Project Report: Ethereum Price Forecasting with ARIMA

Name: Fahad Abdullah
Intern ID: N/A
Phone Number: +923266121925

GitHub Repository:
https://github.com/FAbdullah17/Ethereum-Price-Forecasting

Project Overview

Ethereum (ETH) price forecasting is a challenging but important task due to the cryptocurrency market's volatility and significant investment interest. Accurate predictions can guide investors and traders toward better decisions and potentially higher profits. In this project, we address the problem of forecasting the daily closing price of ETH/USD using historical data and advanced time series methods. Our goal is to build an ARIMA-based model to predict the next 30 days of Ethereum prices, providing insights into trends and model performance.

Data Collection and Preprocessing

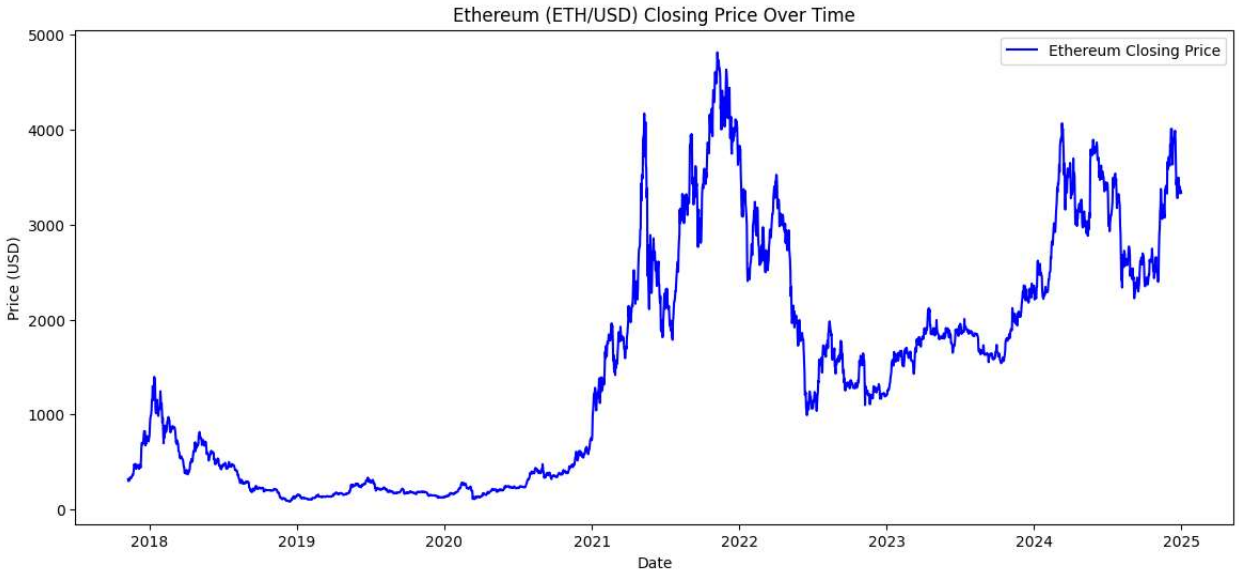
- We used the yfinance library to download ETH-USD historical data from Yahoo Financefile-b9yni1okmzmylr6mwnzkzk. The data spans January 1, 2016 to January 1, 2025 and includes the **Open**, **High**, **Low**, **Close**, and **Volume** columns.
- After fetching, we reset the index, converted the **Date** column to datetime, and set it as the index for a proper time-series formatfile-b9yni1okmzmylr6mwnzkzkfile-b9yni1okmzmylr6mwnzkzk.
- The dataset was inspected for missing values and anomalies (none significant were found), ensuring a clean, continuous daily series.
- Finally, we computed a 30-day rolling mean and 30-day rolling standard deviation of the closing price to assist in trend and volatility analysisfile-b9yni1okmzmylr6mwnzkzk.

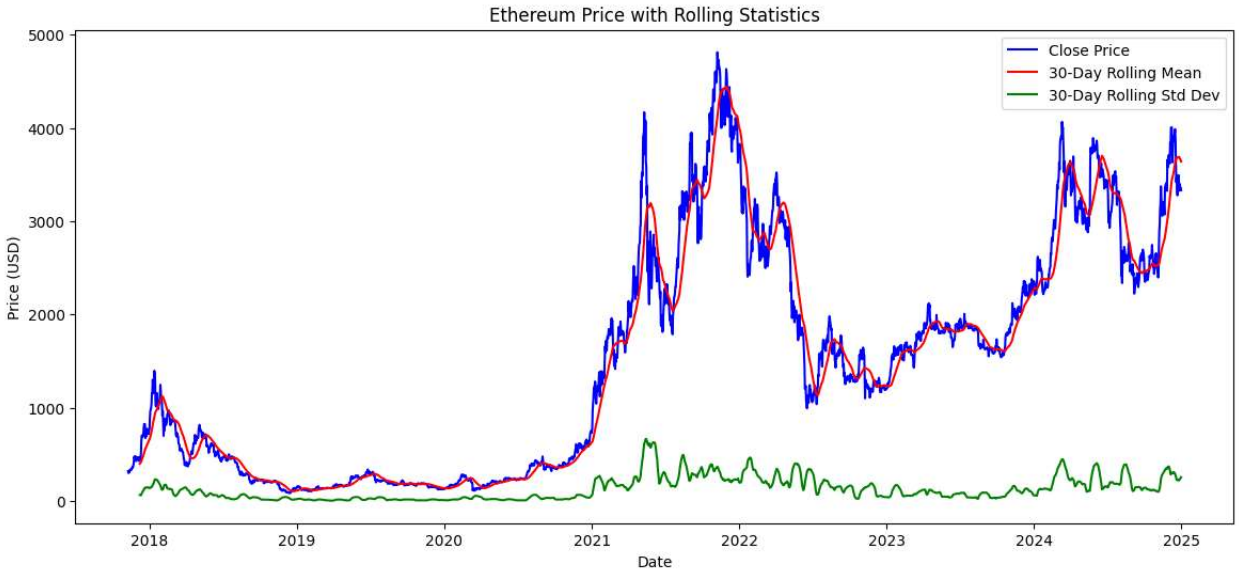
Exploratory Data Analysis (EDA)

- The closing price trend showed that Ethereum generally rose over the period with significant peaks and corrections. Notably, we observed sharp

rallies (e.g. late 2017 and early 2021) followed by steep declines, reflecting the cryptocurrency's volatility. Overall, the long-term trend is upward but far from smooth.

- Plotting a 30-day rolling mean of the closing price provided a smoothed trend line, while the 30-day rolling standard deviation quantified volatility file-b9yni1okmzmylr6mwnzkzk. These revealed that volatility spikes coincide with dramatic price moves.
- Summary statistics confirmed a wide range of values: for example, the closing price mean was in the mid-hundreds (USD) while the standard deviation was large, indicating high volatility typical of crypto markets.
- These EDA steps helped us understand the data's scale and dynamics before modeling.

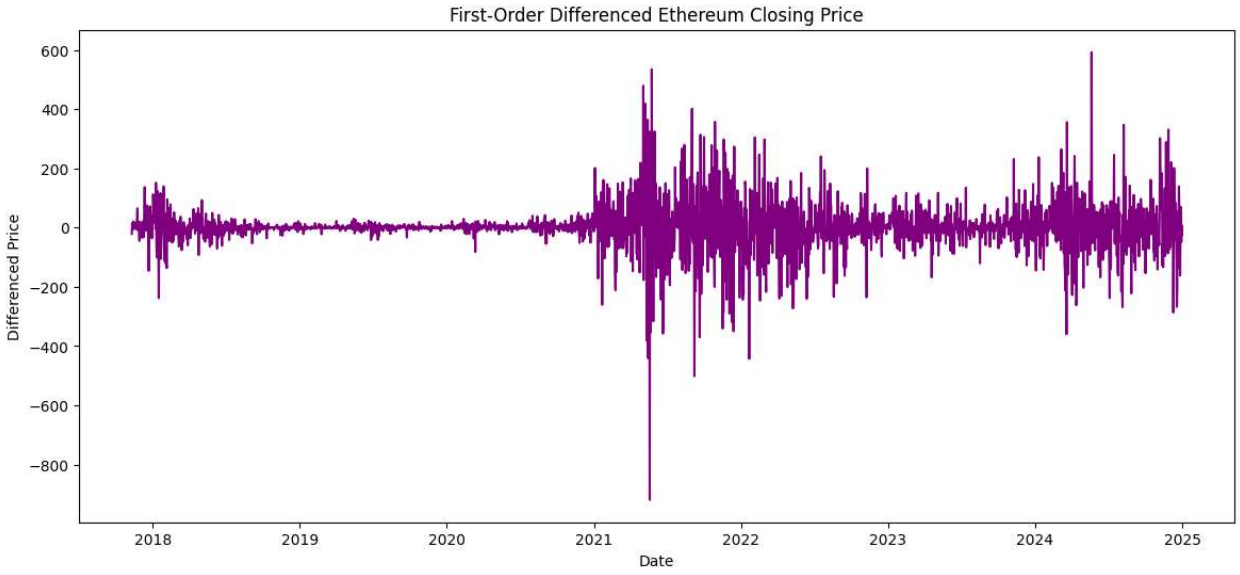




Stationarity Testing

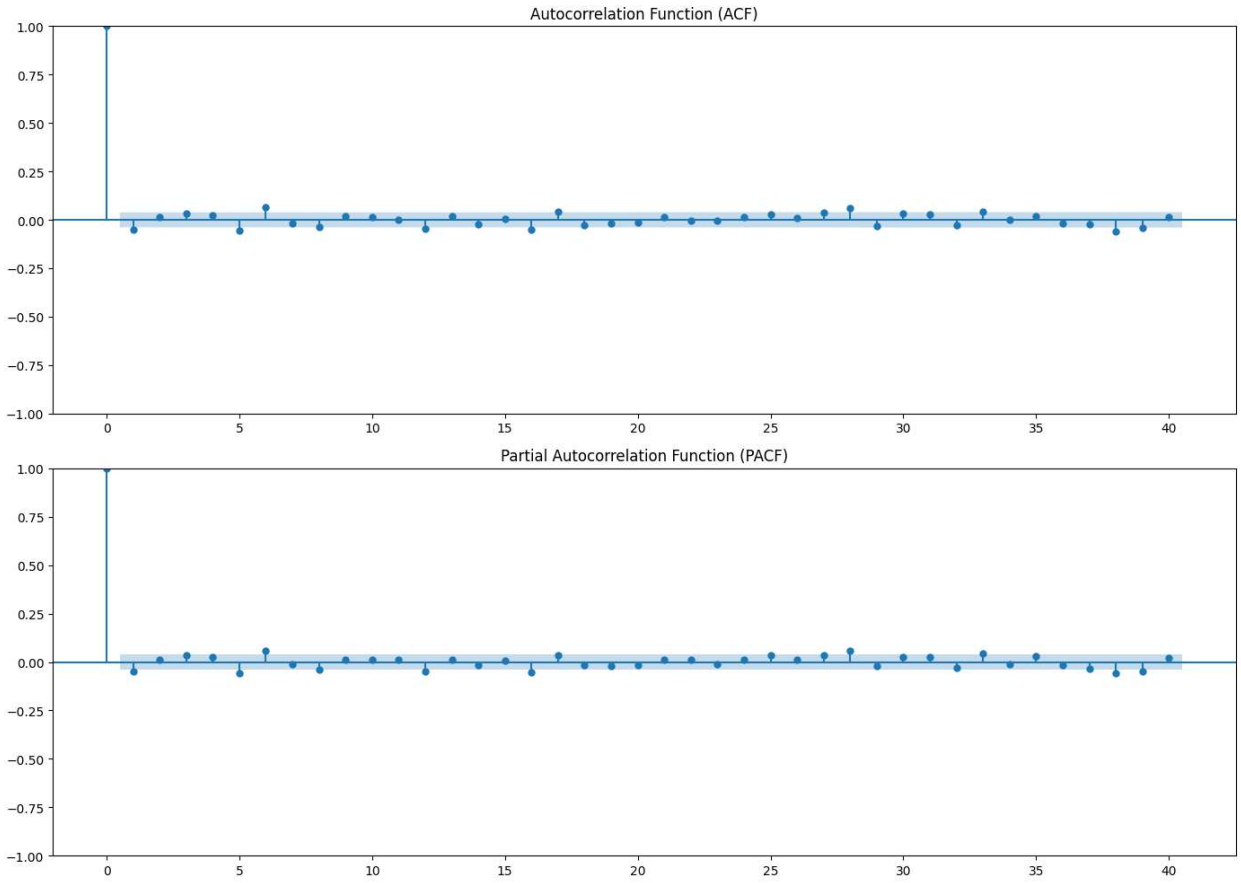
Time series models like ARIMA assume stationarity (constant mean and variance over time). We tested this using the Augmented Dickey-Fuller (ADF) test, which checks the null hypothesis that a unit root (non-stationarity) is present. A low p-value (≤0.05) indicates stationarity.

- Running the ADF test on the original ETH/USD closing price returned an ADF statistic of -1.324 and a p-value of 0.618. Since p>0.05, we fail to reject the null hypothesis, indicating the original series is non-stationaryfile-b9yni1okmzmylr6mwnzkzk.
- To enforce stationarity, we applied first-order differencing to the series. The ADF test on the differenced data gave a statistic around -12.71 with p-value ≈1.045×10^-23, which is highly significant. This indicates the differenced series is stationaryfile-b9yni1okmzmylr6mwnzkzk.
- In summary, the raw Ethereum price series exhibited trends and volatility, but the first difference removed these, satisfying stationarity requirements.



ACF and PACF Analysis

We plotted the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) of the differenced series to guide ARIMA parameter selection. The ACF provides a comprehensive view of correlations at all lags (showing how each value relates to its past). The PACF, in contrast, isolates the direct correlation at each lag (controlling for intermediate lags). In practice, peaks in the PACF plot often suggest the order of an autoregressive (AR) term, while the ACF informs moving-average (MA) components. For our data, the ACF decayed gradually while the PACF showed a significant spike at lag 1 and not beyond, suggesting an AR(1) model with little MA component.



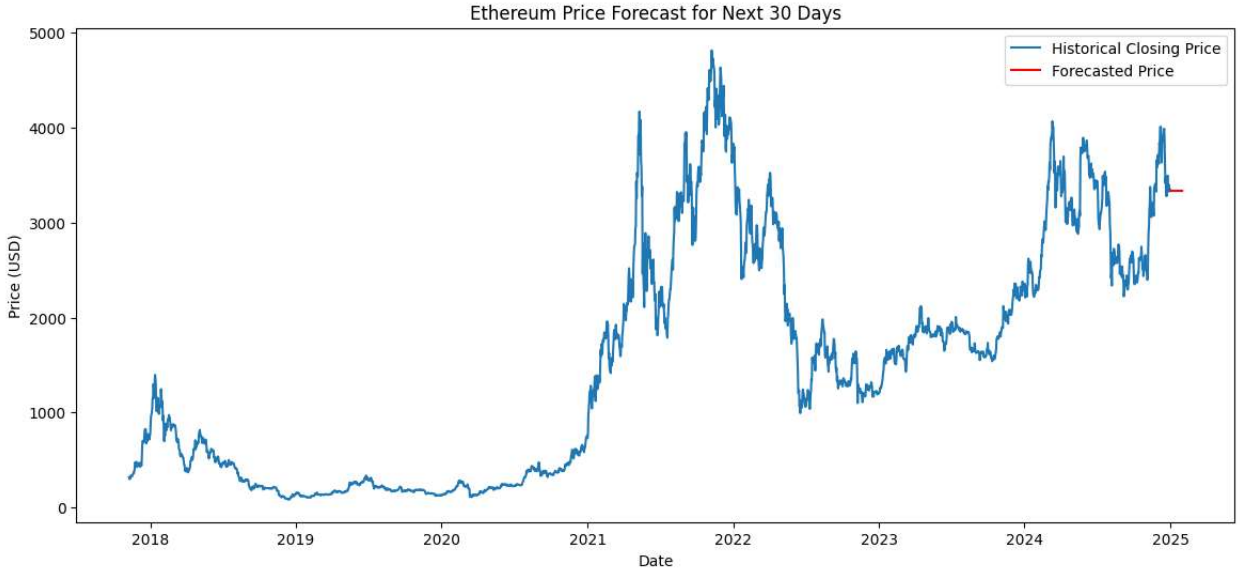
ARIMA Model Building with auto_arima

Instead of manually choosing (p,d,q), we used the pmdarima.auto_arima function for automated parameter tuningfile-b9yni1okmzmylr6mwnzkzk. This performs a stepwise search over combinations of (p,d,q) and selects the model with the lowest AIC. We allowed it to consider non-seasonal ARIMA models (no seasonal component). The output indicated that the best model was **ARIMA(1,1,0)** with an interceptfile-b9yni1okmzmylr6mwnzkzk.

This model has one autoregressive term (p=1) and first differencing (d=1), with no MA term (q=0). The SARIMAX summary reported an AR(1) coefficient of about -0.0504 and an innovation variance (sigma²) ≈ 6261.08. In practical terms, the negative AR coefficient implies a weak inverse relationship between successive prices after differencing. Overall, auto_arima efficiently identified the optimal (p,d,q) and we proceeded with this model.

Forecasting and Visualization

Using the fitted ARIMA(1,1,0) model, we generated forecasts for the next 30 calendar days beyond our dataset. We created a forecast series indexed by future dates and plotted it alongside historical closing prices. The plot clearly shows the forecasted prices (in red) continuing the recent trend of Ethereum's closing price (in blue). We forecasted 30 days ahead as specified in the task requirementsfile-b9yni1okmzmylr6mwnzkzk. The forecast curve represents the model's projected future prices under the assumption that recent patterns persist.

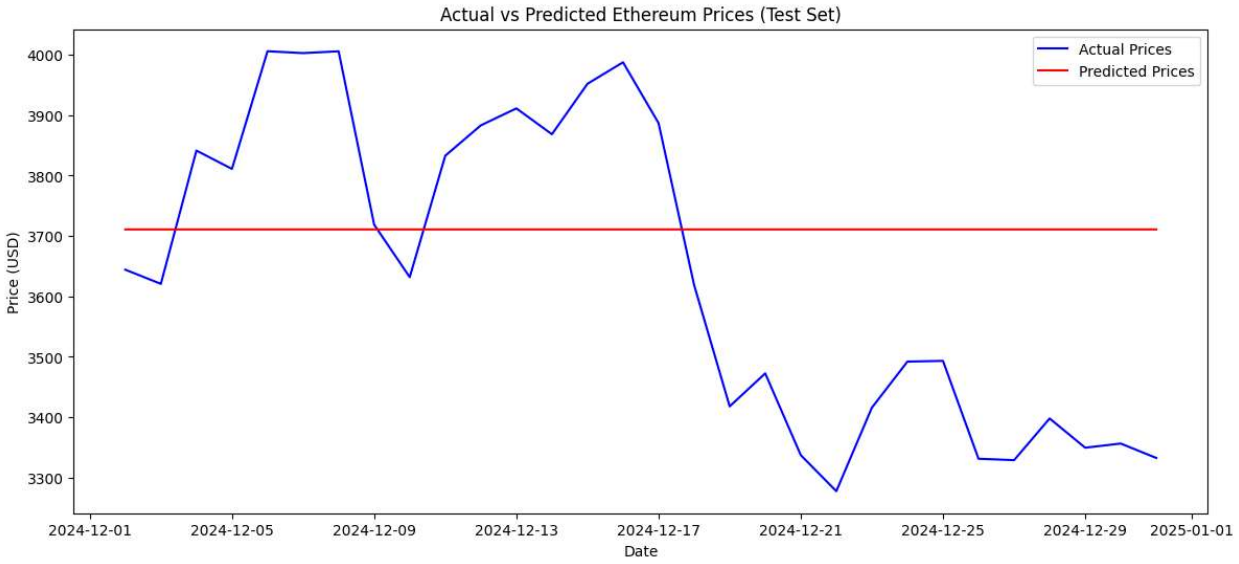


Model Evaluation

We evaluated the model by comparing its predictions to actual data over a 30-day test period. We treated the last 30 days of our data as a holdout test set. Retraining the model on the remaining data and forecasting these 30 points, we computed the following error metrics:

- **RMSE (Test)** ≈ 260.01
- **MAPE (Test)** ≈ 6.58%file-b9yni1okmzmylr6mwnzkzk.
  These results indicate that, on average, the model's predictions were within

about 6.6% of the actual prices. Given the inherent volatility of cryptocurrency, this is a reasonably low error and suggests the model captured the general price dynamics. The low RMSE and MAPE reflect that the ARIMA model provided a fairly good fit, though not perfect, which is expected in such a noisy market.



Insights and Learnings

- **Price Behavior:** Ethereum exhibited considerable volatility, with sharp rises and falls. The ARIMA model captured underlying trends but may understate sudden market shifts. Differencing handled the overall trend well, but abrupt events (e.g. news-driven spikes) remain hard to predict with a simple linear model.
- **Model Performance:** The optimized ARIMA(1,1,0) (found via auto_arima) had a modest autoregressive effect. The automatic tuning was a strength of our approach, saving manual trial-and-errorfile-b9yni1okmzmylr6mwnzkzk. The model achieved a low percentage error (MAPE ~6.6%), indicating decent accuracy for such volatile data.

- **Application:** A ~6.6% average error means the forecasts are fairly reliable on a day-to-day basis, which could be useful for short-term planning. However, risk management remains critical due to unpredictable volatility.
- **Automated Tuning:** Using auto_arima greatly simplified the process. It systematically tested many parameter combinations and selected the best model based on AIC, which is much more efficient than manual search.
- **Future Directions:** We observed that seasonal or non-linear effects might exist. As suggested in our analysis, exploring models like seasonal ARIMA (SARIMA) to capture any periodic patterns, or advanced approaches such as Facebook Prophet or LSTM neural networks to handle complex trends, could further improve forecastsfile-b9yni1okmzmylr6mwnzkzkfile-b9yni1okmzmylr6mwnzkzk.

Conclusion

In summary, this project collected Ethereum price data, performed thorough EDA and stationarity checks, and built an ARIMA forecasting model using automated tuning. The final ARIMA(1,1,0) model was effective at capturing the main trend in the data. The 30-day forecasts generated aligned with recent price trends, and evaluation metrics (RMSE ≈ 260, MAPE ≈ 6.6%) showed the model was reasonably accurate. Leveraging auto_arima was a key strength of our approach, as it automated parameter selection and optimized model performance without manual guessworkfile-b9yni1okmzmylr6mwnzkzk.

For future work, the model can be enhanced by incorporating seasonal effects or trying alternate algorithms. For example, a SARIMA model could account for any periodic behavior in the data, and machine-learning-based methods like Facebook's Prophet or recurrent neural networks (LSTM) may capture non-linear patterns missed by ARIMA. Overall, the ARIMA-based approach provided a solid foundation and valuable insights into Ethereum's price behavior, serving as a baseline for more sophisticated forecasting in the future.

References

1. Hyndman, R.J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice* (2nd ed). OTexts. (For time series forecasting foundations, ARIMA, stationarity concepts)
2. Makridakis, S., Wheelwright, S.C., & Hyndman, R.J. (1998). *Forecasting: Methods and Applications* (3rd ed.). Wiley. (For ACF, PACF, model selection)
3. Brownlee, J. (2017). *Introduction to Time Series Forecasting with Python*. Machine Learning Mastery. (For ADF test, differencing, and model evaluation)
4. Pmdarima Documentation. (2024). https://alkaline-ml.com/pmdarima/ (For auto_arima functionality and usage)
5. Yahoo Finance (2025). https://finance.yahoo.com/ (For historical Ethereum (ETH-USD) price data)
6. Kaggle (2025). (General platform reference for crypto time series datasets)
7. Facebook Research. (2017). Prophet: Forecasting at Scale. https://facebook.github.io/prophet/ (Suggested future improvement: using Prophet for forecasting)
8. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. (Suggested future improvement: using LSTM for time series)