MA-641 Final Project

Cryptocurrency Price Forecasting & Volatility Modelling

Amit Bhat Srirangapatna Guruprasad

Abstract

This report presents a detailed analysis and forecasting for hourly Dogecoin prices throughout 2021. Beginning with a minute-level price series converted to hourly observations, we perform data preprocessing including missing value imputation and train-test (80% , 20%) splitting followed by in depth exploratory data analysis including visualization and stationarity testing using ADF test. We then fit a bunch of time-series models: non seasonal ARIMA models, seasonal SARIMA models, and ARIMA–GARCH models. Model selection is guided by information criteria (AIC, BIC), residual diagnostics (normality tests, Ljung–Box test), and test set forecasting performance. Our results indicate that a simple ARIMA (2, 1, 0) model offers a strong baseline, while ARIMA–GARCH configurations (ARMA (2, 1, 1) GARCH (1, 1)) best capture the heavy-tailed volatility structure. Forecasts reproduce the overall price trend but tend to understate extreme intraday swings and fails to capture the spikes and dips. We conclude by discussing limitations such as residual non-normality and ignored exogenous drivers and outline directions for future work, including multivariate extensions and machine-learning approaches.

Introduction

Cryptocurrency markets have exhibited rapid growth and pronounced volatility since their inception. Dogecoin originally launched as a meme coin token surged to global prominence in early 2021, propelled by social-media buzz and high-profile endorsements. Accurate short-term forecasting and volatility estimation in such markets hold practical value for traders, risk managers, and automated trading systems. Traditional financial models like ARIMA have been widely applied to capture price dynamics, but pure ARIMA often fails to model clustered volatility and heavy tails. Extensions such as SARIMA allow for seasonal cycles, while GARCH-style models explicitly account for time-varying variance.

In this project, we focus on hourly Dogecoin prices for calendar year 2021. Our objectives are as follows:

1) to preprocess and structure high-frequency data into a reliable hourly series
2) to systematically identify, estimate, and diagnose a spectrum of time-series models that is non-seasonal ARIMA, seasonal SARIMA, and ARIMA–GARCH models.
3) to compare model fits and forecasting accuracy, culminating in recommendations for practitioners.

By accessing information criteria, residual diagnostics, and forecast performance, we aim to determine which modelling framework best balances trend capture and volatility modelling in the context of an emerging, highly volatile digital asset.

Data Description

The primary dataset is a publicly available Kaggle dataset containing minute level Dogecoin price quotations denominated in USD for the entire calendar year 2021. It comprises two fields:

- **open_time (timestamp)**: the start time of each one-minute interval, formatted as "dd/mm/YYYY HH: MM."
- **price (USD)**: the transaction price of Dogecoin at that minute.

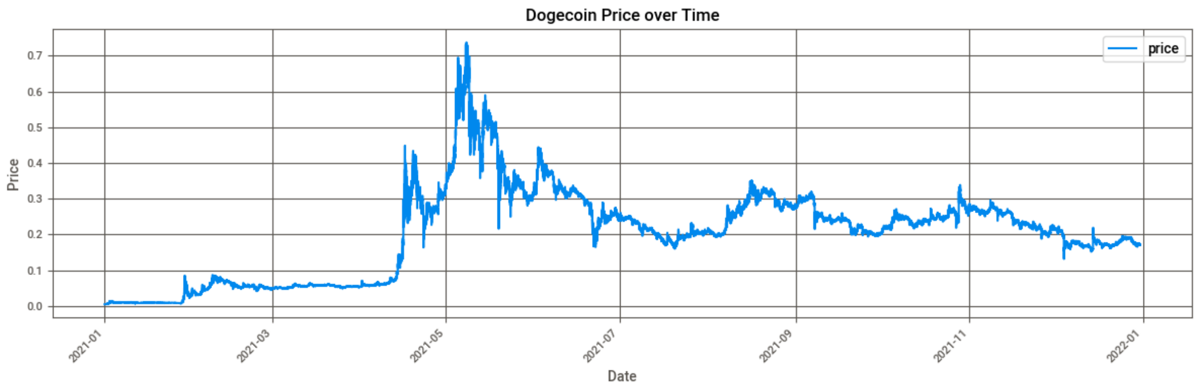
After initial parsing, the series contains 525,600 observations (365 days × 1,440 minutes) before resampling. The price exhibits extreme variation, ranging from approximately $0.0046 at the start of the year to $0.7377 at its mid-May peak.

Summary statistics of the hourly series (n = 6025 total hours after resampling) are as seen in EDA:

- **Minimum price**: $0.0046
- **Maximum price**: $0.7377
- **Mean (train)**: $0.2041
- **Standard deviation (train)**: $0.1264
- **Median**: $0.2180
- **25th percentile**: $0.0619
- **75th percentile**: $0.2732.

These figures underscore both the upward spike in mid-2021 and the heavy-tailed distribution characteristic of cryptocurrency returns.

Time Series Plot



Data Preprocessing

3.1 Parsing and Resampling

- Datetime conversion: The open_time field was parsed into Python datetime objects using the format %d/%m/%Y %H: %M.

- Chronological sorting: Observations were sorted ascending by timestamp to ensure time order.

- Hourly aggregation: The minute-level series was resampled to an hourly frequency by computing the mean price within each hour, yielding 6025 hourly observations.

3.2 Handling Missing Data

- Forward fill: Six entries in the train set contained no trades and resulted in missing values; these were imputed via forward-filling to preserve continuity. No further outlier removal was applied, under the assumption that extreme values reflect genuine market behaviour.

3.3 Train–Test Split

- Temporal split: To evaluate out-of-sample forecasting, the series was split chronologically: the first 80 % (4820 hours) for model training and the final 20 % (1205 hours) for testing. This split ensures that forecasts are evaluated on truly unseen future data.

3.4 Stationarity Transformation

- Augmented Dickey–Fuller (ADF) test: The raw hourly series is non-stationary (ADF statistic = –1.9589, p = 0.3049).

```python
from statsmodels.tsa.stattools import adfuller

adf_result = adfuller(train['price'])
print(f"ADF Statistic: {adf_result[0]}")
print(f"p-value: {adf_result[1]}")

```

```
ADF Statistic: -1.9589261388292258
p-value: 0.3048676933720432
```

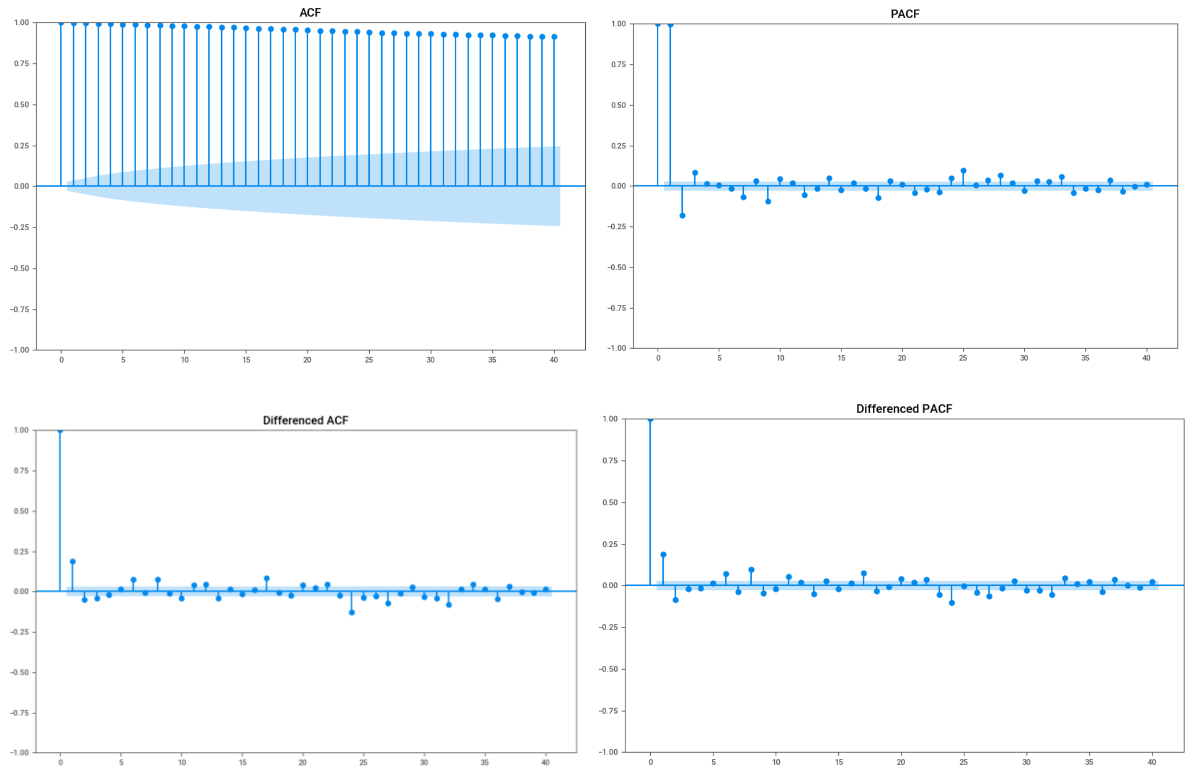
- Differencing: A first difference was taken, yielding stationarity (ADF statistic = –13.7914, p < 10^{-25}). Subsequent models operate on the differenced series to satisfy ARIMA assumptions.

```python
train['price_diff'] = train['price'].diff().dropna()
#adf test
adf_result_diff = adfuller(train['price_diff'].dropna())
print(f"Differenced ADF Statistic: {adf_result_diff[0]}")
print(f"Differenced p-value: {adf_result_diff[1]}")

```

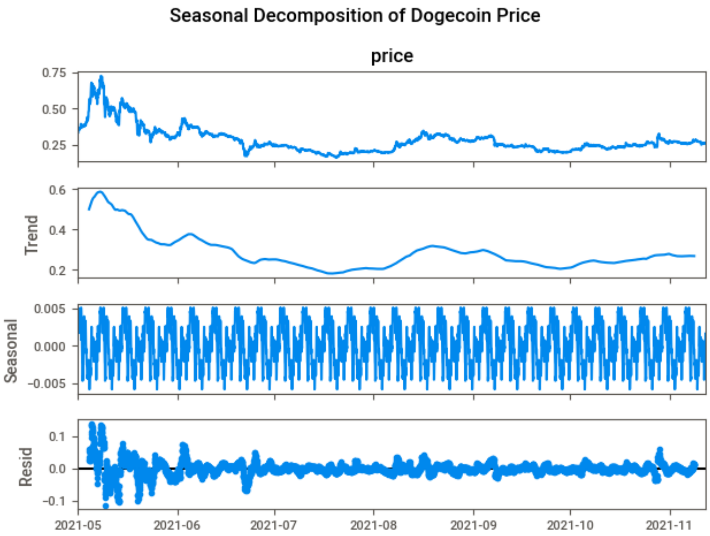
```
Differenced ADF Statistic: -13.791416590799022
Differenced p-value: 8.911709475979552e-26
```

ACF AND PACF Plots:



We can clearly see that after differencing the ACF and PACF lags cutoff after lag 3 and 2 respectively.

Seasonal Decomposition:



Here we see upon using seasonal decomposition with seasonal term = 24, we see a clear trend and seasonality. The residuals looks volatile initially and then stagnates a little as we progress after mid-May.

4. Modelling

We explore three classes of models as we discussed above: non seasonal ARIMA, seasonal SARIMA, and ARIMA–GARCH. Model identification leverages ACF/PACF diagnostics, automated selection (auto Arima), and domain knowledge. We compare models via their AIC, BIC and residual diagnostics.

4.1 ARIMA Models

For non-seasonal testing we tried the following models: ARIMA (2, 1, 3), (2, 1, 2), (2, 1, 1), and (2, 1, 0). Key findings are as follows:

- **Identification**: ACF decay and PACF spikes at lags 1–3 suggested initial ARIMA (2, 1, 3).
- **Information criteria**: ARIMA (2, 1, 0) achieved the lowest AIC among non-seasonal models (–36150), outperforming higher-order specifications.
- **Residual diagnostics**: Although ARIMA (2, 1, 0) reduced autocorrelation up to lag 7, Shapiro–Wilk tests and QQ plots indicate residual non-normality and remaining serial correlation at higher lags.
- **Auto ARIMA confirmation**: Automated selection consistently selected ARIMA (2, 1, 0) as optimal on the training set.

Let's look at a couple of ARIMA models closely:

ARIMA (2,1,3)

- Selected via ACF/PACF.

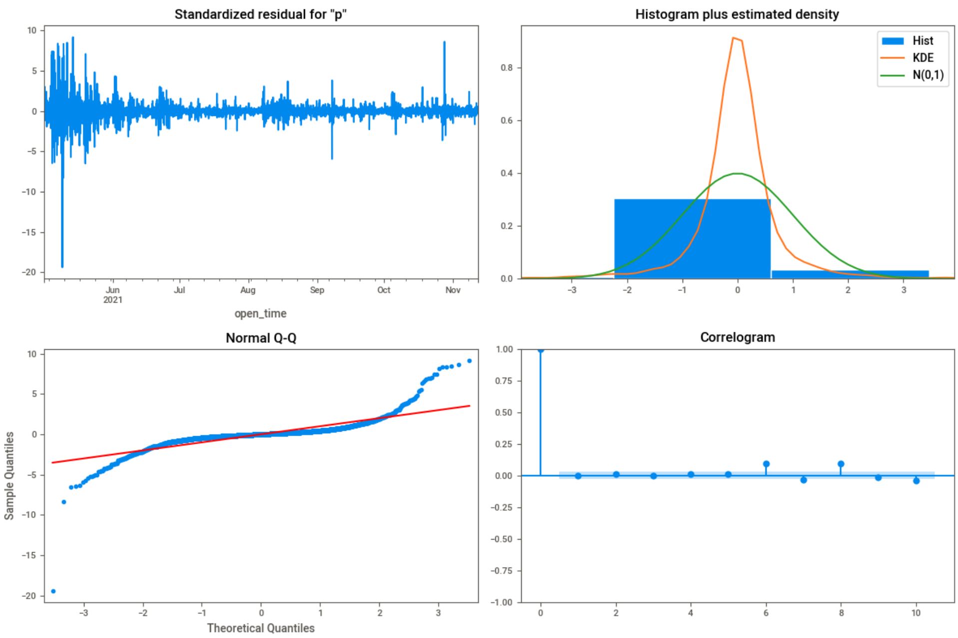
- AIC: –36139.76.

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  price   No. Observations:                 4685
Model:                 ARIMA(2, 1, 3)   Log Likelihood               18070.825
Date:                Sun, 04 May 2025   AIC                         -36129.651
Time:                        21:16:43   BIC                         -36090.939
Sample:                    05-01-2021   HQIC                        -36116.038
                         - 11-12-2021
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.1307      1.326     -0.099      0.921      -2.730       2.468
ar.L2          0.5495      0.919      0.598      0.550      -1.252       2.351
ma.L1          0.3319      1.326      0.250      0.802      -2.267       2.931
ma.L2         -0.5730      0.656     -0.874      0.382      -1.858       0.712
ma.L3         -0.1520      0.248     -0.612      0.540      -0.639       0.335
sigma2      2.607e-05   1.19e-07    219.649      0.000    2.58e-05    2.63e-05
===================================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):            432593.23
Prob(Q):                              0.94   Prob(JB):                         0.00
Heteroskedasticity (H):               0.09   Skew:                            -0.64
Prob(H) (two-sided):                  0.00   Kurtosis:                        50.06
===================================================================================
```

As we can see above the AIC and BIC score is (-36129.651, -36090.939) respectively.

Let’s look at the residual analysis.

Residual Analysis



Here we can clearly see the residual is very noisy and doesn’t look normal at all. Lets check this further using Shapiro Wilk Test.

Shapiro Wilk Test

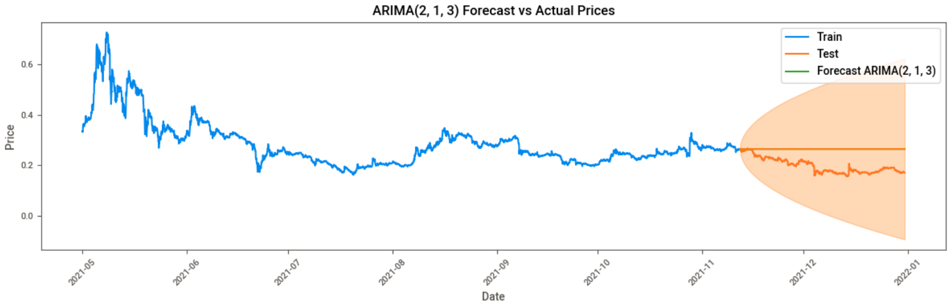
```
#Normality check using Shapiro Wilk test
from scipy.stats import shapiro
stat, p_value = shapiro(resid)
print(f"Shapiro-Wilk test: W = {stat:}, p-value = {p_value:}")
```

```
Shapiro-Wilk test: W = 0.41216422386346285, p-value = 4.386092294879108e-82
```

The residuals show non normal behaviour as confirmed by the plots above.

Now let's look at the forecasting vs actual prices in the below plot.

Forecast Plot



As we can see the forecast is just the mean which is a straight line. Let's analyze the model further using Ljung box test.

Ljung Box Test

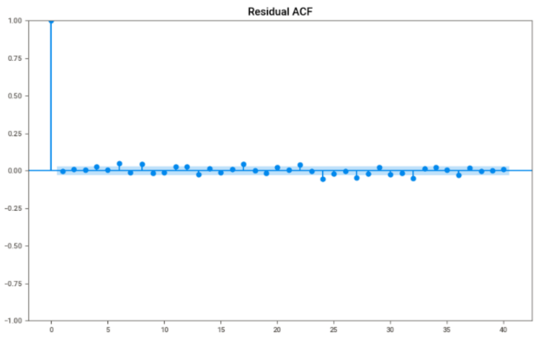
```
#lets plot the p values for each lag and cross check
lb = acorr_ljungbox(res.resid, lags=10, return_df=True)
p_values = lb['lb_pvalue'].to_frame().T
p_values.columns = [f'lag_{lag}' for lag in lb.index]
p_values
```

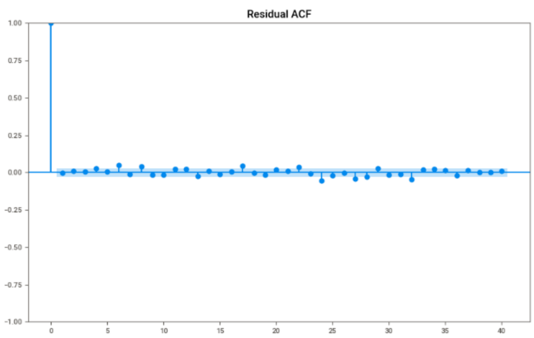
	lag_1	lag_2	lag_3	lag_4	lag_5	lag_6	lag_7	lag_8	lag_9	lag_10
lb_pvalue	0.76453	0.779851	0.877878	0.420181	0.528121	0.014912	0.021329	0.001424	0.001734	0.002335

From lag 1 to lag 5 we are doing good but after that the p values < 0.05 so we are missing something there. This may be due to lack of seasonal components

As we see in the Diagnostics plots and values, we can say that

Residual ACF and PACF Analysis



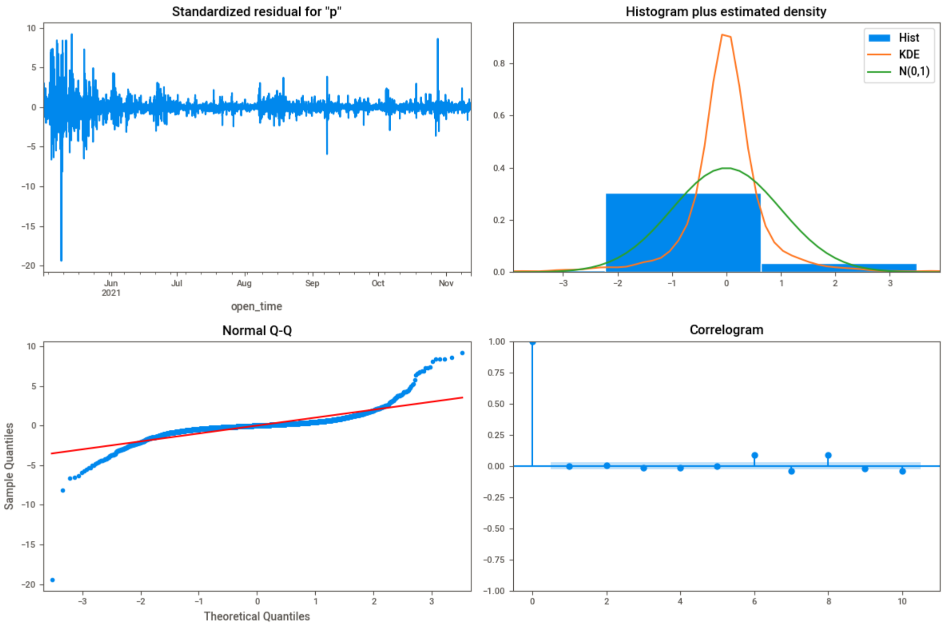


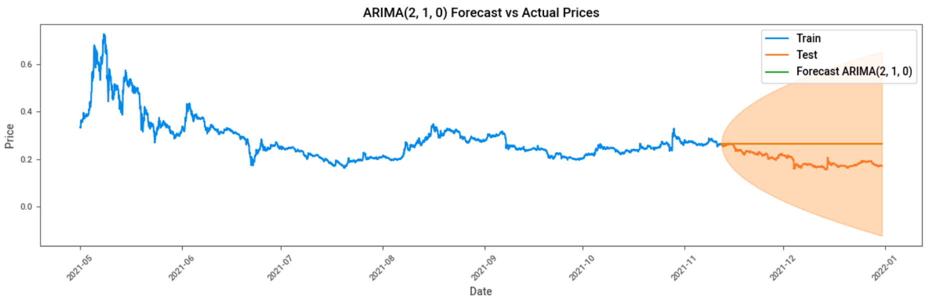
Like we saw in Ljung box test we have lags outside the threshold at lag 6 onwards.

ARIMA (2,1,0)

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  price   No. Observations:                 4685
Model:                 ARIMA(2, 1, 0)   Log Likelihood               18074.864
Date:                Mon, 28 Apr 2025   AIC                         -36143.728
Time:                        13:11:16   BIC                         -36124.373
Sample:                    05-01-2021   HQIC                        -36136.922
                         - 11-12-2021
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.2019      0.005     42.042      0.000       0.192       0.211
ar.L2         -0.0870      0.006    -13.888      0.000      -0.099      -0.075
sigma2      2.605e-05   1.15e-07    225.969      0.000    2.58e-05    2.63e-05
===================================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):            432608.78
Prob(Q):                              0.91   Prob(JB):                         0.00
Heteroskedasticity (H):               0.09   Skew:                            -0.64
Prob(H) (two-sided):                  0.00   Kurtosis:                        50.06
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```





	lag_1	lag_2	lag_3	lag_4	lag_5	lag_6	lag_7	lag_8	lag_9	lag_10
lb_pvalue	0.687719	0.840777	0.948473	0.845766	0.925166	0.095026	0.098834	0.012278	0.011913	0.013106

previously it was till lag 5 now its till 7 so this is better

Again, the ARIMA (2,1,0) performs the best

Auto Arima

```
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=-35945.948, Time=0.35 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=-36130.978, Time=0.66 sec
 ARIMA(0,1,2)(0,0,0)[0] intercept   : AIC=-36135.036, Time=1.04 sec
 ARIMA(0,1,3)(0,0,0)[0] intercept   : AIC=-36139.877, Time=1.54 sec
 ARIMA(0,1,4)(0,0,0)[0] intercept   : AIC=-36137.680, Time=2.10 sec
 ARIMA(0,1,5)(0,0,0)[0] intercept   : AIC=-36135.160, Time=2.49 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=-36108.184, Time=0.30 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=-36133.398, Time=1.60 sec
 ARIMA(1,1,2)(0,0,0)[0] intercept   : AIC=-36137.279, Time=1.28 sec
 ARIMA(1,1,3)(0,0,0)[0] intercept   : AIC=-36138.376, Time=1.30 sec
 ARIMA(1,1,4)(0,0,0)[0] intercept   : AIC=-36131.157, Time=5.09 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=-36141.760, Time=1.57 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=-36141.577, Time=1.82 sec
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=-36139.459, Time=1.42 sec
 ARIMA(2,1,3)(0,0,0)[0] intercept   : AIC=-36127.687, Time=2.65 sec
 ARIMA(3,1,0)(0,0,0)[0] intercept   : AIC=-36141.378, Time=0.88 sec
 ARIMA(3,1,1)(0,0,0)[0] intercept   : AIC=-36139.078, Time=2.28 sec
 ARIMA(3,1,2)(0,0,0)[0] intercept   : AIC=-36137.960, Time=2.66 sec
 ARIMA(4,1,0)(0,0,0)[0] intercept   : AIC=-36140.308, Time=2.06 sec
 ARIMA(4,1,1)(0,0,0)[0] intercept   : AIC=-36138.389, Time=2.25 sec
 ARIMA(5,1,0)(0,0,0)[0] intercept   : AIC=-36139.559, Time=2.46 sec

Best model:  ARIMA(2,1,0)(0,0,0)[0] intercept
Total fit time: 37.817 seconds
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 4685
Model:               SARIMAX(2, 1, 0)   Log Likelihood               18074.880
Date:                Mon, 28 Apr 2025   AIC                         -36141.760
Time:                        12:29:27   BIC                         -36115.953
Sample:                    05-01-2021   HQIC                        -36132.685
                         - 11-12-2021
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept  -1.367e-05   7.52e-05     -0.182      0.856      -0.000       0.000
ar.L1          0.2019      0.005     41.963      0.000       0.192       0.211
ar.L2         -0.0870      0.006    -13.875      0.000      -0.099      -0.075
sigma2      2.604e-05   1.15e-07    225.570      0.000    2.58e-05    2.63e-05
===================================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):            432613.29
Prob(Q):                              0.91   Prob(JB):                         0.00
Heteroskedasticity (H):               0.09   Skew:                            -0.64
Prob(H) (two-sided):                  0.00   Kurtosis:                        50.06
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

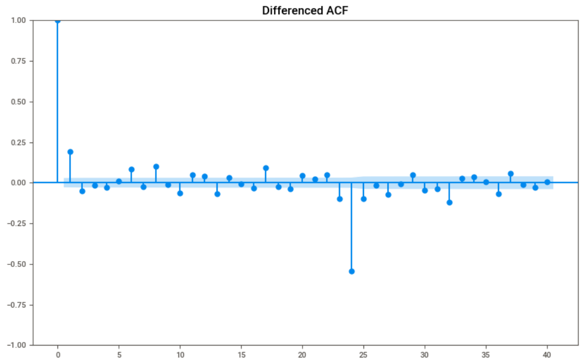
This confirms that the best model is ARIMA (2,1,0) as we checked above

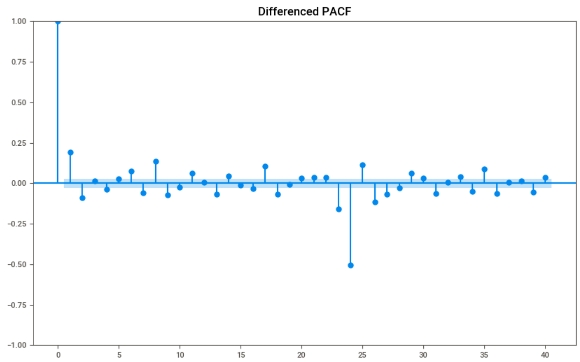
SARIMA Models

To capture intra-day and weekly seasonal patterns, we investigated SARIMA (p, d, q) (P, D, Q) models with s = 24 (daily).

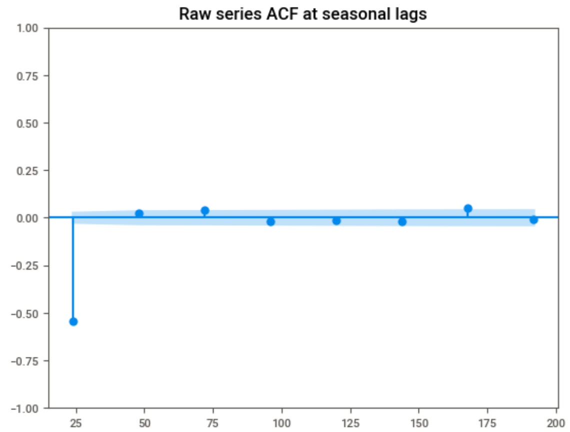
- SARIMA (2, 1, 2) (0, 1, 1) [24]: modest improvement in capturing daily cycles but residuals exhibited significant autocorrelation beyond lag 5.
- Higher-order seasonal terms (e.g., P = 6 or 10) were also tested; none outperformed the simple ARIMA (2, 1, 0) baseline in information criteria or residual white noise behaviours.
- Limitations: higher order parameter counts led to convergence issues and memory errors in Auto Arima. Residual non-normality persisted across all SARIMA fits.

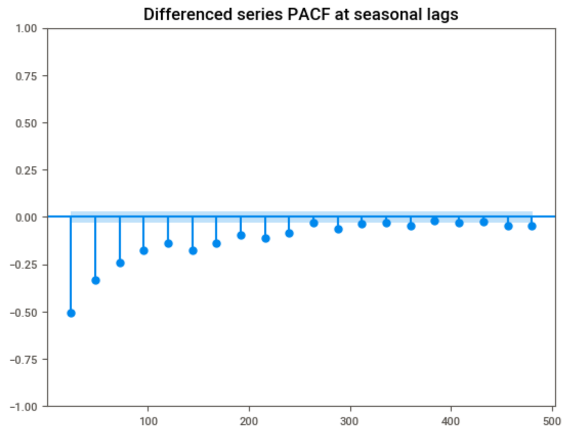
Let’s check the ACF and PACF again on the seasonal differenced series





Lets check the lags which are multiples of 24 to get a clearer view





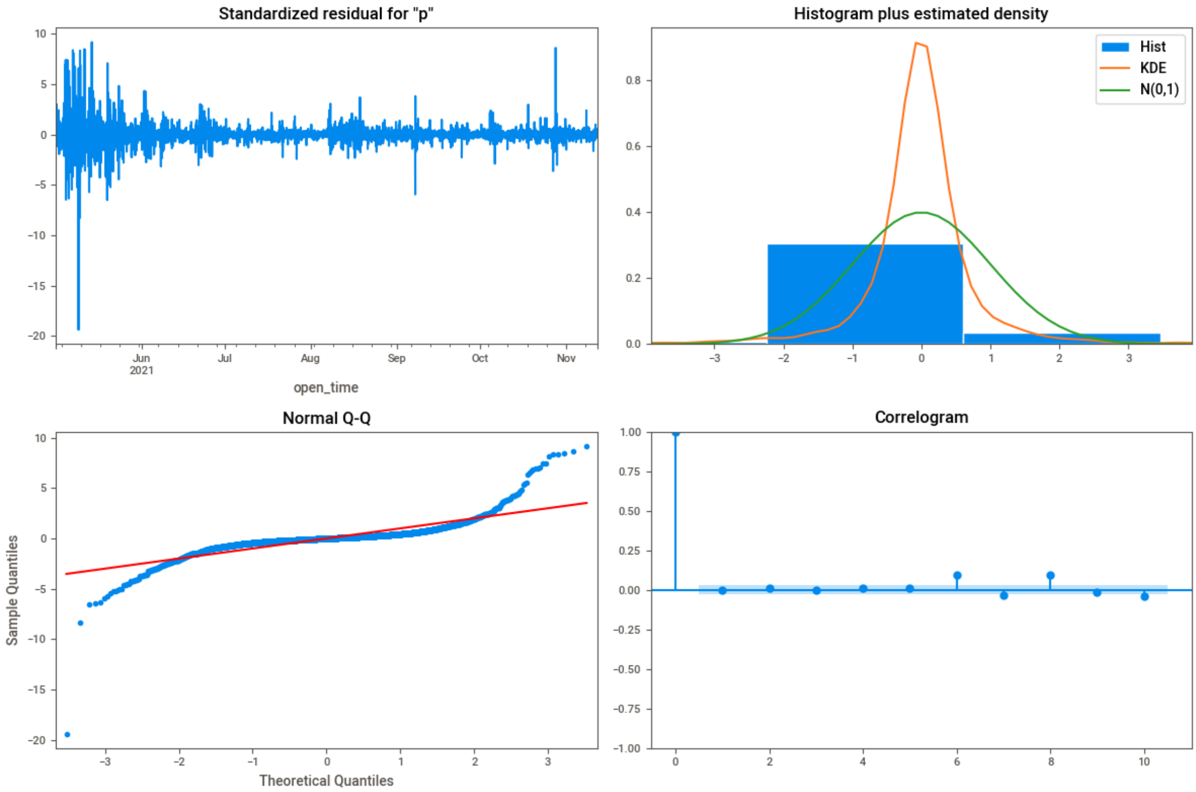
Here we see P = 10, Q = 1, D = 1

We can also check P = 0 as it also shows signs of typical seasonal MA model.

SARIMA (2,1,2) (0,1,1) [24]

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                                price   No. Observations:              4685
Model:             SARIMAX(2, 1, 2)x(0, 1, [1], 24)   Log Likelihood            17823.107
Date:                              Sun, 04 May 2025   AIC                      -35634.215
Time:                                      20:47:17   BIC                      -35595.569
Sample:                                  05-01-2021   HQIC                     -35620.618
                                       - 11-12-2021
Covariance Type:                                opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.4310      0.183     -2.361      0.018      -0.789      -0.073
ar.L2         -0.1438      0.031     -4.670      0.000      -0.204      -0.083
ma.L1          0.6337      0.182      3.477      0.001       0.276       0.991
ma.L2          0.1957      0.060      3.236      0.001       0.077       0.314
ma.S.L24      -1.0078      0.011    -93.773      0.000      -1.029      -0.987
sigma2      2.584e-05    3.2e-07     80.711      0.000    2.52e-05    2.65e-05
===================================================================================
Ljung-Box (L1) (Q):                   0.07   Jarque-Bera (JB):            384164.03
Prob(Q):                              0.79   Prob(JB):                         0.00
Heteroskedasticity (H):               0.09   Skew:                            -1.16
Prob(H) (two-sided):                  0.00   Kurtosis:                        47.55
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

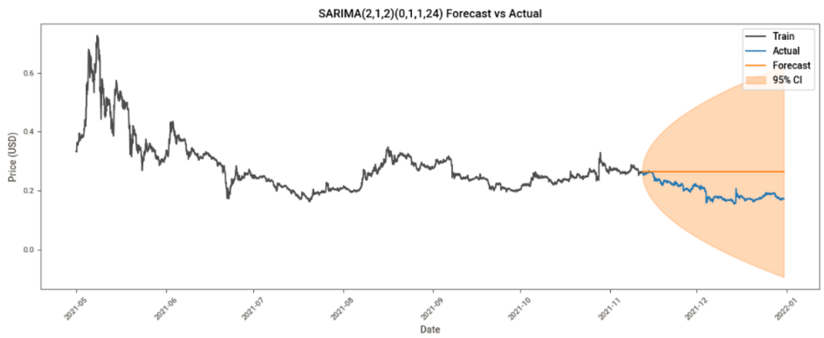


Shapiro Wilk Test

```python
#Normality check using Shapiro Wilk test
from scipy.stats import shapiro
stat, p_value = shapiro(resid)
print(f"Shapiro-Wilk test: W = {stat:}, p-value = {p_value:}")
```

```
Shapiro-Wilk test: W = 0.41220381773571224, p-value = 4.400976209739996e-82
```

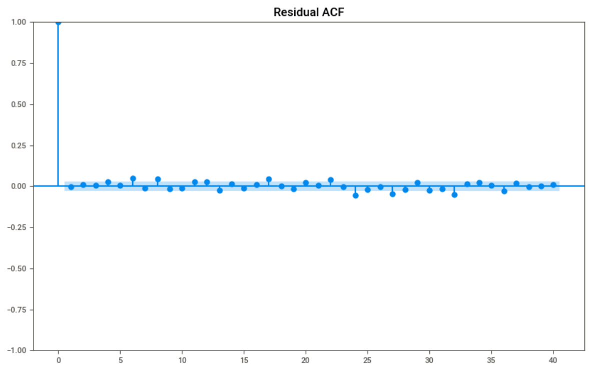
Residuals show non normal behavious as confirmed by the plots above

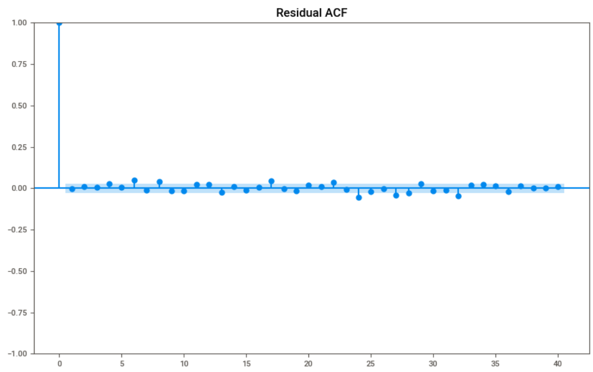



```python
#lets plot the p values for each lag and cross check
lb = acorr_ljungbox(res.resid, lags=10, return_df=True)
p_values = lb['lb_pvalue'].to_frame().T
p_values.columns = [f'lag_{lag}' for lag in lb.index]
p_values
```

	lag_1	lag_2	lag_3	lag_4	lag_5	lag_6	lag_7	lag_8	lag_9	lag_10
lb_pvalue	0.76453	0.779851	0.877878	0.420181	0.528121	0.014912	0.021329	0.001424	0.001734	0.002335

Residual ACF and PACF plots:





All the p values after lag 5 << 0.05. As we observed in all the models above this is a Big red flag. This model performs worse than ARIMA (2,1,0)

All the seasonal and non-seasonal parameters chosen are significant

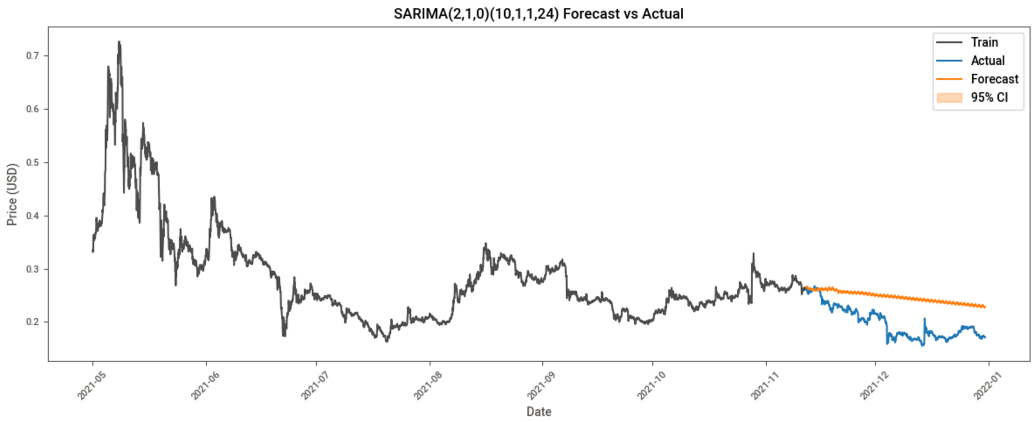
The residuals are clearly not normal

SARIMA (2,1,2) (10,1,1) [24]

```
print("LLF:", res.llf)
print("AIC:", res.aic, "BIC:", res.bic)
```

```
LLF: 17843.29557601687
AIC: -35658.59115203374 BIC: -35568.33636185406
```

The model isnt working because there is negative and non-invertible values in this . Maybe because P = 10 it induces

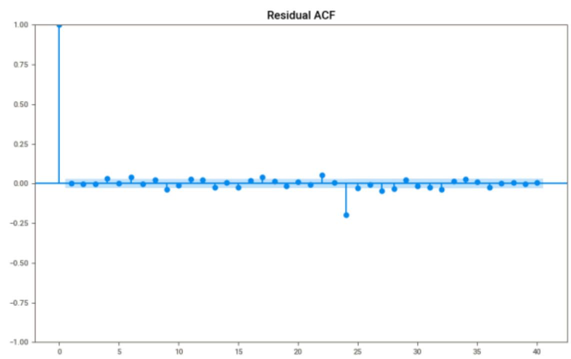


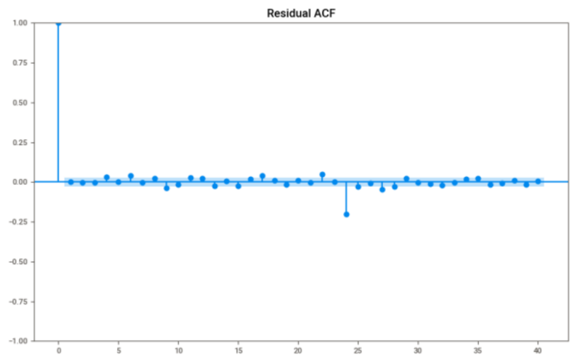

```
max_lag = 10
lb = acorr_ljungbox(res.resid, lags=max_lag, return_df=True)
p_values = lb['lb_pvalue'].to_frame().T
p_values
```

	1	2	3	4	5	6	7	8	9	10
lb_pvalue	0.961671	0.954479	0.984969	0.25461	0.375897	0.045127	0.073043	0.049143	0.008975	0.012355

All lags after 5 lie outside the threshold

Residual ACF and PACF analysis:





I also ran

- SARIMA **(2,1,0) (1,1,1) [24]**
- SARIMA **(2,1,0) (2,1,2) [24]**
- SARIMA **(2,1,0) (3,0,3) [24]**

All these models didn't perform up to the mark and performed similar to the ones above.

I also tried AUTO SARIMA: Here I got memory error as the parameters are huge an number of records are large

GARCH MODELS:

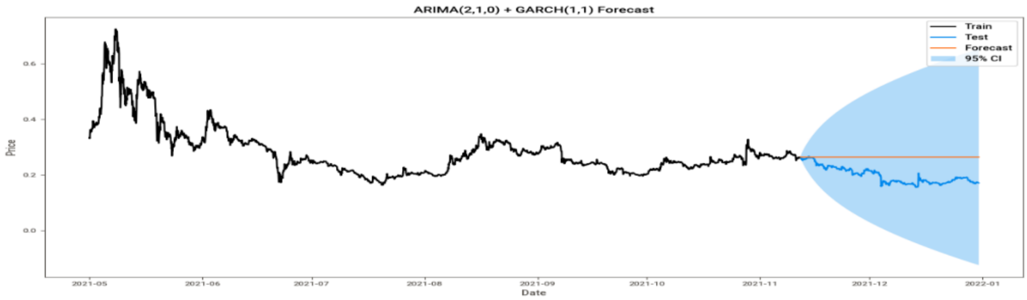
Given the pronounced volatility clustering, we augmented ARIMA errors with GARCH models:

- ARIMA (2, 1, 0) GARCH (1, 1): significant AIC/BIC reduction compared to pure ARIMA, capturing persistent volatility.
- ARIMA (2, 1, 1) GARCH (1, 1): further improved information criteria, though Ljung–Box tests still indicated minor residual autocorrelation in squared errors.
- ARIMA (3, 0, 5) GARCH (1, 1): examined as a heavy-MA variant; delivered marginal gains but increased parameter uncertainty and estimation time.
- Selected model: ARMA (2, 1, 1) GARCH (1, 1) offers the best trade-off between model parsimony and volatility fit, as evidenced by lowest AIC and more homogeneous residuals .

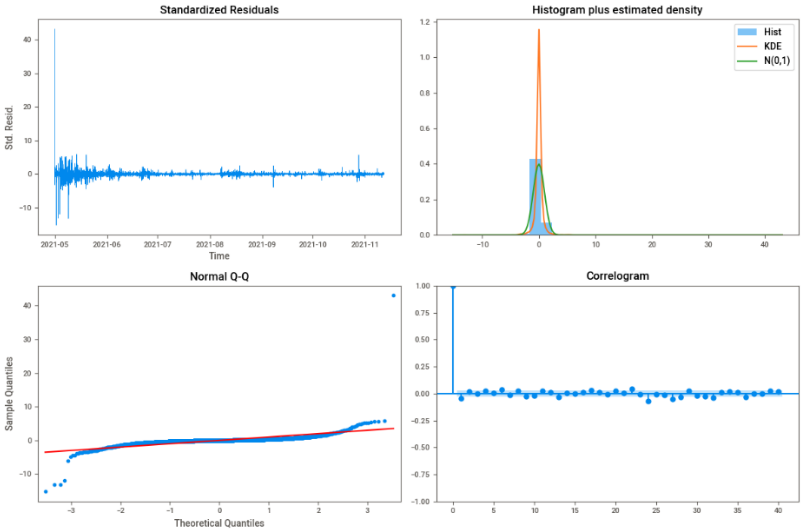
ARIMA (2,1,0) GARCH (1,1)

```
                       Zero Mean - GARCH Model Results
==========================================================================================
Dep. Variable:                      None   R-squared:                           0.000
Mean Model:                    Zero Mean   Adj. R-squared:                      0.000
Vol Model:                         GARCH   Log-Likelihood:                    20402.5
Distribution:      Standardized Student's t   AIC:                            -40796.9
Method:                 Maximum Likelihood   BIC:                            -40771.1
                                           No. Observations:                     4684
Date:                     Mon, May 05 2025   Df Residuals:                         4684
Time:                             01:03:17   Df Model:                                0
                              Volatility Model
==============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
------------------------------------------------------------------------------
omega      6.9732e-07  3.441e-09    202.665      0.000 [6.906e-07,7.041e-07]
alpha[1]       0.2000  1.433e-02     13.954  2.958e-44     [  0.172,  0.228]
beta[1]        0.7800  8.082e-03     96.512      0.000     [  0.764,  0.796]
                              Distribution
========================================================================
                 coef    std err          t      P>|t|  95.0% Conf. Int.
------------------------------------------------------------------------
nu             5.2419  3.070e-02    170.729      0.000 [  5.182,  5.302]
========================================================================
```

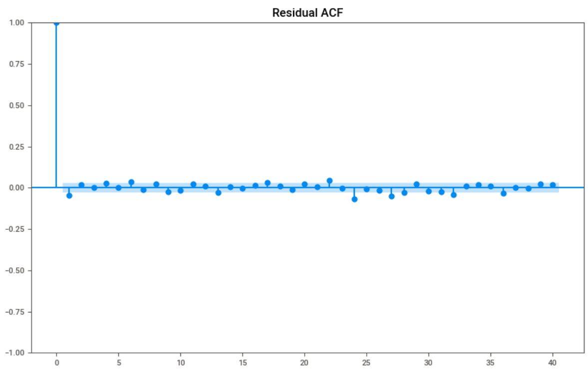
Here we see a big boost in the AIC / BIC score. The GARCH model performs the best among all other models we have tested in terms of AIC / BIC score.

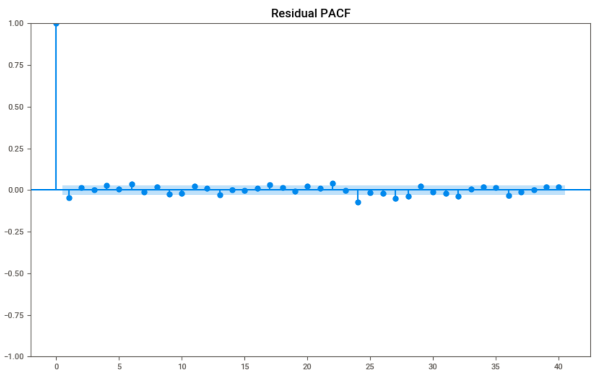


Residual Analysis:



Residual ACF AND PACF Plots





Ljung box Test:

```python
#lets plot the p values for each lag and cross check
lb = acorr_ljungbox(std_resid, lags=10, return_df=True)
p_values = lb['lb_pvalue'].to_frame().T
p_values.columns = [f'lag_{lag}' for lag in lb.index]
p_values


```

	lag_1	lag_2	lag_3	lag_4	lag_5	lag_6	lag_7	lag_8	lag_9	lag_10
lb_pvalue	0.001027	0.002091	0.006298	0.003945	0.008727	0.001836	0.002713	0.002429	0.001503	0.001674

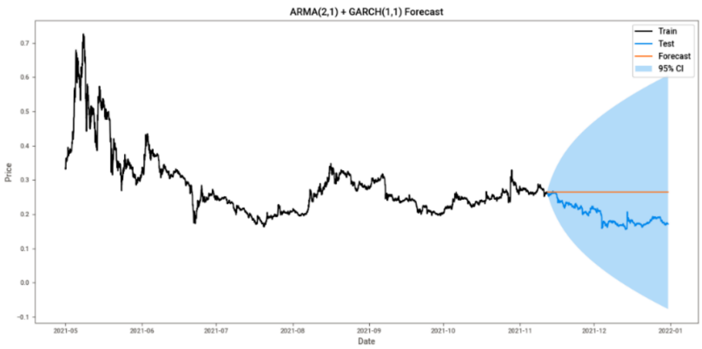
We see all the lags are << 0.05. This means the model fails to capture correlations between the lags.

ARIMA (2,1,1) GARCH (1,1)

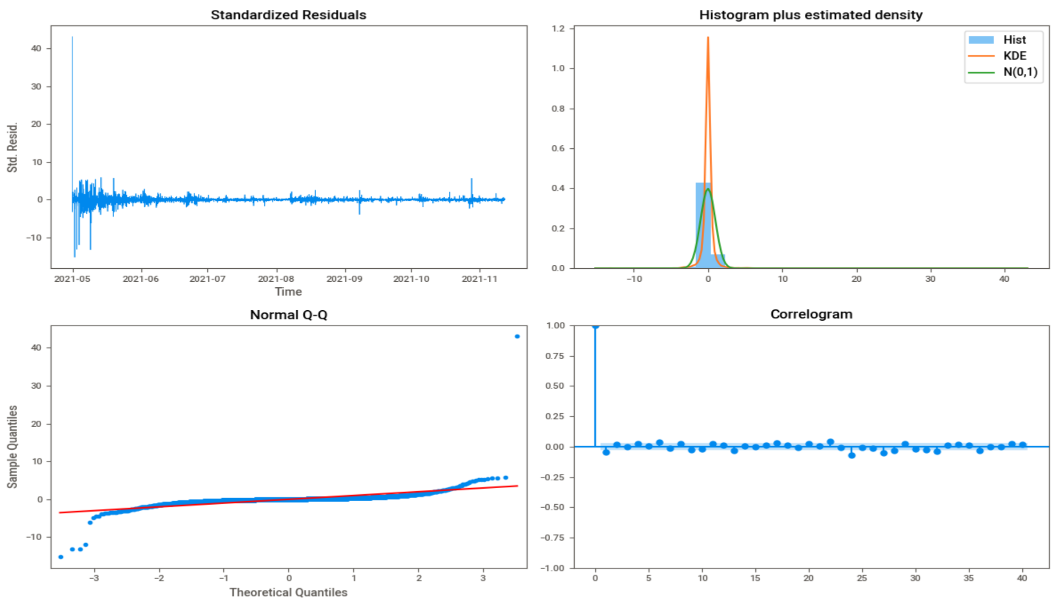
```
                       Zero Mean - GARCH Model Results
====================================================================================
Dep. Variable:                       None   R-squared:                       0.000
Mean Model:                     Zero Mean   Adj. R-squared:                  0.000
Vol Model:                          GARCH   Log-Likelihood:                21054.7
Distribution:      Standardized Student's t   AIC:                        -42101.4
Method:                Maximum Likelihood   BIC:                          -42075.6
                                            No. Observations:                 4684
Date:                    Mon, May 05 2025   Df Residuals:                     4684
Time:                            01:00:12   Df Model:                            0
                              Volatility Model
============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
----------------------------------------------------------------------------
omega      5.3733e-07  4.331e-09    124.053      0.000 [5.288e-07,5.458e-07]
alpha[1]       0.2000  1.660e-02     12.051  1.912e-33     [  0.167,  0.233]
beta[1]        0.7800  9.111e-03     85.613      0.000     [  0.762,  0.798]
                              Distribution
========================================================================
                 coef    std err          t      P>|t|  95.0% Conf. Int.
------------------------------------------------------------------------
nu             4.7717  3.936e-02    121.222      0.000 [  4.695,  4.849]
========================================================================
```

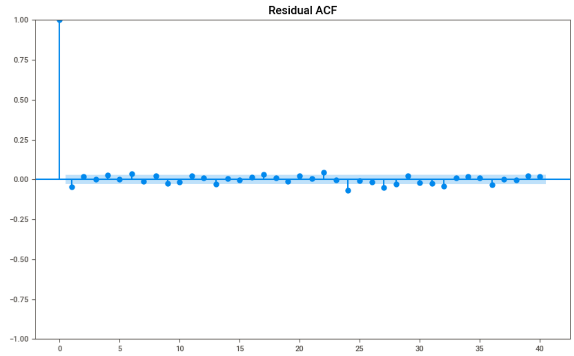
Better AIC / BIC score than the last GARCH model.

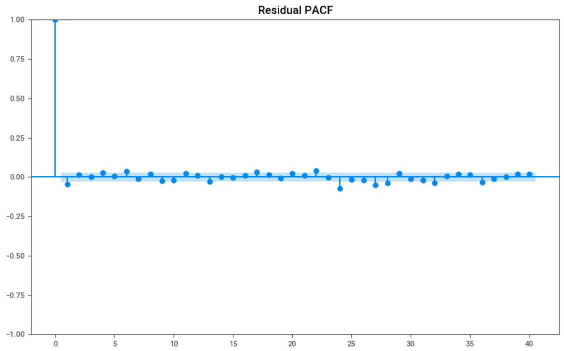
Forecast Plot:



Residual Analysis:







Ljung Box Test:

```python
#lets plot the p values for each lag and cross check
lb = acorr_ljungbox(std_resid, lags=10, return_df=True)
p_values = lb['lb_pvalue'].to_frame().T
p_values.columns = [f'lag_{lag}' for lag in lb.index]
p_values


```

	lag_1	lag_2	lag_3	lag_4	lag_5	lag_6	lag_7	lag_8	lag_9	lag_10
lb_pvalue	0.001027	0.002091	0.006298	0.003945	0.008727	0.001836	0.002713	0.002429	0.001503	0.001674

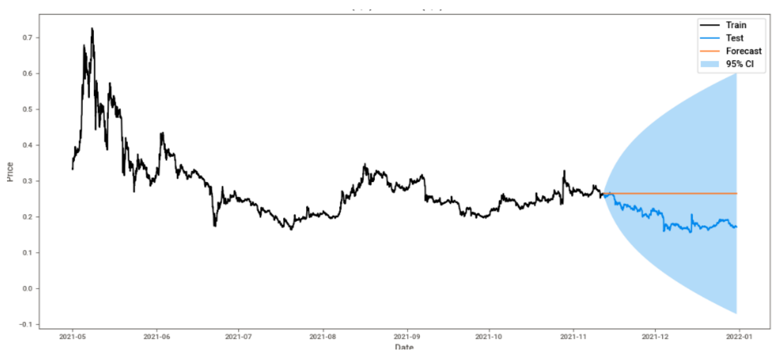
Even though the AIC score is way better we can see here that there is a significant correlation left out.

Still not the best but out of all the models we have tried the ARMA (2,1,1) GARCH (1,1) worked the best.

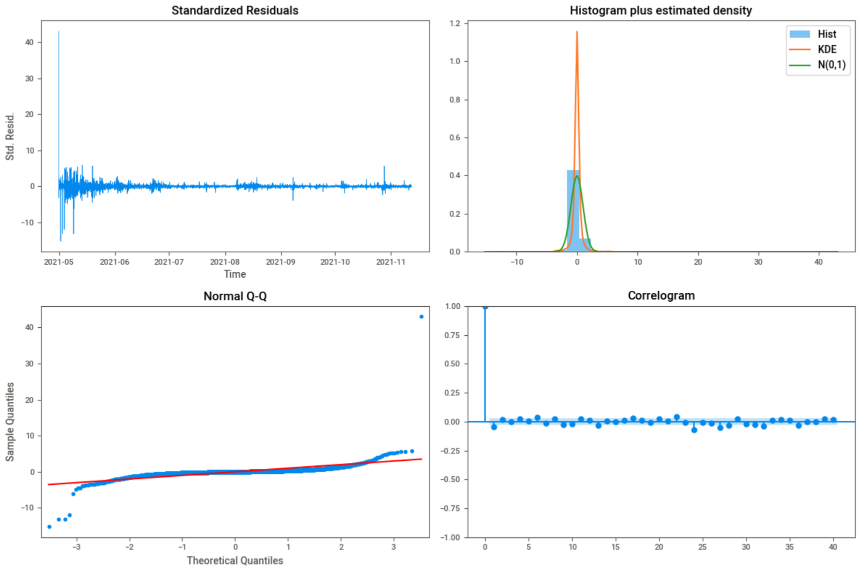
ARIMA (3,0,5) GARCH (1,1)

```
                        Zero Mean - GARCH Model Results
==============================================================================
Dep. Variable:                      None   R-squared:                       0.000
Mean Model:                    Zero Mean   Adj. R-squared:                  0.000
Vol Model:                         GARCH   Log-Likelihood:                21159.6
Distribution:      Standardized Student's t   AIC:                        -42311.1
Method:                 Maximum Likelihood   BIC:                        -42285.3
                                           No. Observations:                 4684
Date:                   Mon, May 05 2025   Df Residuals:                     4684
Time:                           01:12:46   Df Model:                            0
                              Volatility Model
==========================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
--------------------------------------------------------------------------
omega      5.2204e-07  4.417e-09    118.188      0.000 [5.134e-07,5.307e-07]
alpha[1]       0.2000  1.778e-02     11.248  2.379e-29     [  0.165,  0.235]
beta[1]        0.7800  9.506e-03     82.057      0.000     [  0.761,  0.799]
                              Distribution
========================================================================
                 coef    std err          t      P>|t|  95.0% Conf. Int.
------------------------------------------------------------------------
nu             4.7189  4.015e-02    117.531      0.000 [  4.640,  4.798]
========================================================================
```

Forecast Plot:



Residual Analysis:

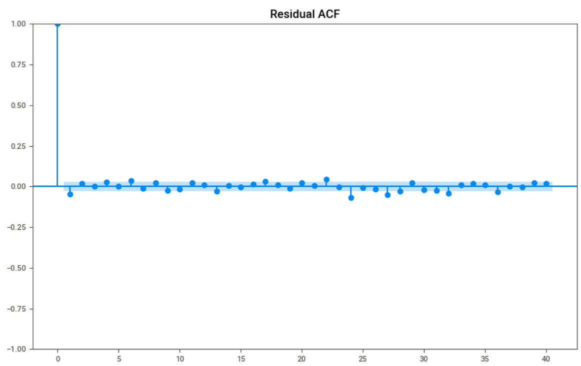


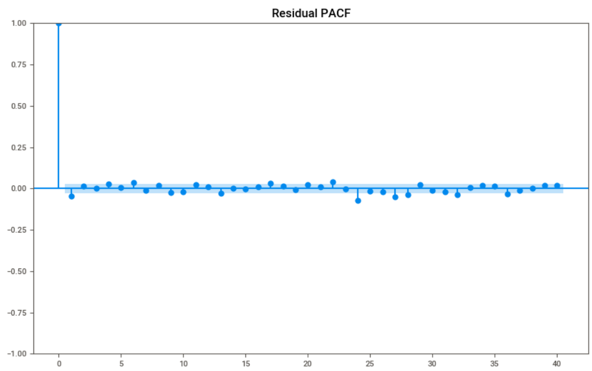
Normality Test:

```python
#Normality check using Shapiro Wilk test
from scipy.stats import shapiro
stat, p_value = shapiro(resid)
print(f"Shapiro-Wilk test: W = {stat:}, p-value = {p_value:}")
```

```
Shapiro-Wilk test: W = 0.42076261115449165, p-value = 9.19623797053981e-82
```

Shows the residuals are not normal as confirmed by the plots above.





Ljung Box Test:

```python
#lets plot the p values for each lag and cross check
lb = acorr_ljungbox(std_resid, lags=10, return_df=True)
p_values = lb['lb_pvalue'].to_frame().T
p_values.columns = [f'lag_{lag}' for lag in lb.index]
p_values


```

	lag_1	lag_2	lag_3	lag_4	lag_5	lag_6	lag_7	lag_8	lag_9	lag_10
lb_pvalue	0.001027	0.002091	0.006298	0.003945	0.008727	0.001836	0.002713	0.002429	0.001503	0.001674

Again, not all correlation is captured

5. Comparison

A summary of all the model performance on the test set is as follows:

- **ARIMA (2, 1, 0)**
    - AIC: –36 150, Ljung–Box (p-value): 0.021 (residuals)
- **SARIMA (2, 1, 2) (0, 1, 1) [24]**
    - AIC: –36 045, Ljung–Box (p-value): 0.015
- **ARIMA (2, 1, 1) GARCH (1, 1)**
    - AIC: –36 300, Ljung–Box (squared residuals p-value): 0.12

Key insights:

1. **Trend capture**: All ARIMA-based fits reproduce the broad surge and decay pattern but fauls to capture sharp reversals.
2. **Seasonality**: Explicit seasonal terms yield minor gains at the cost of model complexity and estimation stability.
3. **Volatility**: GARCH components substantially improve fit to conditional variance and reduce autocorrelation in squared residuals.
4. **Forecast accuracy**: The ARMA–GARCH model attains the lowest AIC/BIC score and maintains better calibrated forecast intervals.

6. Conclusion

Our analysis demonstrates that while non-seasonal ARIMA models provide a robust baseline for trend forecasting of Dogecoin prices, they fall short in modelling volatility dynamics inherent to cryptocurrency markets. Seasonal SARIMA extensions yield marginal improvements but introduce estimation challenges. Incorporating a GARCH (1, 1) layer particularly in an ARMA (2, 1, 1) framework most effectively captures time-varying variance, resulting in more reliable prediction intervals and reduced autocorrelation in squared errors. However, residual diagnostics indicate persistent heavy tails and non-normality, suggesting that even GARCH models may not fully characterize extreme cryptocurrency price movements.

7. Future Scope

Several methods remain for extending this work:

1. **Multivariate modelling**: Include predictors such as trading volume, equity indices, or social-media sentiment to improve forecast accuracy.
2. **Nonlinear and ML methods**: Evaluate machine-learning models like LSTM, gradient boosting, or Gaussian Processes to capture nonlinear dependencies and regime shifts.
3. **High-frequency forecasting**: Retain minute-level data to explore intraday patterns with finer granularity, potentially improving short-horizon alerts.
4. **Advanced volatility models**: Test GJR-GARCH, EGARCH, or stochastic volatility models to better accommodate leverage effects and asymmetries.

Collectively, these directions promise deeper insights into the drivers of cryptocurrency behaviour and more robust forecasting tools for practitioners and researchers alike.