PRICEWISE AI

1. BUSINESS UNDERSTANDING

The main objective of this project was to create a time series model for forecasting economic indicators and commodity prices. Economic indicators such as inflation, exchange rates, and GDP growth were deemed crucial for decision-making by businesses, investors, and policymakers. By accurately forecasting these indicators, businesses could better manage their inventory, adjust their operations, and set prices accordingly. Investors could use these forecasts to make informed investment decisions, and policymakers could use them to set monetary policy.

The project was focused on delivering value to stakeholders who required accurate and timely economic forecasts, including businesses, investors, and policymakers. The project aimed to tackle the real-world problem of accurately forecasting economic indicators and commodity prices. The beneficiaries of this project were businesses, investors, and policymakers who relied on economic data to make informed decisions.

The project aimed to develop models that could reliably predict commodity prices and inflation trends, as well as identify investment opportunities based on these trends. The recommendations derived from these forecasts could help businesses manage costs associated with fluctuations in commodity prices and inflation rates, as well as identify opportunities to reduce these costs.

1.1 Research questions

1. How do changes in inflation rates affect the prices of selected food commodities in Kenya, and can these relationships be modeled accurately using time series methods?
2. What are the major drivers of inflation in Kenya, and how can these drivers be incorporated into forecasting models to improve their accuracy?
3. Can patterns and trends in the historical prices of food commodities in Kenya be used to predict future price movements, and how accurate are these predictions?
4. What is the impact of exchange rate fluctuations on the prices of imported food commodities in Kenya, and can this impact be accurately modeled using time series methods?
5. How can businesses, investors, and policymakers use the insights generated by the project to make informed decisions about pricing, investment, and monetary policy in Kenya?

1.2. Objectives

1.2.1 Main Objective

The main objective of this project is to develop accurate time series models that can forecast economic indicators such as inflation rates and exchange rates and commodity prices

1.2.2 Specific Objectives

1. Explore and clean time series data on economic indicators and commodity prices to identify patterns, trends, and seasonality.
2. Conduct market analysis to identify trends and patterns in commodity prices and inflation rates, and provide investment recommendations to capitalize on the identified opportunities.
3. Develop a web-based application that provides traders and investors with reliable real-time commodity price predictions and continuously updates the model for ongoing accuracy.

1.3 Success Metric

Model Accuracy: The accuracy of the developed time series models will be measured by the Root Mean Squared Error (RMSE) and the project will be considered a success if the time series model has a Root Mean Squared Error for each of the commodities is utmost 5% when making predictions.

2. DATA UNDERSTANDING.

2.1 Data Source

We obtained the datasets Inflation Rates.csv, and Exchange Rates.csv from the Central Bank of Kenya (CBK). CBK is the central monetary authority of Kenya responsible for formulating and implementing monetary policy in the country. The data commodity prices.xlsx is obtained from the Food Security Portal here which is maintained by the International Food Policy Research Institute.

2.2 Properties of the Data

Inflation Rates.csv contains the monthly inflation rates from January 2000 to December 2022. The data is presented as percentages, and the inflation rates are calculated as year-on-year changes in the Consumer Price Index (CPI) for the Kenyan economy. The dataset contains 276 observations.

Exchange Rates.csv contains the daily exchange rates of major currencies (USD, GBP, EUR, and JPY) against the Kenyan Shilling from January 2000 to December 2022. The exchange rates are presented as the number of units of foreign currency that can be exchanged for one Kenyan Shilling. The dataset contains 6,346 observations.

Commodity prices.xlsx contains monthly prices of selected food commodities (maize, beans, rice, and wheat) in Kenya from January 2005 to December 2022. The data is presented in Kenyan Shillings per kilogram, and the dataset contains 216 observations for each commodity.

2.3 Suitability of the Data

The data from CBK and the Food Security Portal is highly relevant to the Kenyan economy and is suitable for the project. Inflation Rates.csv provides insight into the inflation trends in Kenya, which is essential for forecasting commodity prices. Annual GDP.csv helps in understanding the overall performance of the Kenyan economy, and Exchange Rates.csv is crucial in forecasting the future prices of commodities denominated in foreign currencies. Commodity prices.xlsx is essential for understanding the trends and patterns of food commodity prices in Kenya, which is essential for predicting food security.

2.4 Data Limitations

The data from CBK and the Food Security Portal is comprehensive, but there are some limitations. Inflation Rates.csv only covers the period from January 2000 to December 2022, which may not capture long-term trends in inflation. Annual GDP.csv only presents the GDP in nominal terms, which does not account for changes in the prices of goods and services over time. Exchange Rates.csv only covers a limited number of major currencies, which may not reflect the exchange rates of other currencies that may affect commodity prices. Commodity prices.xlsx only covers a limited number of food commodities, which may not reflect the prices of other essential commodities affecting food security in Kenya

2.5 Data Description

The commodity_data had 207 rows and 7 columns, with the first column containing date-time objects and the remaining columns containing float values representing the price of the corresponding commodity at the given date and time. The data frame had some missing values, and the data types of the columns were either datetime64 or float64. It was noted that this may not reflect the prices of other essential commodities affecting food security in Kenya.

The inflation data provided in the data frame had 207 rows and 4 columns. Each row represented a specific month in a year, and the columns provide information on the year, month, annual average inflation, and 12-month inflation rate.

The forex_data DataFrame contained information on the mean, buy, and sell rates of US dollars in comparison to other currencies. The data frame contained 4335 rows and 5 columns. The "Date" column provided the date of the exchange rate, the "Currency" column provided the name of the currency being exchanged with the US dollar, and the "Mean", "Buy", and "Sell" columns provided the average, buying, and selling exchange rates, respectively.

3. DATA PREPARATION

In the data preparation stage, the dataset was checked for duplicates and missing values, and type conversions were made as necessary.

First, the dataset was checked for duplicates using the duplicated() function. The commodity dataset had no duplicates, while the forex_data had 27 duplicated values that were handled in the data wrangling stage. The inflation dataset had one duplicated row that was dropped.

Next, missing values in the commodity dataset was checked using the isna() function. Several columns had missing values except for the date column. The missing values were filled in using the forward fill method with the fillna() function. The inflation_data and forex_data had no missing values.

In the type conversion section, the inflation dataset was converted into the date-time format by concatenating the Year and Month columns and then dropping the original columns. The day of the month was set as the last day of the month using the pd.offsets.MonthEnd(0) function. The resulting dataset was sorted by date.

The commodity dataset was also converted into a date-time format, and the date column was set as the index. The dataset was also renamed to 'date' from 'date ' with the rename() function. The forex_data was converted to date-time format by changing the 'Date' column to the date-time format and then setting it as the index. Finally, the forex_data was resampled to a monthly frequency with average daily exchange rates using the resample() function.

In the 4th stage, the three datasets, commodity_data, inflation_data, and forex_data, were merged into one DataFrame, time_series_data. The first merge combined commodity_data and inflation_data on the 'date' column, and the second merge combined time_series_data with forex_data, again on the 'date' column.

After each merge, the merged DataFrame's first few rows were checked, the 'date' column was set as the index, missing values were checked, and the first few rows of the merged DataFrame were inspected to confirm that it had the correct shape, columns, and data types.

The final merged DataFrame, time_series_data, had 207 rows and 11 columns, which included the price of bread (400g), Refined Vegetable oil (1L), Cows Milk(Fresh, Pasteurized) -500ML, Diesel (1L), Maize meal(2kg), Gasoline (1L), Annual Average Inflation, 12-Month Inflation, Mean, Buy, and Sell.

Finally, missing values in the final DataFrame were checked, which returned that there were no missing values in the DataFrame. The 'date' column was then set as the index of the DataFrame, and it was checked that the columns had the correct data types.

4.0 EXPLORATORY DATA ANALYSIS

To gain a better understanding of our dataset, we conducted various exploratory data analyses. We began with univariate analysis, which involved examining the price trends of bread, vegetable oil, milk, diesel, maize meal, gasoline, inflation, and exchange rates over the years. We then created a multipanel plot to display all six graphs, enabling us to discern trends over time.

For bivariate analysis, we also plotted a multipanel graph to explore the relationship between all commodities and inflation. Moving on to multivariate analysis, we used line charts to visualize changes in commodity prices and exchange rates over time. Additionally, we analyzed commodity prices over time and created a panel plot to visualize commodity prices, exchange rates, and inflation over time. Finally, we plotted a correlation matrix to visually identify the relationships between the different variables in the dataset and identify any significant correlations that may exist between the variables.

This exploratory data analysis allowed us to gain a deeper understanding of the dataset and its trends.

5. Modelling

5.1 Checking For Stationarity

In this section, the stationarity of the time series data was checked using the Augmented Dickey-Fuller (ADF) test. A function adf_test was defined to perform the ADF test on the data, which printed out the ADF statistic, p-value, and critical values. The adf_test function was called on each relevant column of the time series data, including bread, vegetable oil, milk, diesel, maize meal, gasoline, inflation, and exchange rate.

The ADF test was conducted on each column, and the results indicated that the time series data for bread, vegetable oil, milk, diesel, maize meal, and gasoline were non-stationary, as their ADF statistics exceeded the critical values. In contrast, the time series data for inflation was stationary, as its ADF statistic was below the critical values. The ADF test result for the exchange rate was uncertain because its ADF statistic was close to zero, and the p-value was higher than 0.05.

5.2 Differencing to achieve stationarity

The primary objective was to transform the non-stationary time series data columns into a stationary form. This was accomplished by applying differencing to the selected columns, followed by the application of the Augmented Dickey-Fuller (ADF) test to evaluate stationarity. The ADF test scrutinized if the ADF statistic fell below the critical values, which led to the acceptance of the hypothesis of stationarity. The results were aggregated in a dictionary and presented in a tabular form, which showed that all the columns attained stationarity following the application of differencing, as evidenced by the ADF statistic being lower than the critical threshold of 0.05.

5.3 Splitting data into train and test set

In the next step, the time series data was partitioned into training and test sets. The train_data and test_data variables were created to represent the training and test sets, respectively. The training set consisted of the data from the start of the time series until September 1st, 2022, while the test set contained the data from September 1st, 2022, onwards. This splitting approach ensured that the model was trained on past data and tested on future data, which is a best practice in time series analysis.

5.4 Models

5.4.1 Arima Model

In this section, the ARIMA model was used to make predictions on the test set for each commodity in the time series data. The values for p, d, and q were defined, and all possible combinations of these values were generated. For each commodity, the RMSE value was calculated for each combination of (p, d, q), and the combination with the lowest RMSE value was selected as the best fit for the ARIMA model. The RMSE values and corresponding (p, d, q) values for each commodity were stored in a dictionary and presented as a table. The results showed the RMSE values for each commodity and the corresponding values of (p, d, q) that minimized the RMSE value.after obtaining the best (p, d, q) values for each commodity, the ARIMA model was used to predict the values for the test set. The actual and predicted values were then plotted to visualize and compare the trends. These plots can help to evaluate the performance of the ARIMA model for each commodity.

5.5 Model Combinations

In this section, different model combinations (SARIMA, SES, HWES, and ARIMA) were tested to determine which combination produced the best results for the given time series data. A dictionary was defined to store the RMSE values for each model. The aim was to identify the combination of models that best captured the differences in the patterns of the data.

5.5.1 Sarima Model

In this section, the SARIMA model was used to make predictions on the test set for each commodity in the time series data. The SARIMA model, which is an extension of the ARIMA model that takes into account seasonality in the data, was fitted for each commodity with (1,1,1) and (1,1,1,12) as the order and seasonal order, respectively. The endogenous variable was selected as the train data for each commodity. The performance of the SARIMA model was evaluated on the test set by making predictions and calculating the RMSE value between the predicted values and the actual values. The RMSE values for each commodity and corresponding SARIMA model were stored in a dictionary and presented as a table. The results showed the RMSE values for each commodity and the performance of the SARIMA model on the given time series data. Based on the results, it was concluded that the SARIMA model performed better for some commodities (e.g., Milk (500ML)) than for others (e.g., Vegetable Oil (1L)). These findings suggested that the SARIMA model may not be the best choice for modeling all commodities in the time series data.

5.5.2 Simple Exponential Smoothing (SES) Model

In this section, a Simple Exponential Smoothing (SES) model was fitted to each column of the time series data. The model was used to make predictions on the test set for each commodity, and the root mean square error (RMSE) was calculated between the predicted values and the actual values to evaluate the model's performance. The SES model is a forecasting method that assigns exponentially decreasing weights over time to the previous observations, giving more weight to recent observations. For each commodity, the column was selected as the endogenous variable, and an SES model was fit using the training data. The model was then evaluated on the test set by making predictions and calculating the RMSE value. The results showed that the SES model performed better for some commodities (e.g., Milk (500ML)) than for others (e.g., Vegetable Oil (1L)). These findings suggest that the SES model may not be the best choice for modeling all commodities in the time series data. The RMSE values for each commodity and the corresponding SES model were stored in a dictionary and presented as a table.

5.5.3 Holt-Winters Exponential Smoothing (HWES) Model

In this section, the Holt-Winters Exponential Smoothing (HWES) model was used to make predictions on the test set for each commodity in the time series data. The HWES model is a variation of the Simple Exponential Smoothing (SES) model that takes into account seasonality and trend in the data. For each commodity, the endogenous variable was selected as the train data and an HWES model was fit with a seasonal period of 12, additive trend, and additive seasonality. The model was then evaluated on the test set by making predictions and calculating the RMSE value between the predicted values and the actual values. The RMSE values for each commodity and corresponding HWES model were stored in a dictionary and presented as a table. The results showed the RMSE values for each commodity and the performance of the HWES model on the given time series data. Based on the results, it can be concluded that the HWES model outperformed the SARIMA and SES models for most commodities in the time series data.

5.5.4 Arima Model

We then fitted an ARIMA (AutoRegressive Integrated Moving Average) model on each time series column in the given dataset. The order of the ARIMA model used was (1,1,1).

The ARIMA models were trained on the training data, and the performance was evaluated on the test data using RMSE (Root Mean Squared Error). The lower the RMSE, the better the model's performance.

The results showed the RMSE values for each time series column, where the lowest RMSE values indicated better performance of the model. Compared to SES and HWES models, the

ARIMA model generally performed better for most of the time series columns except for the "Exchange Rate (USD)" and "Inflation" columns where HWES performed better.

5.5.5 Models Comparison

We created a data frame to compare the RMSE (Root Mean Squared Error) values of each of the four models (SARIMA, SES, HWES, and ARIMA) on each of the time series columns. This data frame showed the RMSE values for each column and each model.
Next, we tried to determine the best model for each column by finding the model with the lowest RMSE value. The model with the lowest RMSE was considered the best fit for the time series data. For example, for the "Bread(400g)" column, the SARIMA model performed the best, while for the "Vegetable Oil (1L)" column, the HWES model performed the best.

5.5.6 Model Fitting

After all this, we fitted the best model for each time series using the entire training set, made predictions on the testing set, and evaluated the model performance by comparing the predicted values with the actual values to calculate the RMSE value for each model.

Then, we created a plot of the actual and predicted values for each time series using the best model. The plot allowed us to visualize the accuracy of the model predictions and how well they fit the actual data.

The results of the model fitting and prediction showed that the RMSE value varied depending on the time series and the model used. For instance, the SARIMA model performed best for the "Bread(400g)" and "Milk(500ML)" columns, while the HWES model performed best for the "Vegetable Oil (1L)", "Diesel (1L)", "Gasoline (1L)", and "Inflation" columns. The Exchange Rate (USD) column had the lowest RMSE with the SARIMA model. The plots of actual and predicted values for each time series allowed us to visualize the accuracy of the model predictions and how well they fit the actual data.

5.6 Model Forecasting

At this point, we aimed to forecast future values for each time series in the dataset using the best model selected for each series to fit the model. We fitted the models and forecasted the future values for the next 12 months.

The forecasted values were stored in the forecast_df DataFrame, and the index of the DataFrame was set to be the forecasted dates. The forecasted dates started from 2023-03-31 and went up to 2024-02-29 with a frequency of one month.

Finally, the forecast_df DataFrame showed the forecasted values for each time series for the next 12 months. The data frame showed the forecasted values for Bread (400g), Vegetable Oil (1L), Milk (500ML), Diesel (1L), Maize meal (2kg), Gasoline (1L), Inflation, and Exchange Rate

(USD). The forecasts generated by our models can be used for decision-making and planning purposes.

5.7 LSTM MODEL

Finally, an LSTM model was fitted to the data due to its numerous advantages. First and foremost Ability to handle long-term dependencies: Traditional RNNs have a problem with vanishing gradients, which makes it difficult for them to capture long-term dependencies in time series data. LSTMs are designed to address this issue by using a more complex architecture that includes a memory cell, input gate, output gate, and forget gate. These components enable LSTMs to selectively remember and forget information over long periods of time. Secondly, LSTMs can handle input sequences of varying lengths and can output sequences of different lengths, which is important for time series analysis where the length of the input and output sequences can vary. Thirdly, time series data can be noisy and contain outliers, missing values, or other types of irregularities. LSTMs are effective at handling noisy data and can learn to filter out irrelevant information. Time series data often contains non-linear relationships between the input features and the target variable. LSTMs are good at capturing non-linear relationships in data and can learn complex patterns that might be difficult to identify using traditional statistical methods. This particular model produced the best possible RMSE for our models and was one of the most accurate predictions.

6 . MINIMAL VIABLE PRODUCT

After conducting a thorough evaluation of various models, including ARIMA, SARIMA, HWES, and SES, the LSTM (Long Short-Term Memory) model was chosen as the Minimum Viable Product (MVP) for the time series project. The decision was primarily based on its superior performance, specifically, its low RMSE (Root Mean Squared Error) values, which indicates a high level of accuracy.

The LSTM model demonstrated an ability to capture complex patterns and relationships within the time series data, allowing for more precise predictions of future values. Its ease of implementation and minimal preprocessing requirements made it a practical and cost-effective choice for an MVP.

The LSTM model was able to learn quickly from a small amount of training data and generate accurate predictions, which is crucial for an MVP. Its efficiency and accuracy make it a promising solution for the time series project.

7 . CONCLUSION

1. Based on the results of this project, our time series model can provide valuable insights into commodity prices and economic indicators such as inflation rates and exchange rates. The developed models show accuracy with an RMSE of less than 5% and can be used to forecast prices for the next 12 months with a reasonable degree of confidence.
2. Investors and commodity buyers can use these forecasts to make informed decisions regarding their investments or purchasing decisions. For instance, they can take advantage of opportunities identified by the market analysis to buy or sell commodities at the right time to maximize profits.
3. This web-based application developed as part of this project provides traders and investors with reliable, real-time commodity price predictions, which can be continuously updated for ongoing accuracy. This will enable them to stay on top of the market and make informed decisions based on the latest information.
4. Overall, our project highlights the potential of time series modeling in providing valuable insights into the dynamics of commodity prices and economic indicators, and how these insights can be leveraged to inform decision-making for investors and commodity buyers.

7. RECOMMENDATIONS

1. Traders and Investors: By providing traders and investors with reliable real-time commodity price predictions, we can help them make better investment decisions. They can use our predictions to identify market trends and capitalize on opportunities for profit. This can be especially beneficial for traders and investors who specialize in commodities and are looking for ways to gain an edge in the market.
2. Farmers and Producers: The time series commodity price predictions can be useful for farmers and producers who rely on the prices of commodities like maize meal and diesel to make business decisions. By tracking the prices of these commodities, farmers can adjust their production schedules and ensure they have a steady supply of essential goods to sell at the market. This can help them maximize profits and ensure the sustainability of their business.
3. Monitor inflation rates: Since inflation rates can have a significant impact on the prices of commodities, it is important to keep a close eye on them so that Investors can use the forecasts to anticipate changes in inflation rates and make adjustments to their investments accordingly. High inflation rates can lead to higher prices of commodities, and therefore, it may be wise to invest in commodities that are likely to rise in price during inflationary periods.
4. Government Agencies: our time series commodity price predictions can be valuable to government agencies responsible for regulating the prices of essential commodities. By providing accurate forecasts, these agencies can better manage supply and demand, prevent price spikes, and ensure that essential goods are accessible to all citizens.