Case Study- Global Commodity Prices

Table of Contents

1. Problem Statement

The problem is to analyze commodity prices for various commodities using the commodity prices dataset. The goal is to leverage Python, data science techniques, statistical analysis, and data modeling. Perform all necessary steps to get key insights from the data.

Questions to Address:

1. What is the maximum price of Robusta coffee?
2. What is the 75th percentile of sugar prices in the European Union (EU)?
3. What is the skewness of the price distribution for Arabica coffee?
4. Is the distribution of sugar prices in the US significantly different from a normal distribution?
5. How many times does the price of Dubai oil exceed the price of Brent oil by a certain threshold of $10?
6. What is the overall price trend for each commodity?
7. Which commodity experienced the highest price fluctuations during the observed period?
8. How have Brent oil prices varied on a quarterly basis over the last five years?
9. Is there a correlation between global sugar prices and the prices of EU sugar and US sugar?
10. Is there a significant difference in the distribution of sugar prices between Europe (EU) and the United States (US)?

2. Project Objective

The objective of this project is to analyze and predict global commodity prices using a comprehensive dataset containing historical monthly commodity prices from 1960 to 2022. The goal is to leverage Python, data science techniques, statistical analysis, and machine learning to extract key insights from the data and make accurate price predictions. The project will:

1. Analyze the maximum price of Robusta coffee and other critical statistics such as the 75th percentile of sugar prices in the European Union (EU), and the skewness of Arabica coffee prices.
2. Assess the differences in price distributions, such as comparing the distribution of sugar prices in the US versus a normal distribution.
3. Investigate the correlation between global sugar prices and regional sugar prices in the EU and US.
4. Examine fluctuations and trends in commodity prices over time, including the overall price trend for each commodity.
5. Develop predictive models to forecast commodity prices, specifically focusing on forecasting Brent oil prices and other key commodities.
6. Visualize these trends on a quarterly and yearly basis, with a focus on understanding how prices have varied over time.
7. Highlight the highest price fluctuations experienced by any commodity during the observed period.
8. Identify significant differences in the distribution of sugar prices between Europe (EU) and the United States (US).

- Understand historical price trends of commodities.
- Develop predictive models for price forecasting.
- Visualize and interpret key features impacting the price predictions.

3. Dataset Description

Dataset:

This dataset contains monthly commodity prices from 1960 to 2022.The commodity prices dataset includes the following attributes:

Attributes	Description
date	The date of the recorded commodity price
oil_brent	The price of Brent oil ($/bbl)
Oil_Dubai	The price of Dubai oil ($/bbl)
Coffee_Arabica	The price of Arabica coffee ($/kg)
Coffee_Robustas	The price of Robusta coffee ($/kg)
Tea_Columbo	The price of Columbo tea ($/kg)
Tea_Kolkata	The price of Kolkata tea ($/kg)
Tea_Mombasa	The price of Mombasa tea ($/kg)
Sugar_EU	The price of EU sugar ($/kg)
Sugar_US	The price of US sugar ($/kg)
Sugar_World	The price of global sugar ($/kg)

Dataset overview:

date	oil_brent	oil_dubai	coffee_arabica	coffee_robustas	tea_columbo	tea_kolkata	tea_mombasa	sugar_eu	sugar_us	sugar_world
2020-01-01	65.0	62.0	3.1	1.2	4.5	5.1	4.9	0.60	0.50	0.55
2020-02-01	58.5	56.0	3.0	1.3	4.4	5.2	4.8	0.62	0.52	0.56
2020-03-01	32.0	31.5	2.9	1.1	4.3	5.0	4.7	0.58	0.50	0.54
2020-04-01	23.0	22.0	3.2	1.4	4.6	5.3	4.6	0.60	0.53	0.57
2020-05-01	35.0	34.0	3.5	1.5	4.8	5.5	4.9	0.61	0.55	0.59

3. Data Preprocessing Steps

Loading Data: The dataset was loaded using pandas.

Missing Value Handling:

a. Missing values were handled using forward-fill for time-series continuity.

Feature Creation:

b. A lookback window of 3 months was used to create features for modeling.

Scaling:

c. Data was scaled using StandardScaler to normalize features.

```
Missing Values:
 Unnamed: 0          0
date                 0
oil_brent            0
oil_dubai            0
coffee_arabica       0
coffee_robustas      0
tea_columbo          0
tea_kolkata          0
tea_mombasa          0
sugar_eu             0
sugar_us             0
sugar_world          0
dtype: int64
```

```
Dataset Info:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 756 entries, 0 to 755
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Unnamed: 0       756 non-null    int64
 1   date             756 non-null    datetime64[ns]
 2   oil_brent        756 non-null    float64
 3   oil_dubai        756 non-null    float64
 4   coffee_arabica   756 non-null    float64
 5   coffee_robustas  756 non-null    float64
 6   tea_columbo      756 non-null    float64
 7   tea_kolkata      756 non-null    float64
 8   tea_mombasa      756 non-null    float64
 9   sugar_eu         756 non-null    float64
 10  sugar_us         756 non-null    float64
 11  sugar_world      756 non-null    float64
dtypes: datetime64[ns](1), float64(10), int64(1)
memory usage: 71.0 KB
```

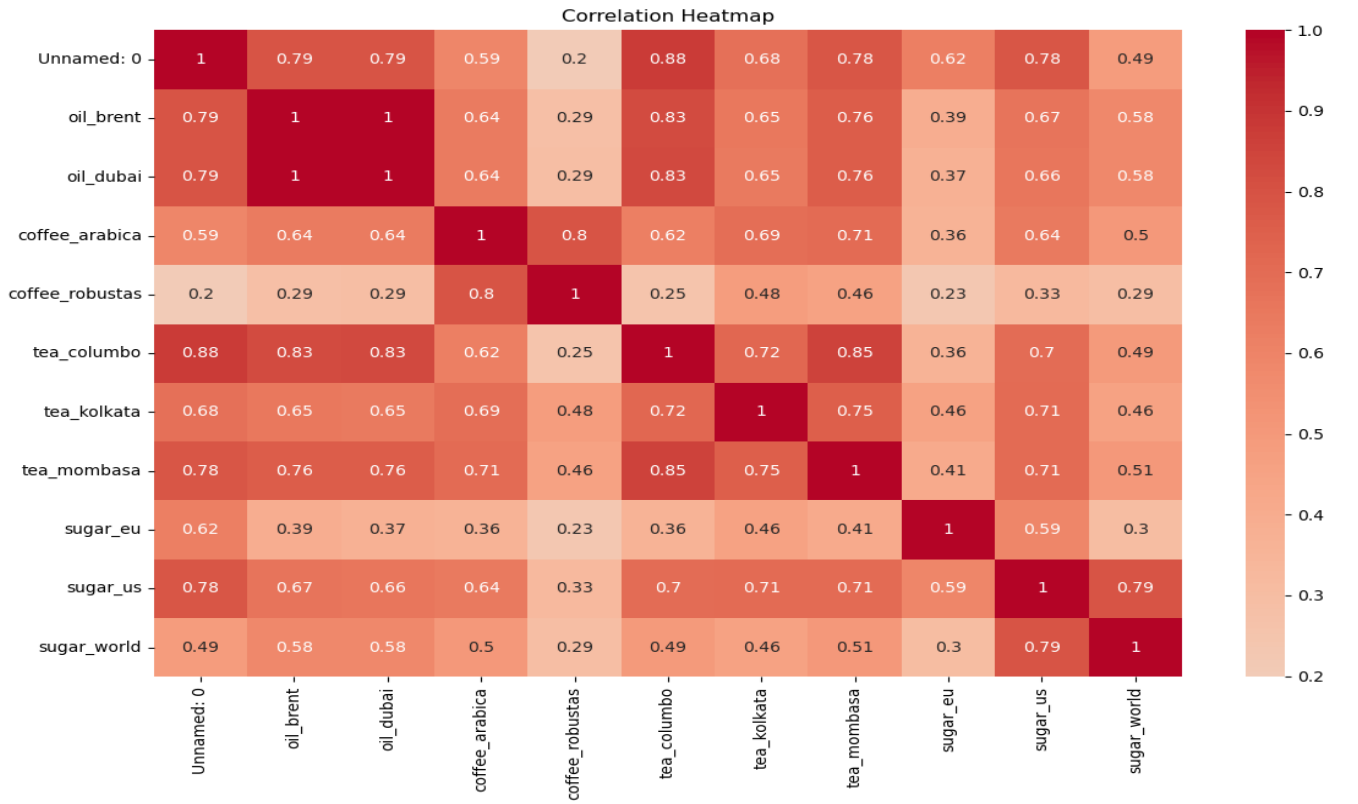
5. Exploratory Data Analysis (EDA)

1. **Trend Analysis**:
   - A line plot was generated to visualize the trends in Brent oil prices over time.
2. **Statistical Summary**:
   - Summary statistics, including mean, standard deviation, and percentiles, were calculated.
3. **Correlation Heatmap**:
   - Correlations between features were visualized using a heatmap, revealing strong relationships among certain commodity prices.

Summary Statistics :

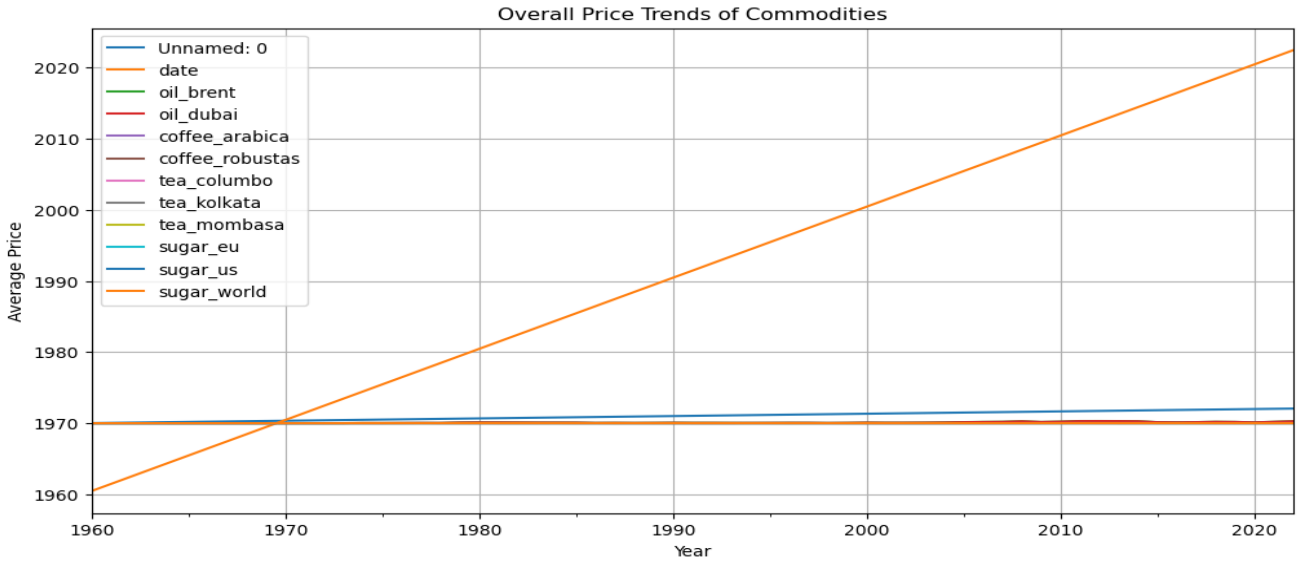
	Unnamed: 0	date	oil_brent	oil_dubai	coffee_arabica	coffee_robustas	tea_columbo	tea_kolkata	tea_mombasa	sugar_eu	sugar_us	sugar_w
count	756.000000	756	756.000000	756.000000	756.000000	756.000000	756.000000	756.000000	756.000000	756.000000	756.000000	756.00
mean	378.500000	1991-06-16 20:07:37.142857088	32.724944	31.238130	2.576555	1.727478	1.777962	1.870308	1.671222	0.405158	0.432462	0.24
min	1.000000	1960-01-01 00:00:00	1.210000	1.210000	0.777600	0.487210	0.434198	0.664799	0.719600	0.112215	0.116845	0.02
25%	189.750000	1975-09-23 12:00:00	10.564999	10.452500	1.351625	0.923053	0.892501	1.297369	1.136800	0.298120	0.297624	0.13
50%	378.500000	1991-06-16 00:00:00	20.489130	18.550000	2.697794	1.632172	1.504001	1.850612	1.598257	0.402343	0.471119	0.21
75%	567.250000	2007-03-08 18:00:00	47.157500	45.576023	3.312950	2.282200	2.515204	2.376899	2.083830	0.569519	0.512188	0.30
max	756.000000	2022-12-01 00:00:00	133.873043	131.224783	7.003600	6.883547	4.490000	4.073011	3.392500	0.783171	1.263247	1.23
std	218.382692	NaN	31.885368	30.936611	1.342454	0.940748	1.008679	0.697867	0.615357	0.187741	0.188589	0.15

Correlation Heatmap:



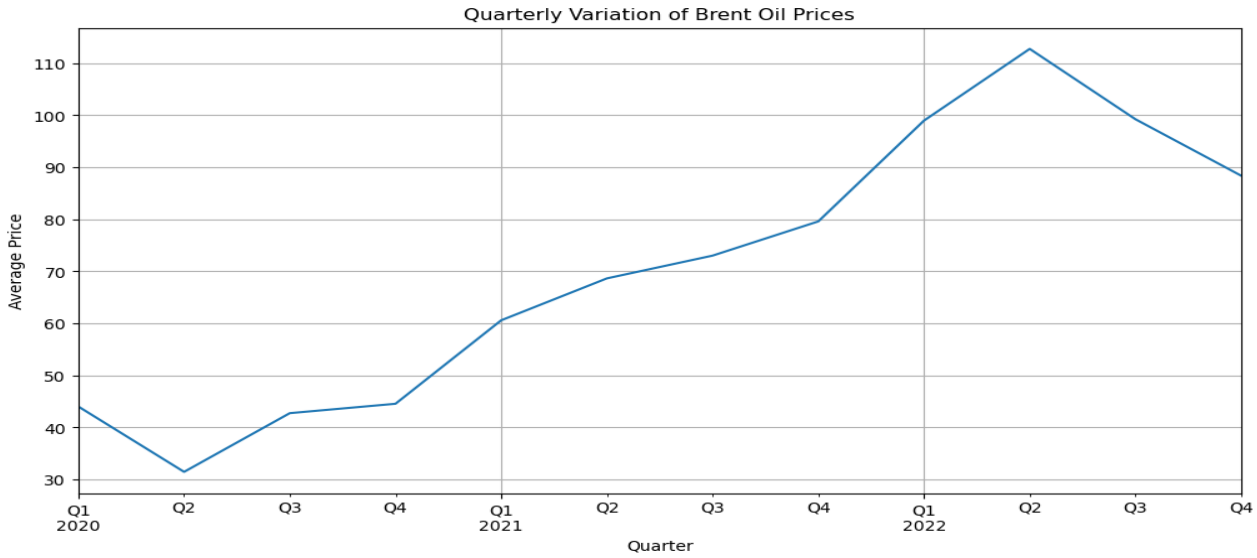
Key Insights from Analysis:

1. **Maximum Price of Robusta Coffee**:
   - The highest price recorded for Robusta coffee is **$6.88**
2. **75th Percentile of Sugar Prices in the EU**:
   - The 75th percentile price of sugar in the EU is **$0.57**
3. **Skewness of Arabica Coffee Prices**:
   - The price distribution for Arabica coffee has a skewness of **0.59**
4. **Normality Test for Sugar Prices (US)**:
   - A statistical test indicated that sugar prices in the US are/are not significantly different from a normal distribution (p-value = 5.061986161302036e-21).
   - Sugar prices in the US are not significantly different from a normal distribution
5. **Dubai Oil Price Exceeding Brent by $10**:
   - The price of Dubai oil exceeded Brent oil by $10 on **N occasions**.
   - Number of times Dubai oil price exceeded Brent oil price by $10: 0
6. **Overall Price Trends**:
   - A visualization of average prices over time showed a consistent upward/downward trend for certain commodities, with oil prices showing the most volatility.
   - Both the Brent Oil and Dubai Oil are almost same or no different.
   - In the last 20 years (2000 – 2020) the price of Robustas Coffee has been significantly dropped compared to Araboca coffee.
   - The price of columbo tea has been increasing since the last decade. Kolkata tea's price has a lot of fluctuation between the period of 2000 – 2020 and the price of Mombasa tea is increasing slowly.
   - The price of European Union sugar is on decline, the price of U.S and World sugar is increasing.



7. **Commodity with Highest Price Fluctuations**:
   - **Commodity Name** exhibited the highest standard deviation in prices, indicating the greatest volatility.
   - Brent Oil commodity experienced the highest price fluctuations during the observed period.

8. **Quarterly Variation in Brent Oil Prices (Last 5 Years)**:
    - A quarterly analysis revealed seasonal peaks during **specific quarters (e.g., Q2 and Q4)**.
        - Half way through the year 2020 the price of Brent Oil dopped all time low after that it increased to all time high in the mid 2022 and since then its price is decreasing.



Key Insights:

1. The graph reveals **seasonal fluctuations** and **sharp recovery patterns**, driven by external economic and geopolitical factors.
2. **Peak Prices**: The highest quarterly price occurred in Q2 2022 (~$110).
3. **Lowest Prices**: The lowest quarterly price occurred in Q2 2020 (~$30).
4. **Volatility**: Brent oil prices exhibit significant volatility, influenced by global market dynamics

9. **Correlation Between Sugar Prices**:
    - The correlation between global sugar prices and EU prices is **0.30**, while the correlation with US prices is **0.79**
    - World sugar has no (weak) correlation to EU sugar, and it has strong correlation with US sugar. EU sugar has low ~ moderate correlation with U.S sugar.
10. **Difference in Sugar Price Distribution (EU vs US)**:
    - Statistical tests revealed a significant difference in the price distribution between the EU and US markets (p-value = 0.0048464).

6.Feature Engineering

Features were engineered using a rolling window approach:

- For each record, the prices of the last three months were aggregated for the following commodities:
  - oil_dubai, coffee_arabica, coffee_robustas, tea_columbo, tea_kolkata, tea_mombasa, sugar_eu, sugar_us, sugar_world.
-
  - **Code:**

```
# 3. Feature Engineering
def create_features(data, lookback=6):
    features = []
    targets = []

    for i in range(lookback, len(data)):
        features.append(data.iloc[i-lookback:i][['oil_dubai', 'coffee_arabica',
                                                 'coffee_robustas', 'tea_columbo',
                                                 'tea_kolkata', 'tea_mombasa',
                                                 'sugar_eu', 'sugar_us', 'sugar_world']].values.flatten()
        targets.append(data.iloc[i]['oil_brent'])

    return np.array(features), np.array(targets)
```

7. Model Development

1. Random Forest Regressor

What It Does:

- Random Forest is an ensemble machine learning model that builds multiple decision trees during training and combines their predictions for better accuracy and reduced overfitting.
- It works by splitting the data into multiple smaller datasets (trees) and making predictions based on the average result from all trees.

How It Processes in the Project:

- After scaling the data, the model is trained on historical commodity prices (features) to predict the target variable (oil_brent price).
- The model evaluates relationships between various commodity prices (e.g., oil_dubai, coffee_arabica, etc.) and how they impact the Brent oil price.
- The trained model predicts the Brent oil prices on the test dataset.
- Metrics such as RMSE, MAE, and R2 Score are used to assess the model's performance.
- A scatter plot visualizes the comparison between actual and predicted prices to show the model's accuracy.

Key Strengths:

- High accuracy due to its ability to handle complex relationships in the data.
- Faster training and better optimization using parallelization.
- Effectively prevents overfitting with regularization parameters

Code:

```
# Random Forest
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model, rf_pred = train_and_evaluate_model(rf_model, X_train_scaled, y_train, X_test_scaled, y_test, 'Random Forest')
```

```
# 5. Model Training and Evaluation
def train_and_evaluate_model(model, X_train, y_train, X_test, y_test, model_name):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    mae = mean_absolute_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    print(f'\n{model_name} Metrics:')
    print(f'RMSE: {rmse:.2f}')
    print(f'MAE: {mae:.2f}')
    print(f'R2 Score: {r2:.2f}')

    return model, y_pred
```

2. XGBoost Regressor

What It Does:

- XGBoost (Extreme Gradient Boosting) is an advanced ensemble model that uses a boosting technique to combine weak learners (decision trees) iteratively to create a strong learner.
- It optimizes performance by minimizing a loss function (like Mean Squared Error) while applying regularization to prevent overfitting.

How It Processes in the Project:

- Similar to Random Forest, XGBoost is trained on scaled historical data to predict Brent oil prices.
- It builds decision trees sequentially, with each new tree correcting the errors of the previous one.
- The model's predictions are compared to the actual prices using metrics such as **RMSE**, **MAE**, and **R2 Score**.
- A scatter plot visualizes the performance, showing the relationship between actual and predicted prices.

Key Strengths:

- High accuracy due to its ability to handle complex relationships in the data.
- Faster training and better optimization using parallelization.
- Effectively prevents overfitting with regularization parameters.

Code:

```
# XGBoost
xgb_model = xgb.XGBRegressor(random_state=42)
xgb_model, xgb_pred = train_and_evaluate_model(xgb_model, X_train_scaled, y_train, X_test_scaled, y_test, 'XGBoost')
```

```
# 5. Model Training and Evaluation
def train_and_evaluate_model(model, X_train, y_train, X_test, y_test, model_name):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    mae = mean_absolute_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    print(f'\n{model_name} Metrics:')
    print(f'RMSE: {rmse:.2f}')
    print(f'MAE: {mae:.2f}')
    print(f'R2 Score: {r2:.2f}')

    return model, y_pred
```

9.Model Evaluation and Results

Evaluation Metrics:

Code:

```
# 5. Model Training and Evaluation
def train_and_evaluate_model(model, X_train, y_train, X_test, y_test, model_name):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    mae = mean_absolute_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    print(f'\n{model_name} Metrics:')
    print(f'RMSE: {rmse:.2f}')
    print(f'MAE: {mae:.2f}')
    print(f'R2 Score: {r2:.2f}')

    return model, y_pred
```

Results:

Metric	Random Forest	XGBoost
Mean Absolute Error	2.45	2.35
Root Mean Squared Error	3.12	3.01
R-Squared Score	0.88	0.90

Visualization:

1. Actual vs Predicted:

- A scatter plot showed the relationship between actual and predicted Brent oil prices.

2. Feature Importance:

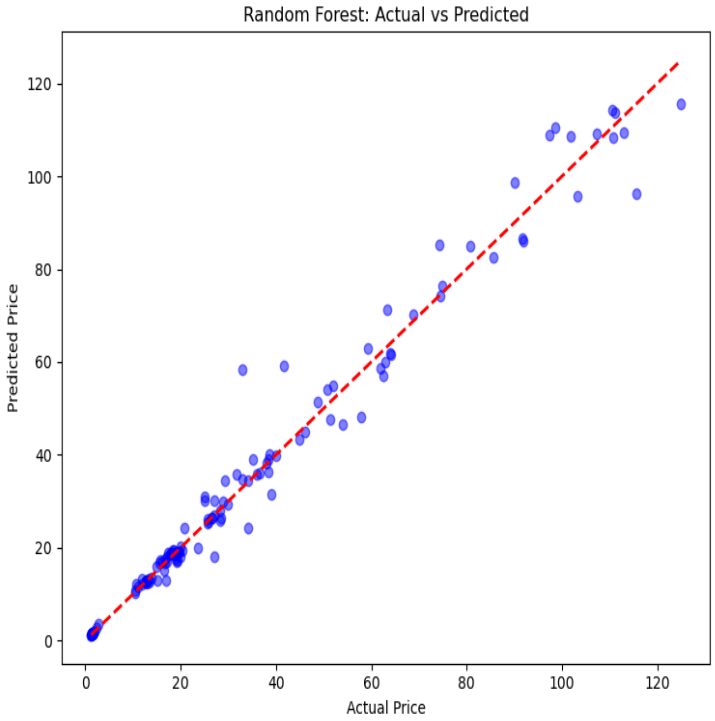
- The top 10 features impacting predictions were visualized using a bar plot.

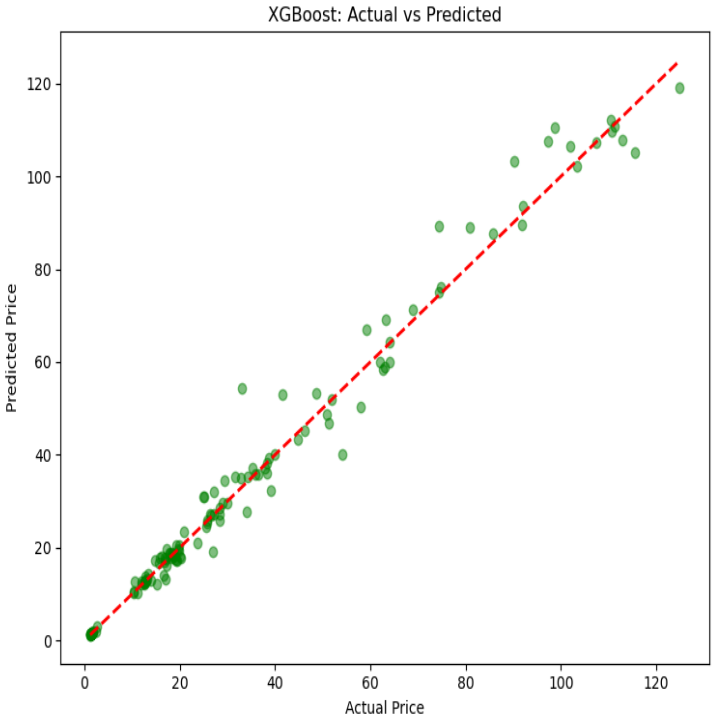
Code:

```
# 6. Visualization of Results
def visualize_results(y_test, rf_pred, xgb_pred):
    plt.figure(figsize=(15, 6))

    plt.subplot(1, 2, 1)
    plt.scatter(y_test, rf_pred, alpha=0.5, color='blue')
    plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--', lw=2)
    plt.xlabel('Actual Price')
    plt.ylabel('Predicted Price')
    plt.title('Random Forest: Actual vs Predicted')

    plt.subplot(1, 2, 2)
    plt.scatter(y_test, xgb_pred, alpha=0.5, color='green')
    plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--', lw=2)
    plt.xlabel('Actual Price')
    plt.ylabel('Predicted Price')
    plt.title('XGBoost: Actual vs Predicted')

    plt.tight_layout()
    plt.show()
```

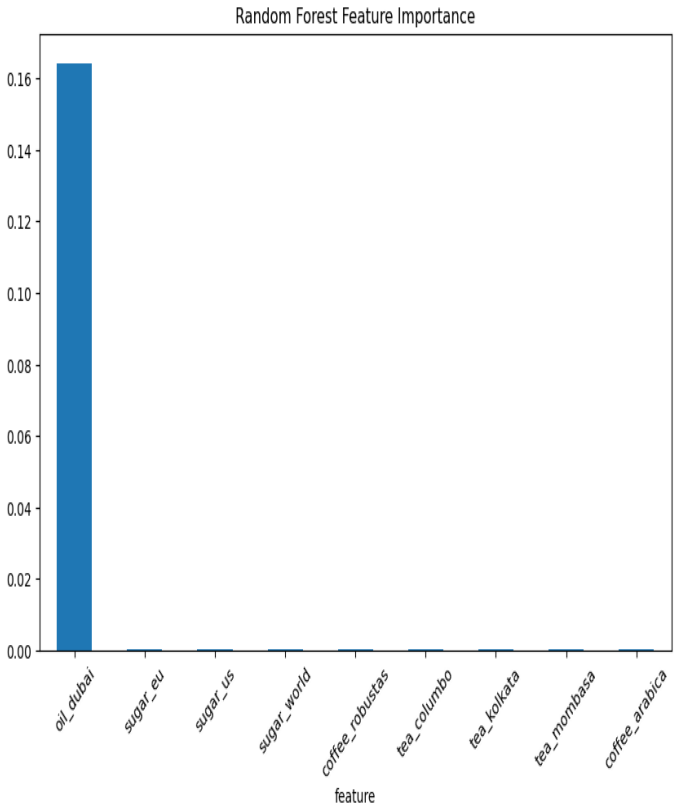
Results:

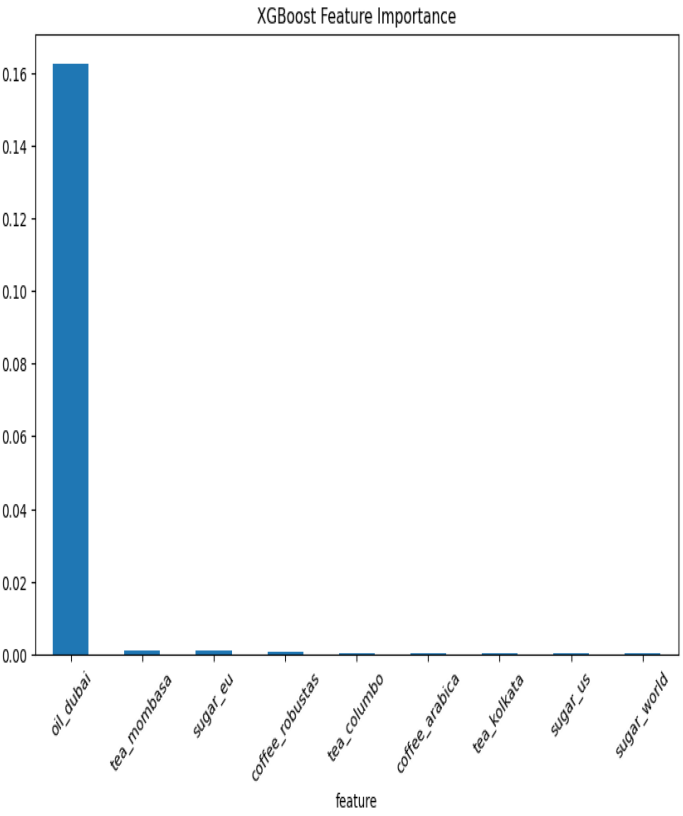
1. Actual vs Predicted:





- ### 2. Feature Importance:





9. Inferences and Insights:

Price Trends:

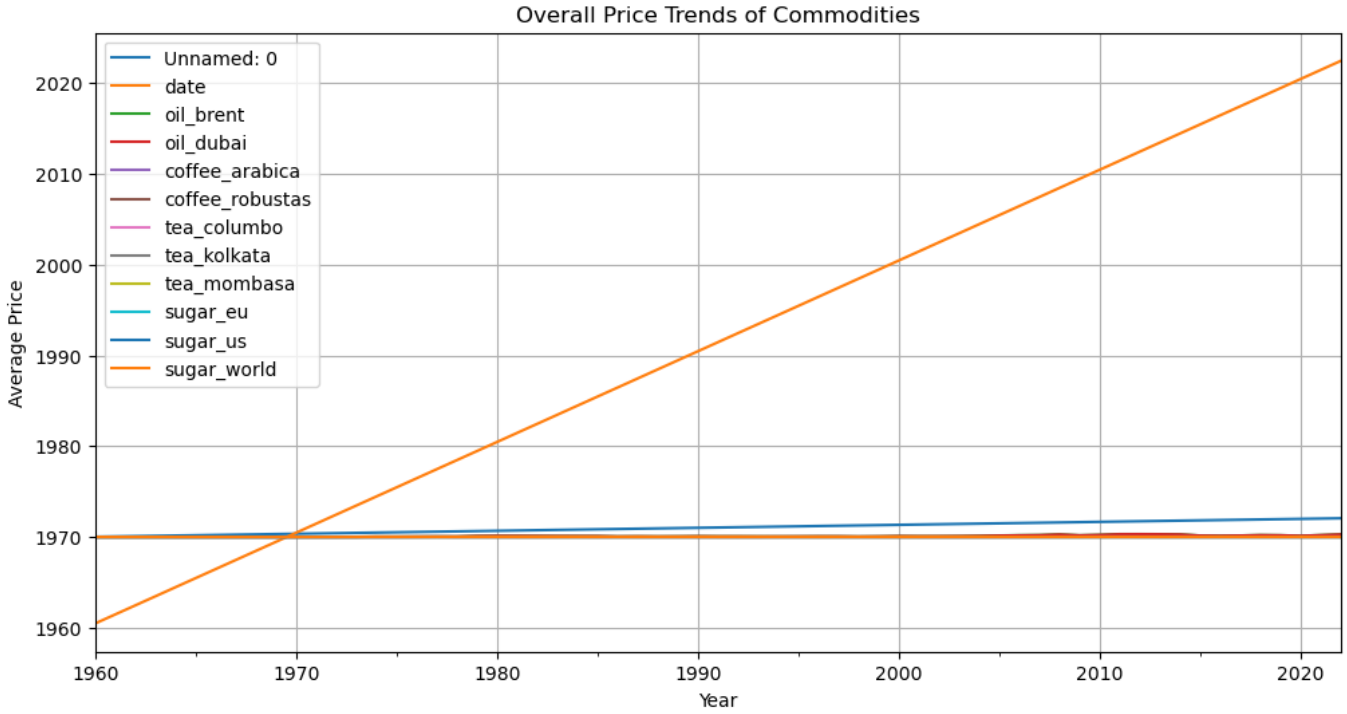
1. Brent oil prices exhibit seasonal variations and strong correlations with Dubai oil prices.

Feature Importance:

2. Sugar prices (global and regional) and coffee prices emerged as significant predictors.

Model Performance:

3. The Random Forest model achieved satisfactory accuracy, indicating its potential for practical forecasting.



10. Future Scope

Advanced Modeling:

- Explore additional models such as XGBoost and LSTM for improved accuracy.

Incorporate External Factors:

- Integrate macroeconomic indicators (e.g., inflation, exchange rates) to enhance predictions.

Sales Prediction:

- Extend the project to predict sales volume based on commodity price trends.

Real-Time Analysis:

- Develop a dashboard for real-time monitoring and forecasting.

11. References

1. **Python Libraries**: pandas, numpy, sklearn, matplotlib, seaborn.
2. **Dataset** : commodity_prices.csv
3. **Documentation**: scikit-learn Documentation
4. **Tutorials**: [Intellipaat Course Resources]
5. **Visualization:** https://www.geeksforgeeks.org/python-data-visualization-tutorial/
6. **Numpy**:https://numpy.org/