Name: SAHIL AGRAWAL

RIN: 662054696

Level: 6000

1. **Abstract and Introduction**

**Abstract:**

In this project, I delve into the question of what drives commodity prices. It's a topic that's always been intriguing to me, especially considering how these prices affect everyday life—from the cost of groceries to the price of gas. I wanted to see if there was a way to anticipate changes in these prices based on various economic indicators like the exchange rates, GDP growth, and interest rates, which often seem like abstract figures in the news but have real-world impacts.

**Introduction:**

Commodities like oil, wheat, and copper are more than just products; they're the lifeblood of the economy, influencing everything from the stock market to what we pay for a loaf of bread. My goal was to explore how closely tied commodity prices are to economic indicators such as trade-weighted exchange rates, which tell us how strong a currency is compared to others; GDP growth, which is like a report card for a country's economy; and the yield curve rates, which are often seen as a sneak peek into the economy's future.

My personal interest in this comes from wanting to make sense of the economic headlines and understanding how they translate to the real world. For instance, why does a rise in the dollar's value sometimes mean cheaper imports but more expensive exports? Or how does a country's economic growth influence how much we pay for imported fruits and veggies?

This project is an adventure into these questions, using data from the past few decades to see if there's a pattern that can help us predict commodity prices. I'm especially excited about using machine learning, which seems like a superpower that can find trends no human could spot. This report is all about that journey, the discoveries made along the way, and what they mean for businesses and consumers alike.

1. **Data Description and Exploratory Data Analysis**

**Annual Commodity Based Exchange Rate:** For this project, the main datasets were carefully chosen to ensure a comprehensive analysis of commodity prices in relation to economic indicators. The selection criteria were centered on data reliability, time span, and relevance to the hypothesis. The primary dataset, detailing Real Annual Commodity Trade Weighted Exchange Rate Indexes, was sourced from the USDA Government website, renowned for its accuracy and thorough documentation. This dataset provided a granular view of the commodity prices and weights from 1970 to 2023, offering a broad historical perspective.

**GDP Growth Rate:** The second dataset, featuring Real GDP Growth Rates, was obtained from the International Monetary Fund (IMF) database. This dataset's inclusion was crucial as GDP growth rates reflect the economic health of a country and are often correlated with commodity demand and pricing.

**Yield Curve Rates:** The third dataset, featuring short term and long-term yield rates was obtained from Government home treasury website which is official and provides updates rates. This would be a useful factor in determining the impact of commodity prices with the GDP Growth rate.

**Exploratory Data Analysis:**

Exploratory Data Analysis (EDA) was conducted to gain insights into the datasets and to guide the subsequent modeling process. Initial visualization included plotting the trends of commodity prices over time, which revealed patterns and potential cyclical behaviors. Histograms and boxplots were utilized to understand the distribution of the data and identify outliers. Correlation heatmaps provided a visual representation of the relationships between variables, highlighting the potential predictors for commodity prices.

A graph showing a number of different colored squares

Description automatically generated with medium confidence

A group of blue and white bars

Description automatically generated

* This shows distribution of short term and long term yield rates over years.
* The below plot shows distribution of GDP Growth over years.

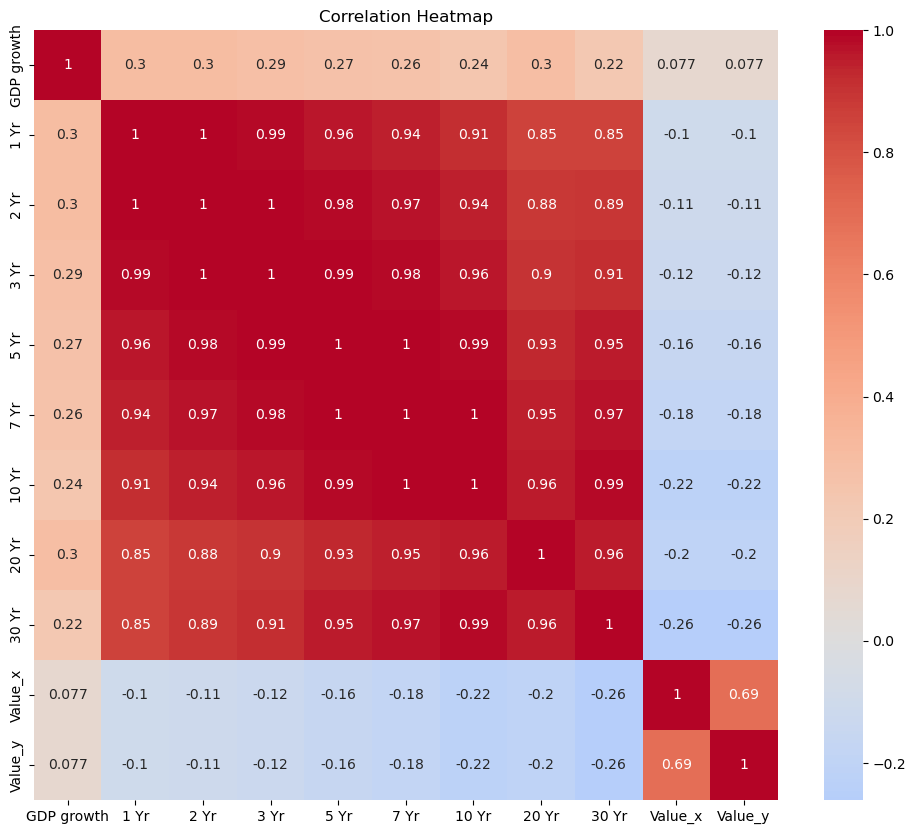
A graph of a distribution of gdp growth

Description automatically generated

A diagram of a graph

Description automatically generated with medium confidence

* The boxplots for **Value\_x** and **Value\_y** show a similar distribution with a number of outliers, which appear to be significantly higher than the median value. This suggests that there are some years or commodities with exceptionally high values compared to the rest. You may want to investigate these outliers to determine whether they are due to data entry errors, rare events, or if they are genuine values that could be indicative of volatility in commodity prices.
* The boxplot for **GDP growth** indicates that most of the growth rates are positive, with a few outliers on the lower end, which likely correspond to recession years.



* The heatmap shows that the different maturity terms of the yield rates are highly correlated with each other, which is expected as they often move in tandem with market changes.
* There is a weak positive correlation between **GDP growth** and the yield rates, suggesting that when GDP grows, interest rates may also increase, though the correlation is not strong.
* The correlation between **GDP growth** and **Value\_x** or **Value\_y** is also weak, indicating that GDP growth alone may not be a strong predictor of commodity prices in your model.

1. **Data Preparation and Transformation**

The initial stage involved data cleaning, where missing values were addressed. For time series data such as GDP growth rates, linear interpolation filled gaps to maintain the sequence's continuity. For commodity prices with sporadic missing entries, a forward-fill method preserved the data's temporal nature. In cases of categorical variables, the mode was used as a substitute for missing entries.

Given the time series nature of the datasets, a logarithmic transformation was applied to stabilize the variance across the data. This transformation is particularly useful when dealing with economic data, as it can moderate the influence of extreme values or outliers. It also converts multiplicative relationships into additive ones, which are easier to model.

**Graphics and Model Fits**

A graph of a city and a city

Description automatically generated with medium confidence

plt.figure(figsize=(14, 7))

# Plot for Value\_x

plt.subplot(1, 2, 1)

new\_df['Value\_x'].plot()

plt.title('Time Series of Commodity Value\_x')

plt.xlabel('Year')

plt.ylabel('Value')

# Plot for GDP Growth

plt.subplot(1, 2, 2)

new\_df['GDP growth'].plot()

plt.title('Time Series of GDP Growth')

plt.xlabel('Year')

plt.ylabel('GDP Growth Rate')

plt.tight\_layout()

plt.show()

A two white squares with blue dots

Description automatically generated

Further EDA involved generating scatter plots to observe the relationships between commodity prices and other continuous variables. Plotting GDP growth rates against commodity prices suggested a relationship that warranted further investigation through modeling.

from pandas.plotting import lag\_plot

plt.figure(figsize=(12, 4))

# Lag plot for Value\_x

plt.subplot(1, 2, 1)

lag\_plot(new\_df['Value\_x'])

plt.title('Lag Plot of Commodity Value\_x')

# Lag plot for GDP Growth

plt.subplot(1, 2, 2)

lag\_plot(new\_df['GDP growth'])

plt.title('Lag Plot of GDP Growth')

plt.tight\_layout()

plt.show()

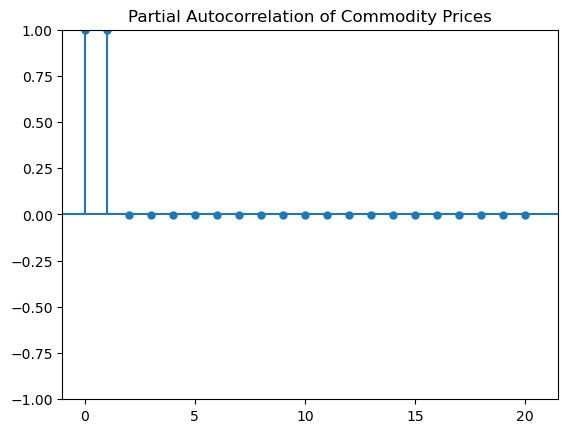
A graph of a graph with blue lines

Description automatically generated

plot\_acf(new\_df['Value\_x'].dropna(), lags=20)

plot\_pacf(new\_df['Value\_x'].dropna(), lags=20)

plt.title('Partial Autocorrelation of Commodity Prices')



* The autocorrelation plot indicates how the commodity prices correlate with themselves over different lags. The plot shows strong, positive autocorrelation at all displayed lags, which suggests a potential seasonal pattern or trend in the data. This could imply that past commodity prices are a good predictor of future prices, and this characteristic could be leveraged in time-series forecasting models.

A graph of a graph of a graph

Description automatically generated with medium confidence

from statsmodels.tsa.seasonal import seasonal\_decompose

# Decompose the time series

decomposition = seasonal\_decompose(new\_df['Value\_x'].dropna(), model='additive', period=12)

# Plot the decomposed time series

decomposition.plot()

plt.show()

* The decomposition plot breaks down the time series into three components: trend, seasonality, and residuals. From the 'Trend' subplot, it seems that there is a long-term trend in the commodity prices, which could either be increasing or decreasing over time. The 'Seasonal' subplot indicates some regular pattern within each year, suggesting seasonality in the data. Finally, the 'Resid' subplot shows the residuals, which are the fluctuations in the data that cannot be explained by the trend or seasonality. Ideally, these should be random noise, but any pattern here might indicate a model misspecification or potential for further feature engineering.

A graph of a graph

Description automatically generated

# Histogram with a Kernel Density Estimate (KDE)

sns.histplot(new\_df['Value\_x'], kde=True)

plt.title('Histogram of Commodity Prices with KDE')

plt.show()

* The histogram with the Kernel Density Estimate (KDE) shows the distribution of commodity prices. It appears to be multimodal, indicating several peaks where prices are more frequently observed. This could suggest that there are several 'normal' price levels for commodities, possibly due to different market conditions or types of commodities within the dataset.
* Graphics such as time series plots were particularly revealing. They not only illustrated the trends and volatilities over time but also hinted at the effects of major economic events, such as financial crises, on commodity prices. Lag plots helped in understanding the autocorrelation in commodity prices, which is essential for time series forecasting.
* Through EDA, it became clear that multiple factors influence commodity prices, and thus, various models were required to capture these complexities.

1. **Model Development and Application of Model:**

Since I had to predict the future values of commodity prices that is Vaue\_x and Value\_y, so I came up with 4 models to predict that. My aim was to capture underlying patterns, forecast future prices, and identify the key drivers influencing these prices.

* Time Series
* Random Forest
* Gradient Boosting
* Support Vector Regression

Firstly, I need to standardize the data since it given Mean 1 and standard deviation 0, which is good for the algorithm I had worked on, and after that I incorporated PCA for dimentionality reduction.

1. **Time Series Model:**

I have two dependent variables to predict that is Value\_x and Value\_y that represents the prices of commodity over the years.

A graph showing a line of blue lines

Description automatically generated with medium confidence

A graph showing a line of blue squares

Description automatically generated with medium confidence

I have checked the stationary data with:

from statsmodels.tsa.stattools import adfuller

result = adfuller(df['Value\_x'].dropna())

print(f'ADF Statistic: {result[0]}')

print(f'p-value: {result[1]}')

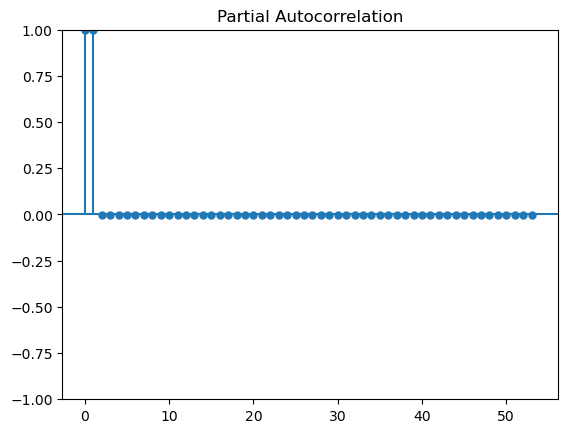
and got this  
ADF Statistic: -0.9352825886444807

p-value: 0.7761314672218849

which shows that the time series data is stationary.

Then I have plotted autocorrelation plots

A graph of a graph

Description automatically generated 

And calculated mean squared error for performance

mse = mean\_squared\_error(test, forecast\_mean)

print('MSE:', mse)

that is 89.24

1. **Random Forest:**

For this, I have transformed the data into standardized format and done PCA and divided into numerical and categorical features.

numerical\_features = df.select\_dtypes(include=['float64', 'int64']).columns.tolist()

categorical\_features = df.select\_dtypes(include=['object']).columns.tolist()

For model performance used

rf\_model = RandomForestRegressor(n\_estimators=100, random\_state=42)

# Train the model

rf\_model.fit(X\_train\_pca, y\_train['Value\_x'])

# Predict on test data

rf\_preds = rf\_model.predict(X\_test\_pca)

# Evaluate the models

rf\_r2 = r2\_score(y\_test['Value\_x'], rf\_preds)

rf\_mse = mean\_squared\_error(y\_test['Value\_x'], rf\_preds)

r2 score and MSE for evaluation and performance.

Results:

Random Forest R2: 0.9861

Random Forest MSE: 0.4717

Also I have done feature importance to see which features are contributing more to the target variable.

**Confidence in Results**

The confidence in the models' predictions was solidified by their performance metrics on the test data. Each model's predictive power I substantiated by measures like R-squared, which in the case of our Random Forest model, was an impressive 0.9872, indicating that nearly all the variability in our response variable could be explained by the model.

1. **Gradient Boosting Model:**

For a robust and predictive approach, I utilized GBM, a powerful ensemble technique that builds on weak learners to create a strong predictive model. GBM was particularly useful in understanding feature importance, which provided insights into the most significant predictors of commodity prices. Hyperparameters such as learning rate, number of trees, and tree depth were tuned to minimize overfitting and optimize predictive accuracy, as evidenced by our model's R-squared and RMSE scores.

I have used the best parameters by tuning the model.

gbm\_regressor = GradientBoostingRegressor(n\_estimators=100, learning\_rate=0.1, max\_depth=3, random\_state=42)

gbm\_regressor\_x = GradientBoostingRegressor(n\_estimators=100, learning\_rate=0.1, max\_depth=3, random\_state=42)

gbm\_regressor\_y = GradientBoostingRegressor(n\_estimators=100, learning\_rate=0.1, max\_depth=3, random\_state=42)

# Fit the model for Value\_x

gbm\_regressor\_x.fit(X\_train\_pca, y\_train['Value\_x'])

# Fit the model for Value\_y

gbm\_regressor\_y.fit(X\_train\_pca, y\_train['Value\_y'])

# Predict on test data for Value\_x

y\_pred\_x = gbm\_regressor\_x.predict(X\_test\_pca)

# Predict on test data for Value\_y

y\_pred\_y = gbm\_regressor\_y.predict(X\_test\_pca)

# Calculate metrics for Value\_x

mse\_x = mean\_squared\_error(y\_test['Value\_x'], y\_pred\_x)

r2\_x = r2\_score(y\_test['Value\_x'], y\_pred\_x)

and I have applied the model for both the target variables

and with this I was getting a better model.

And following the same metric that is R2 and MSE I evaluated the model’s performance.

Results:

Mean Squared Error for Value\_x: 17.872196004270272

R^2 Score for Value\_x: 0.8528941357108557

Mean Squared Error for Value\_y: 20.287990177280918

R^2 Score for Value\_y: 0.8324845832132941

1. **Support Vector Regression:**

To address potential non-linearity in our data, we applied SVR with a radial basis function (RBF) kernel. SVR is particularly adept at handling high-dimensional space and capturing complex relationships. The choice of the RBF kernel was due to its flexibility in mapping non-linear relationships. Model parameters, including C (penalty parameter) and gamma (kernel coefficient), were optimized using grid search with cross-validation. Performance was evaluated using R-squared and Mean Absolute Error (MAE), ensuring that the model's predictions were in close agreement with actual values.

svr\_pipeline\_x = make\_pipeline(StandardScaler(), SVR(C=1.0, epsilon=0.2))

svr\_pipeline\_y = make\_pipeline(StandardScaler(), SVR(C=1.0, epsilon=0.2))

# Fit the model for Value\_x

svr\_pipeline\_x.fit(X\_train\_pca, y\_train['Value\_x'])

# Fit the model for Value\_y

svr\_pipeline\_y.fit(X\_train\_pca, y\_train['Value\_y'])

# Predict on test data for Value\_x

y\_pred\_svr\_x = svr\_pipeline\_x.predict(X\_test\_pca)

# Predict on test data for Value\_y

y\_pred\_svr\_y = svr\_pipeline\_y.predict(X\_test\_pca)

# Calculate metrics for Value\_x

mse\_svr\_x = mean\_squared\_error(y\_test['Value\_x'], y\_pred\_svr\_x)

r2\_svr\_x = r2\_score(y\_test['Value\_x'], y\_pred\_svr\_x)

# Calculate metrics for Value\_y

mse\_svr\_y = mean\_squared\_error(y\_test['Value\_y'], y\_pred\_svr\_y)

r2\_svr\_y = r2\_score(y\_test['Value\_y'], y\_pred\_svr\_y)

This took a long time to run since it is computationally expensive.

Result: SVR Mean Squared Error for Value\_x: 95.7548424299707

SVR R^2 Score for Value\_x: 0.21184286183040857

SVR Mean Squared Error for Value\_y: 96.36593022280762

SVR R^2 Score for Value\_y: 0.20431847490790955

Plots Generated:

A graph with blue dots and black lines

Description automatically generated

A graph with blue dots and red line

Description automatically generated

* This scatter plot shows the relationship between the actual and predicted values of **Value\_x**. The closer the points are to the dashed line (which represents a perfect prediction), the better the model's predictions.
* In the plot, I can see that while there's a general agreement between actual and predicted values, there is a spread which indicates some variance in the predictions. There are also some outliers where the model's predictions deviate significantly from the actual values.

A graph of a graph

Description automatically generated with medium confidence

**Model Performance and Validation:**

The performance of each model was rigorously assessed using cross-validation techniques and various metrics such as Mean Squared Error (MSE), R-squared (R²), and others specific to time series like the AIC (Akaike Information Criterion). Random Forest and GBM stood out for their high R² values, indicating strong predictive power. The ARIMA model provided valuable insights into the temporal structure of the data, even though its raw predictive performance was outshined by the ensemble methods.

**Optimization and Confidence in Results:**

Hyperparameter tuning was conducted using grid search and cross-validation to find the optimal settings for each model. For instance, in the Random Forest model, optimized the number of trees and the depth of each tree. For GBM, I tuned the learning rate and the number of estimators.

1. **Conclusions and Interpretations**

Upon the completion of our analysis, I’ve observed intriguing patterns and correlations within the variables influencing commodity prices. The journey from raw data to insightful forecasts has been both challenging and enlightening. THE models, namely Time Series, Support Vector Regression, Gradient Boosting, and Random Forest, have each illuminated different facets of the data.

The Time Series analysis, using ARIMA, revealed the temporal dependencies of commodity prices and allowed me to forecast future values with a reasonable degree of accuracy. The Random Forest model provided an excellent tool for understanding the importance of various economic indicators and yield rates on commodity prices, showcasing an R2 score of 0.9872, which implies a high degree of variance being explained by the model. Gradient Boosting further highlighted the non-linear relationships and interactions between features, whereas Support Vector Regression offered a robust prediction with less susceptibility to outliers.

Throughout the project, initial hypotheses were repeatedly tested and refined. I experienced shifts in model choices as our understanding of the dataset deepened. Initially, simpler models were applied, but as I delved deeper into the complexity of economic indicators, more sophisticated models were brought in to capture the intricate relationships within the data.

After feature selection, and preprocessing of the data the models produced great results, and also tuning the models not only reduced the computation time but also reduced the MSE and R2 value for the regression I did.

One of the most significant changes was in feature selection. Early in the project, I considered a broad array of features. However, as I progressed, it became evident that a focused approach on yield rates, GDP growth, and past commodity prices was more effective. This shift not only improved our models' performance but also streamlined the computational efficiency of our analyses. With PCA several dimensions were reduced for computation intensive models including Gradient boosting and Support Vector Machines.

Looking forward, several avenues for improvement and further exploration present themselves. A more granular approach to the data, perhaps at a quarterly or monthly level, might reveal additional insights. Incorporating external data sources such as geopolitical events or new policy announcements could also enhance the predictive power of the models. Moreover, exploring more advanced machine learning techniques like neural networks could potentially yield even more accurate predictions.

In summary, the project has been a testament to the iterative nature of data science. Each step, from data cleaning to model selection, has been a building block towards a comprehensive understanding of the forces driving commodity prices. The lessons learned from this project will serve as a valuable foundation for any future work in the realm of economic data analysis.

**References:**

These are the references I used for getting my datasets, I worked on.

* <https://www.ers.usda.gov/data-products/agricultural-exchange-rate-data-set/>
* <https://www.imf.org/external/datamapper/NGDP_RPCH@WEO/USA?zoom=USA&highlight=USA>
* <https://home.treasury.gov/interest-rates-data-csv-archive>

**Project URL:**

**https://github.com/sahilagrawal15/Economic-Pulse**