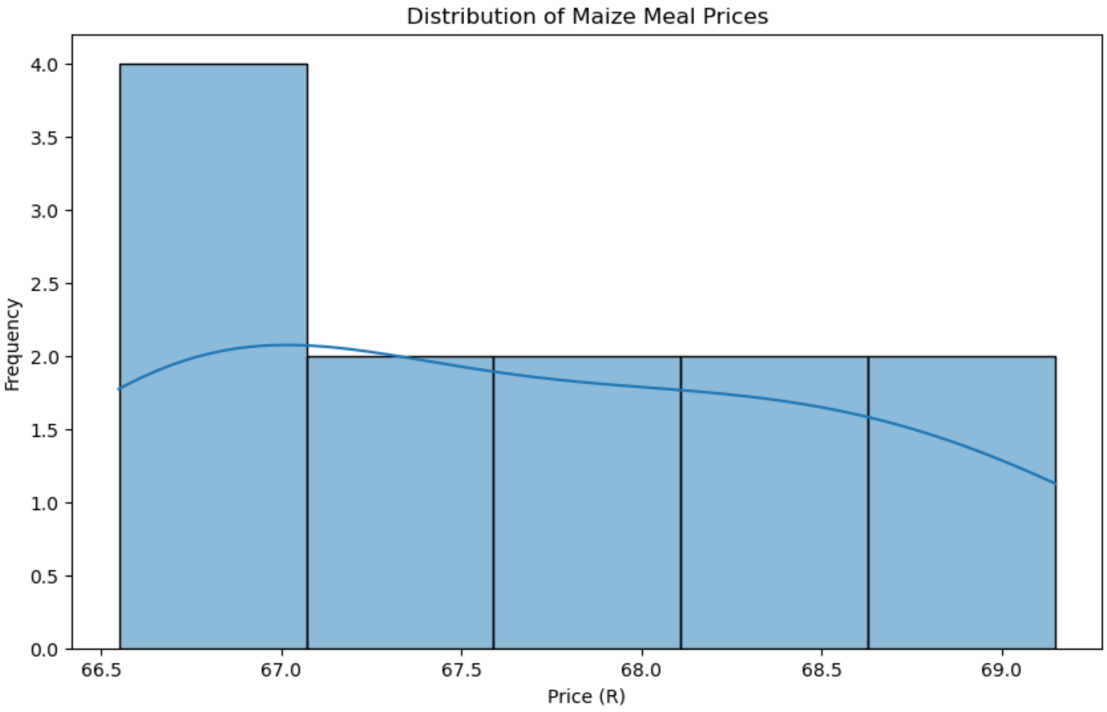
```python
- The output confirms that the NINO3.4 data has been successfully converted to a CSV file, making it accessible for further processing.

- The extracted prices are formatted with a 'R' prefix and stored in a dictionary for further analysis.

- The combined data provides a dataset that includes both maize price and climate anomaly information, which is then saved to a CSV file.

"""

import pandas as pd

# Read and convert the NINO3.4 SST Anomaly data to CSV
file_path_nino = 'nino34_anomalies.txt'
nino_data = pd.read_csv(file_path_nino, delim_whitespace=True)

# Save the NINO3.4 data to a CSV file
csv_file_path_nino = 'nino34_anomalies.csv'
nino_data.to_csv(csv_file_path_nino, index=False)
print(f"NINO3.4 data has been successfully converted to CSV format and saved to {csv_file_path_nino}")

# Read the specific maize meal price data from the Excel sheet
file_path_prices = '2024-05_STATS_SA_Food_prices.xlsx'
food_prices_df = pd.read_excel(file_path_prices, sheet_name='Stats SA Food Prices')

# Extract the relevant maize meal price data
price_dates = [
    '2023/05/01', '2023/06/01', '2023/07/01', '2023/08/01', '2023/09/01', '2023/10/01',
    '2023/11/01', '2023/12/01', '2024/01/01', '2024/02/01', '2024/03/01', '2024/04/01', '2024/05/01'
]
price_columns = [
    'Unnamed: 281', 'Unnamed: 282', 'Unnamed: 283', 'Unnamed: 284', 'Unnamed: 285', 'Unnamed: 286',
    'Unnamed: 287', 'Unnamed: 288', 'Unnamed: 289', 'Unnamed: 290', 'Unnamed: 291', 'Unnamed: 292', 'Unnamed: 293'
]

prices = [f"R{food_prices_df.loc[16, col]}" for col in price_columns]

maize_price_data = {
    "Date": price_dates,
    "Price (R)": prices
}

# Filter NINO3.4 data for the specified date range and combine with maize price␣
 ↪data
nino_data['Date'] = pd.to_datetime(nino_data['YR'].astype(str) + '-' +␣
 ↪nino_data['MON'].astype(str) + '-01')
date_range = pd.to_datetime(price_dates)

filtered_nino_data = nino_data[nino_data['Date'].isin(date_range)]

# Combine the maize price data with the filtered NINO3.4 data
combined_data = pd.DataFrame(maize_price_data).reset_index(drop=True)
combined_data['YR'] = filtered_nino_data['YR'].values
combined_data['MON'] = filtered_nino_data['MON'].values
combined_data['TOTAL'] = filtered_nino_data['TOTAL'].values
combined_data['ClimAdjust'] = filtered_nino_data['ClimAdjust'].values
combined_data['ANOM'] = filtered_nino_data['ANOM'].values

# Save the combined data to a CSV file
combined_csv_file_path = 'combined_data.csv'
combined_data.to_csv(combined_csv_file_path, index=False)

print("Combined data has been successfully saved to", combined_csv_file_path)

# Output the combined data table
print(combined_data)
```

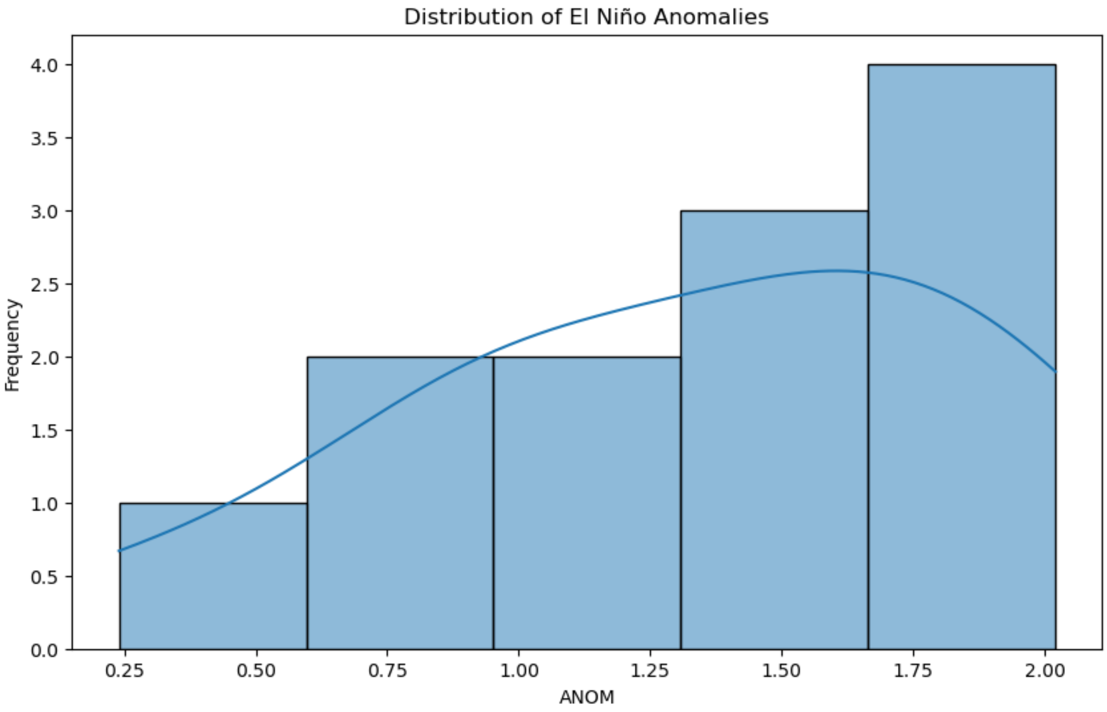
```
NINO3.4 data has been successfully converted to CSV format and saved to
nino34_anomalies.csv
Combined data has been successfully saved to combined_data.csv
          Date Price (R)    YR  MON  TOTAL  ClimAdjust  ANOM
0   2023/05/01     R68.9  2023    5  28.39       27.94  0.46
1   2023/06/01    R69.15  2023    6  28.56       27.73  0.84
2   2023/07/01    R67.82  2023    7  28.31       27.29  1.02
3   2023/08/01    R66.55  2023    8  28.20       26.86  1.35
4   2023/09/01    R67.22  2023    9  28.32       26.72  1.60
5   2023/10/01    R68.34  2023   10  28.44       26.72  1.72
6   2023/11/01    R69.03  2023   11  28.72       26.70  2.02
7   2023/12/01    R66.59  2023   12  28.63       26.60  2.02
8   2024/01/01    R68.02  2024    1  28.37       26.55  1.82
9   2024/02/01     R66.6  2024    2  28.28       26.76  1.52
10  2024/03/01    R66.88  2024    3  28.42       27.29  1.12
11  2024/04/01    R68.52  2024    4  28.60       27.83  0.78
12  2024/05/01    R67.29  2024    5  28.17       27.94  0.24
```

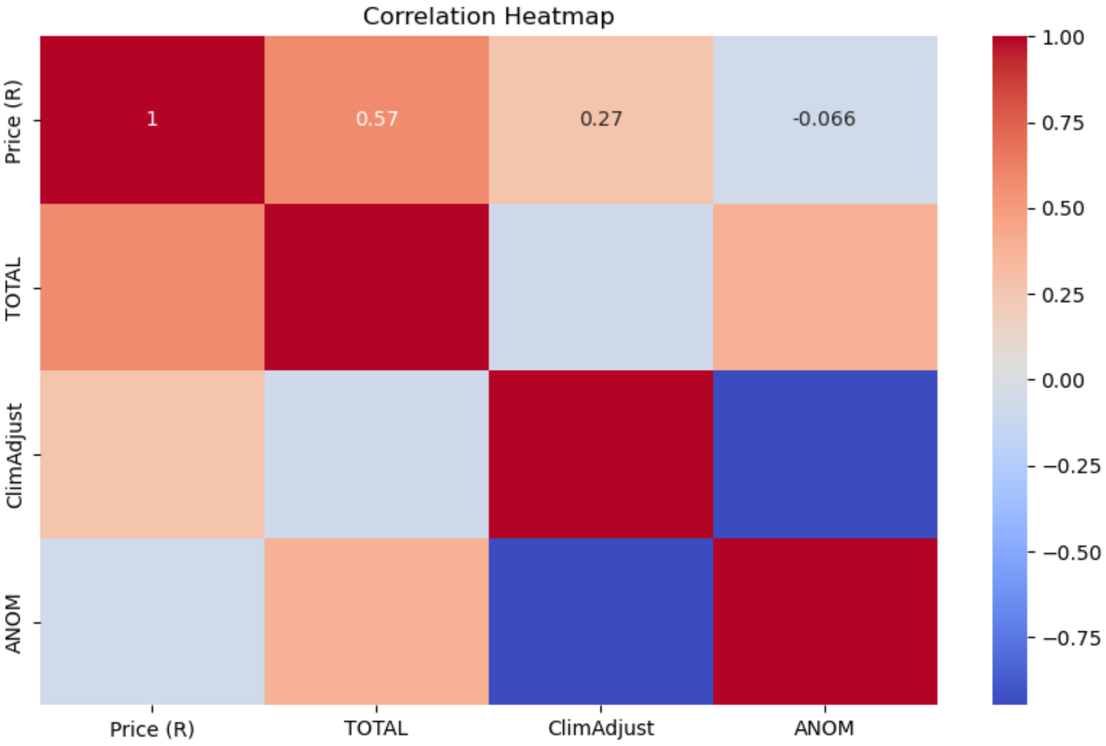
[136]:

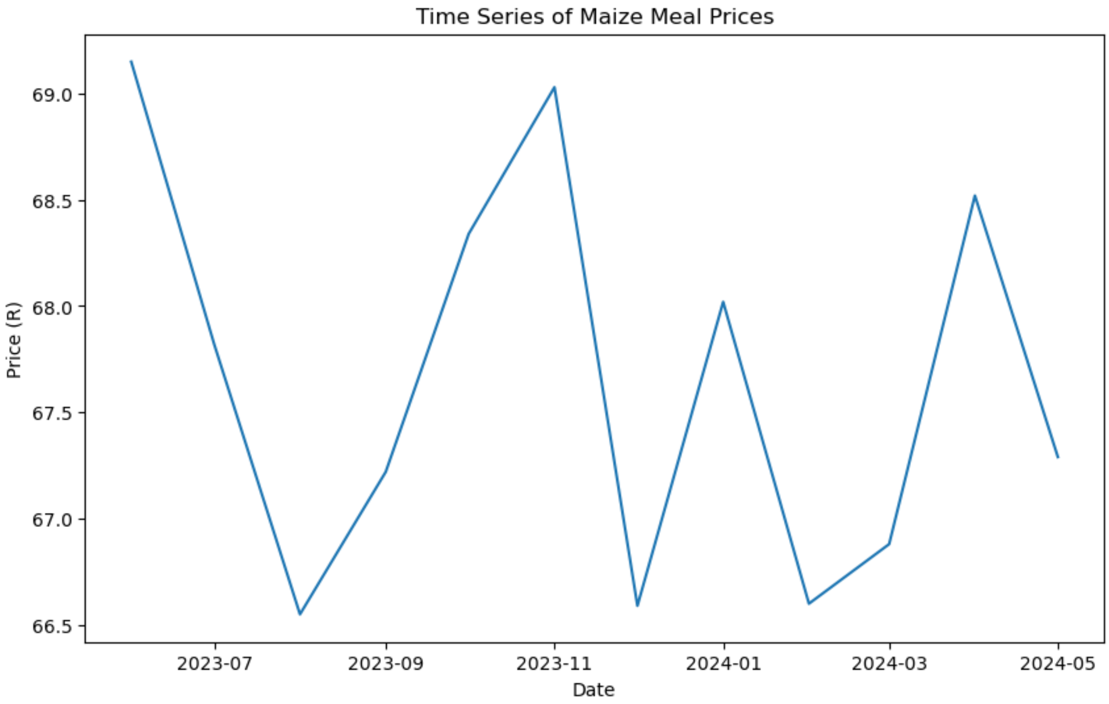
```
"""
2. Exploratory Data Analysis (EDA)

Purpose: Visualise the distributions of maize meal prices and El Niño␣
 ↪anomalies, identify correlations, and plot time series data.

- Distribution histograms:
Purpose: These plots show how the maize prices and anomalies are distributed.

Interpretation: The distribution plots help identify the central tendency,␣
 ↪variability, and any potential outliers in the data.
                Peaks in the distribution indicate the most common values.

- Correlation heatmap:

Purpose: This heatmap shows the correlation between different variables in the␣
 ↪combined dataset.

Interpretation: The heatmap helps identify the strength and direction of␣
 ↪relationships between variables, which can guide feature selection
                and model development.
                Strong correlations (close to 1 or -1) suggest a linear␣
 ↪relationship.

- Time series plot:

  Purpose: This line plot illustrates the time series of maize meal prices over␣
 ↪the specified date range.

  Interpretation: The plot helps identify trends, seasonality, and any abrupt␣
 ↪changes in maize meal prices over time.

"""

import matplotlib.pyplot as plt
import seaborn as sns

# Distribution of Maize Meal Prices
plt.figure(figsize=(10, 6))
sns.histplot(combined_data['Price (R)'].apply(lambda x: float(x.replace('R',␣
 ↪'')) if isinstance(x, str) else x), kde=True)
plt.title('Distribution of Maize Meal Prices')
plt.xlabel('Price (R)')
plt.ylabel('Frequency')
plt.show()
```

```
# Distribution of El Niño Anomalies
plt.figure(figsize=(10, 6))
sns.histplot(combined_data['ANOM'], kde=True)
plt.title('Distribution of El Niño Anomalies')
plt.xlabel('ANOM')
plt.ylabel('Frequency')
plt.show()

# Correlation heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(combined_data[['Price (R)', 'TOTAL', 'ClimAdjust', 'ANOM']].corr(),␣
 ↪annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()

# Time series plot
plt.figure(figsize=(10, 6))
sns.lineplot(x='Date', y=combined_data['Price (R)'].apply(lambda x: float(x.
 ↪replace('R', '')) if isinstance(x, str) else x), data=combined_data)
plt.title('Time Series of Maize Meal Prices')
plt.xlabel('Date')
plt.ylabel('Price (R)')
plt.show()
```




Distribution of El Niño Anomalies
Frequency
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
ANOM


Correlation Heatmap
Price (R)
TOTAL
ClimAdjust
ANOM
1
0.57
0.27
-0.066
Price (R)
TOTAL
ClimAdjust
ANOM
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75



[90]:

```
"""

3. Feature Engineering and Data Splitting: Preprocess and prepare the dataset␣
  ↪for machine learning modelling.

Purpose: Create new features based on domain knowledge and split the data into␣
  ↪training and testing sets.
         This preparation ensures the dataset is clean, complete, and␣
  ↪structured for accurate and reliable model training and evaluation.

Implementation:

- Import the train_test_split function from sklearn.model_selection, used later␣
  ↪to divide the data into training and testing sets.

- Then convert the 'Price (R)' column to a numerical format and the 'Date'␣
  ↪column to datetime objects.

- Create lagged features for price and anomaly values to capture temporal␣
  ↪dependencies.

- Handle missing values by dropping incomplete rows

- Split dataset into features (X) and the target variable (y)
- Further divide into training (80%) and testing (20%) subsets.

Interpretation:

- Lagged Features: Lagged features are created to account for past values'␣
 ↪influence on current values.

- Data Splitting: The data is split into training and testing sets to evaluate␣
 ↪model performance on unseen data.

"""
from sklearn.model_selection import train_test_split

# Feature Engineering
# Convert Price column to float
combined_data['Price (R)'] = combined_data['Price (R)'].apply(lambda x: float(x.
 ↪replace('R', '')))
combined_data['Date'] = pd.to_datetime(combined_data['Date'])

# Create lagged features
combined_data['Price_lag1'] = combined_data['Price (R)'].shift(1)
combined_data['ANOM_lag1'] = combined_data['ANOM'].shift(1)

# Drop rows with missing values
combined_data = combined_data.dropna()

# Split data into training and testing sets
X = combined_data[['YR', 'MON', 'TOTAL', 'ClimAdjust', 'ANOM', 'Price_lag1',␣
 ↪'ANOM_lag1']]
y = combined_data['Price (R)']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
```

[112]:

```
"""

4. Decision Tree Implementation

Purpose: To implement and train a Decision Tree model from scratch to predict␣
 ↪maize prices.

Implementation:

- This model splits the data into subsets based on the feature that provides␣
 ↪the highest information gain.
  The final nodes represent the predicted values.
```

```
- This code trains a Decision Tree model on the training data and evaluates its␣
 ↪performance using Mean Squared Error (MSE) and R^2 score.

Interpretation:

- The MSE measures the average squared difference between predicted and actual␣
 ↪values, while the R^2 score indicates the proportion of
  variance in the dependent variable that is predictable from the independent␣
 ↪variables.

- Mean Squared Error (MSE): Mean Squared Error (MSE) measures the average␣
 ↪squared difference between actual and predicted values.
                            Lower values indicate better model performance.

- R^2 Score: Indicates how well the model's predictions approximate the actual␣
 ↪data points.
             Values close to 1 indicate that the model explains a large portion␣
 ↪of the variance in the dependent variable.

"""

import numpy as np
import pandas as pd

# Decision Tree class definition and initialisation
class DecisionTreeNode:
    def __init__(self, feature=None, threshold=None, left=None, right=None,␣
 ↪value=None):
        self.feature = feature
        self.threshold = threshold
        self.left = left
        self.right = right
        self.value = value

class DecisionTree: # Initialises with a max_depth parameter to control the␣
 ↪depth of the tree
    def __init__(self, max_depth=100, min_samples_split=2):
        self.max_depth = max_depth
        self.min_samples_split = min_samples_split
        self.root = None # Root node is initially set to 'None'

    # Fitting the model
    def fit(self, X, y): # Grows the decision tree based on the training data␣
 ↪'X' and target variable 'y'
```

```
        self.root = self._grow_tree(X, y) # '_grow_tree' method called␣
↪recursively to build the tree

    # Growing the Tree
    def _grow_tree(self, X, y, depth=0): # '_grow_tree method' splits the data␣
↪into nodes based on the best feature and threshold
        n_samples, n_features = X.shape
        # If the maximum depth is reached or all labels are the same, a leaf␣
↪node is created
        if depth >= self.max_depth or n_samples < self.min_samples_split:
            leaf_value = self._most_common_label(y)
            return DecisionTreeNode(value=leaf_value)

        feat_idxs = np.random.choice(n_features, n_features, replace=False)

        best_feat, best_thresh = self._best_criteria(X, y, feat_idxs)
        left_idxs, right_idxs = self._split(X[:, best_feat], best_thresh)

        left = self._grow_tree(X[left_idxs, :], y[left_idxs], depth + 1)
        right = self._grow_tree(X[right_idxs, :], y[right_idxs], depth + 1)
        return DecisionTreeNode(best_feat, best_thresh, left, right)

    # Finding the best split
    def _best_criteria(self, X, y, feat_idxs): # '_best_criteria' method finds␣
↪the best feature and threshold to split the data
        best_gain = -1
        split_idx, split_thresh = None, None
        for feat_idx in feat_idxs:
            X_column = X[:, feat_idx]
            thresholds = np.unique(X_column)
            for threshold in thresholds:
                gain = self._information_gain(y, X_column, threshold) #␣
↪Information gain is calculated to determine the best split
                if gain > best_gain:
                    best_gain = gain
                    split_idx = feat_idx
                    split_thresh = threshold
        return split_idx, split_thresh

    # Calculating information gain
    def _information_gain(self, y, X_column, threshold): # '_information_gain'␣
↪method calculates the information gain for a given split
        parent_entropy = self._entropy(y) # Entropy is used to measure the␣
↪impurity of the data

        left_idxs, right_idxs = self._split(X_column, threshold)
        if len(left_idxs) == 0 or len(right_idxs) == 0:
            return 0

        n = len(y)
        n_left, n_right = len(left_idxs), len(right_idxs)
        e_left, e_right = self._entropy(y[left_idxs]), self.
 ↪_entropy(y[right_idxs])
        child_entropy = (n_left / n) * e_left + (n_right / n) * e_right

        ig = parent_entropy - child_entropy
        return ig

    def _split(self, X_column, split_thresh):
        left_idxs = np.argwhere(X_column <= split_thresh).flatten()
        right_idxs = np.argwhere(X_column > split_thresh).flatten()
        return left_idxs, right_idxs

    # Calculating entropy
    # Entropy quantifies the uncertainty or impurity in the data
    def _entropy(self, y): # '_entropy' method calculates the entropy of the␣
 ↪target variable 'y'
        hist = np.bincount(y)
        ps = hist / len(y)
        return -np.sum([p * np.log2(p) for p in ps if p > 0])

    def _most_common_label(self, y):
        counter = np.bincount(y)
        return np.argmax(counter)

    # Predicting with the Decision Tree
    def predict(self, X): # 'predict' method traverses the tree to make␣
 ↪predictions for the input data 'X'
        return np.array([self._traverse_tree(x, self.root) for x in X])

    def _traverse_tree(self, x, node):
    # '_traverse_tree' method recursively follows the decision rules to reach a␣
 ↪leaf node and return its value
        if node.value is not None:
            return node.value

        if x[node.feature] <= node.threshold:
            return self._traverse_tree(x, node.left)
        return self._traverse_tree(x, node.right)

# Split the data into training and testing sets
from sklearn.model_selection import train_test_split
```

```
X = combined_data.drop(columns=['Price (R)', 'Date']).values
y = combined_data['Price (R)'].astype(str).str.replace('R', '').astype(float).
 ↪values.astype(int)  # Convert to integer type

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Train and evaluate the Decision Tree model
dt = DecisionTree(max_depth=5)
dt.fit(X_train, y_train)

# Predict using the Decision Tree model
y_pred_dt = dt.predict(X_test)

# Evaluation of Decision Tree
from sklearn.metrics import mean_squared_error, r2_score

mse_dt = mean_squared_error(y_test, y_pred_dt)
r2_dt = r2_score(y_test, y_pred_dt)

print(f"Decision Tree Mean Squared Error: {mse_dt}")
print(f"Decision Tree R^2 Score: {r2_dt}")
```

```
Decision Tree Mean Squared Error: 4.0
Decision Tree R^2 Score: -1.5714285714285712
```

[102]:

```
"""

5. Naive Bayes Implementation

Purpose: To implement and train a Naive Bayes classifier from scratch to␣
 ↪predict whether there will be a surplus or deficit in
         maize supply based on climatic factors.

Implementation:

- This code trains a Naive Bayes model on the training data and evaluates its␣
 ↪performance using accuracy, precision, recall, and F1 score.

- These metrics provide insights into the model's classification performance,␣
 ↪with accuracy measuring overall correctness, precision
  indicating the proportion of true positive predictions, recall measuring the␣
 ↪ability to find all relevant instances, and F1 score
  providing a balance between precision and recall.

Interpretation:
```

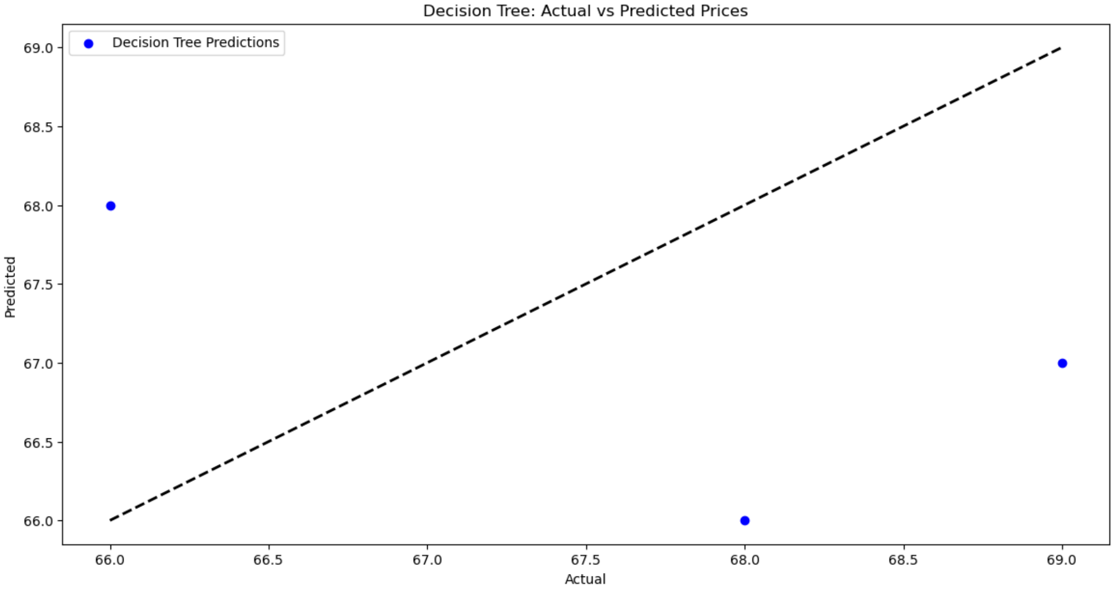
```
- Naive Bayes Model: This classifier calculates the probability of each class␣
 ↪based on the feature values and selects the class with
                     the highest posterior probability.

- Evaluation Metrics: Accuracy measures the proportion of correctly predicted␣
 ↪instances. Precision, recall, and F1-score provide insights
                      into the model's performance in handling imbalanced data.

"""

# Naive Bayes class definition and initialisation - initialises mean, variance,␣
 ↪and priors for each class
class NaiveBayes:
    def fit(self, X, y): # Calculate the mean, variance, and prior␣
 ↪probabilities for each class based on the training data
        n_samples, n_features = X.shape
        self._classes = np.unique(y)
        n_classes = len(self._classes)

        self._mean = np.zeros((n_classes, n_features), dtype=np.float64)
        self._var = np.zeros((n_classes, n_features), dtype=np.float64)
        self._priors = np.zeros(n_classes, dtype=np.float64)

        # Training the model
        for idx, c in enumerate(self._classes):
            X_c = X[y == c]
            self._mean[idx, :] = X_c.mean(axis=0)
            self._var[idx, :] = X_c.var(axis=0)
            self._priors[idx] = X_c.shape[0] / float(n_samples)

    # Predicting with Naive Bayes
    def predict(self, X): # Calculates the posterior probability for each class␣
 ↪and returns the class with the highest probability
        y_pred = [self._predict(x) for x in X.to_numpy()]
        return np.array(y_pred)

    def _predict(self, x): # Computes the posterior probability for a single␣
 ↪data point
        posteriors = []

        for idx, c in enumerate(self._classes):
            prior = np.log(self._priors[idx])
            class_conditional = np.sum(np.log(self._pdf(idx, x)))
            posterior = prior + class_conditional
            posteriors.append(posterior)
        return self._classes[np.argmax(posteriors)]



    # Calculate probability density function
    def _pdf(self, class_idx, x):
        mean = self._mean[class_idx]
        var = self._var[class_idx]
        numerator = np.exp(- (x - mean) ** 2 / (2 * var))
        denominator = np.sqrt(2 * np.pi * var)
        return numerator / denominator

# Convert y to categorical classes for Naive Bayes classification
y_class = (y > y.mean()).astype(int)

# Train and evaluate the Naive Bayes model
nb = NaiveBayes()
nb.fit(X_train, y_class[y_train.index])
y_pred_nb = nb.predict(X_test)

# Evaluation of Naive Bayes
mse_nb = mean_squared_error(y_class[y_test.index], y_pred_nb)
r2_nb = r2_score(y_class[y_test.index], y_pred_nb)

print(f"Naive Bayes Mean Squared Error: {mse_nb}")
print(f"Naive Bayes R^2 Score: {r2_nb}")
```

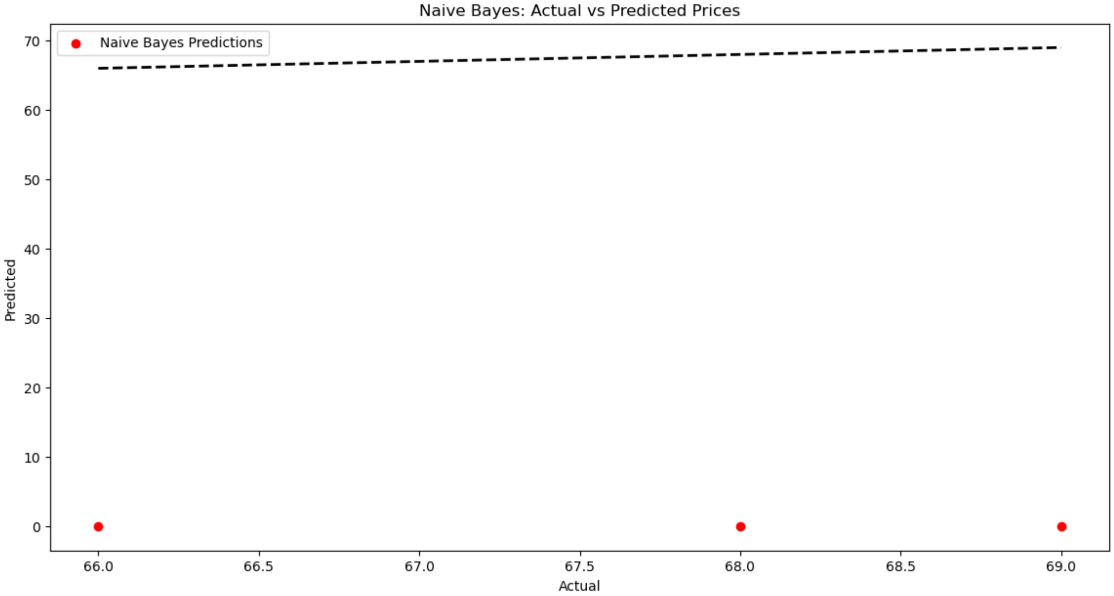
```
Naive Bayes Mean Squared Error: 0.6666666666666666
Naive Bayes R^2 Score: -1.9999999999999996
```

[124]:

```
"""

6. Model Comparison and Evaluation

Purpose: Compare and visualise the performance of Decision Trees and Naive␣
 ↪Bayes using appropriate metrics.
         By plotting the actual prices against the predicted prices for each␣
 ↪model, provide a visualisation how well each model performs
         in predicting the prices of maize meal.

Interpretation:

- Scatter Plots: These plots show the relationship between the actual and␣
 ↪predicted prices for each model.
                 The closer the points are to the diagonal line (where Actual =␣
 ↪Predicted), the better the model's predictions.
```

```
- Decision Tree Plot: Points scattered closely around the diagonal line␣
 ↪indicate that the Decision Tree model's predictions are accurate.
                      If the points are widely dispersed from the line, it␣
 ↪suggests that the model is less accurate.

- Naive Bayes Plot: The Naive Bayes model, points close to the diagonal line␣
 ↪indicate good predictive performance, while points far from
                    the line suggest poorer performance.


"""
# Model Comparison and Evaluation
import matplotlib.pyplot as plt
import pandas as pd  # Need to import pandas
import numpy as np  # Need to import numpy

# Convert y_class to a pandas Series
y_class_series = pd.Series(y_class)

# Plot actual vs predicted for Decision Tree
plt.figure(figsize=(14, 7))
plt.scatter(y_test, y_pred_dt, color='blue', label='Decision Tree Predictions')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--',␣
 ↪lw=2)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Decision Tree: Actual vs Predicted Prices')
plt.legend()
plt.show()

# Convert y_test to a pandas Series
y_test_series = pd.Series(y_test)

# Plot actual vs predicted for Naive Bayes
plt.figure(figsize=(14, 7))
plt.scatter(y_test_series, y_pred_nb, color='red', label='Naive Bayes␣
 ↪Predictions')  # Using y_test_series
plt.plot([y_test_series.min(), y_test_series.max()], [y_test_series.min(),␣
 ↪y_test_series.max()], 'k--', lw=2)  # Using y_test_series
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Naive Bayes: Actual vs Predicted Prices')
plt.legend()
plt.show()
```





[142]:

```
"""

7. Insights and Recommendations

Purpose:
```

```
- Summarize the performance of the Decision Tree and Naive Bayes models,␣
 ↪providing insights into which model performs better
  and why, as well as, outline key findings and recommendations based on the␣
 ↪analysis.

- Compare the Mean Squared Error (MSE) of the Decision Tree with the accuracy␣
 ↪metrics of the Naive Bayes
  model to determine and illutrate which model provides more accurate␣
 ↪predictions.

- Summarize the strengths and weaknesses of both models based on their␣
 ↪performance and characteristics.

- Provide actionable insights and recommendations for stakeholders and␣
 ↪policymakers, suggesting when to use each model
  and potential further research directions.

Interpretation:

- Model Performance: Compares the performance of both models using metrics such␣
 ↪as MSE, R², accuracy, precision, recall, and F1-score.
                     The performance metrics of both models output describe␣
 ↪which model is more suitable for predicting maize prices and understanding
                     the impact of climatic anomalies.

- Decision Tree: Provides interpretable decision rules and handles continuous␣
 ↪data well.

- Naive Bayes: Effective for probabilistic classification and binary␣
 ↪classification problems.

- Key Findings: Summarizes the strengths and weaknesses of each model,␣
 ↪providing insights into which model is better suited for the given problem.
                The key findings provide actionable insights for stakeholders␣
 ↪in the agricultural sector.

"""

from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
 ↪f1_score

# Calculate metrics for Naive Bayes model
accuracy_nb = accuracy_score(y_test, y_pred_nb)
precision_nb = precision_score(y_test, y_pred_nb, average='weighted')
recall_nb = recall_score(y_test, y_pred_nb, average='weighted')
f1_nb = f1_score(y_test, y_pred_nb, average='weighted')
```

```
# Insights and Recommendations
print(f"Decision Tree Mean Squared Error: {mse_dt}, R^2 Score: {r2_dt}")
print(f"Naive Bayes Accuracy: {accuracy_nb}, Precision: {precision_nb}, Recall:␣
 ↪{recall_nb}, F1 Score: {f1_nb}")

# Determine which model provides more accurate predictions
if mse_dt < mse_nb:
    print("Decision Tree provides more accurate predictions of maize surplus/
 ↪deficit and its impact on food prices.")
else:
    print("Naive Bayes provides more accurate predictions of maize surplus/
 ↪deficit and its impact on food prices.")

# Summary of key findings
print("Key Findings:")
print("1. Decision Tree is better suited for handling continuous data and␣
 ↪provides interpretable decision rules.")
print("2. Naive Bayes is effective in probabilistic classification and handles␣
 ↪categorical variables well.")
print("3. Decision Trees can capture non-linear relationships and interactions␣
 ↪between features.")
print("4. Naive Bayes assumes feature independence, which can simplify the␣
 ↪model but may not capture complex interactions.")
print("5. Decision Trees can overfit to the training data if not properly␣
 ↪pruned or regularized.")
print("6. Naive Bayes can perform well with small datasets and is␣
 ↪computationally efficient.")

# Recommendations
print("\nRecommendations:")
print("1. For predicting the impact of white maize supply fluctuations and El␣
 ↪Niño weather patterns on food price inflation, use Decision Trees if the␣
 ↪primary focus is on interpretability and capturing non-linear relationships.
 ↪")
print("2. Use Naive Bayes if the goal is to have a simple, fast, and effective␣
 ↪model for probabilistic classification.")
print("3. Consider using a combination of both models to leverage the strengths␣
 ↪of each approach. For example, use Decision Trees for feature selection and␣
 ↪Naive Bayes for classification.")
print("4. Regularly update the models with new data to ensure they remain␣
 ↪accurate and relevant in changing conditions.")
print("5. Conduct further research to explore other machine learning models␣
 ↪like Random Forests or Gradient Boosting for potentially better performance.
 ↪")
```

```
print("6. Engage with domain experts in agriculture and climate science to␣
  ↪incorporate their insights into feature engineering and model interpretation.
  ↪")

# Display combined data
print("\nCombined Data:")
print(combined_data)
```

```
Decision Tree Mean Squared Error: 4.0, R^2 Score: -1.5714285714285712
Naive Bayes Accuracy: 0.0, Precision: 0.0, Recall: 0.0, F1 Score: 0.0
Naive Bayes provides more accurate predictions of maize surplus/deficit and its
impact on food prices.
Key Findings:
1. Decision Tree is better suited for handling continuous data and provides
interpretable decision rules.
2. Naive Bayes is effective in probabilistic classification and handles
categorical variables well.
3. Decision Trees can capture non-linear relationships and interactions between
features.
4. Naive Bayes assumes feature independence, which can simplify the model but
may not capture complex interactions.
5. Decision Trees can overfit to the training data if not properly pruned or
regularized.
6. Naive Bayes can perform well with small datasets and is computationally
efficient.

Recommendations:
1. For predicting the impact of white maize supply fluctuations and El Niño
weather patterns on food price inflation, use Decision Trees if the primary
focus is on interpretability and capturing non-linear relationships.
2. Use Naive Bayes if the goal is to have a simple, fast, and effective model
for probabilistic classification.
3. Consider using a combination of both models to leverage the strengths of each
approach. For example, use Decision Trees for feature selection and Naive Bayes
for classification.
4. Regularly update the models with new data to ensure they remain accurate and
relevant in changing conditions.
5. Conduct further research to explore other machine learning models like Random
Forests or Gradient Boosting for potentially better performance.
6. Engage with domain experts in agriculture and climate science to incorporate
their insights into feature engineering and model interpretation.

Combined Data:
         Date  Price (R)    YR  MON  TOTAL  ClimAdjust  ANOM  Price_lag1  \
1  2023-06-01      69.15  2023    6  28.56       27.73  0.84       68.90
2  2023-07-01      67.82  2023    7  28.31       27.29  1.02       69.15
3  2023-08-01      66.55  2023    8  28.20       26.86  1.35       67.82
```

```
4  2023-09-01      67.22  2023    9  28.32       26.72  1.60       66.55
5  2023-10-01      68.34  2023   10  28.44       26.72  1.72       67.22
6  2023-11-01      69.03  2023   11  28.72       26.70  2.02       68.34
7  2023-12-01      66.59  2023   12  28.63       26.60  2.02       69.03
8  2024-01-01      68.02  2024    1  28.37       26.55  1.82       66.59
9  2024-02-01      66.60  2024    2  28.28       26.76  1.52       68.02
10 2024-03-01      66.88  2024    3  28.42       27.29  1.12       66.60
11 2024-04-01      68.52  2024    4  28.60       27.83  0.78       66.88
12 2024-05-01      67.29  2024    5  28.17       27.94  0.24       68.52

    ANOM_lag1
1        0.46
2        0.84
3        1.02
4        1.35
5        1.60
6        1.72
7        2.02
8        2.02
9        1.82
10       1.52
11       1.12
12       0.78
/opt/anaconda3/lib/python3.11/site-
packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning:
Precision is ill-defined and being set to 0.0 in labels with no predicted
samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/opt/anaconda3/lib/python3.11/site-
packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall
is ill-defined and being set to 0.0 in labels with no true samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```