Time Series Forecasting of Ethereum (ETH) Prices using ARIMA Model

Preet Bobde

November 8, 2024

1 Introduction

Time series forecasting is a crucial technique for predicting future values based on past observations. In this report, we apply the ARIMA model to predict the future prices of Ethereum (ETH) using daily price data from the ETH/USDT trading pair. The ARIMA model is particularly useful for modeling time series data where past values have an autocorrelated structure, which is typical in financial markets.

2 Understanding Data

The dataset used in this study contains daily prices of Ethereum against the US Dollar (ETH/USDT), including open, high, low, close prices, and trading Volume.

The dataset is now ready for analysis, and we proceed by performing exploratory data analysis (EDA) to understand the trends and patterns in the data.

2.1 Data Visualization

One of the first steps in EDA is to plot the closing prices to observe the general trend of the Ethereum market over time. We use the Matplotlib library for visualizing the data.

This visualization helps us identify any trends, seasonality, or volatility in the ETH prices.

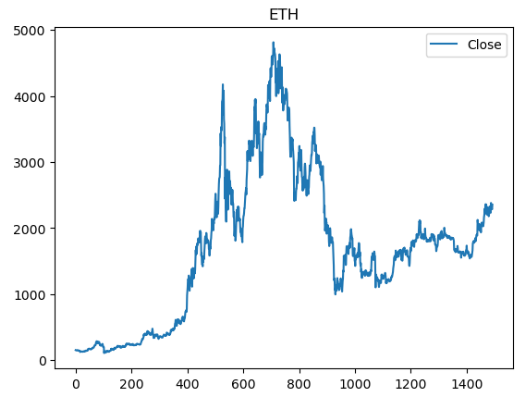

Figure 1: ETHUSDT Price from 01/01/2019 to 31/12/2022

3 Exploratory Data Analysis (EDA)

Before building the model, we need to analyze the data further to identify patterns. We focus on the cumulative returns and the autocorrelation of ETH prices.

3.1 Cumulative Returns

Cumulative returns are a useful metric to assess the total return on an asset over time. It is calculated by taking the cumulative sum of returns from the asset price. We plot the cumulative returns of ETH to see how the price has evolved over time.

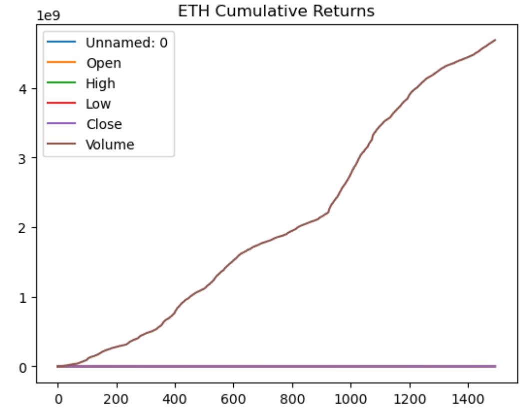

Figure 2: Cumulative return plot

3.2 Autocorrelation and Lag Plot

Autocorrelation indicates how much the current price depends on past values, and the lag plot helps visualize this relationship. We plot the lag plot of the 'Open' price series with a lag of 5 days to investigate this further.

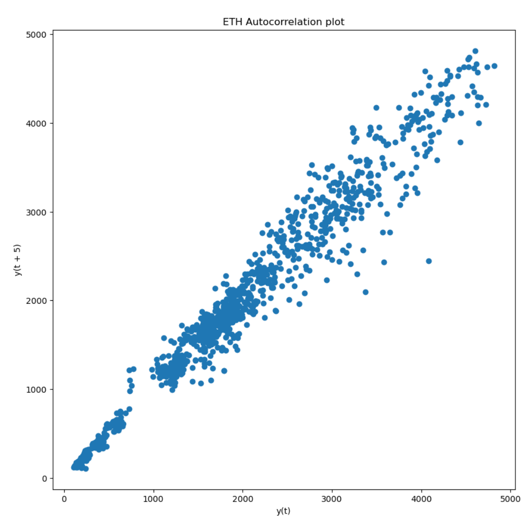

Figure 3: Lag Plot of the 'Open' Price with 5-Day Lag

These plots provide insight into the dependence of the ETH prices on previous values, which is crucial for building a time series model.

4 ARIMA Model

The ARIMA model consists of three main components:

- AutoRegressive (AR) term: The relationship between an observation and a specified number of lagged observations.
- Integrated (I) term: The differencing of raw observations to make the time series stationary.
- Moving Average (MA) term: The relationship between an observation and a residual error from a moving average model applied to lagged observations.

4.1 Splitting the Data

We split the data into training and test sets, with 80% of the data used for training and the remaining 20% for testing. This split allows us to evaluate the model's performance on unseen data.

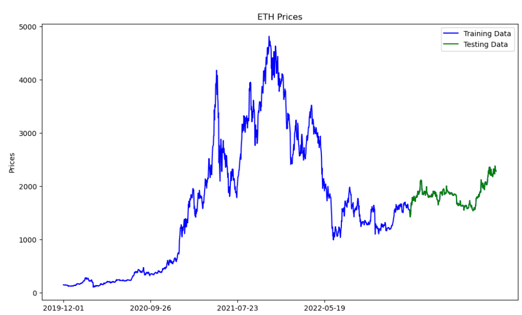

Figure 4: Train Test Split

4.2 Fitting the ARIMA Model

We fit an ARIMA model to the training data. In this case, we use an ARIMA model with parameters $p = [1]$, $d = 1$, and $q = 1$. These parameters represent the order of the AR, I, and MA components, respectively. After fitting the model, we make predictions for the test set.

The Mean Squared Error (MSE) is calculated to evaluate the accuracy of the predictions. A lower MSE indicates better performance.

4.3 SMAPE Calculation

Symmetric Mean Absolute Percentage Error (SMAPE) is another important evaluation metric. It is especially useful in financial forecasting because it expresses the error as a percentage of the actual value.

The formula for SMAPE is:

$$\text{SMAPE} = \frac{1}{n} \sum_{i=1}^{n} \frac{|y_{\text{true},i} - y_{\text{pred},i}|}{\frac{|y_{\text{true},i}|+|y_{\text{pred},i}|}{2}} \times 100$$

5 Model Evaluation

After obtaining the forecasts, we evaluate the model's performance using both MSE and SMAPE. The model's ability to generalize to unseen data is assessed by comparing the predictions to the actual values in the test set.

5.1 Evaluation Results

For the ARIMA model with parameters $p = 1$, $d = 1$,$q = 1$ and $step = 1$ we obtain the following evaluation metrics:

5.2 Evaluation Results

For the ARIMA model with parameters $p = 1$, $d = 1$, $q = 1$, and step $= 1$, we obtained the following evaluation metrics:

Forecast Length (l)	**Mean Squared Error (MSE)**	**SMAPE (%)**
1	1986.551	1.634
2	38019.051	8.173
3	47350.653	9.109

Table 1: Evaluation Metrics for ARIMA Model with $p = 1$, $d = 1$, $q = 1$

These results suggest that the model performs reasonably well, but there is still room for improvement.

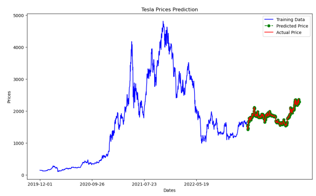

Figure 5: ARIMA (1,1,1) for l = 1

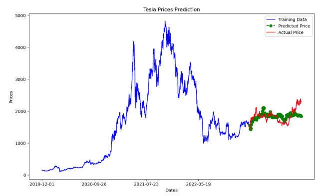

Figure 6: ARIMA (1,1,1) for l = 2

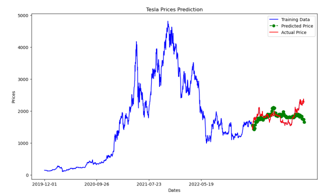

Figure 7: ARIMA (1,1,1) for l = 3

6 Hyperparameter Tuning

The ARIMA model's performance depends significantly on the choice of its hyperparameters: p, d, and q. We conduct a grid search over a range of values for these parameters to find the optimal combination.

6.1 Grid Search

We perform a grid search over the values $p \in [0, 1, 2, 3]$, $d \in [0, 1, 2, 3]$, and $q \in [0, 1, 2, 3]$, evaluating the model's performance for each combination of parameters.

We can generate heatmaps for both MSE and SMAPE to visualize the impact of different hyperparameter combinations on model performance.

7 Conclusion

In this report, we successfully used the ARIMA model to forecast Ethereum (ETH) prices. We performed data preprocessing, exploratory data analysis, model fitting, and evaluation. Our model was able to produce reasonably accurate predictions, as evaluated by both MSE and SMAPE. Further tuning of the model's hyperparameters could improve its accuracy. Future work may explore other models like SARIMA or machine learning-based methods for better performance.