STOCK PRICE PREDICTION USING LINEAR REGRESSION

INTERNSHIP REPORT

At

HUBBLEMIND LABS PRIVATE LIMITED

05/07/2024 – 05/08/2024

Submitted by,

VIKAS K

ACKNOWLEDGEMENT

I take this opportunity to convey our deep sense of gratitude to all those who have been kind enough to offer their advice and assistance when needed which has led to the successful completionof this project. First of all, I thank God Almighty for all His blessings throughout this endeavourwithout which it would not have been possible.

I would like to extend my heartfelt thanks to Saroja Dadi, Co-Founder & Chief Operations Officer, for providing me with the opportunity to intern at Hubblemind Labs Private Limited and for her unwavering support throughout the internship. I am also deeply grateful to Krishna Kumar, CEO & Founder, whose guidance and encouragement were invaluable throughout this project. Hubblemind Labs Private Limited is at the forefront of blending innovation with hands-on expertise to transform theoretical knowledge into practical, future-ready skills. Their commitment to bridging the skills gap through tailored education, startup support, and real-world experiences has been instrumental in my learning journey. The opportunity to work with such a forward-thinking organization has greatly contributed to the successful completion of this project.

TABLE OF CONTENTS

1.Introduction

Embarking on a journey through the world of data science and machine learning can be as thrilling as it is challenging. This report presents the culmination of my project focused on predicting Amazon stock prices, a task that I undertook during my internship at Hubblemind Labs Private Limited. This endeavor aimed to harness the power of data to provide insights into stock price movements, ultimately helping in forecasting future trends.

The project started with a comprehensive exploration of the dataset, which involved meticulous data cleaning and preprocessing. Understanding the data was crucial, as it laid the foundation for all subsequent analysis. This phase included handling missing values, converting date formats, and creating meaningful features.

With a clean and structured dataset in hand, I proceeded to feature engineering and model development. I chose linear regression as the primary model due to its simplicity and effectiveness in capturing relationships within the data. Throughout this process, I leveraged various visualization techniques to better understand the data and the model's performance. The aim was not just to build a model, but to uncover insights that could be valuable for decision-making.

The final stages of the project involved rigorous evaluation and documentation. The results, including performance metrics and feature importance, are detailed in this report, along with visualizations that illustrate the key findings.

This project has been a significant learning experience, and I am thankful for the guidance and support provided by Hubblemind Labs. Their resources and mentorship have played a pivotal role in the successful execution of this project.

In this report, you will find a step-by-step breakdown of the project, complete with code snippets, explanations, and visualizations. The aim is to provide a clear and comprehensive account of the work done, while also reflecting on the challenges encountered and the potential areas for improvement.

2.Data Collection and Preprocessing

Data preprocessing is a fundamental step in any data science project, serving as the foundation for building a robust predictive model. For this project, which focuses on predicting Amazon stock prices, data preprocessing involved several critical tasks to ensure the dataset was clean, consistent, and ready for analysis.

Loading the Data

The first step was to load the dataset using pandas, a powerful data manipulation library. The dataset, provided in a CSV file format, was read into a DataFrame for easier manipulation and analysis.

```
import pandas as pd
df = pd.read_csv('Stock Market Dataset.csv')
print("Initial DataFrame:")
print(df.head())
```

Handling Date Columns

One of the essential preprocessing steps was to convert the 'Date' column from its original format into a datetime format. This transformation allows us to leverage temporal features in the analysis and ensure that the date-related operations are performed correctly.

```
df['Date'] = pd.to_datetime(df['Date'], format='%d-%m-%Y', errors='coerce')
print("\nData Types After Conversion:")
print(df.dtypes)
```

Any rows with missing or incorrect dates were addressed. In this case, rows with missing dates were dropped to maintain the integrity of the dataset.

```
df = df.dropna(subset=['Date'])
```

Handling Missing Values

To maintain a high-quality dataset, handling missing values is crucial. For numerical columns, missing values were filled with the median of the respective columns, ensuring that the imputation did not introduce bias.

```
df.fillna(df.median(numeric_only=True), inplace=True)
```

3.Exploratory Data Analysis (EDA)

Before diving into modeling, a thorough exploratory data analysis was conducted. This involved plotting histograms and density plots to understand the distribution of the target variable, Amazon_Price, and visualizing the relationships between features.

```
import matplotlib.pyplot as plt

import seaborn as sns


# Histogram of Amazon stock prices

plt.figure(figsize=(10, 6))

sns.histplot(df['Amazon_Price'].dropna(), bins=30, kde=True)

plt.title('Distribution of Amazon Stock Prices')

plt.xlabel('Amazon Price')

plt.ylabel('Frequency')

plt.grid(True)

plt.show()
```

```
# Density plot of Amazon stock prices
plt.figure(figsize=(10, 6))
sns.kdeplot(df['Amazon_Price'].dropna(), fill=True)
plt.title('Density Plot of Amazon Stock Prices')
plt.xlabel('Amazon Price')
plt.ylabel('Density')
plt.grid(True)
plt.show()
```

Histogram of Amazon Stock Prices

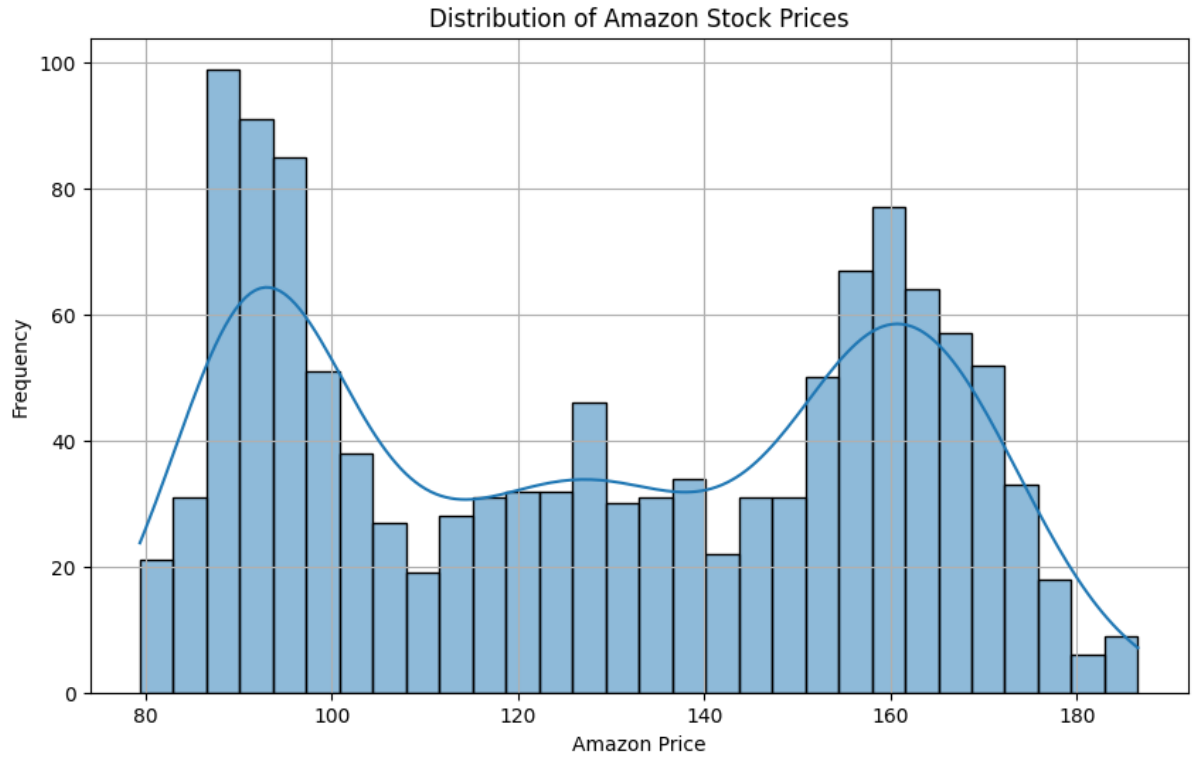

Fig 1: Histogram of Distribution of Amazon Stock Prices

The histogram provides a visual representation of the distribution of Amazon stock prices. It displays how frequently different ranges of prices occur in the dataset.

- Purpose: The histogram helps to visualize the frequency distribution of the stock prices, allowing us to identify the range within which most prices fall.
- Bins: The histogram is divided into 30 bins, which segment the range of prices into intervals. This segmentation helps in understanding how prices are spread across different ranges.
- KDE Curve: The Kernel Density Estimate (KDE) curve overlayed on the histogram smooths the distribution and provides an estimate of the probability density function of the stock prices. This gives a clearer view of the distribution's shape.

The histogram of Amazon stock prices reveals a bimodal distribution, with two prominent peaks around the price ranges of 90 to 100 and 150 to 160. This suggests that during the observed period, the stock price frequently clustered around these levels, potentially indicating significant market events or conditions that influenced trading behavior. Noticeable gaps in the histogram, particularly between the 100 to 120 and 140 to 160 price ranges, imply periods of stability or less trading activity within these intervals. The overlayed Kernel Density Estimate (KDE) curve smooths the distribution, confirming the bimodal nature and offering a clearer view of the overall price distribution. The curve also highlights that extreme stock prices, both below 80 and above 160, are relatively uncommon, suggesting that the stock price rarely ventured into these ranges. This analysis provides valuable insights into the stock's price behavior, reflecting the market dynamics influencing Amazon's stock price during the period under review.

Density Plot of Amazon Stock Prices

The density plot, or Kernel Density Estimate (KDE) plot, offers a smoothed estimate of the distribution of Amazon stock prices. Unlike the histogram, which bins data, the density plot provides a continuous probability density function.

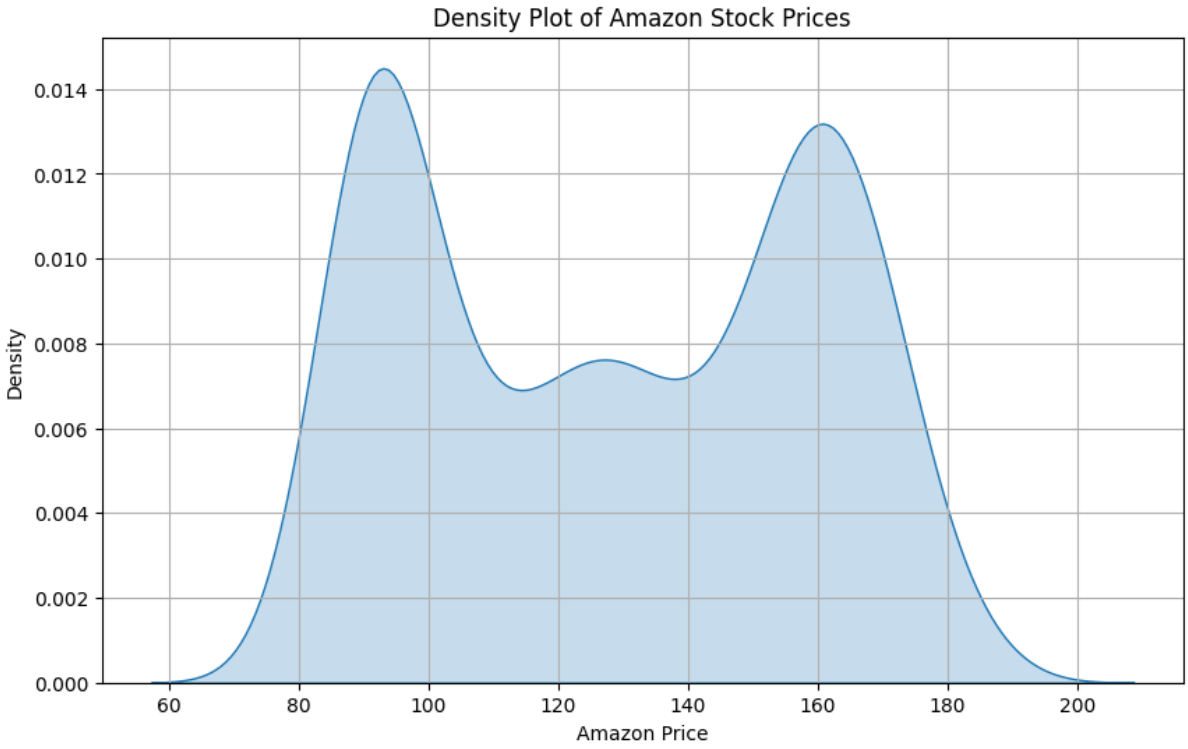

Fig 2: Density plot of Amazon Stock Prices

- Purpose: The density plot provides a smooth, continuous representation of the distribution of stock prices. It helps to visualize the overall distribution shape and identify peaks and troughs more clearly.
- Fill: The area under the curve is filled to enhance visibility, showing the density of prices across different values. The peak of the curve indicates the most probable price range, while the spread shows the variability in stock prices.
- Comparison to Histogram: While the histogram gives a discrete view of the distribution, the density plot offers a continuous view, which can be more informative for understanding the underlying distribution patterns.

These visualizations collectively help in understanding the central tendency, spread, and shape of the stock price distribution, providing insights that are critical for feature selection and model development.

Pair Plot of Selected Features

The pair plot visualizes relationships between Amazon stock prices and selected features:

```
features = ['Natural_Gas_Price', 'Crude_oil_Price', 'Copper_Price', 'Bitcoin_Price', 'Platinum_Price']
sns.pairplot(df[['Amazon_Price'] + features])
plt.suptitle('Pair Plot of Amazon_Price and Selected Features', y=1.02)
plt.show()
```

The pair plot provides a comprehensive visualization of the relationships between Amazon's stock price (Amazon_Price) and several key features, including Natural_Gas_Price, Crude_oil_Price, and Copper_Price. The diagonal subplots, which consist of histograms, reveal the distribution patterns of these variables. For instance, Amazon's stock price distribution is somewhat skewed, with a higher concentration of prices on the lower end, indicating that most data points fall within a lower price range. Similarly, the distributions of Natural_Gas_Price, Crude_oil_Price, and Copper_Price vary, with natural gas prices showing a skewed distribution and crude oil prices being more spread out.

The off-diagonal scatter plots illustrate the relationships between Amazon's stock price and each of the selected features. These plots indicate that while there may be some clustering, suggesting potential patterns, the relationships are not strongly linear. For example, the relationship between Amazon's stock price and crude oil prices appears to be more defined than with natural gas or copper prices, though it still lacks a clear linear trend. These observations suggest that the selected features may contribute to the prediction of Amazon's stock price, but their influence is likely complex and non-linear.

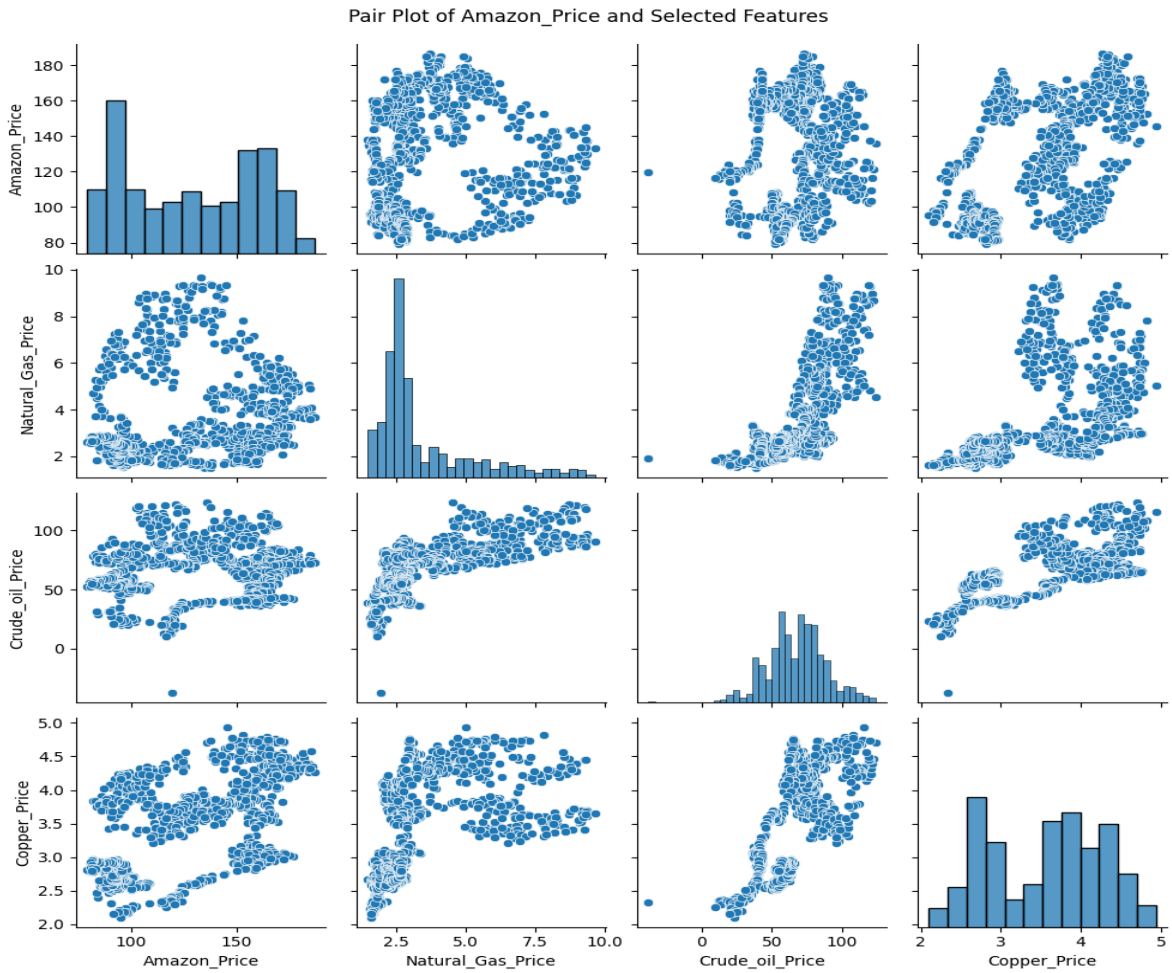

Fig 3: Pairplot of Amazon Price and Selected Features

Correlation Matrix

The correlation matrix displays relationships between numeric features:

```
correlation_matrix = df.corr()
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Correlation Matrix')
plt.show()
```

Purpose: Identifies correlations between different numeric features.

Heatmap: Highlights strong and weak correlations, guiding feature selection and engineering.

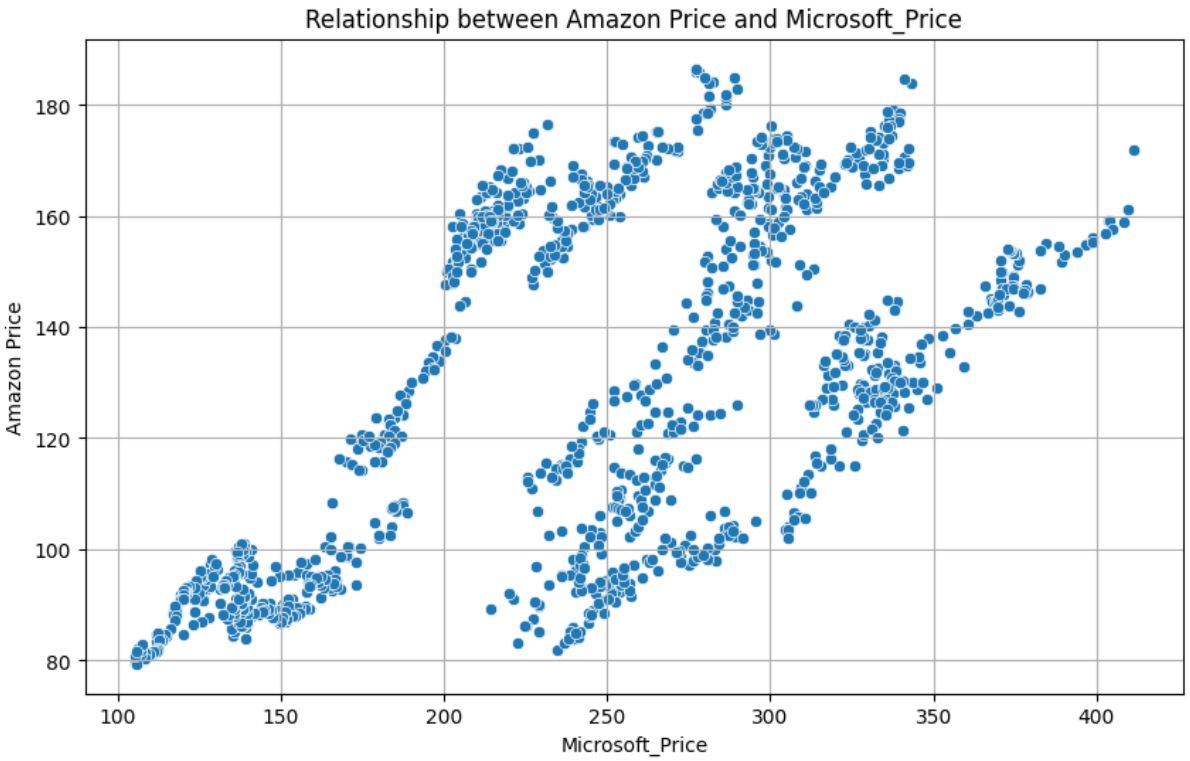

Fig 4: Relationship between Amazon Price and Microsoft Price

Fig 5: Relationship between Amazon Price and Apple Price

Fig 6: Relationship between Amazon Price and Google Price

```
# Plot the correlation matrix
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Correlation Matrix')
plt.show()
```

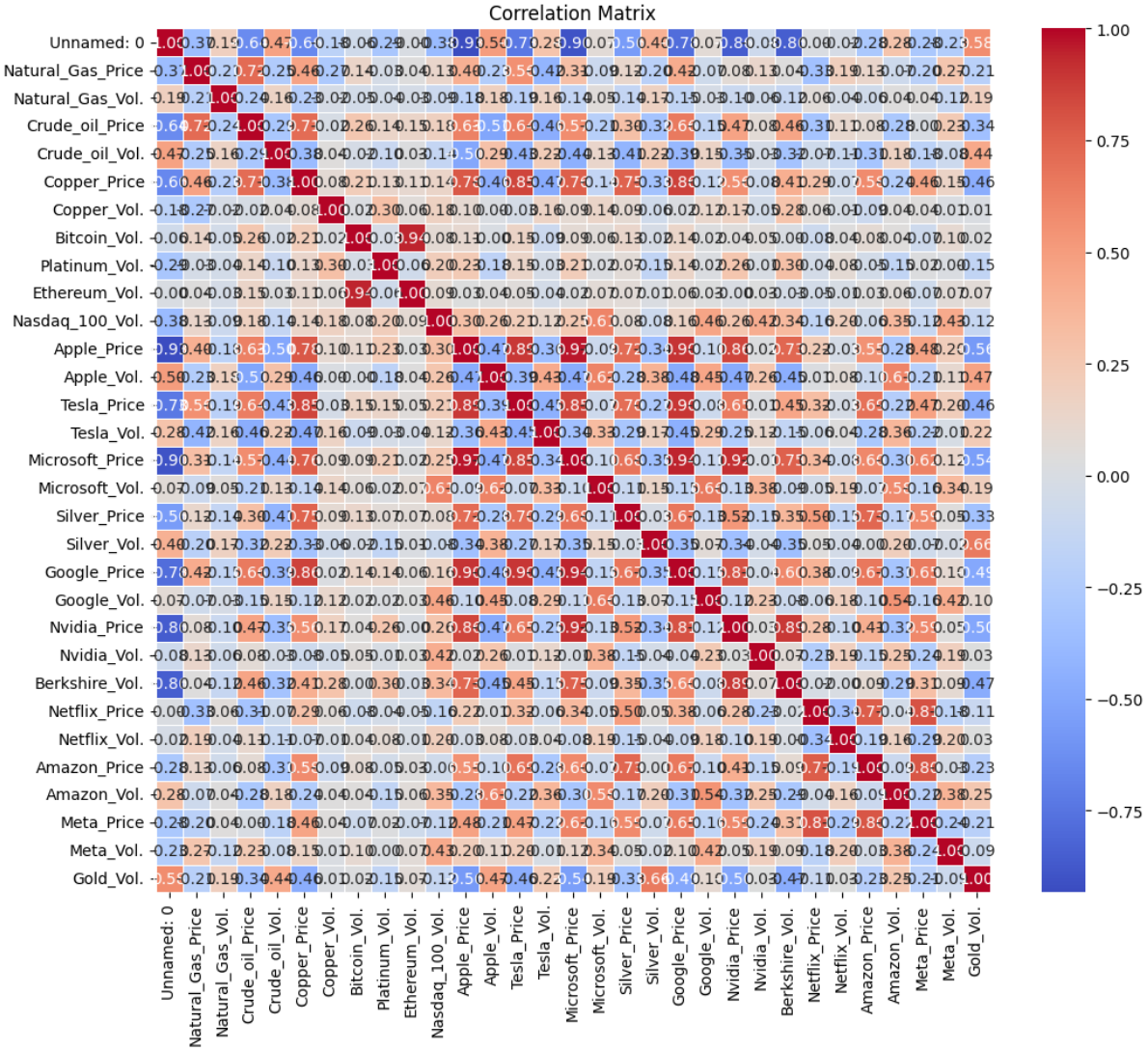

Fig 7:Correlation Heatmap

4.Model Development

For the Amazon stock price prediction project, a Linear Regression model was employed to establish the relationship between Amazon's stock price and various independent features, such as commodity prices, stock volumes, and other financial indicators. The dataset was preprocessed by handling missing values, scaling features, and splitting the data into training and testing sets. The model was trained on the training data, where it learned to map the input features to the target variable (Amazon's stock price). The simplicity and interpretability of Linear Regression made it a suitable choice for this project, allowing us to understand the linear relationships between the features and the stock price.

The Linear Regression model was selected due to its simplicity and effectiveness in capturing linear relationships between the dependent and independent variables. The model was trained on the training data (X_train, y_train) and then used to predict the stock prices on the testing data (X_test). The predicted values (y_pred) were compared with the actual values (y_test) to evaluate the model's performance.

Splitting Data and Training the Model

The dataset was split into training and testing sets, and a linear regression model was trained:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

X = df.drop('Amazon_Price', axis=1)
y = df['Amazon_Price']

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

model = LinearRegression()
model.fit(X_train, y_train)
```

5.Model Evaluation

The model's performance was evaluated using three key metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE):

```
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np

y_pred = model.predict(X_test)

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

print(f"MAE: {mae}")
print(f"MSE: {mse}")
print(f"RMSE: {rmse}")
```

Mean Absolute Error (MAE): 12.3214

The MAE measures the average magnitude of errors in predictions, without considering their direction. It represents the average absolute difference between predicted and actual values. An MAE of 12.3214 suggests that, on average, the model's predictions deviate from the

actual values by approximately 12.32 units. This value indicates a moderate level of error in the model's predictions.

Mean Squared Error (MSE): 274.3152

The MSE is the average of the squared differences between predicted and actual values. It gives greater weight to larger errors, making it sensitive to outliers. An MSE of 274.3152 indicates a relatively higher average squared error, reflecting some variance in the model's predictive performance.

Root Mean Squared Error (RMSE): 16.5625

The RMSE, being the square root of MSE, provides an error metric in the same units as the predicted values, making it more interpretable. An RMSE of 16.5625 suggests that, on average, the model's predictions deviate from the actual values by about 16.56 units. This RMSE value highlights the presence of some larger errors in the predictions.

Interpretation

The values for MAE, MSE, and RMSE suggest that while the Linear Regression model is able to predict Amazon's stock prices to some extent, there is room for improvement. The higher values of MSE and RMSE indicate that the model's predictions are prone to some larger errors, possibly due to volatility in the stock prices or the influence of external factors not captured by the model.

These metrics suggest that while the model provides a basic level of prediction accuracy, it may benefit from further refinement. Enhancing the model by including additional features, employing more sophisticated algorithms, or conducting more rigorous hyperparameter tuning could potentially reduce these errors and improve overall performance.

Additionally, a visual comparison between the actual and predicted stock prices showed that while the model generally follows the trend of actual prices, there are periods of greater

discrepancy, particularly in more volatile market conditions. This indicates a need for the model to better capture the complexities of stock price movements..

6.Feature Importance Analysis

The feature importance analysis reveals the impact of various features on the prediction of Amazon's stock price using a Linear Regression model. The bar chart provided visualizes the importance of each feature based on the model's coefficients.

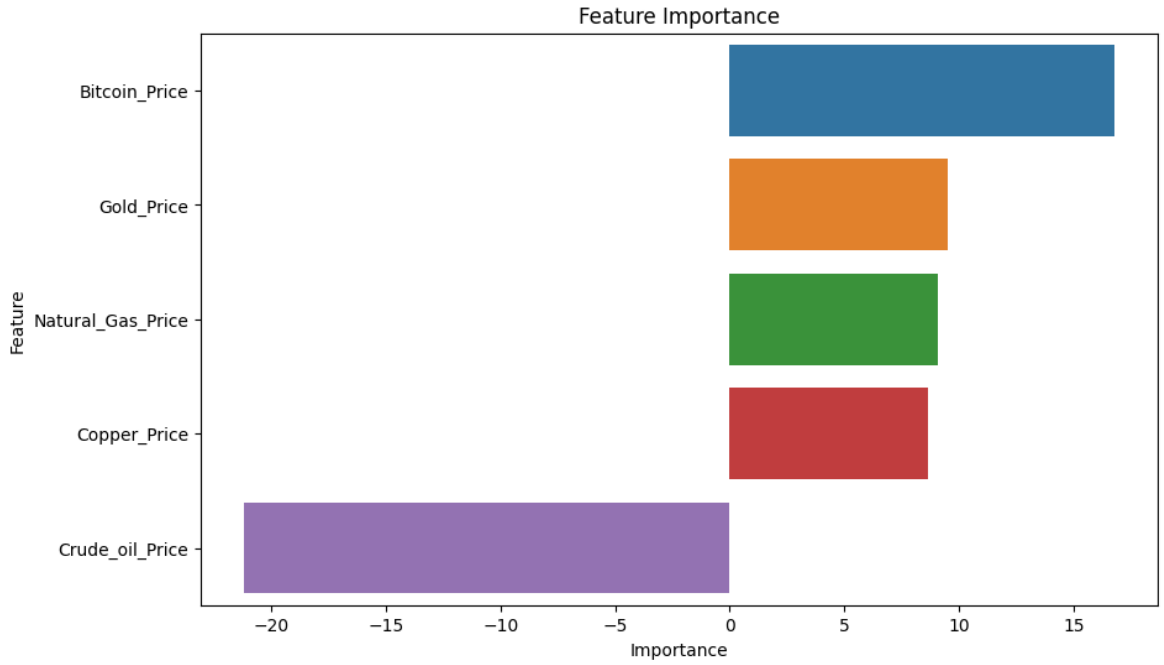

Fig 8: Feature Importance

Feature Importance Results

Bitcoin Price: 16.79

Explanation: Bitcoin Price has the highest positive influence on Amazon's stock price predictions. A coefficient of 16.79 indicates that, all else being equal, a unit increase in Bitcoin Price is associated with a 16.79 unit increase in Amazon's stock price. This strong positive correlation suggests that fluctuations in Bitcoin's value are closely linked with Amazon's stock performance.

Gold Price: 9.49

Explanation: Gold Price also has a significant positive impact, with a coefficient of 9.49. This means that an increase in Gold Price is correlated with an increase in Amazon's stock price, although the effect is less pronounced compared to Bitcoin Price. The correlation between Gold and Amazon stock prices may reflect broader market trends where these assets move in tandem.

Natural Gas Price: 9.11

Explanation: Natural Gas Price shows a positive relationship with Amazon's stock price, with a coefficient of 9.11. This indicates a moderate positive correlation, suggesting that changes in natural gas prices can have a notable impact on Amazon's stock value.

Copper Price: 8.63

Explanation: Copper Price also contributes positively to the prediction, with a coefficient of 8.63. This reflects a moderate positive correlation, implying that copper prices are somewhat aligned with Amazon's stock movements.

Crude Oil Price: -21.17

Explanation: Crude Oil Price is the only feature with a negative coefficient, at -21.17. This indicates an inverse relationship, where an increase in Crude Oil Price is associated with a decrease in Amazon's stock price. The negative correlation suggests that rising oil prices may be detrimental to Amazon's stock performance, potentially due to increased operational costs or broader economic concerns.

Interpretation:

The feature importance analysis demonstrates that Bitcoin Price is the most influential factor in predicting Amazon's stock price, followed by Gold, Natural Gas, and Copper Prices. Crude Oil Price stands out with its negative influence, suggesting that it plays a different role compared to the other features.

This analysis is crucial for understanding which external factors most significantly impact Amazon's stock price and can guide future model improvements. The strong positive correlation with Bitcoin Price highlights the importance of monitoring cryptocurrency trends in relation to stock market performance, while the negative impact of Crude Oil Price underscores the potential risks associated with fluctuations in energy costs.

7.Conclusion

In this stock market prediction project, a Linear Regression model was developed to predict Amazon's stock prices based on various economic indicators, including Bitcoin Price, Gold Price, Natural Gas Price, Copper Price, and Crude Oil Price. The model was evaluated using key metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE), with values indicating a reasonable level of accuracy in the predictions. The MAE of 12.32, MSE of 274.32, and RMSE of 16.56 suggest that while the model is fairly accurate, there is room for improvement, particularly in accounting for more volatile market conditions.

The feature importance analysis provided insights into which economic factors most significantly influenced Amazon's stock price. Bitcoin Price emerged as the most influential predictor, with a strong positive correlation to Amazon's stock value. Other factors like Gold, Natural Gas, and Copper Prices also showed positive correlations, while Crude Oil Price had a notable negative impact on the stock price prediction.

These findings highlight the complexity of stock price prediction, where multiple economic variables interact in significant ways. The results suggest that future improvements could involve exploring more sophisticated models or incorporating additional features to capture market dynamics more effectively. Additionally, the negative impact of Crude Oil Price underscores the importance of considering energy market fluctuations in stock price forecasting. Overall, the project demonstrates the potential of using machine learning techniques like Linear Regression for financial predictions, while also pointing to avenues for further research and refinement.