Springboard DSC

Capstone Project 3

Time Series Forecasting Cardano(ADA)

Hastings Reeves

August, 2021

(1) Introduction

The cryptocurrency market has provided retail investors with an incredible return on investment for the last decade. The adoption of blockchain technology as a secure platform for financial transactions has seen numerous companies attempt to provide meaningful services. Cardano, through the use of it's blockchain technology, provides investors the unique capability of securing transactions personally, with no middle man, and with the same processing speeds as Visa and Mastercard in terms of smart contracts they are able to produce per day. By creating a decentralized system for financial transactions, Cardano is attempting to shift buying power from the hands of the wealthy to the working class across the globe.

It is the purpose of this capstone project to compare market trends between Cardano's native token (ADA), Bitcoin(BTC), and Ethereum(ETH) with the intent of predicting future price action for Cardano. Time series forecasting algorithms ARIMA, PMD ARIMA, and FB Prophet will be utilized in the Preprocessing and Modeling portions of this project. Key stakeholders are myself and my business partner as we seek information regarding Cardano and it's blockchain/ecosystem. Overall, predicting future price action for such a volatile currency is possible.

Project Link:

https://github.com/threeves91/Springboard/tree/main/Capstone%20Project%203

(2.1) Data Acquisition and Wrangling

The Kaggle competition website contained CSV files of three dozen cryptocurrency tokens ranging from BTC and ETH to LINK and DOT. For the purposes of this project, I retrieved the CSV files for BTC, ETH, and ADA, each containing the following information:

SNo : Serial Number

Date : date of observation

Open : Opening price on the given day

High : Highest price on the given day

Low : Lowest price on the given day

Close : Closing price on the given day

Volume : Volume of transactions on the given day

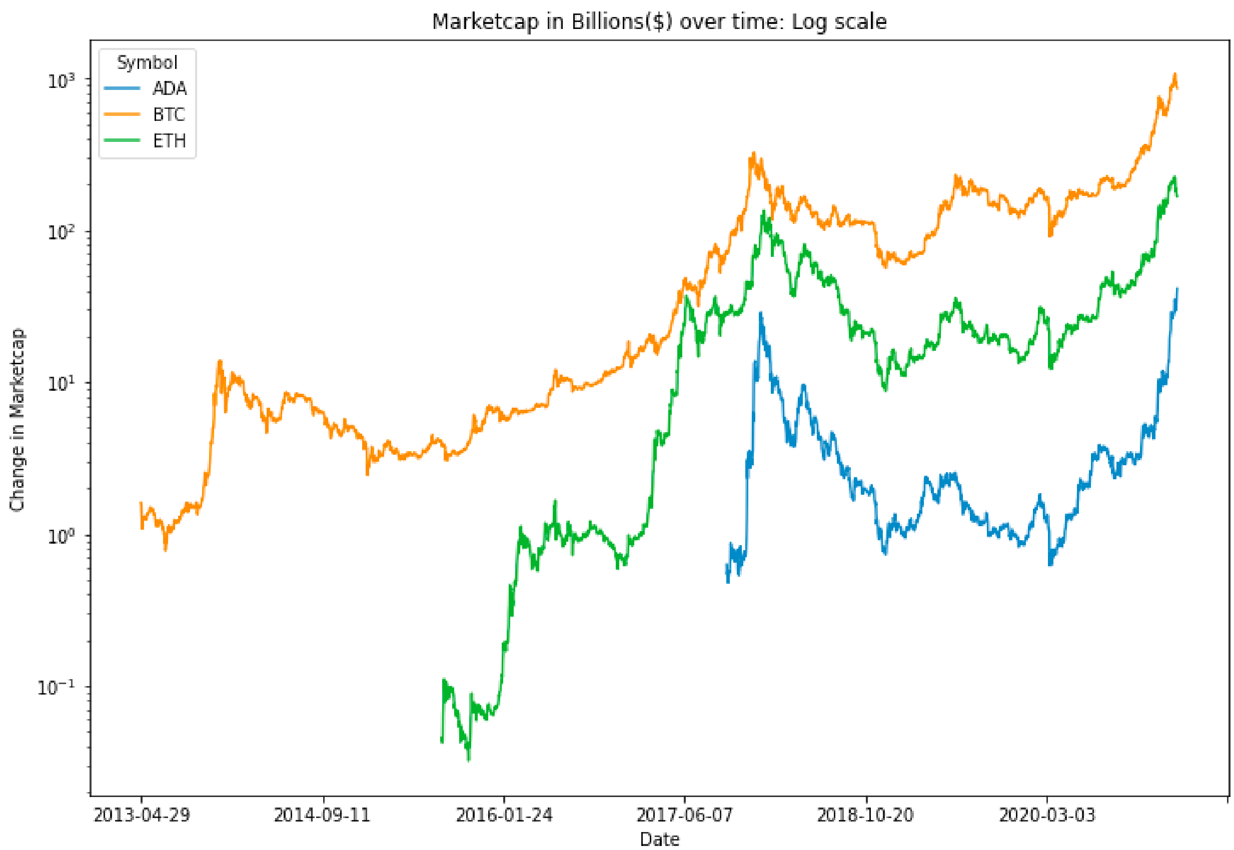
Market Cap : Market capitalization in USD

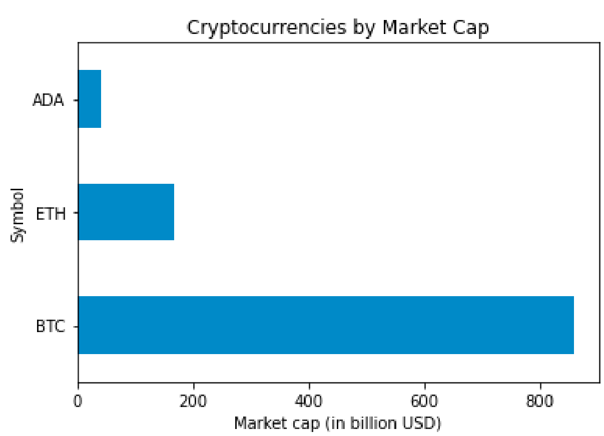
For this univariate time series forecasting problem, the focus was on the ‘Close’ price for each given ‘Date’ in the table. The granularity of the ‘Date’ column was split up into 24 hour increments, which was confirmed to have no gaps or missing dates. While the buying and selling of cryptocurrencies does not have a closing price as the markets don’t open or close per se, the ‘Close’ column was used as the target variable.

(2.2) Storytelling and Inferential Statistics

Within the exploratory data analysis phase, it was especially important to focus on three things: Market Cap comparisons between the three currencies, Transaction

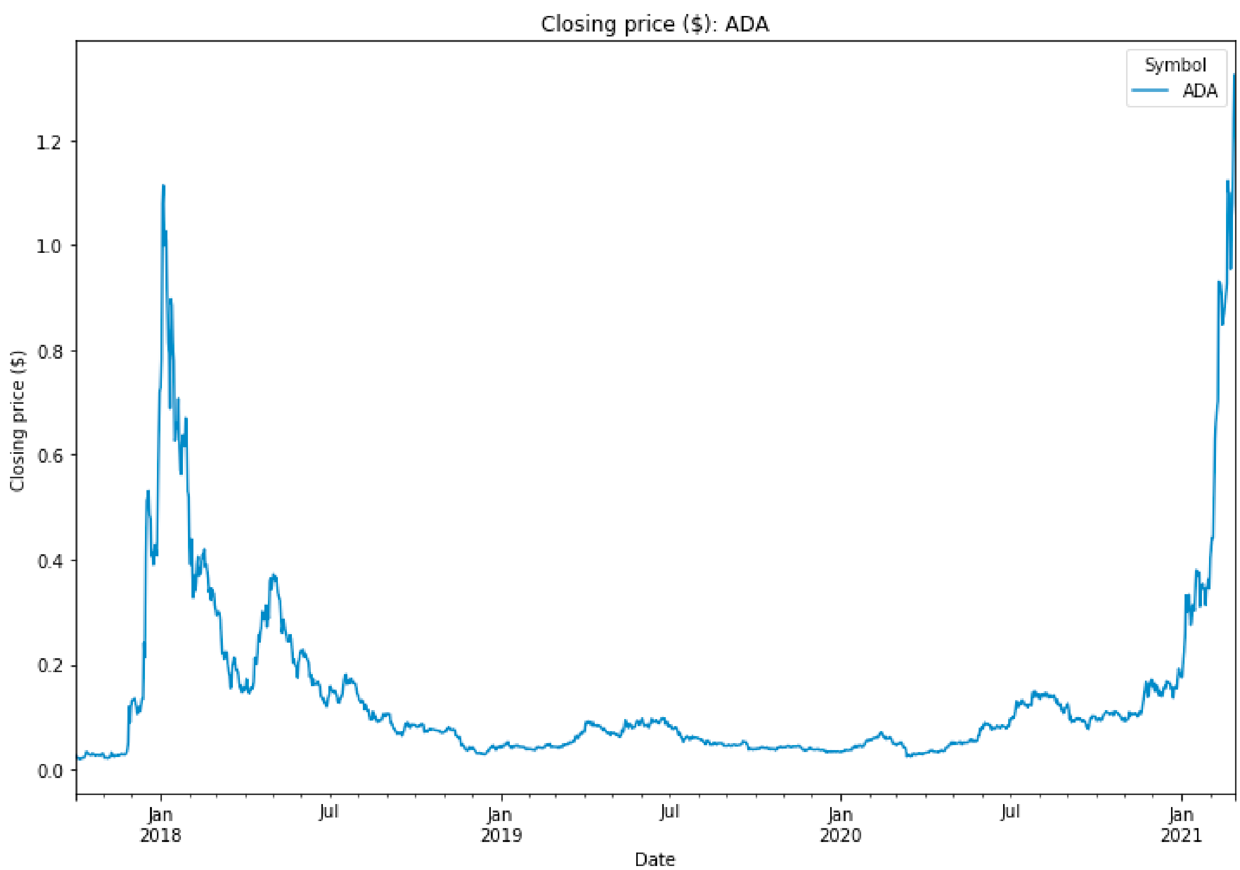
Volume comparisons between the three currencies, ADA price action over the course of it's existence. In terms of market cap, as is expressed in the following graphical representation, BTC dominates the market, accounting for over $850 billion worth of the roughly $2 trillion market.





As is explained in the preceding bar graph, BTC leads the way in terms of percentage of total market cap in comparison to ETH and ADA, so we can infer that BTC movements tend to see other cryptocurrencies follow in its path. In any decision made regarding cryptocurrency investing, it would be primarily important to be aware of the price action of BTC.

When exploring the price action of Cardano, it was insightful to see a graphical representation of the change in price over time.



There is a slight upward trend, dominated by two enormous peaks in the dataset. There is no seasonality, and it is clear there is no stationarity in the data. It was important, as a part of the exploratory data analysis, to have a good understanding of what features could use tuning within the ARIMA model itself in order to provide the best predicted values. ARIMA does best on stationary, seasonal data with a clear trend. With this dataset being quite a ways away from that initially, our work was cut out for us.

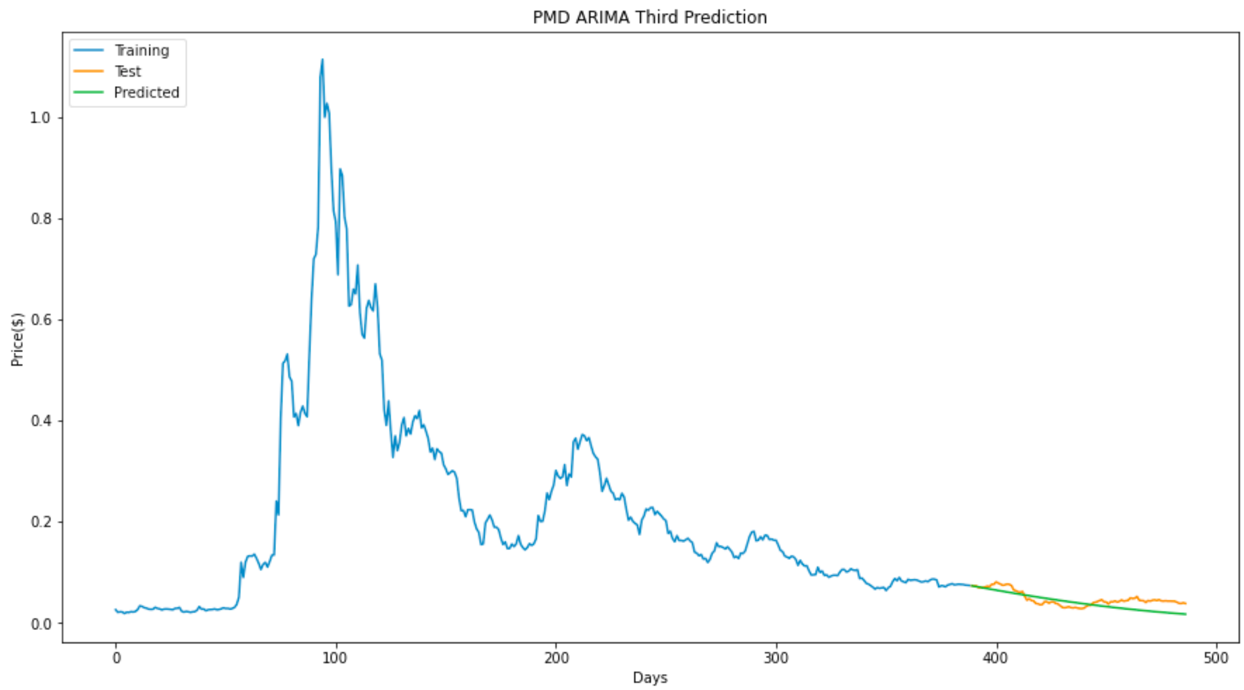
(2.3) Baseline Modeling

Using ARIMA as the baseline model it was important to achieve stationarity before attempting to run anything through the algorithm. This was achieved through logarithmic transformation of the data as well as differencing the data once. It is important to note that with the necessity of differencing the data, we should set the D value for the ARIMA model to 1, knowing this allows for stationarity in the data. I then used an 80/20 train/test split and began working on the training data set. From here, I defined a function called Evaluate_Arima_Model that returned the mean absolute error as a performance metric for the ARIMA model. I then passed this function into another function that iterates through a range of values from 0 to 3 for each P, D and Q value, and returned the model with the best MAE as the Best Arima. The results was an ARIMA(1,0,2) which I knew was not the best model, as I knew the data needed to be differenced. Nevertheless, I substituted the D value for 1 and continued on. I intended on using two performance metrics as key indicators for model optimization going forward; the mean absolute percentage error and a distribution plot of the residuals. The MAPE for this model was over 620%, which indicated the difficulty the model was having fitting predicted values to actual values. The distribution of the residuals confirmed the difficulty, and left us nowhere to go but up in the modeling phase of the project.

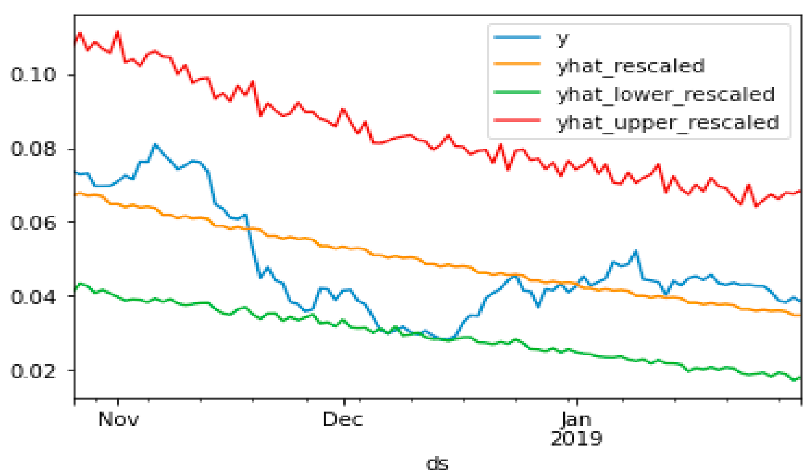
(2.4) Extended Modeling

There were several key issues I knew I would need to address going into this phase. Firstly, the time frame was simply to0 expansive for such a volatile data set, and it would be beneficial to focus on a smaller period of time in which there was a well defined trend. Following this logic, I chose to run a PMD ARIMA and FB Prophet model on the full time frame, a shortened time frame that focused on the obvious valley between the two peaks, and the first peak found in the data set. The Peak Time Frame was from 10/01/2017-1/31/2019 and the Valley Time Frame was from 6/30/2018-7/21/2020.

For the PMD ARIMA, we saw a noticeable improvement for the MAPE at 34% in the full time frame model. It is much simpler to allow the PMD ARIMA function to find an optimal model and then tune hyperparameters from there as opposed to attempting to decipher P, D, and Q values as I did in the ARIMA model. The Valley time frame performed marginally worse with a MAPE of 35% and the Peak time frame, see on next page, was the best of all PMD ARIMA models at 28%. As is obvious from the graphical representation, the model did notably better in part due to the well defined, downward trend in the data. This is a common theme I found throughout this project, that within a trend it is easier to predict future price action. Conversely, it is difficult for ARIMA or PMD ARIMA to account for sudden changes in trend.



After completing this portion of the modeling phase, I moved onto FB Prophet, using the same time frames. Here we saw similar results, with the models performing better than the baseline and the peak time frame providing the best model of models with a MAPE of 22%. See below.



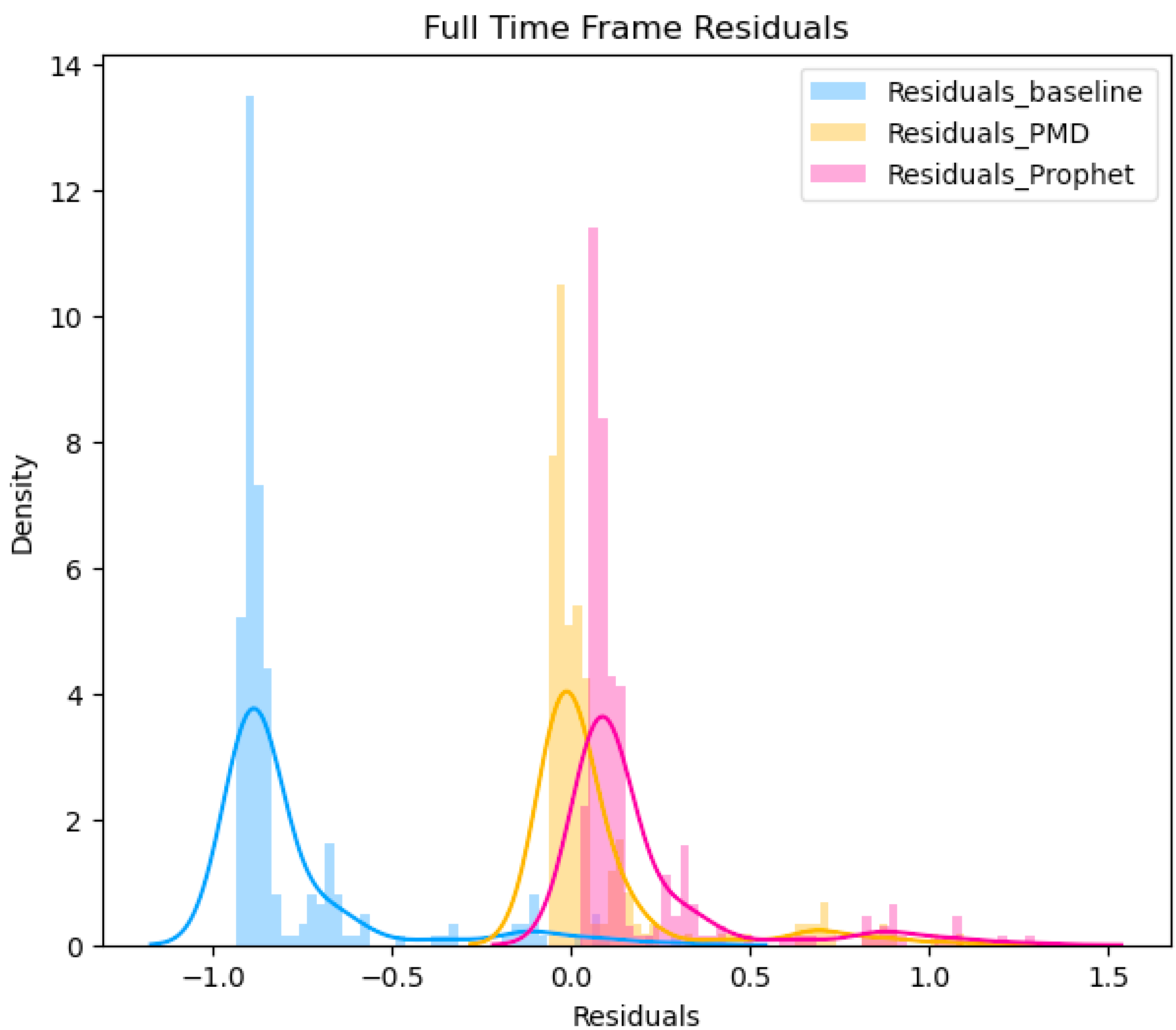
Importantly, all of the data points fell within the upper and lower bounds with the fitted prediction line being unable to account for the smaller fluctuation but still finding the general trend.

(3) Findings

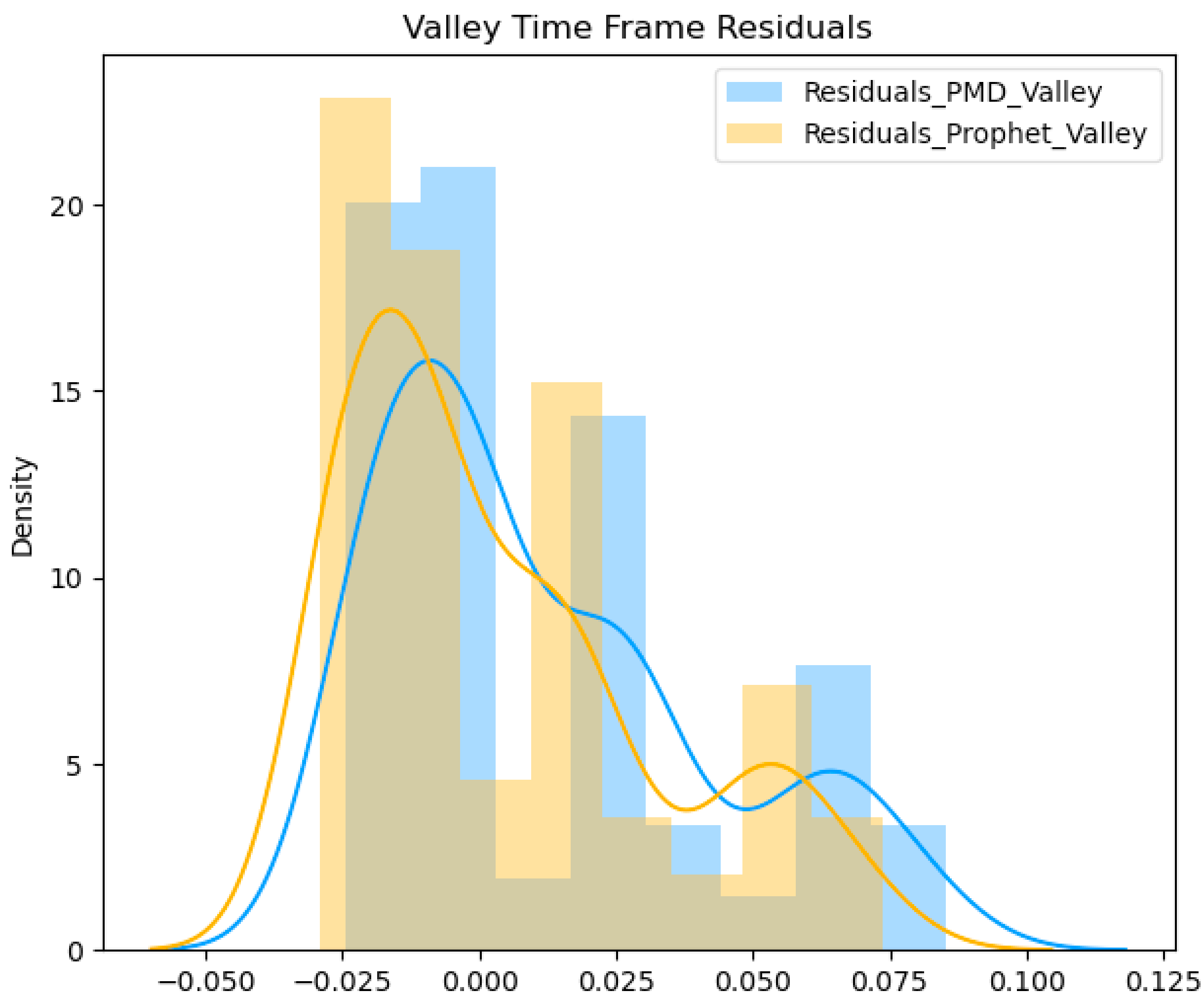
Below you will find a table containing the MAPE's for each model used in this notebook as well as a distribution plot of residuals, broken into three groups: Full time frame, Valley time frame, and Peak time frame.

	Model	**MAPE**
6	Prophet_Peak_MAPE	0.2199
3	PMD_Peak_MAPE	0.2828
1	PMD_MAPE	0.3403
2	PMD_Valley_MAPE	0.3515
5	Prophet_Valley_MAPE	0.3952
4	Prophet_MAPE	0.7444
0	Baseline_MAPE	6.2671

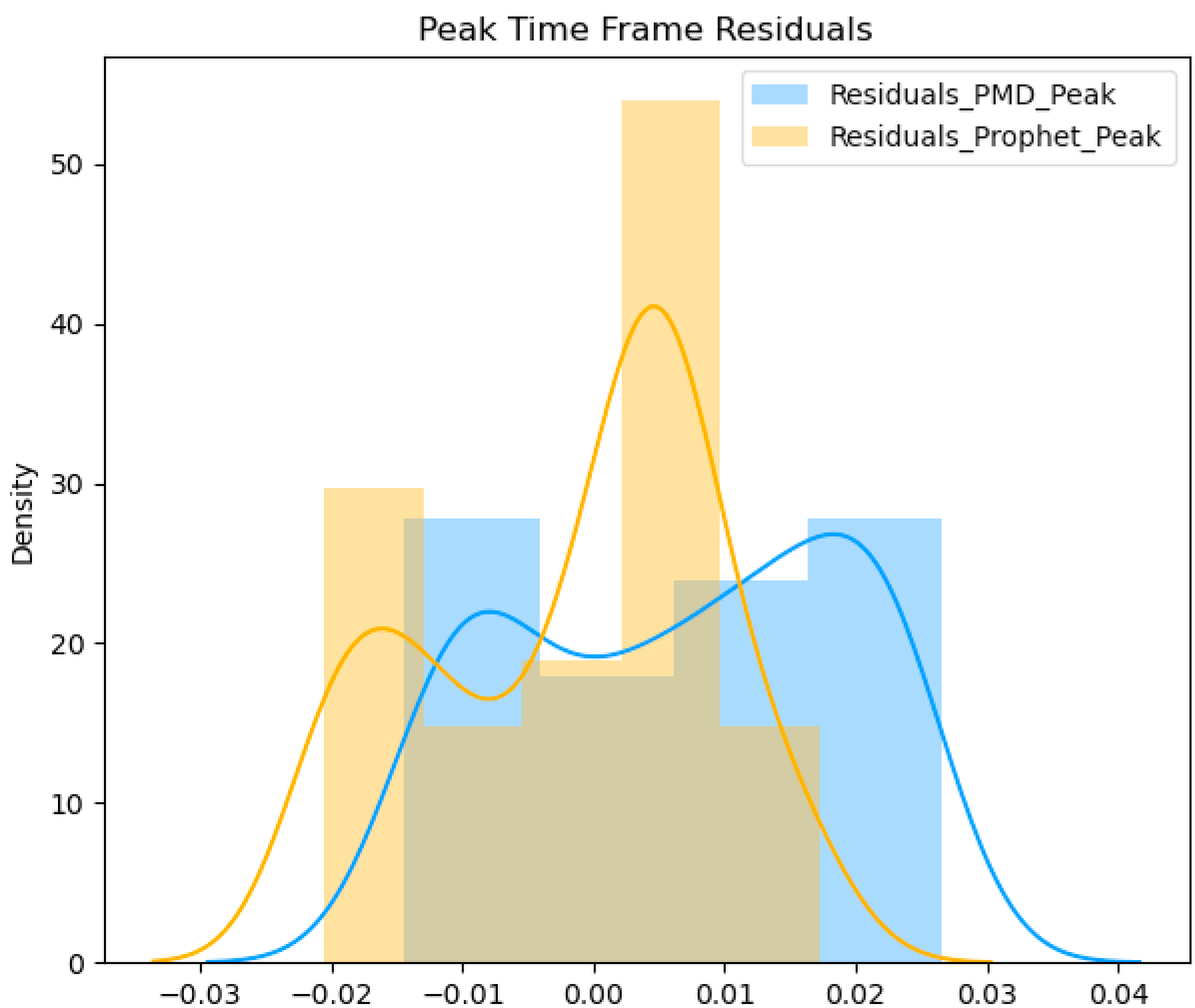
Firstly, we can see that the peak time frames performed the best of all models, with the FB prophet MAPE edging out the PMD ARIMA by 6%. All models following the baseline were a vast improvement when comparing the MAPE of each. In the distribution of the residuals, one can see the full time frame below. While the PMD ARIMA and Prophet models were much closer to a normal distribution, all three had significant tails to the right.



In the following residuals plot, we note the valley time frame has a similar issue with a right tail although both the PMD ARIMA and Prophet model were closer to 0 than the full time frame as a whole.



Finally, as is seen on the following page, the best distribution of the residuals were found in the peak time frames, with the Prophet model being closer to a normal distribution.



In conclusion, the Prophet model focusing on the Peak time frame had the best MAPE at 22% and boasted the closest residual plot resembling normal distribution. It can thus be inferred that the future price action of Cardano can be predicted with a certain degree of confidence by the FB Prophet model. However, further studying and potential adding of features could boost the model performance even more.

(4) Conclusions and Recommendations

While FB Prophet did provide us with a decent set of predicted values, it would still have a hard time predicting a drastic change in direction. Proceeding forward it would be insightful to use a multivariate time series forecasting model where we factor in transaction volume, market cap and an hourly granularity. There could be some feature engineering with the transaction volume based on whether or not the transaction was ‘buying’ or ‘selling’ the currency. It would also be incredibly helpful to use NLP in conjunction with twitter mentions of this currency to determine market sentiment for that currency on a given day.

Overall, Cardano seems like an asset in an upward trend, and as Bitcoin continues to seek a more prevalent role in our society, ecosystems like Cardano have the potential to thrive as they provide a crucial asset. Using FB Prophet in conjunction with noting the macro trends of BTC, one could come to a reasonable conclusion about future price action of Cardano, and find good entry prices accordingly.

Sources:

Hyndman, Rob. J, Athanasopoulos, George. ‘Forecasting: Principals and Practice’.

https://github.com/mkosaka1/AirPassengers_TimeSeries/blob/master/Time_Series.ipynb

https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.auto_arima.html

https://pythondata.com/forecasting-time-series-data-with-prophet-part-1/

https://github.com/urgedata/pythondata/blob/master/fbprophet/fbprophet_part_one.ipynb

https://towardsdatascience.com/predicting-the-future-with-facebook-s-prophet-bdfe11af10ff

https://towardsdatascience.com/time-series-forecasting-with-pycaret-regression-module-237b703a0c63