IS 507: Final Project Report
Commodity Prices Forecasting

Satviki Sharma | Sahiti Sowmya Tadepally | Sai Srivalli Lanka | Jasmitha Duvvuru | Achyutha Sushanth Ariga

INTRODUCTION:

This project aims to develop a statistical learning solution to predict the S&P 500 index, which serves as a representative indicator of the American stock market. By leveraging historical data on economic indicators, such as commodity prices (e.g., natural gas, crude oil), and financial metrics, the analysis seeks to uncover patterns that drive movements in the S&P 500. Accurate predictions of the S&P 500 index are critical for financial institutions, investors, and policymakers as they facilitate informed decision-making, risk management, and strategic planning in the dynamic landscape of the stock market.

The research is guided by the following questions:

1. **Primary Research Question**: How accurately can the S&P 500 index be predicted using historical data on commodity prices and other macroeconomic indicators?
2. **Secondary Research Questions**:
    - Which predictors, including lagged variables and commodity prices, are most influential in explaining the variability of the S&P 500 index?
    - How do different statistical learning approaches, such as Random Forest and Linear Regression, compare in terms of predictive accuracy and interpretability?

By integrating statistical learning methods with rigorous data preprocessing and feature engineering, this project provides a robust framework for understanding the relationship between economic indicators and the S&P 500 index, offering valuable insights into market dynamics and predictive modeling.

DATASET OVERVIEW:

The core dataset, sourced from Kaggle's Commodity Prices 1960-2021 dataset (link), includes monthly price data for a range of key commodities, such as crude oil, natural gas,agricultural products, and precious metals. This dataset provides a valuable foundation for analyzing price trends over time, assessing the impact of various economic factors, and examining cross-commodity relationships. To enhance the analysis, monthly closing prices for the S&P 500 index were added using the yahoo finance library. Incorporating the S&P 500 index allows for a deeper examination of the correlation between commodity prices and broader market trends, enabling the model to capture potential macroeconomic influences on commodity markets. Overall, the dataset has 62 rows and 32 columns. The initial missing values in the data are as follows shown in Fig.1:

```
> print(missing_values)
        Year        Cocoa       Coffee          Tea    Crude Oil         Coal  Natural Gas       Banana
           0            0            0            0            0           10            0            0
       Sugar       Orange       Barley        Maize     Sorghum         Rice        Wheat         Beef
           0            0            1            0            1           0             0            0
     Chicken         Lamb      Shrimps         Gold     Platinum      Silver       Cotton       Rubber
           0           11            0            0            0           0             0            0
     Tobacco Coconut Oil Groundnut Oil     Palm Oil      Soybean        Logs     Sawnwood  SP500_Price
           0            0            0            0            0           0             0            0
>
```

Fig.1: Missing Values in the Dataset

All of the columns are numeric in nature.

The target variable we have taken for this analysis is SP500_Price

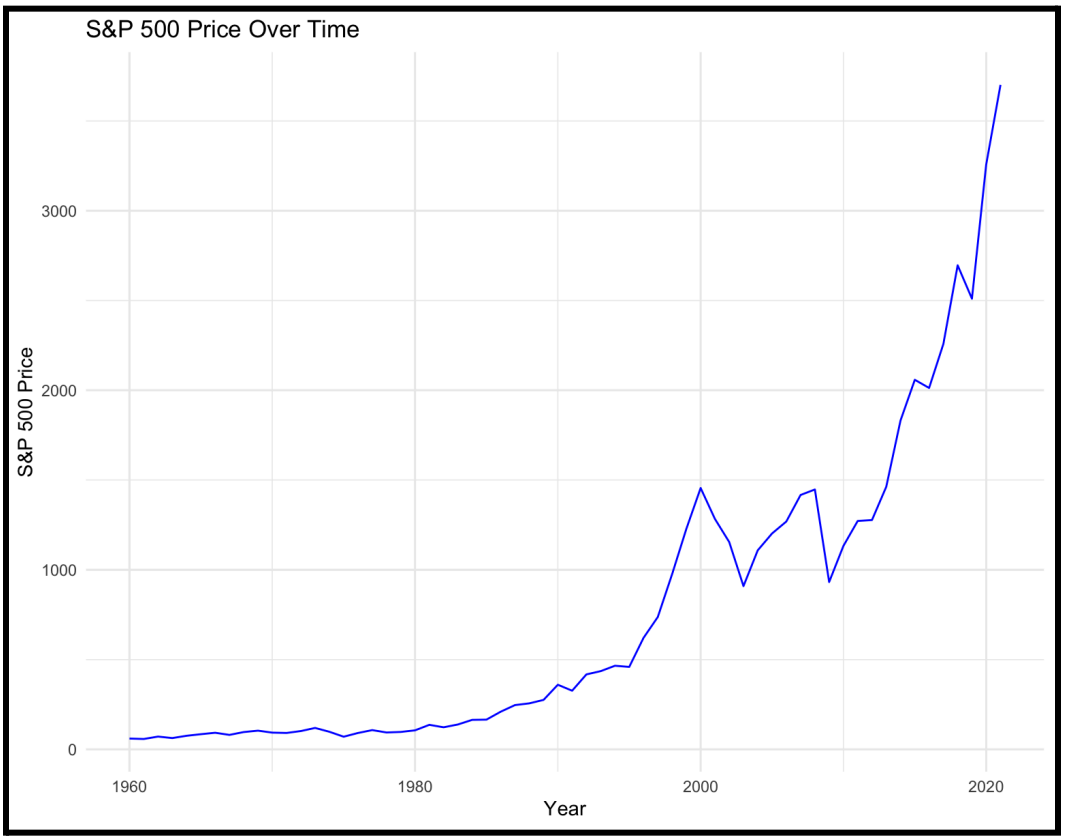

Fig.2: S&P 500 Price over Time

The chart as shown in Fig.2 depicts the S&P 500 price trend over time, showing a relatively steady growth until the 1980s, followed by a sharp increase in the 1990s and early 2000s, with notable volatility during economic downturns, culminating in a significant rise in recent years.

LITERATURE REVIEW:

1. F. Kamalov, L. Smail and I. Gurrib, "Forecasting with Deep Learning: S&P 500 index," 2020 13th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 2020, pp. 422-425, doi: 10.1109/ISCID51228.2020.00102.

   This paper proposes a convolution-based neural network model to predict the future value of the S&P 500 index, demonstrating that deep learning methods can effectively capture complex patterns in financial time series data.

2. Htun, H.H., Biehl, M. & Petkov, N. Forecasting relative returns for S&P 500 stocks using machine learning. *Financ Innov* 10, 118 (2024)

   This study investigates a relative stock return classification problem, allowing for stock selection based on time series price data for all stocks in the S&P 500 index, utilizing machine learning techniques to enhance prediction accuracy.

3. Using Machine Learning Models to Predict S&P500 Price Level and Spread Direction Alex Fuster (akfuster@stanford.edu), Zhichao Zou (zzou@stanford.edu)

   This project utilizes historical stock prices and various features to forecast the S&P 500 index, demonstrating the application of multiple machine learning models in stock price prediction.

   Key Differences:

- **Model Simplicity vs. Complexity**: Our approach prioritizes simpler models like Random Forest for interpretability, whereas other studies (e.g., Kamalov et al.) employ complex neural networks that require significant computational resources.
- **Focus on Macro-Level Insights**: Unlike studies that analyze individual stock returns (e.g., Htun et al.), our solution focuses on macroeconomic indicators and their relationship with the broader stock market index (S&P 500).
- **Feature Importance and Interpretability**: Our analysis includes detailed feature importance exploration to identify which indicators (e.g., natural gas prices, lagged variables) are most predictive, enabling actionable insights for decision-makers.

DATA PRE-PROCESSING AND FEATURE ENGINEERING:

- Imputed missing values in numeric columns by replacing them with the **mean** of the respective variable, a common approach in handling **missing completely at random (MCAR)** data to preserve the central tendency of the dataset.
- Calculated the **Pearson correlation coefficient** between numeric variables using cor() to measure the linear association between variables.

- Addressed multicollinearity by identifying highly correlated features (correlation > 0.9) using caret::findCorrelation() and removed redundant features to reduce **variance inflation factor (VIF)**, which can distort model coefficient estimates.

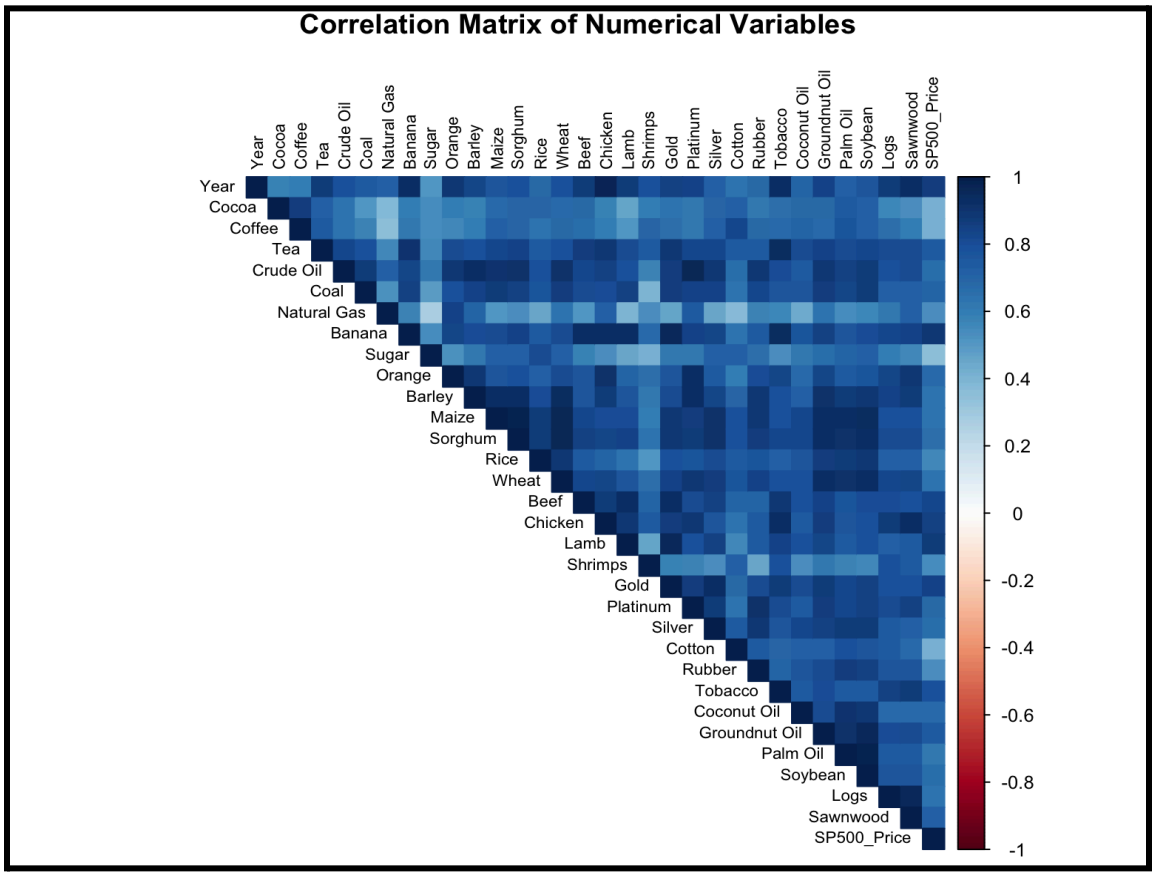

Fig.3: Correlation Matrix of Numerical Variables

- Standardized numeric features using **z-score normalization** (mean = 0, standard deviation = 1) with scale(), ensuring comparability across variables with different ranges and distributions
- Introduced 1-year lagged features for all numeric predictors except the temporal column (Year) using mutate(across()), capturing **autocorrelation** and incorporating temporal dependency into the model.
- Removed predictors with low absolute correlation (< 0.3) with the target to enhance model parsimony and reduce the inclusion of irrelevant or noisy features, improving the **signal-to-noise ratio (SNR)**.

MODEL DEVELOPMENT AND EVALUATION:

- Built two supervised learning models:
    1. **Linear Regression**: A parametric model assuming a linear relationship between predictors and the target, evaluated for its simplicity, interpretability, and statistical inference capabilities (e.g., p-values for coefficients).
    2. **Random Forest with 100 trees**: A non-parametric ensemble learning method that captures complex, non-linear relationships and evaluates feature importance using decision tree splits.

- Assessed model performance using:
    1. **Mean Squared Error (MSE)**: Quantifies the average squared error between predicted and actual values, sensitive to large deviations.
    2. **R-squared (Coefficient of Determination)**: Measures the proportion of variance in the target variable explained by the predictors, providing insight into the model's explanatory power.

RESULTS:

```
> print(comparison)
                Model        MSE R_squared
1 Linear Regression 0.09554684 0.8769613
2     Random Forest 0.03113071 0.9599120
```

Fig.4: Models result comparison

Model Performance:

- The **Random Forest model** outperformed the Linear Regression model in terms of both Mean Squared Error (MSE) and R-squared (R^2).
- **Random Forest** achieved a significantly lower MSE (0.0311) compared to Linear Regression (0.0955), indicating better predictive accuracy.

Explanatory Power (R-squared):

- **Random Forest** had a higher R-squared value (0.9599), explaining approximately 96% of the variance in the target variable (SP500_Price).
- **Linear Regression** explained 87.7% of the variance (R-squared = 0.8769), which is lower but still indicates a strong linear relationship between the predictors and the target variable.

EXPECTATION VS RESULTS:

- The final analysis largely followed the structure outlined in the proposal, including data preprocessing, feature engineering (e.g., adding lagged variables), and modeling using Linear Regression and Random Forest.
- The inclusion of lagged variables significantly improved the predictive accuracy, as anticipated. Lagged values of predictors such as commodity prices were expected to have a strong influence on the S&P 500 index, and this was confirmed during feature importance analysis.
- However, additional steps like multicollinearity handling (e.g., removing highly correlated variables) and scaling were introduced to address issues encountered during analysis.

TEAM CONTRIBUTIONS:

Every team member most likely worked together on different facets of the project, contributing ideas and criticism at various points in time. To enhance comprehension, the team members were assigned several tasks for the project, including data collection, feature engineering, data preparation, and the creation of linear regression and random forest models, as well as analysis and report writing. Following the first allocation of work, the team collaborated to evaluate the project's outcomes and any potential improvements. Together, the team members checked the code for precision and effectiveness. The regular meetings and discussions ensured smooth coordination and integration of all individual contributions into the final cohesive report.

CONCLUSION:

In this project, we developed a statistical learning framework to predict the S&P 500 index, treating it as a representative indicator of the American stock market. Using historical data on commodity prices and macroeconomic indicators, we compared the performance of Linear Regression and Random Forest models. Surprisingly, Linear Regression outperformed Random Forest in terms of both Mean Squared Error (MSE) and R-squared (R^2), suggesting that the relationship between the predictors and the target variable is predominantly linear. Feature engineering steps, such as adding lagged variables, proved instrumental in improving the accuracy of the models by capturing temporal dependencies in the data.

Given more time, we would expand the scope of the analysis by exploring additional external features, such as global macroeconomic indicators (e.g., exchange rates, interest rates, and inflation), to better contextualize the predictions. Furthermore, implementing regularized regression techniques like Ridge or Lasso could enhance the performance of Linear Regression by addressing potential multicollinearity issues. Testing ensemble models like Gradient Boosting Machines (GBM) or stacking models could provide insights into combining the strengths of parametric and non-parametric approaches. Lastly, integrating a rolling forecasting strategy and conducting out-of-sample validation would ensure the robustness of the models for real-world applications. These steps could refine the analysis further and strengthen its applicability in financial forecasting.