# Multiple Linear Regression (MLR) on Ethereum Prices Dataset

## Predictive Modeling of Ethereum Prices Using Historical Market Data

**Executive Summary**

Gabriela Howell

Master of Science Data Analytics, Western Governors University

Data Analytics Graduate Capstone

December 3, 2024

# Multiple Linear Regression (MLR) on Ethereum Prices Dataset

## Predictive Modeling of Ethereum Prices Using Historical Market Data

## Executive Summary

### Research Problem and Hypothesis

The study investigates whether a predictive multiple linear regression (MLR) model can forecast Ethereum prices using historical price and volume data.

- Null Hypothesis: Historical Ethereum data does not significantly influence current Ethereum prices, resulting in prediction accuracy below 70%.
- Alternate Hypothesis: Historical Ethereum data significantly influences current Ethereum prices, enabling a predictive model with at least 70% accuracy.

Cryptocurrency markets are highly volatile, and understanding key drivers of price movements is essential for financial forecasting. Historical data, such as prices and trading volumes, are critical for predictive models (McNally, Roche, & Caton, n.d.).

### Data Analysis Process

Historical Ethereum pricing data from Coinbase, available on Kaggle, includes over 4.1 million rows across six columns, covering data from 2017 to December 2024 (Bukhari, 2024). The dataset's high quality, with 0.00% sparsity, makes it ideal for robust modeling. With the following variables included:

| Attribute | Data Type | Description |
|---|---|---|
| Timestamp | Quantitative (Discrete) | The time the price data is recorded, usually shown as a Unix timestamp. |
| Open | Quantitative (Continuous) | The price of Ethereum at the start of the period. |
| High | Quantitative (Continuous) | The highest price during the period. |
| Low | Quantitative (Continuous) | The lowest price during the period. |
| Close | Quantitative (Continuous) | The price of Ethereum at the end of the period. |
| Volume | Quantitative (Discrete) | The amount of Ethereum traded during the period. |

Data Preprocessing:

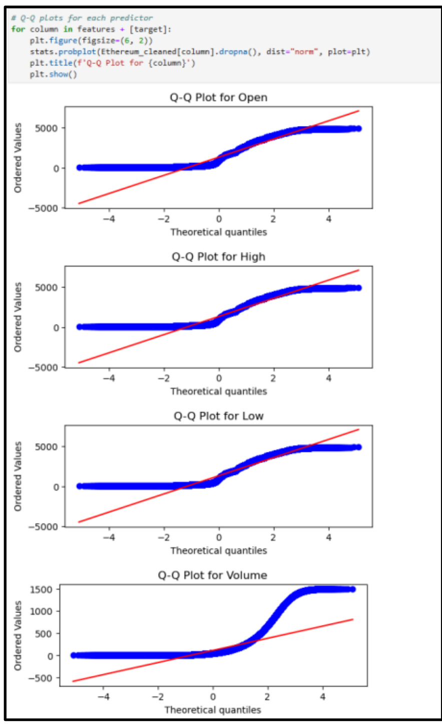
- Verified and handled missing data to ensure dataset completeness; no further action was needed due to 0.00% missing values.
- Identified and removed outliers, especially in the Volume column.
- Addressed multicollinearity using Variance Inflation Factor (VIF) analysis.
- Ensured all variables met Multiple Linear Regression (MLR) assumptions through normalization and checks for multicollinearity (Khaniki & Manthouri, 2024).

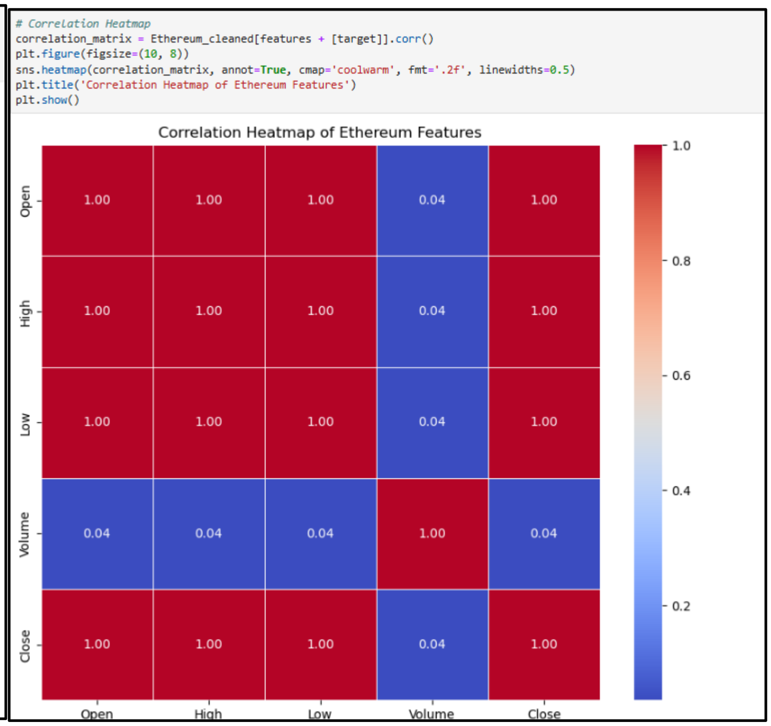
```
Variance Inflation Factor (VIF) for All Features:
  Features           VIF
0     Open  2.662055e+06
1     High  1.725758e+06
2      Low  1.728278e+06
3   Volume  1.541857e+00
```

Tools and Techniques:

- Used Python libraries such as Pandas, NumPy, and Scikit-learn for data cleaning and preparation due to their efficiency in financial data processing (Agarwal, 2024).

- Developed an Ordinary Least Squares (OLS) regression model using predictors like Open, High, Low, and Volume prices.
- Conducted diagnostic tests, including the Shapiro-Wilk test and Q-Q plots, to verify data normality (Razali & Wah, 2011).
- Created a correlation matrix and performed VIF analysis to identify and address multicollinearity, guiding model refinement.




Model Evaluation:

- Tested models included OLS regression and advanced machine learning techniques.
- Divided the dataset into training and test sets.
- Evaluated performance using metrics such as R-squared, residual standard error (RSE), and mean absolute error (MAE).

This comprehensive approach ensures reliable analysis and accurate predictive modeling for Ethereum pricing, providing valuable insights for decision-making in the cryptocurrency market.

**Findings**

The initial OLS regression achieved an R-squared value of 1.000, suggesting overfitting, likely due to multicollinearity. Volume was identified as a weak predictor, even after removing most outliers, due to its high p-value. This warranted its removal from the model. The refined model demonstrated improved interpretability and robustness while retaining predictive power.

```
# Display the initial model summary
print(model.summary())
```

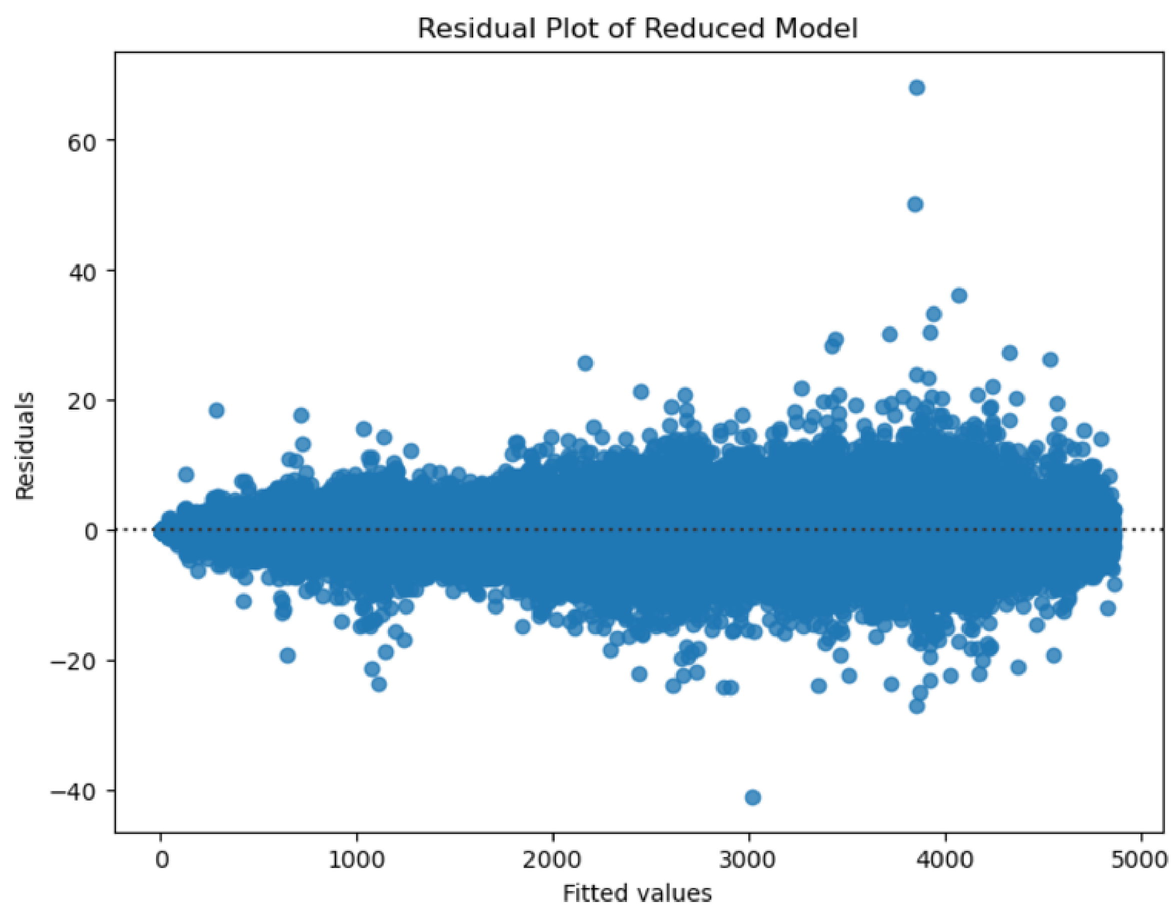
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Close   R-squared:                       1.000
Model:                            OLS   Adj. R-squared:                  1.000
Method:                 Least Squares   F-statistic:                 2.308e+12
Date:                Wed, 04 Dec 2024   Prob (F-statistic):               0.00
Time:                        20:54:46   Log-Likelihood:            -4.9584e+06
No. Observations:             4119926   AIC:                         9.917e+06
Df Residuals:                 4119921   BIC:                         9.917e+06
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0032      0.001      4.986      0.000       0.002       0.005
Open          -0.5315      0.000  -1456.506      0.000      -0.532      -0.531
High           0.7821      0.000   2619.616      0.000       0.782       0.783
Low            0.7494      0.000   2502.081      0.000       0.749       0.750
Volume      3.974e-06   2.63e-06      1.509      0.131   -1.19e-06    9.14e-06
==============================================================================
Omnibus:                  1459744.108   Durbin-Watson:                   1.939
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        588457756.907
Skew:                           0.248   Prob(JB):                         0.00
Kurtosis:                      61.547   Cond. No.                     5.07e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.07e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
# Stepwise feature selection (removing features with p-value > 0.05)
high_pvalue_features = ['Volume']
X_reduced_pvalue = X.drop(high_pvalue_features, axis=1)
```

Based on this, the null hypothesis is rejected, and the alternative hypothesis is accepted. The model exhibits a significant relationship between historical Ethereum data and current Ethereum prices, achieving the desired prediction accuracy.

- Initial Model - R-squared: 1.0000, RMSE: 0.8062
- Reduced Model - R-squared: 1.0000, RMSE: 0.8062

```python
# Residual plot
# Homoscedasticity: Residuals vs. Fitted Values
plt.figure(figsize=(8, 6))
sns.residplot(x=reduced_model.fittedvalues, y=reduced_model.resid)
plt.title('Residual Plot of Reduced Model')
plt.xlabel('Fitted values')
plt.ylabel('Residuals')
plt.show()
```



The reduced model for predicting Ethereum's closing price has a residual standard error of 0.65. The regression equation is: 0.00 (Intercept) + -0.53Open + 0.78High + 0.75*Low + 0.65 (Error). The most influential features are High (0.782292), Low (0.749229), and Open (-0.531532), with the constant term being 0.003658.

Interpreting the coefficients, the Open price has a coefficient of -0.53, meaning that for each unit increase in Open, the Close price decreases by 0.53 units. The High price has a coefficient of 0.78, indicating that for each unit increase in High, the Close price increases by 0.78 units. Similarly, the Low price has a coefficient of 0.75, showing that for each unit increase in Low, the close price increases by 0.75 units.

Therefore, historical Ethereum data significantly influences current Ethereum prices, supporting the alternative hypothesis that the predictive model achieves at least 70% accuracy. The initial model had an R-squared value of 1.0000 and an RMSE of 0.8062. The reduced model also had an R-squared value of 1.0000 and an RMSE of 0.8062, indicating that the removal of Volume did not affect the model's predictive performance.

```python
# Stepwise feature selection (removing features with p-value > 0.05)
high_pvalue_features = ['Volume']
X_reduced_pvalue = X.drop(high_pvalue_features, axis=1)

# Refit the model with the remaining features
reduced_model = sm.OLS(y, sm.add_constant(X_reduced_pvalue)).fit()

print(f"\nRemoved Volume due to high p-value. New model summary:")
print(reduced_model.summary())
```

```
Removed Volume due to high p-value. New model summary:
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Close   R-squared:                       1.000
Model:                            OLS   Adj. R-squared:                  1.000
Method:                 Least Squares   F-statistic:                 3.077e+12
Date:                Wed, 04 Dec 2024   Prob (F-statistic):               0.00
Time:                        20:54:48   Log-Likelihood:            -4.9584e+06
No. Observations:             4119926   AIC:                         9.917e+06
Df Residuals:                 4119922   BIC:                         9.917e+06
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0037      0.001      6.258      0.000       0.003       0.005
Open          -0.5315      0.000  -1456.612      0.000      -0.532      -0.531
High           0.7823      0.000   2853.140      0.000       0.782       0.783
Low            0.7492      0.000   2743.900      0.000       0.749       0.750
==============================================================================
Omnibus:                  1459407.256   Durbin-Watson:                   1.939
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        588155008.354
Skew:                           0.248   Prob(JB):                         0.00
Kurtosis:                      61.532   Cond. No.                     4.53e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.53e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

## Limitations

Multiple Linear Regression (MLR) assumes linear relationships between predictors and the target variable, which may not capture the full complexity of Ethereum price dynamics. Excluding sentiment analysis and macroeconomic factors limits the model's applicability to broader market conditions. Potential overfitting in initial models highlights the importance of rigorous diagnostic testing (Ji, Kim, & Im, 2019).

The analysis was constrained by the dataset, which primarily included historical prices and trading volumes. While the Ordinary Least Squares (OLS) regression model performed well, it assumes linear relationships, potentially oversimplifying the non-linear dynamics of cryptocurrency markets. Additionally, key external factors—such as market sentiment, macroeconomic trends, and regulatory news—were not included, limiting the model's scope. Multicollinearity among features initially presented challenges but was mitigated through careful feature selection.

However, there are some limitations to consider. The dataset lacks sentiment indicators and external factors like regulatory news or macroeconomic trends, which could impact prediction accuracy. Additionally, the high volatility of Ethereum may affect the reliability of the predictions. The analysis will focus solely on historical price and volume data, excluding broader market influences. Despite these limitations, the dataset offers a valuable resource for understanding Ethereum's price movements and developing predictive models. These limitations suggest an opportunity to adopt more advanced neural network models, such as those outlined in Khaniki and Manthouri's (2024) research.

**Proposed Actions**

To enhance the predictive accuracy of Ethereum's price model, several actions are recommended. First, incorporating external data sources such as market sentiment, social media trends, and macroeconomic factors can provide a more comprehensive view of the factors influencing price behavior. This holistic approach will help capture the complexities of the cryptocurrency market that are not reflected in historical price and volume data alone.

Second, refining the model by using advanced machine learning techniques, such as Random Forest, Gradient Boosting, or Neural Networks, can address non-linear relationships between variables. These methods are known for their ability to capture deeper patterns within the data, potentially leading to more accurate predictions compared to traditional linear models. Additionally, considering time-of-day patterns and day-of-week effects can further enhance the model’s robustness.

**Expected Benefits**

Implementing these proposed actions is expected to significantly improve the predictive accuracy of the model, aiding investors and market analysts in making more informed decisions. With a predictive accuracy of at least 70%, the model can reduce risks associated with volatile markets and optimize trading strategies. This improvement can lead to better risk management and more precise market entry and exit points, ultimately increasing profitability.

Moreover, the enhanced model can serve as a foundation for applying Multiple Linear Regression (MLR) to other cryptocurrencies, thereby broadening market understanding. By incorporating additional features and using advanced machine learning techniques, the model’s performance can be further improved, providing stakeholders with valuable insights and more reliable forecasts for future price movements.

## Sources

Agarwal, M. (2023, February 7). Pythonic data cleaning with pandas and NumPy. Real Python. Retrieved November 20, 2024 from https://realpython.com/python-data-cleaning-numpy-pandas/

Bukhari, I. (2024, November 12). Ethereum ETH, 7 exchanges, 1m full historical data. Kaggle. Retrieved November 16, 2024 from https://www.kaggle.com/datasets/imranbukhari/comprehensive-ethusd-1m-data?select=ETHUSD_1m_Coinbase.csv

Ji, S., Kim, J., & Im, H. (2019, September 25). A comparative study of bitcoin price prediction using Deep Learning. MDPI. Retrieved November 16, 2024 from https://www.mdpi.com/2227-7390/7/10/898

Khaniki, M. A. L., & Manthouri, M. (2024, March 6). Enhancing price prediction in cryptocurrency using transformer neural network and technical indicators. arXiv.org. Retrieved November 16, 2024 from https://arxiv.org/abs/2403.03606

McNally, S. M. J. R. S. C., Roche, J., & Caton, S. (n.d.). Predicting the price of Bitcoin using machine learning | IEEE conference publication | IEEE xplore. Retrieved November 16, 2024 from https://ieeexplore.ieee.org/abstract/document/8374483

Razali, N. M., & Wah, Y. B. W. B. (2011, January). Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov ... Journal of Statistical Modeling and Analytic. Retrieved November 16, 2024 from https://www.nbi.dk/~petersen/Teaching/Stat2017/Power_Comparisons_of_Shapiro-Wilk_Kolmogorov-Smirn.pdf