CAPSTONE PROJECT 3 ‘PREDICTING BITCOIN PRICE’

SPRINGBOARD – DSC

By Igor Verevkin

August 2021

PROBLEM IDENTIFICATION

BITCOIN

- First and largest cryptocurrency
- $40,000 per one BTC on August 1, 2021
- $787 billions total market capitalization
- Extremely volatile

GOAL

- Build predictive models using time series forecasting algorithms
- Predict Bitcoin price for one month

DATA ACQUISITION

SOURCE

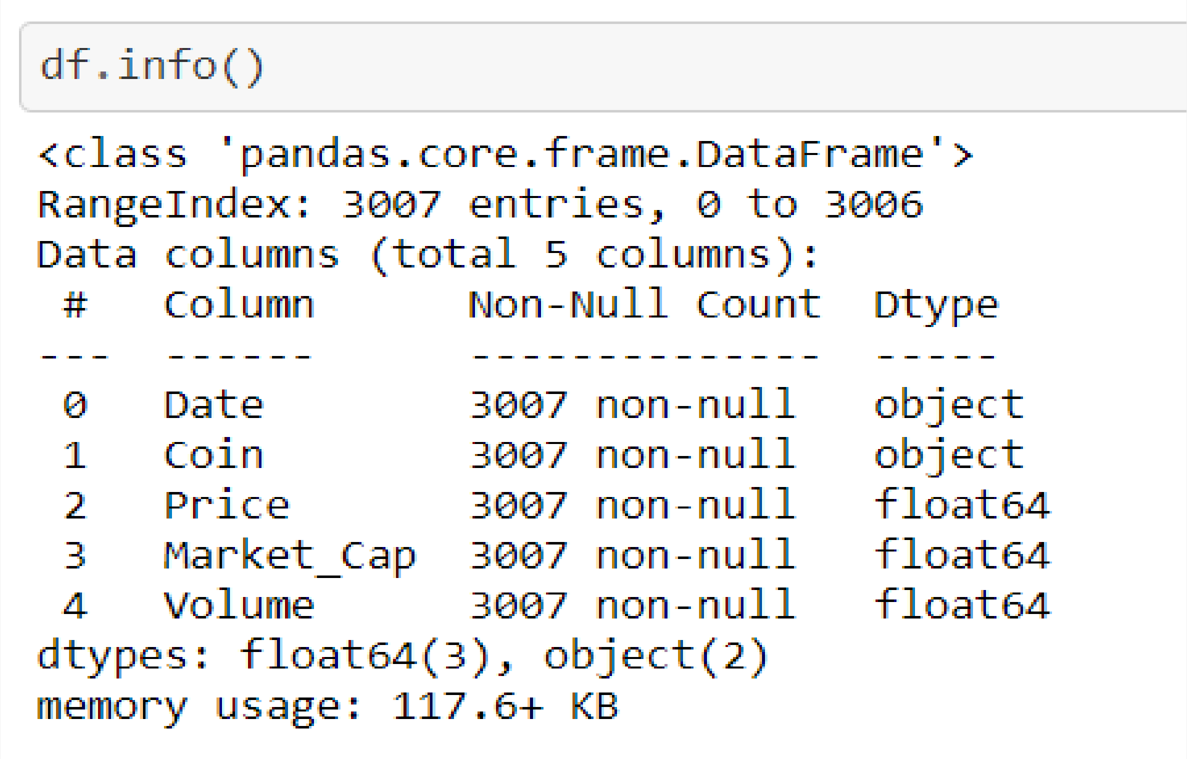
- CoinGecko cryptocurrency exchange

DATA WRANGLING

- Dataset
  - Observations: Bitcoin’s daily performance from April 2013 to July 2021
  - Target Feature: Bitcoin’s price
- No missing values

DATA INFO

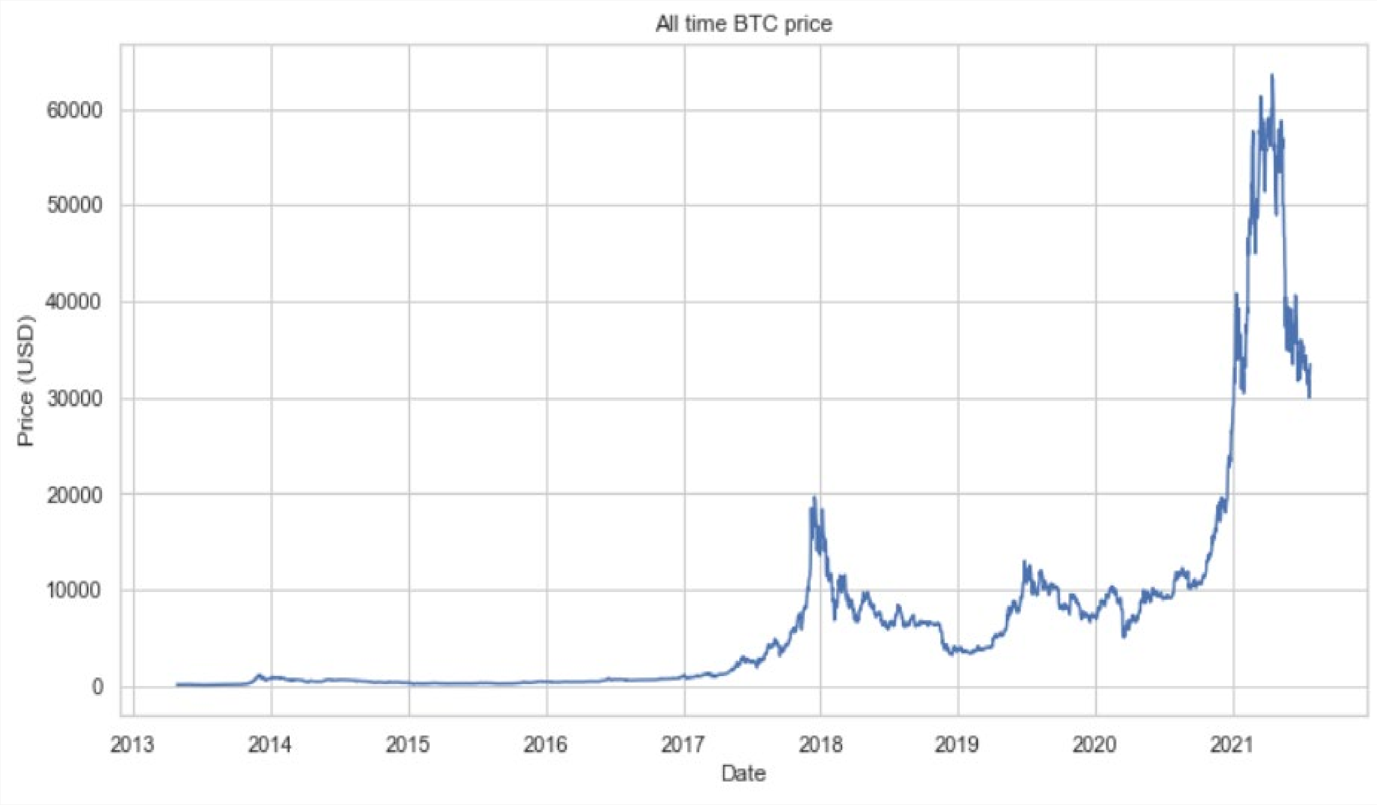
- We have 3007 observations and 5 features



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3007 entries, 0 to 3006
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Date        3007 non-null   object
 1   Coin        3007 non-null   object
 2   Price       3007 non-null   float64
 3   Market_Cap  3007 non-null   float64
 4   Volume      3007 non-null   float64
dtypes: float64(3), object(2)
memory usage: 117.6+ KB
```

BITCOIN'S PRICE CHART 2013-2021



EXPLORATORY DATA ANALYSIS

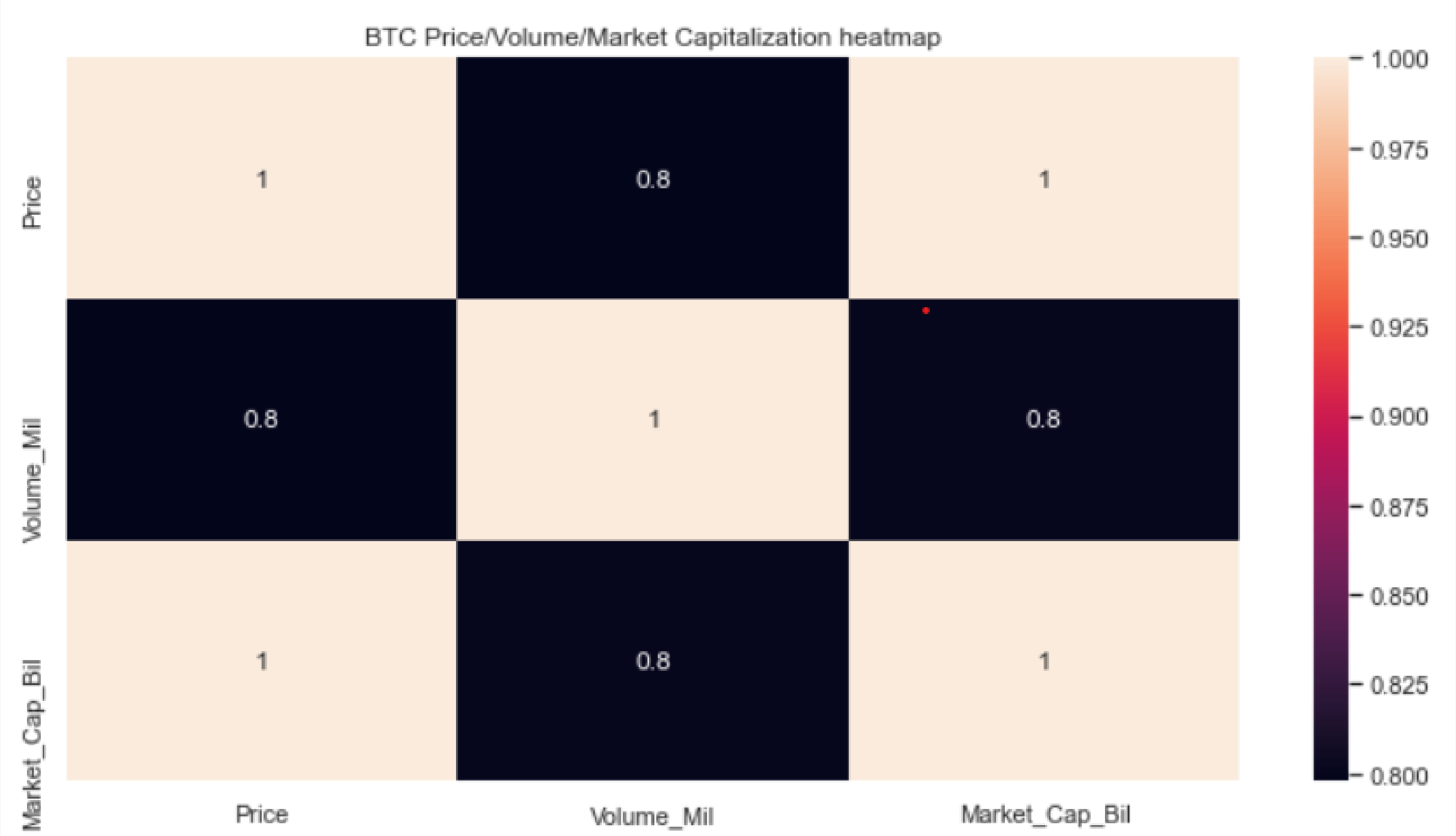
QUESTIONS

- Feature-to-target correlation?
- Is time series a random walk?
- Is time series stationary?
- Is time series autocorrelated?

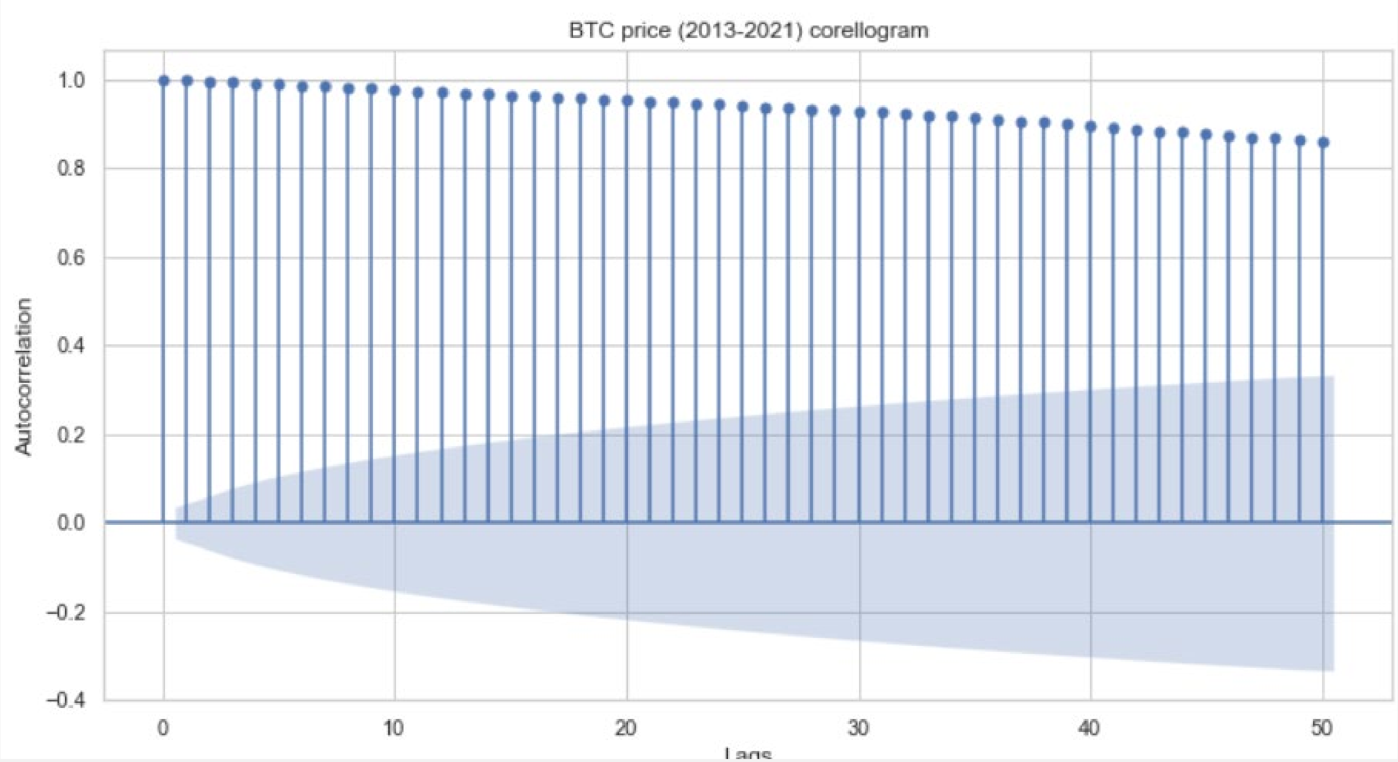
RANDOM WALK

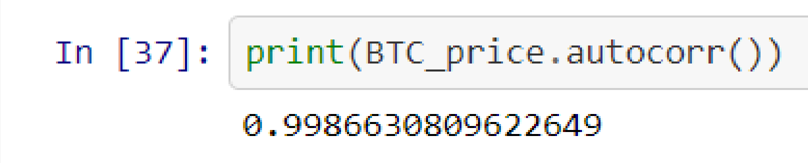
- Random Walk - mathematical object, known as a stochastic or random process, that describes a path that consists of a succession of random steps on some mathematical space such as the integers.
- In other words, we would like to find out if the price is random or not, because if it is indeed random then there is no way we can predict it.

FEATURE CORRELATION



AUTOCORRELATION





```
In [37]: print(BTC_price.autocorr())

         0.9986630809622649
```

TESTING FOR STATIONARITY

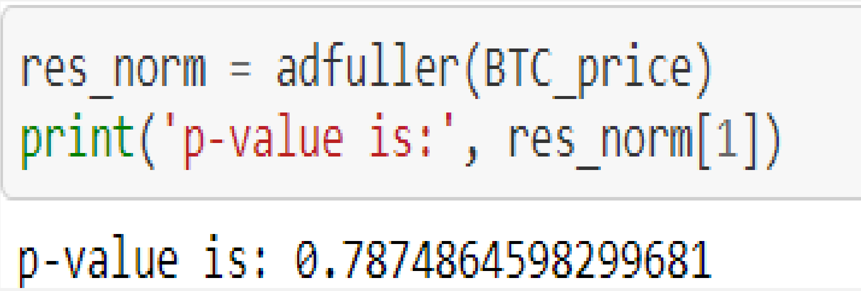
- Null hypothesis: Non-Stationarity exists in BTC Price
- Alternative hypothesis: Stationarity exists in BTC Price

AUGMENTED DICKEY-FULLER TEST

- Helps to test if the process is stationary
- The more negative the ADF statistic, the more confidently we can reject Null hypothesis

AUGMENTED DICKEY-FULLER TEST

- P-value is very high with respect to the threshold of 0.05
- Can not confidently reject null hypothesis
- BTC price is NOT stationary



```
res_norm = adfuller(BTC_price)
print('p-value is:', res_norm[1])
```

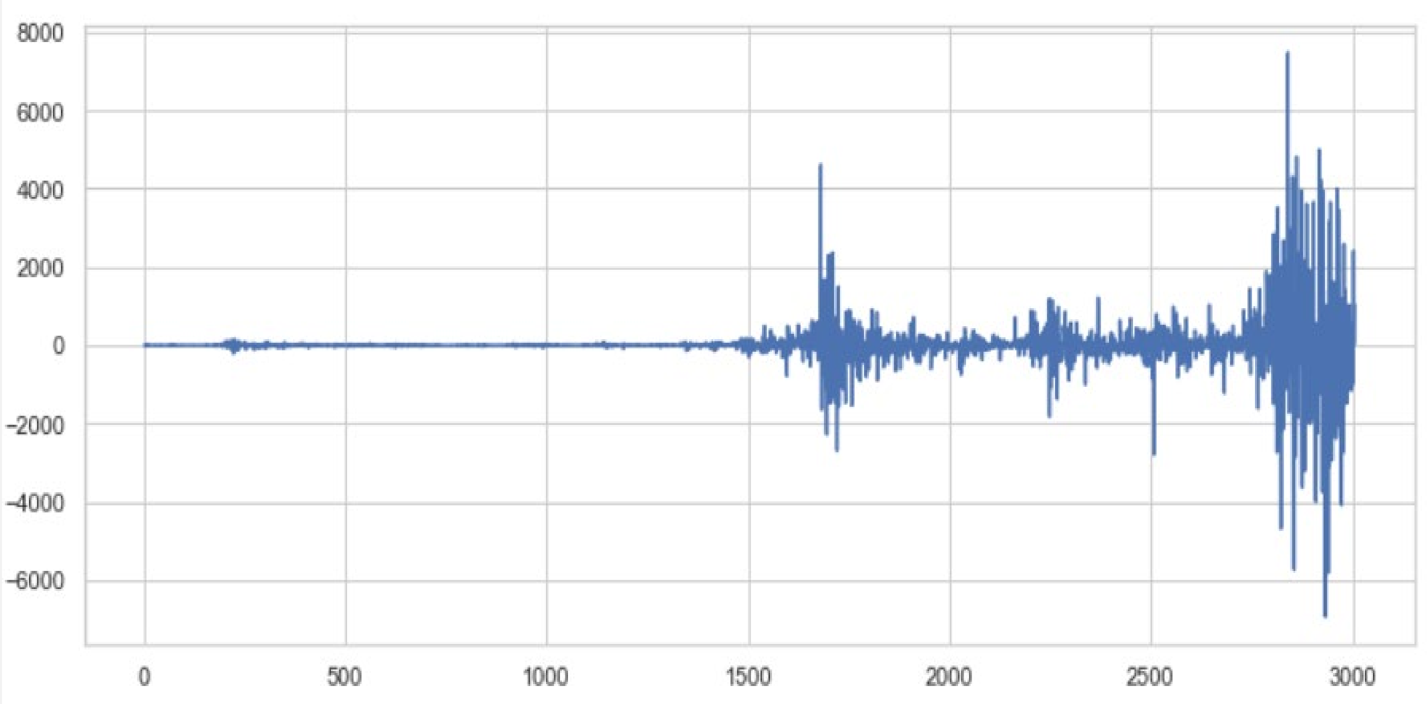
```
p-value is: 0.7874864598299681
```

BASELINE MODELING

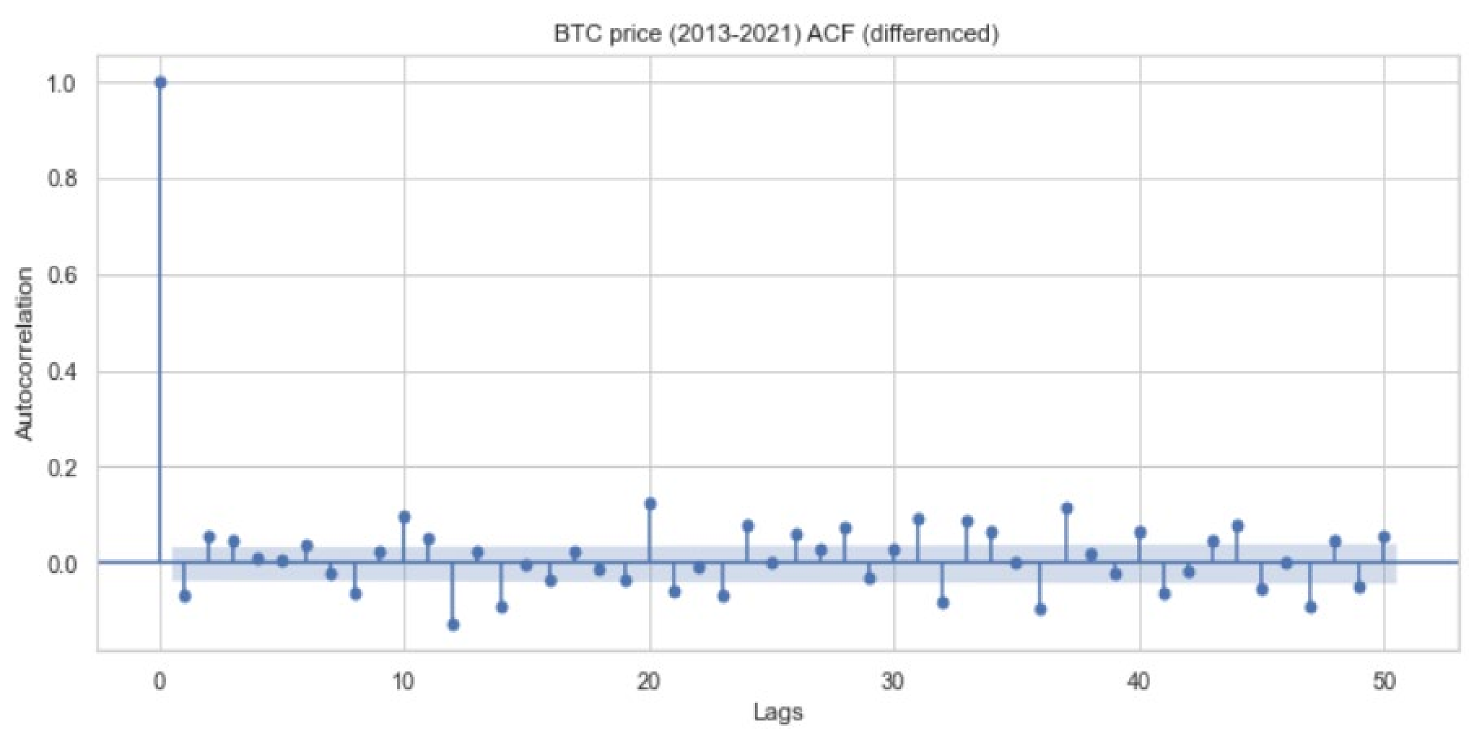
- Problem – time series forecasting (regression)
- Baseline model – ARIMA
- ARIMA order derived from ACF, PACF and differentiation
- Performance metrics – Mean Absolute Percent Error (MAPE), r-squared
- Goal – minimize MAPE

DIFFERENCING

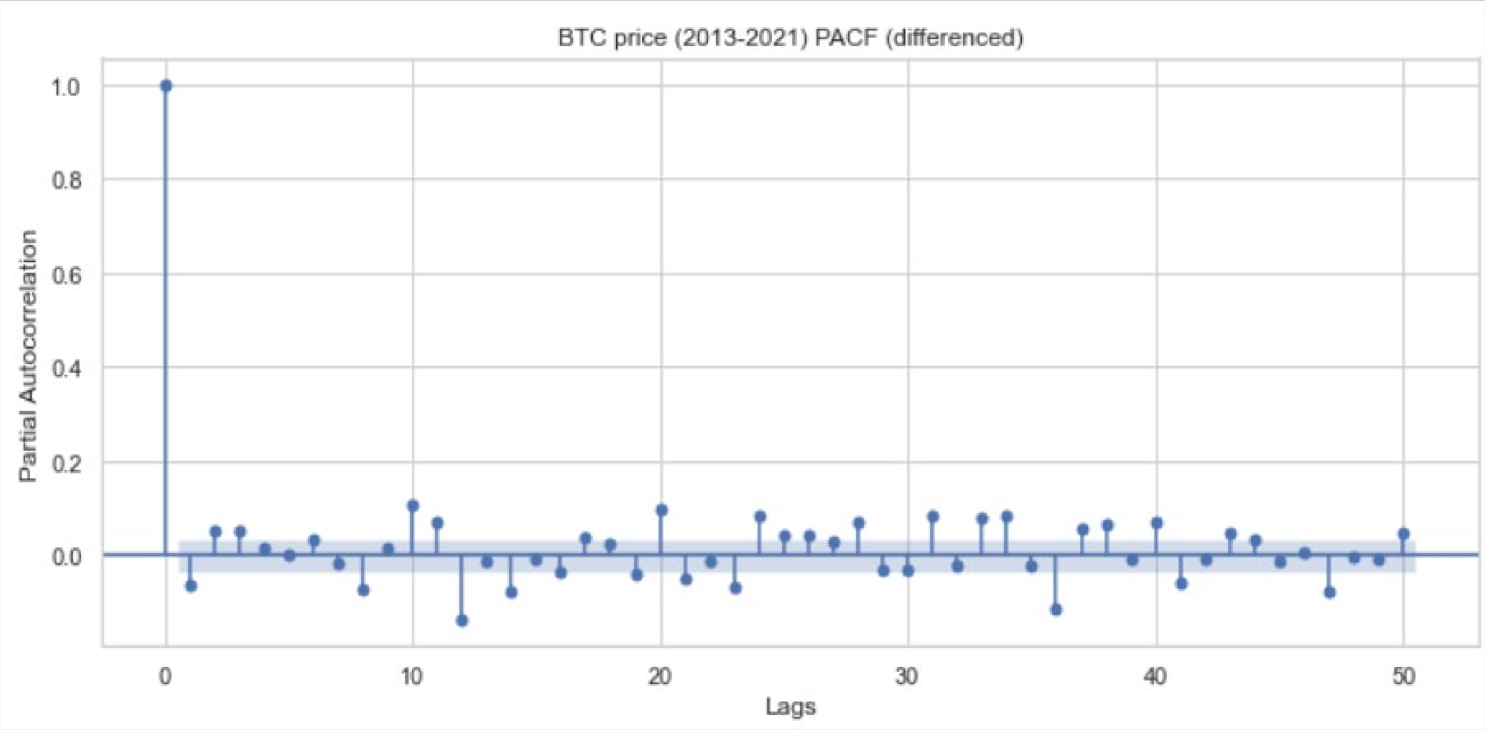
DIFFERENCED BTC PRICE (order = 1)



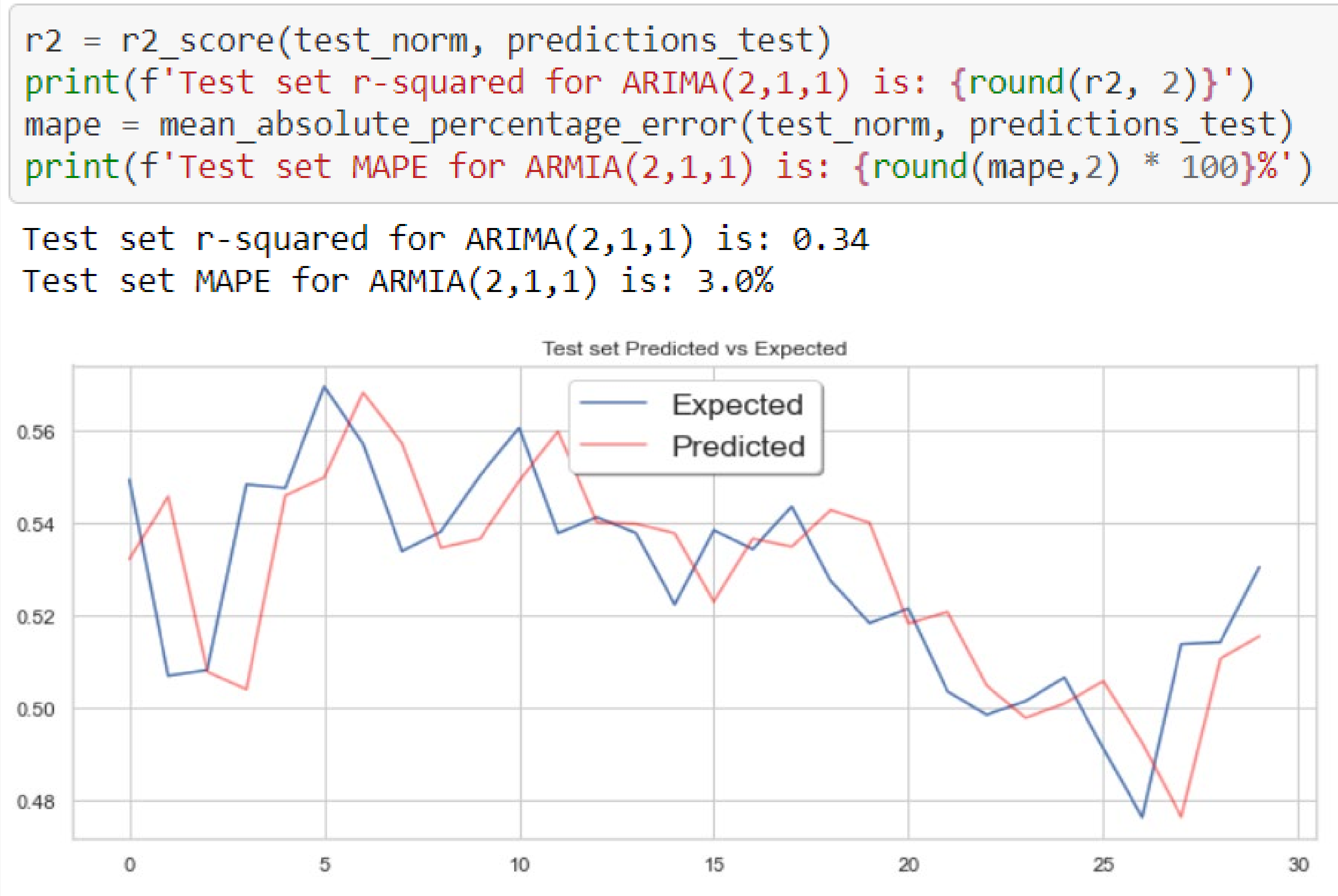
ACF PLOT


BTC price (2013-2021) ACF (differenced)
Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
Lags

PACF PLOT


BTC price (2013-2021) PACF (differenced)
Partial Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
Lags

ARIMA (2,1,1)



```
r2 = r2_score(test_norm, predictions_test)
print(f'Test set r-squared for ARIMA(2,1,1) is: {round(r2, 2)}')
mape = mean_absolute_percentage_error(test_norm, predictions_test)
print(f'Test set MAPE for ARMIA(2,1,1) is: {round(mape,2) * 100}%')
```

```
Test set r-squared for ARIMA(2,1,1) is: 0.34
Test set MAPE for ARMIA(2,1,1) is: 3.0%
```

EXTENDED MODELING

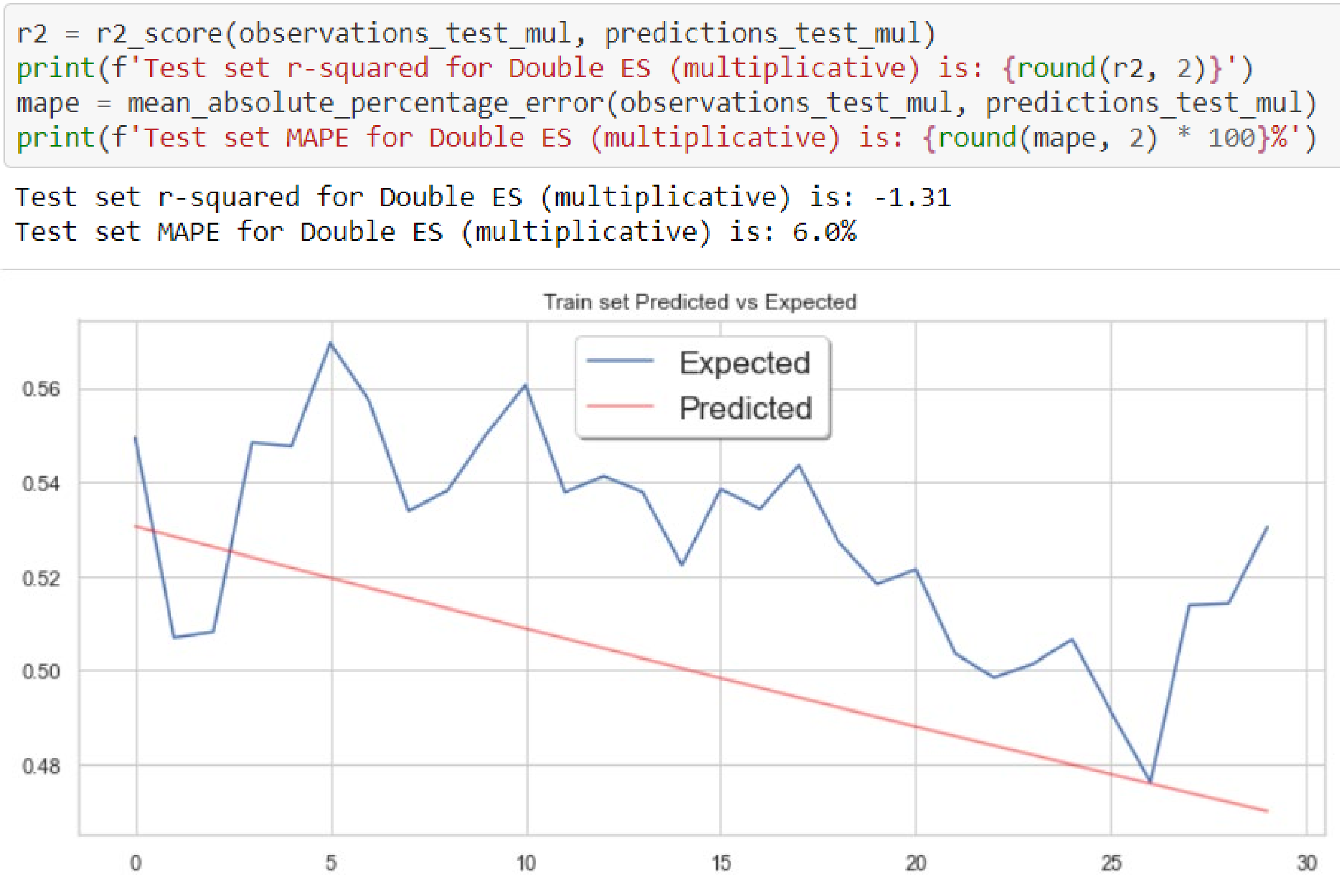
GOAL

- Build additional models
- Compare their performance with baseline model

MODELS

- Exponential Smoothing
- Facebook Prophet
- PyCaret

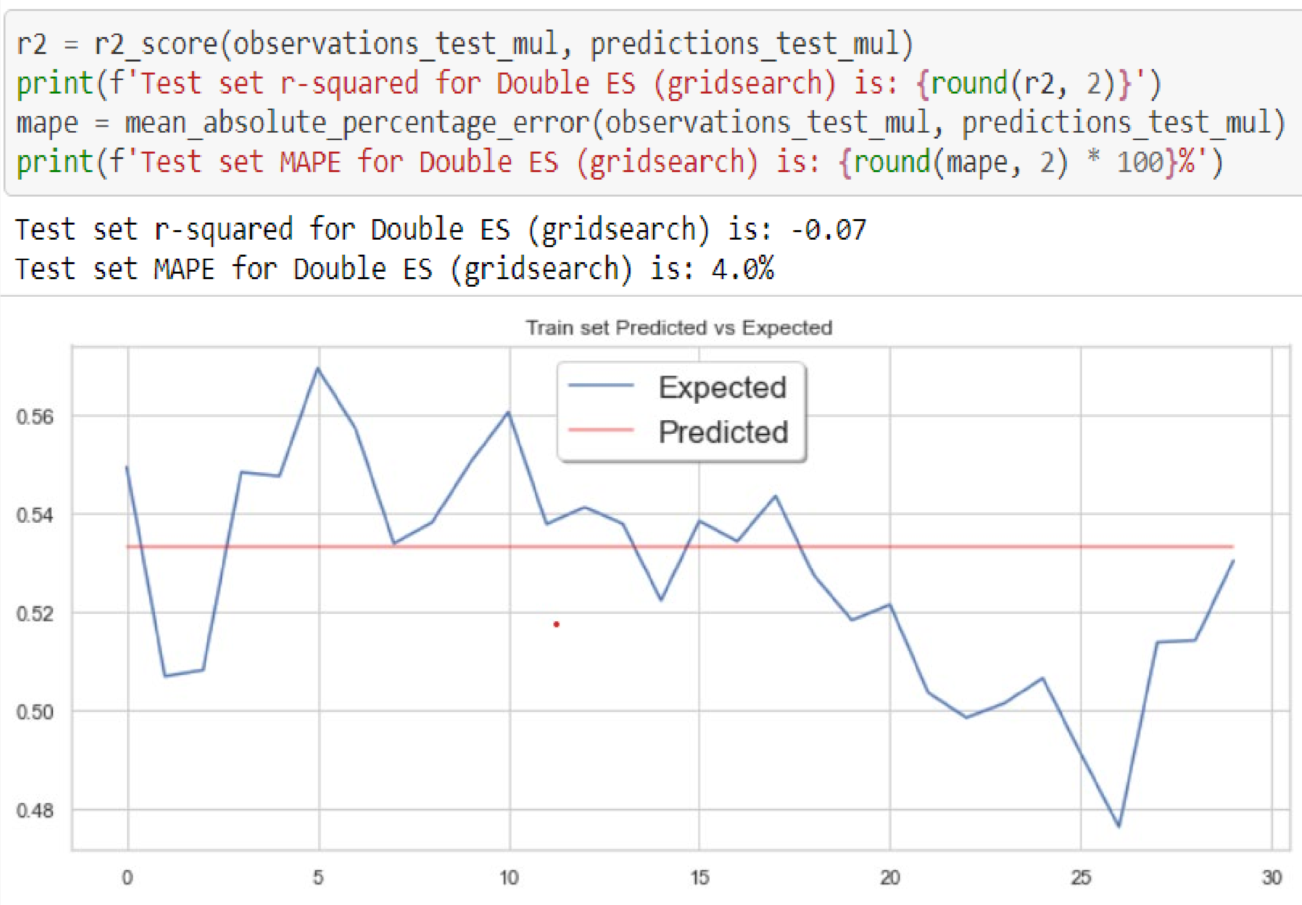
EXPONENTIAL SMOOTHING (NOT TUNED)



```
r2 = r2_score(observations_test_mul, predictions_test_mul)
print(f'Test set r-squared for Double ES (multiplicative) is: {round(r2, 2)}')
mape = mean_absolute_percentage_error(observations_test_mul, predictions_test_mul)
print(f'Test set MAPE for Double ES (multiplicative) is: {round(mape, 2) * 100}%')
```

```
Test set r-squared for Double ES (multiplicative) is: -1.31
Test set MAPE for Double ES (multiplicative) is: 6.0%
```

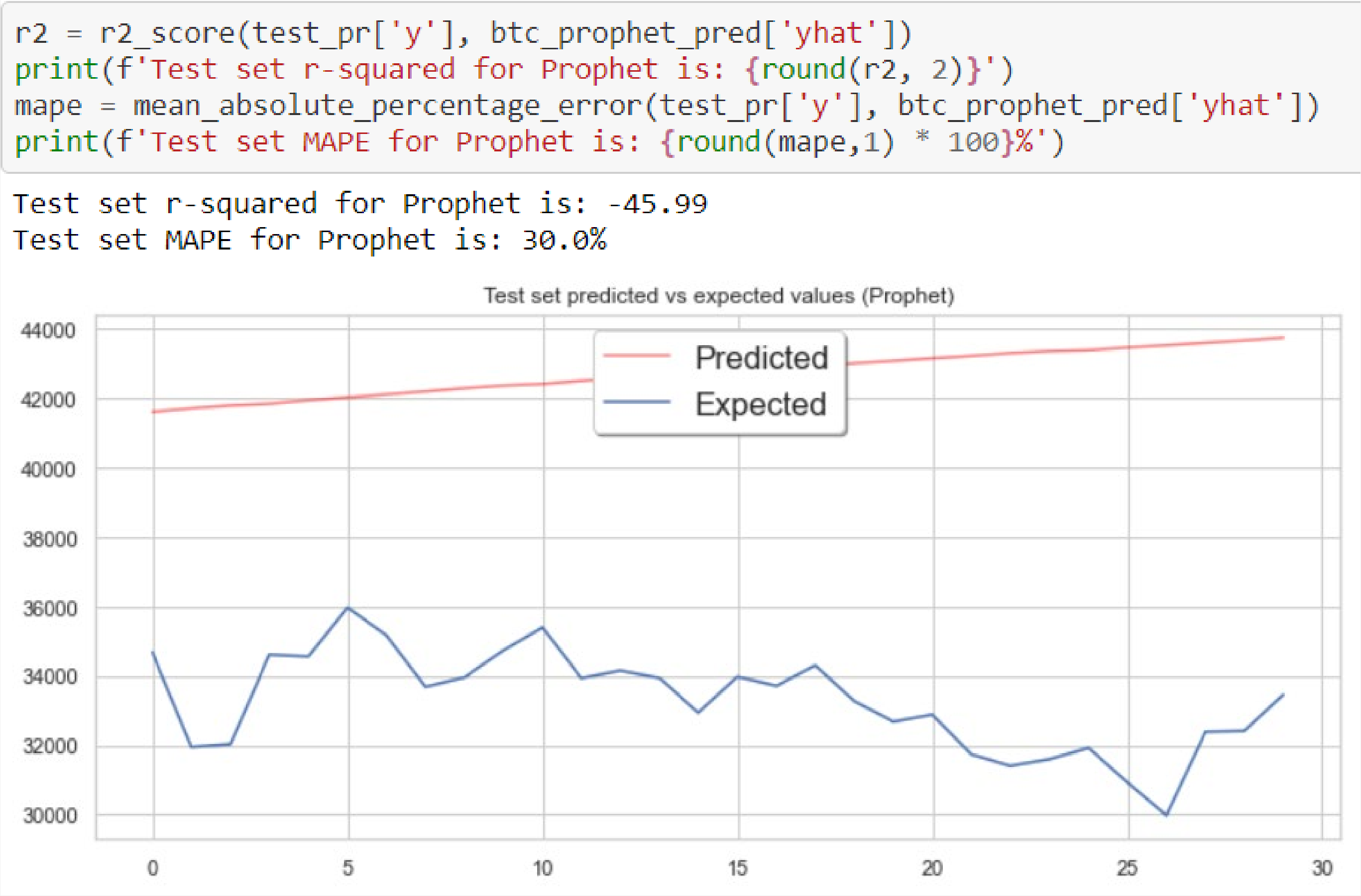
EXPONENTIAL SMOOTHING (TUNED)



```
r2 = r2_score(observations_test_mul, predictions_test_mul)
print(f'Test set r-squared for Double ES (gridsearch) is: {round(r2, 2)}')
mape = mean_absolute_percentage_error(observations_test_mul, predictions_test_mul)
print(f'Test set MAPE for Double ES (gridsearch) is: {round(mape, 2) * 100}%')
```

```
Test set r-squared for Double ES (gridsearch) is: -0.07
Test set MAPE for Double ES (gridsearch) is: 4.0%
```

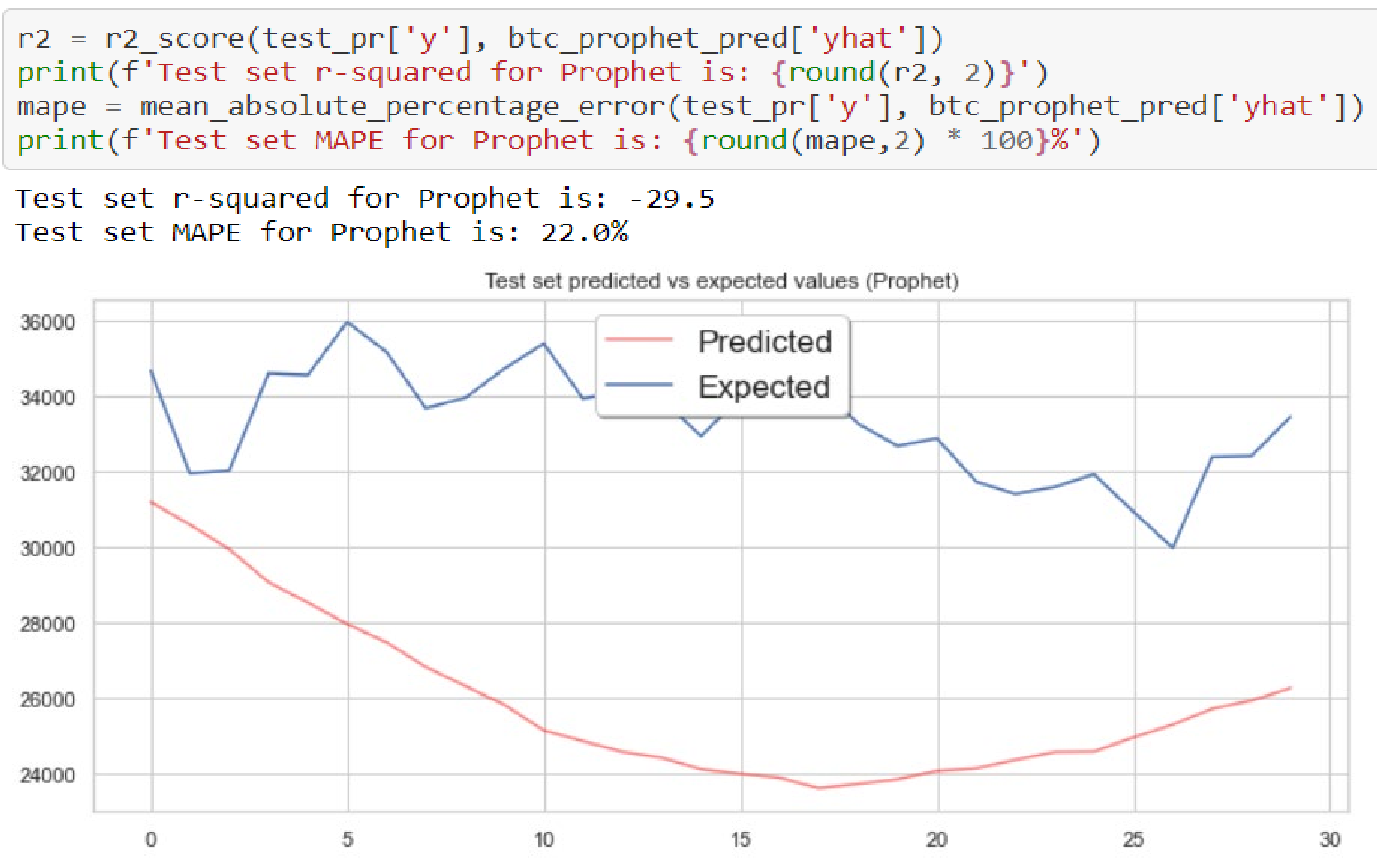
FACEBOOK PROPHET (NOT TUNED)



```
r2 = r2_score(test_pr['y'], btc_prophet_pred['yhat'])
print(f'Test set r-squared for Prophet is: {round(r2, 2)}')
mape = mean_absolute_percentage_error(test_pr['y'], btc_prophet_pred['yhat'])
print(f'Test set MAPE for Prophet is: {round(mape,1) * 100}%')
```

```
Test set r-squared for Prophet is: -45.99
Test set MAPE for Prophet is: 30.0%
```

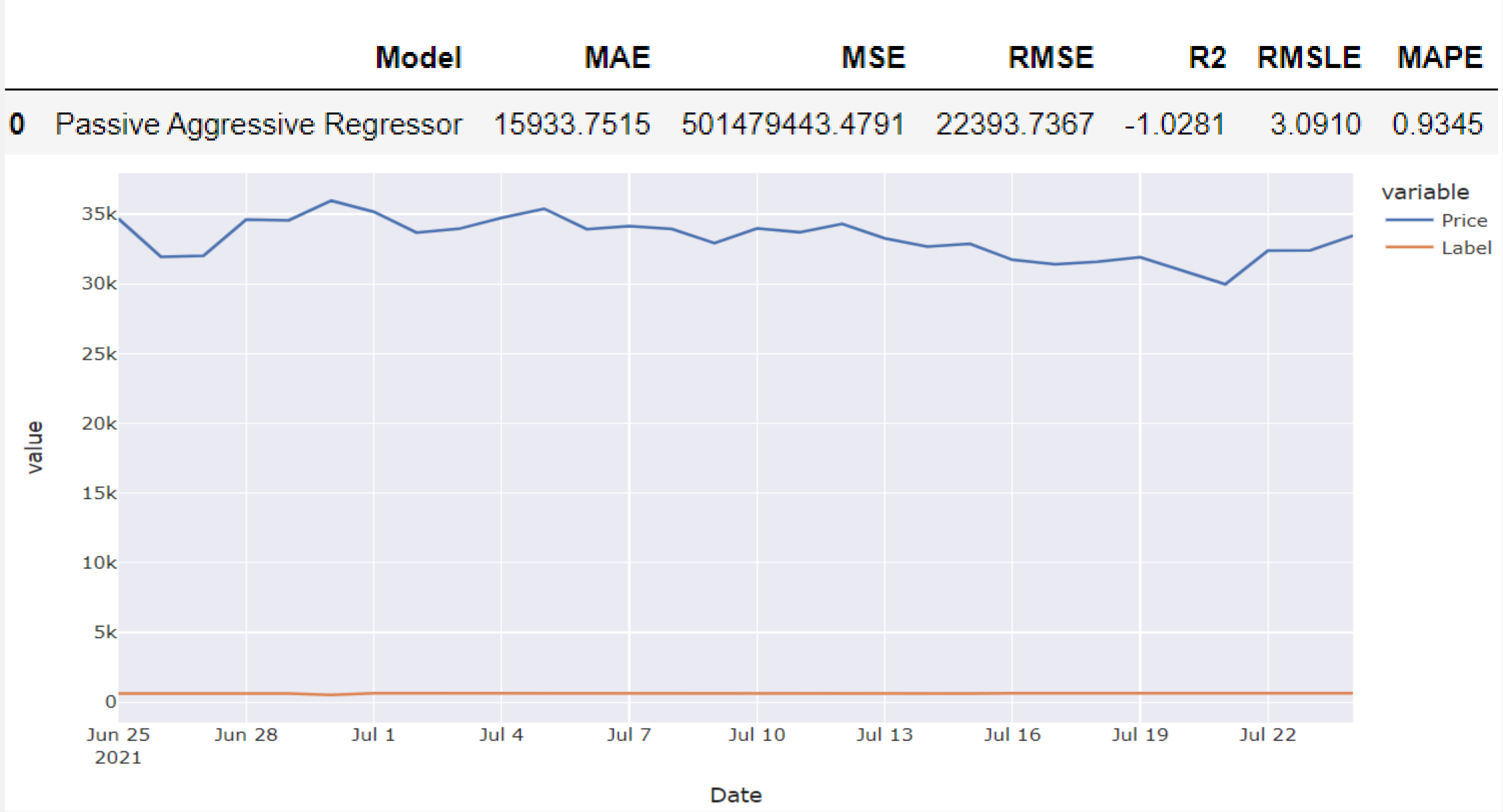
FACEBOOK PROPHET (TUNED)



```
r2 = r2_score(test_pr['y'], btc_prophet_pred['yhat'])
print(f'Test set r-squared for Prophet is: {round(r2, 2)}')
mape = mean_absolute_percentage_error(test_pr['y'], btc_prophet_pred['yhat'])
print(f'Test set MAPE for Prophet is: {round(mape,2) * 100}%')
```

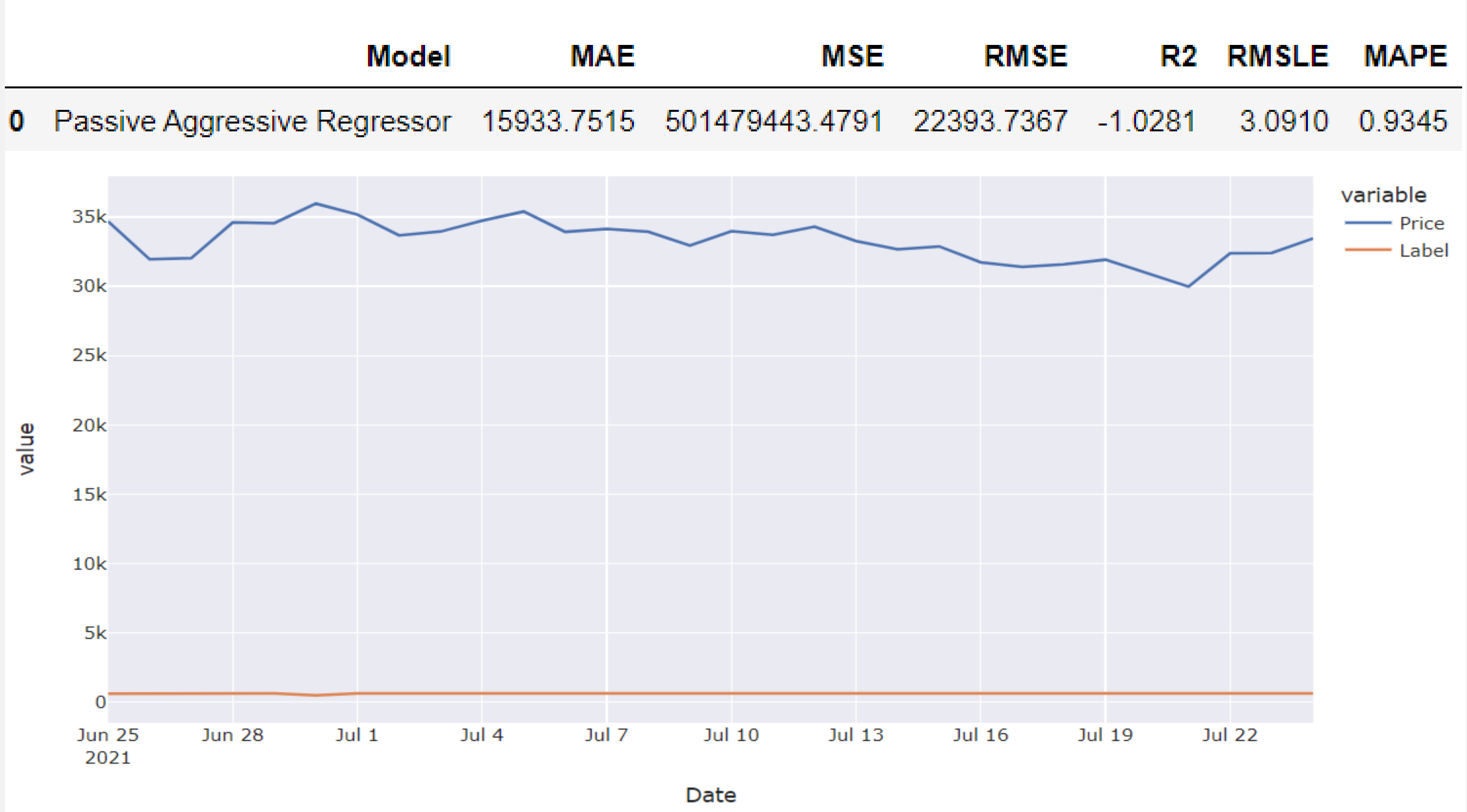
```
Test set r-squared for Prophet is: -29.5
Test set MAPE for Prophet is: 22.0%
```

PYCARET PASSIVE AGGRESSIVE REGRESSOR (NOT TUNED)



	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Passive Aggressive Regressor	15933.7515	501479443.4791	22393.7367	-1.0281	3.0910	0.9345

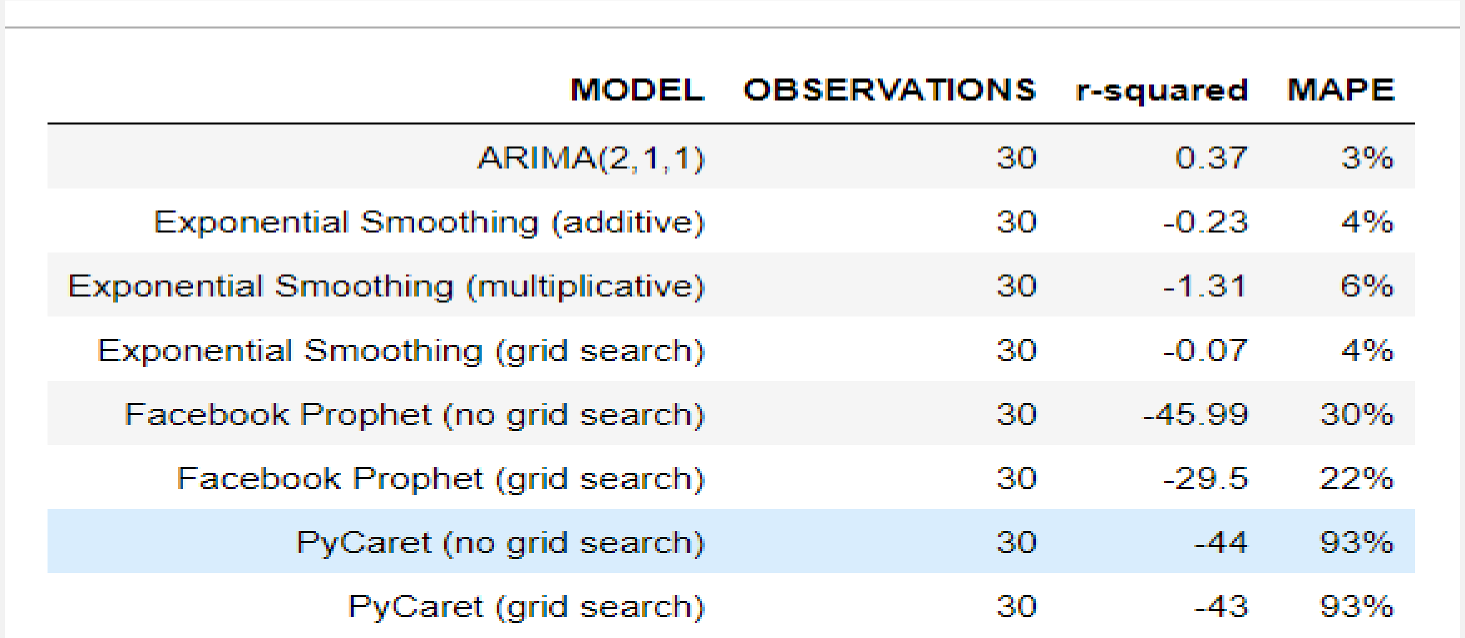
PYCARET PASSIVE AGGRESSIVE REGRESSOR (TUNED)



	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Passive Aggressive Regressor	15933.7515	501479443.4791	22393.7367	-1.0281	3.0910	0.9345

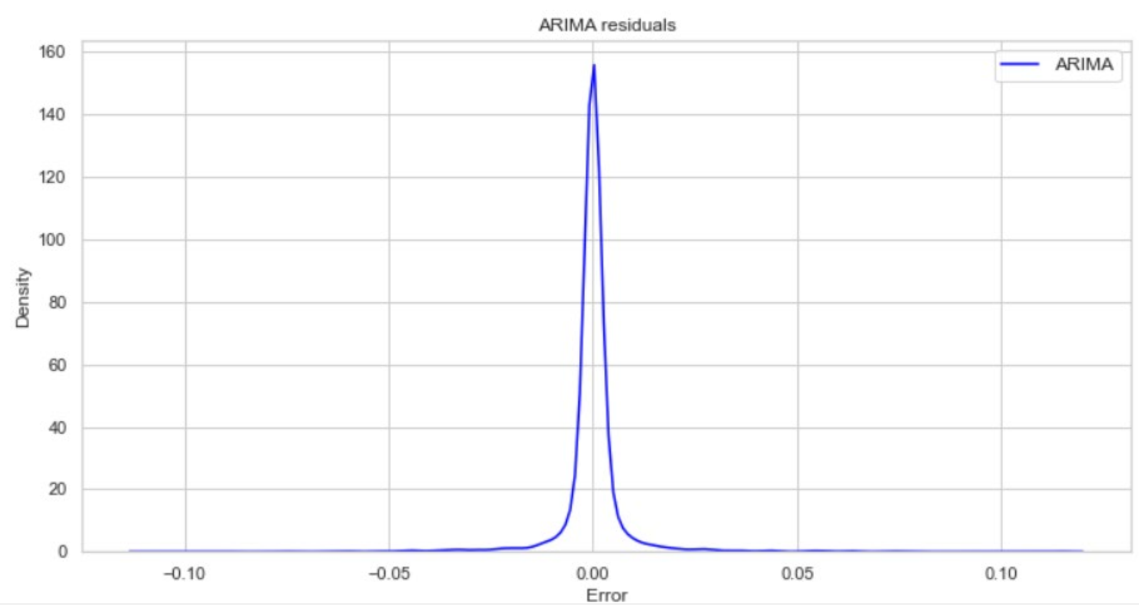
MODELING RESULTS

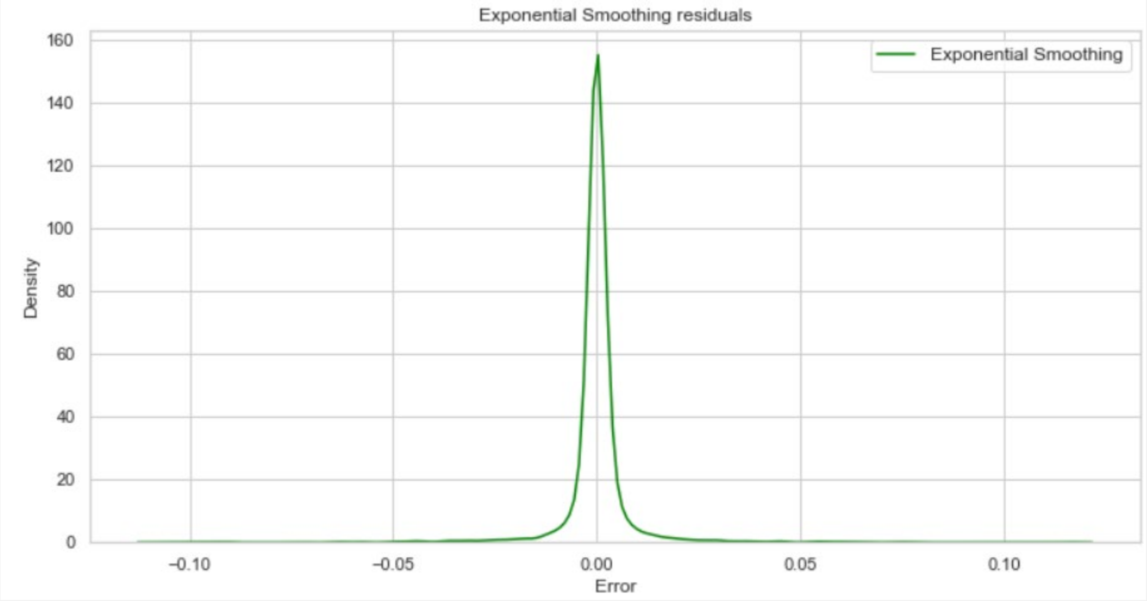
- ARIMA(2,1,1) and Exponential Smoothing showed low MAPE
- Only ARIMA(2,1,1) actually captured the structure of the time series



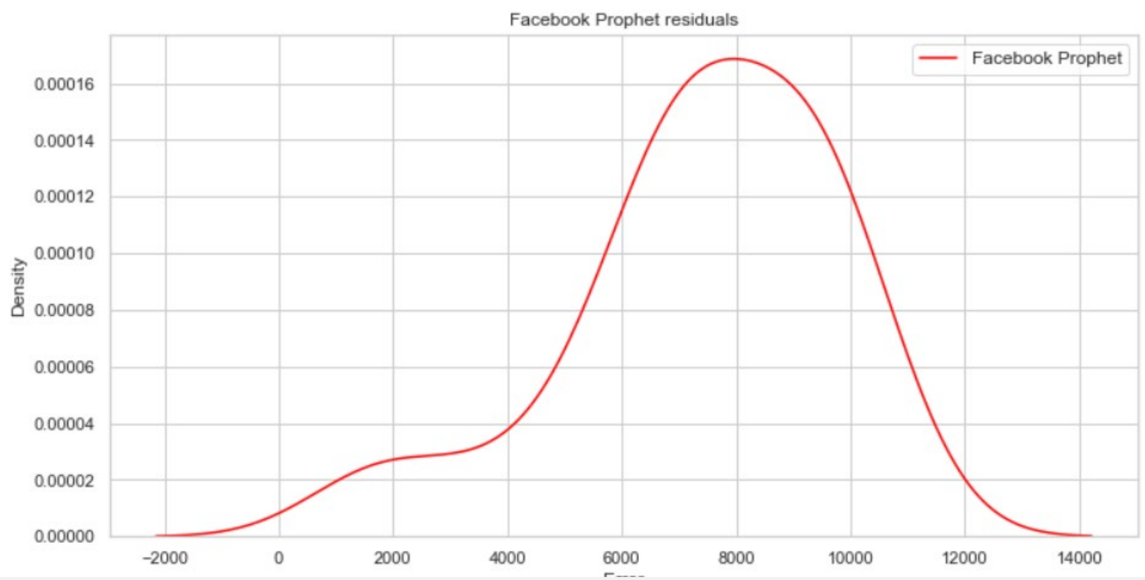
MODEL	OBSERVATIONS	r-squared	MAPE
ARIMA(2,1,1)	30	0.37	3%
Exponential Smoothing (additive)	30	-0.23	4%
Exponential Smoothing (multiplicative)	30	-1.31	6%
Exponential Smoothing (grid search)	30	-0.07	4%
Facebook Prophet (no grid search)	30	-45.99	30%
Facebook Prophet (grid search)	30	-29.5	22%
PyCaret (no grid search)	30	-44	93%
PyCaret (grid search)	30	-43	93%

RESIDUALS PLOTS (ARIMA AND EXPONENTIAL SMOOTHING)

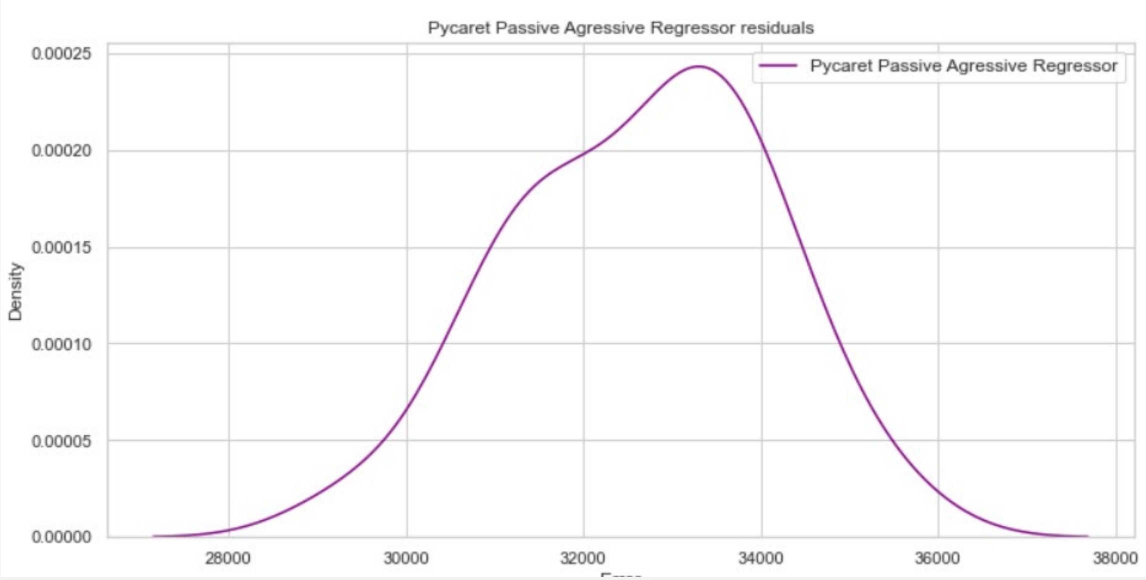




RESIDUALS PLOTS (PROPHET AND PYCARET)



Facebook Prophet residuals



Pycaret Passive Agressive Regressor residuals

CONCLUSIONS

- ARIMA(2,1,1) showed low MAPE and accurately captured the structure of the time series
- Exponential Smoothing showed low MAPE but failed to accurately capture the structure of the time series
- Both Prophet and PyCaret showed high MAPE and failed to accurately capture the structure of the time series

FUTURE WORK

- Apply neural networks
- Perform sentiment analysis of bitcoin-related social media accounts
- Estimate worst-case scenarios for the upper bound and lower bound of the errors using test set for each of the models

RECOMMENDATIONS

- Use ARIMA (2,1,1) for production implementation
- Be cautious since no statistical model can predict human psychology

USEFUL LINKS

- MAPE - https://en.wikipedia.org/wiki/Mean_absolute_percentage_error
- Random Walk - https://en.wikipedia.org/wiki/Random_walk
- Autocorrelation - https://en.wikipedia.org/wiki/Autocorrelation
- Partial Autocorrelation - https://en.wikipedia.org/wiki/Partial_autocorrelation_function
- Stationarity - https://en.wikipedia.org/wiki/Stationary_process
- Hypothesis testing - https://en.wikipedia.org/wiki/Statistical_hypothesis_testing

USEFUL LINKS

- ARIMA - https://www.statsmodels.org/stable/generated/statsmodels.tsa.arima_model.ARIMA.html
- Exponential Smoothing - https://www.statsmodels.org/stable/examples/notebooks/generated/exponential_smoothing.html
- Facebook Prophet - https://facebook.github.io/prophet/
- PyCaret - https://pycaret.org/