Crypto Currency Price Prediction Using Azure Auto-Machine Learning

by

Hung Kai Ho, M.S.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

1.1 Motivation

Machine Learning (ML) is a tool that enables us to better understand our data and acquire data-driven decisions using algorithmic methods and statistical learning. This technology has revolutionized all sectors of global economy and is rapidly evolving. However, most of the ML process can be computationally expensive and requires significant upfront investment for the necessary hardware to obtain useful information within reasonable time. Fortunately, these problems can be solved by utilizing cloud computing services that are optimized for ML uses.

The motivation of this report is to employ one of the major ML cloud computing services: Azure Machine Learning Studio and exploring its utilities to forecast financial data.

1.2 Outline of the Research Approach

The goal of this effort is to explore the utility of Auto-Machine Learning tool (AutoML) in the Microsoft Azure cloud computing services. This work used crypto currencies' end of day prices as dataset and used regression and time-series models in the AutoML to estimate the future closing price.

The report analyzed three different crypto currencies using AutoML and selected the models with the minimum Normalized Root Mean Square Error (NRMSE) for forecasting. The selected models are later used to forecast the price with various forecasting lengths.

1.3 Organization of the report

The organization of this thesis follows the work’s progression order, which is broken down into eight chapters. The first chapter introduces the motivation of applying Machine Learning for Crypto currency price prediction, the outline and organization of this study, and assumptions of the report.

Chapter 2 describes the methodology of this research. It covers the data selection and cleaning process, as well as the functionality and the steps to set up AutoML.

Chapter 3 provides a detailed explanation of the regression models experiment and the results. The results show how the regression models perform with various types of crypto currency with different forecasting lengths.

Chapter 4 presents an in-depth explanation of the time-series models experiment and the results. The results show how the time-series models perform with various types of crypto currency with different forecasting lengths.

Chapter 5 summarizes the results of this study and presents the research finding and suggestions for using AutoML and using ML models for price forecasting. additionally, this chapter discusses the limitation of the research and recommendation for future topics.

CHAPTER II

METHODOLOGY AND DATA COLLECTION

This report utilized the use of Azure Machine Learning Studio to create the forecasting models. The AutoML tool in the studio is an application designed to automatically train and tune model using a target metric. This report used crypto currency price data to showcase the usability and predictive results of the AutoML tool.

2.1 Data selection and collection

This report used three differently type of crypto currencies: Ether($ETH), Bitcoin($BTH), and Binance Coin ($BNB). These three crypto currencies (CC) have different use case. The utilities of these currencies can possibly affect their price movement. $ETH is the currency on the Ethereum Network, which is a platform that allows developers and creators to build applications using blockchain technology. $BTC is the first developed crypto currency intended to be used a decentralized medium of exchange. However, as the popularity of the CC grows and institutional money enters the market, $BTC is becoming a reserve asset for future price appreciation. $BNB is a CC issued by Binance crypto currency exchange. $BNB is designed to be a utility token within the Binance exchange. $BNB holders can access discounted trading fees and developers can use it to fund applications on Binance smart chain.

The report collected these three CC price with Pandas Python library from Yahoo finance. The duplicated data is then removed. For this report the models were trained to predict the closing price, therefore, the daily high, mid, low, open, and close price data are removed to prevent models from predicting the next day price using intraday price data. Since the final models are tasked to forecast the next month’s price, the last 30 days of the data are trimmed off. finally, the data are uploaded to the Azure studio dataset for the model training.

2.2 Models creation in Azure Machine Learning Studio

The AutoML tool in Microsoft Azure Machine Learning Studio allows users to create machine learning models using the studio's build-in ML algorithms. The users can setup the machine learning models using the studio interface or execute the processes with Python. Furthermore, the AutoML tool offer two types of ML models: regression analysis and time-series analysis. For this report both analysis methods are used for forecasting $ETH, $BTC, $BNB price.

Since all the models are executed on the Azure server, a computer instance is required before setting up AutoML processes. For the purpose of the report the standard virtual machine setup is recommended by Azure studio. Subsequently, the date column, training data column, cross validation, and iteration stopping parameters are specified in the AutoML configuration. The model performance is then gauged by the normalized root mean squared error.

2.3 Testing and forecasting results generation

After the final models are created, they are then deployed on the Azure ML studio for forecasting. For the purpose of this report, the models are tasked to forecast 30 days after the last date of the training dataset. The generated forecast results are compared against actual CC price data and exhibit in both graphical and quantitative (in NRMSE and average percentage error) manners.

CHAPTER III

REGRESSION MODELS

3.1 Model description

The models in this chapter are built with the AutoML regression analysis tool. The AutoML tool is capable of training and discovering the best model based on the dataset and training configuration without the need of implementing regression ML models individually.

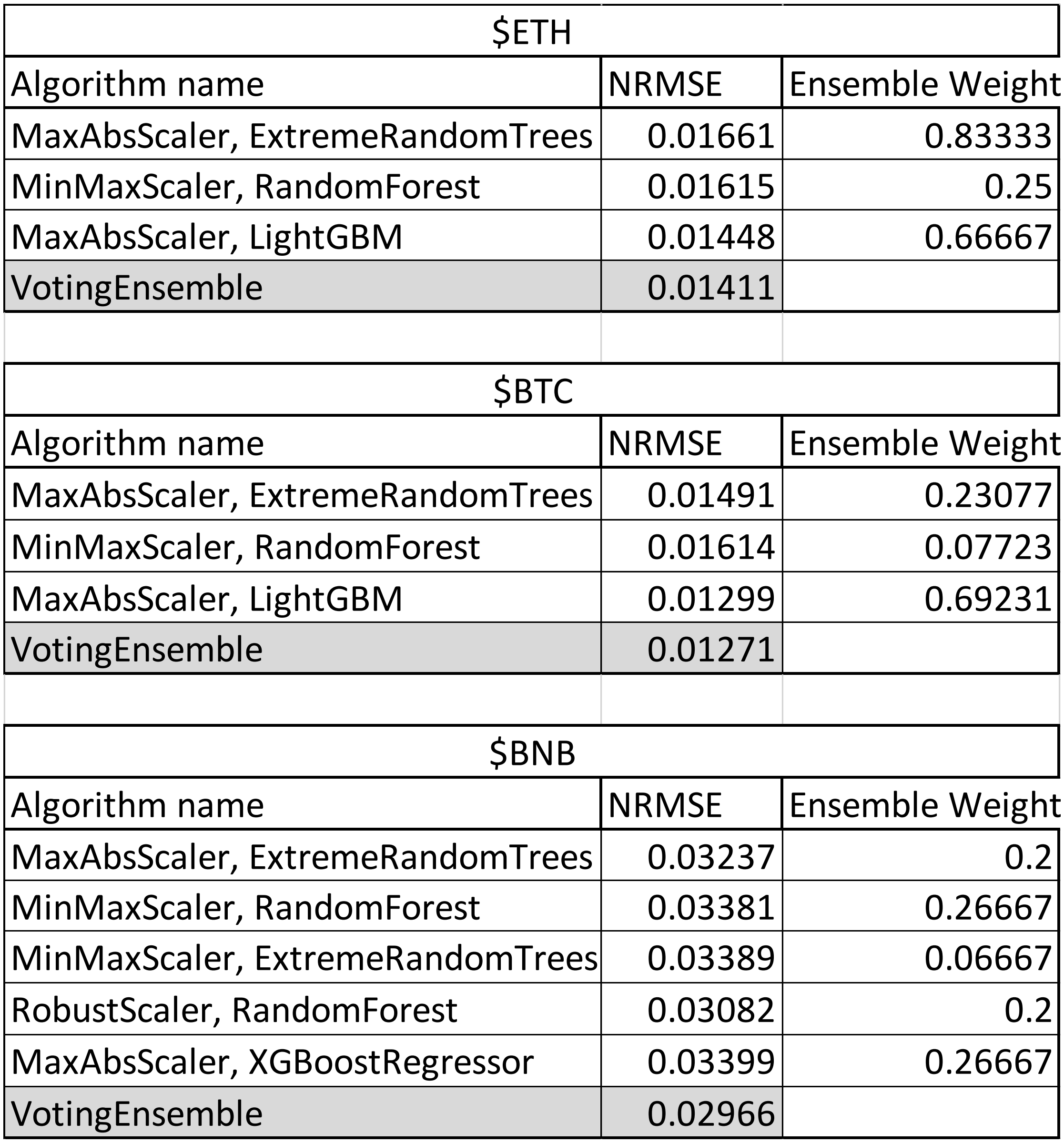
The AutoML is first supplied with the dataset and specified time column and training column ($ETH, $BTC, and $BNB in our case). Subsequently, the validation and iteration parameters are set based on the designated computational power and modeling time. Lastly, AutoML trains several (in our case ~45) regression models based on the given parameters and evaluates (based on NRMSE) then selects the best three models to create the final model using voting ensemble learning. The voting ensemble model is then setup to forecast three different CC data with time frame of one day, one week, and one month.

3.2 model results

3.2.1 Training results

Figure 4.1 shows the NRMSE values of the best models built with voting ensemble for each CC data and the composition and weights of each ensemble algorithms. Form the data shown, $ETH and $BTC have the same algorithm composition with various weight distribution. On the contrary $BNB has a different algorithm composition in its voting ensemble model. The NRMSE value for $BNB is also relatively greater compared to $ETH and $BTC.

One possible explanation is that both $ETH and $BTC have been showing price volatility since 2017, whereas $BNB only has price volatility in the past year since 2015.



$ETH		
Algorithm name	NRMSE	Ensemble Weight
MaxAbsScaler, ExtremeRandomTrees	0.01661	0.83333
MinMaxScaler, RandomForest	0.01615	0.25
MaxAbsScaler, LightGBM	0.01448	0.66667
VotingEnsemble	0.01411	

$BTC		
Algorithm name	NRMSE	Ensemble Weight
MaxAbsScaler, ExtremeRandomTrees	0.01491	0.23077
MinMaxScaler, RandomForest	0.01614	0.07723
MaxAbsScaler, LightGBM	0.01299	0.69231
VotingEnsemble	0.01271	

$BNB		
Algorithm name	NRMSE	Ensemble Weight
MaxAbsScaler, ExtremeRandomTrees	0.03237	0.2
MinMaxScaler, RandomForest	0.03381	0.26667
MinMaxScaler, ExtremeRandomTrees	0.03389	0.06667
RobustScaler, RandomForest	0.03082	0.2
MaxAbsScaler, XGBoostRegressor	0.03399	0.26667
VotingEnsemble	0.02966	

Figure 3.1 Voting Ensemble and the Ensemble Weight results in NRMSE

3.2.2 Forecasting results

The price forecast results of each crypto currency are compared with the actual price data and shown in figure 3.2, 3.3 and 3.4. Since the models stopped receiving new data during the forecasting period, degradation in accuracy can be observed. Furthermore, the forecasting results are relatively steady compared to the actual price. In figure 4.5 the forecasting errors are quantified by using NRMSE and average percentage errors.

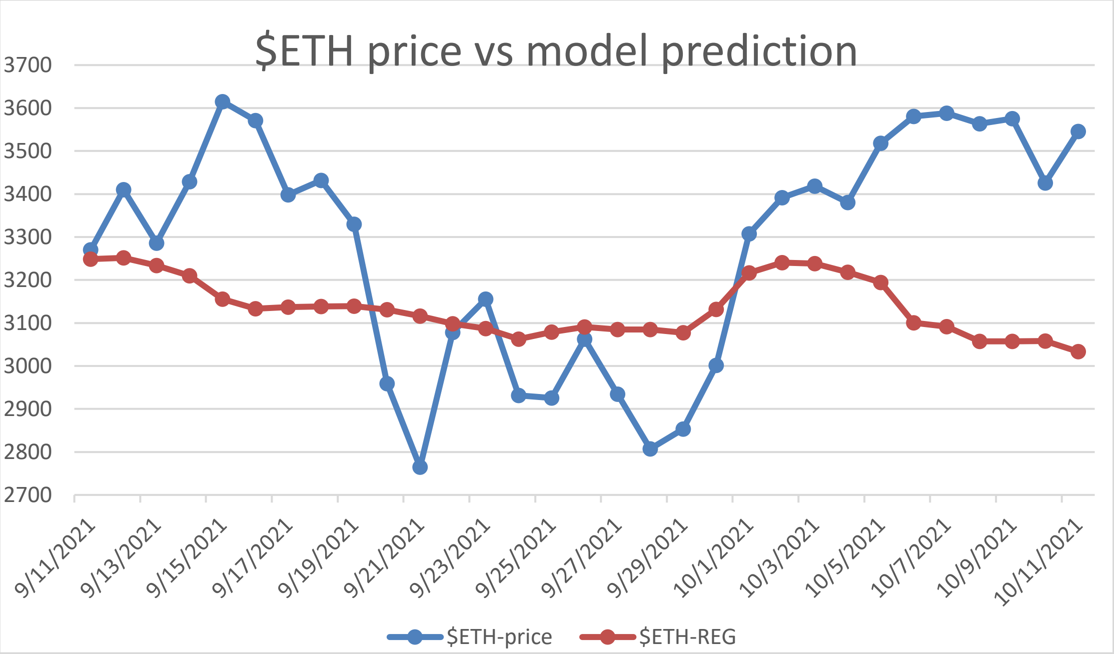


Figure 3.2 $ETH price vs model prediction

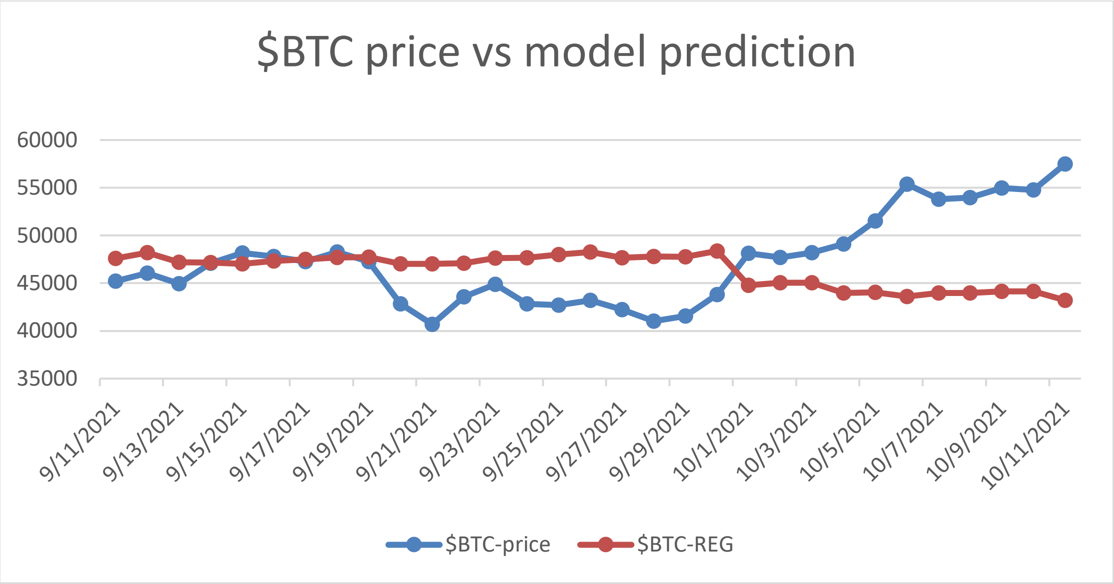


Figure 3.3 $BTC price vs model prediction

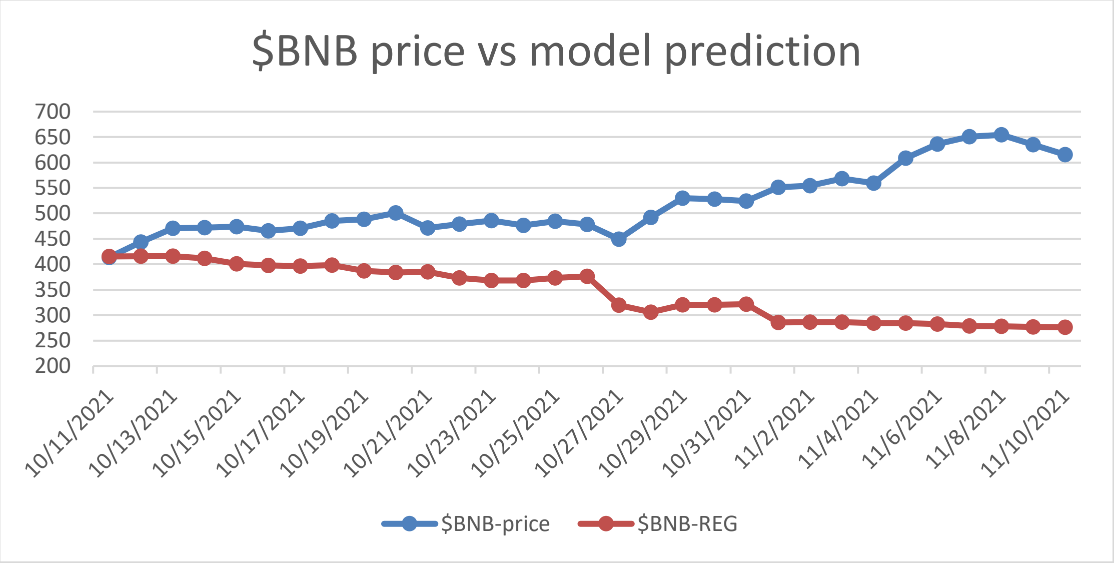
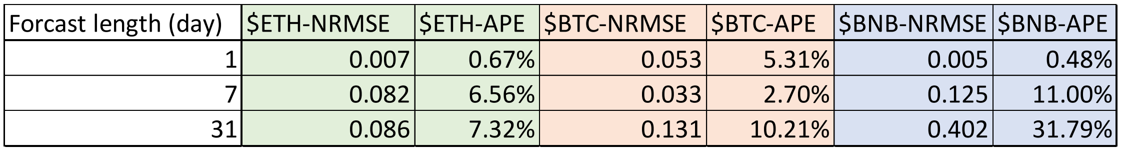


Figure 3.4 $BNB price vs model prediction



Forcast length (day)	$ETH-NRMSE	$ETH-APE	$BTC-NRMSE	$BTC-APE	$BNB-NRMSE	$BNB-APE
1	0.007	0.67%	0.053	5.31%	0.005	0.48%
7	0.082	6.56%	0.033	2.70%	0.125	11.00%
31	0.086	7.32%	0.131	10.21%	0.402	31.79%

Figure 4.5 NRMSE and average percentage error (APE) of the models with different forecast length

CHAPTER IV

TIME SERIES MODELS

4.1 Model description

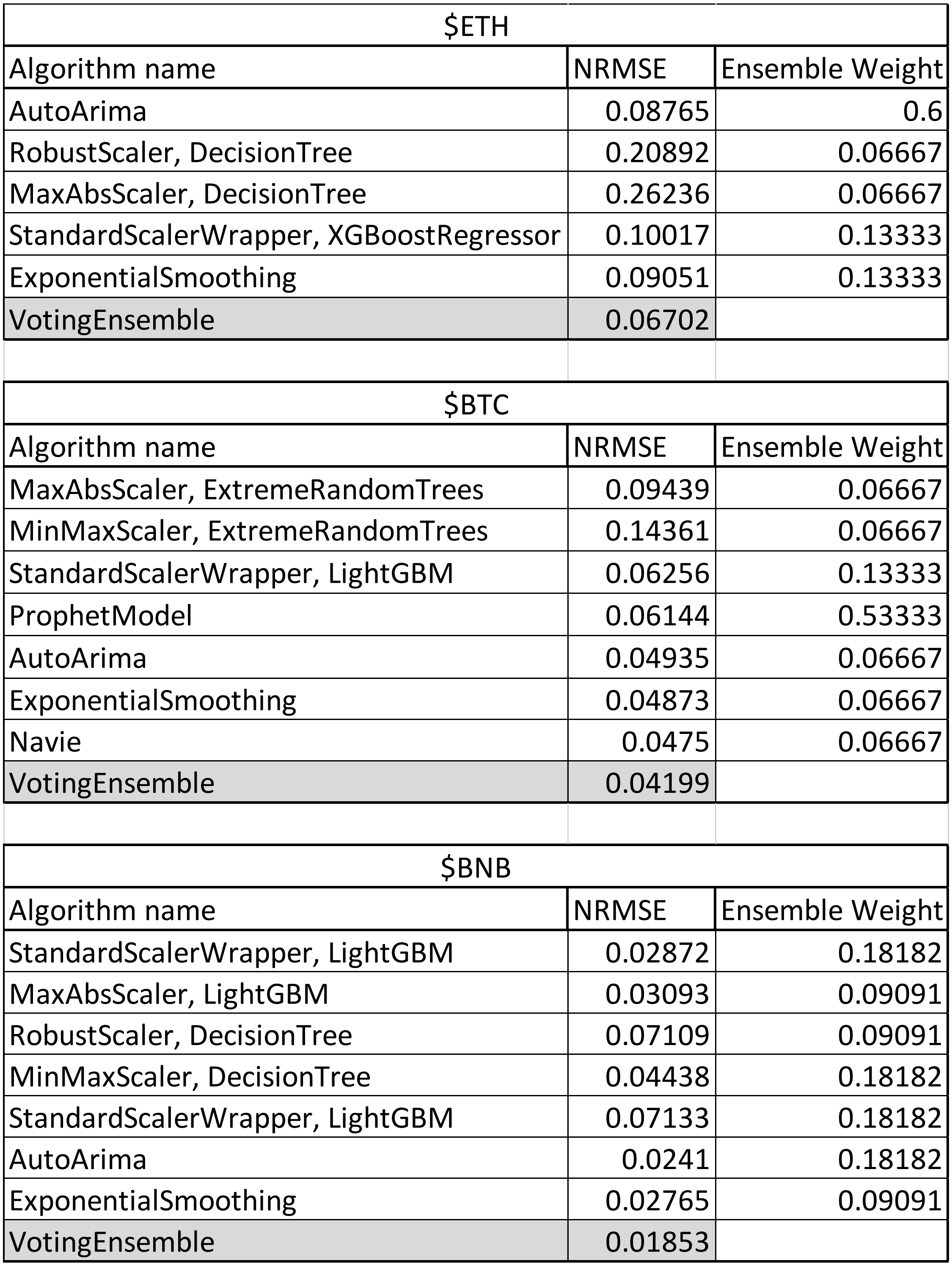
The models in this chapter are built with the AutoML time series analysis tool. In AutoML time series analysis, past time series values are pivoted to become additional dimensions for the regressor with other predictors. This method has shown good results when forecasting sales, exchange rate, and other non-stationary data.

The AutoML is first supplied with the dataset and specified time column and training column ($ETH, $BTC, and $BNB in our case). Subsequently, the time series analysis validation and iteration parameters are set based on the designated computational power and modeling time. Lastly, AutoML trains several time series and regression models based on the given parameters and evaluates (based on NRMSE) then selects the best three models to create the final model using voting ensemble learning. The voting ensemble model is then setup to forecast three different CC data with time frame of one day, one week, and one month.

4.2 Model results

4.2.1 Training results

Figure 4.1 shows the NRMSE values of the best models built with voting ensemble for each CC data and the composition and weights of each ensemble algorithms. Form the data shown, time series ensemble learning incorporates more algorithms in the final models compare to the regression ensemble models. Furthermore, the composition of the algorithms is vastly different among three CC predictions.



$ETH		
Algorithm name	NRMSE	Ensemble Weight
AutoArima	0.08765	0.6
RobustScaler, DecisionTree	0.20892	0.06667
MaxAbsScaler, DecisionTree	0.26236	0.06667
StandardScalerWrapper, XGBoostRegressor	0.10017	0.13333
ExponentialSmoothing	0.09051	0.13333
VotingEnsemble	0.06702	

$BTC		
Algorithm name	NRMSE	Ensemble Weight
MaxAbsScaler, ExtremeRandomTrees	0.09439	0.06667
MinMaxScaler, ExtremeRandomTrees	0.14361	0.06667
StandardScalerWrapper, LightGBM	0.06256	0.13333
ProphetModel	0.06144	0.53333
AutoArima	0.04935	0.06667
ExponentialSmoothing	0.04873	0.06667
Navie	0.0475	0.06667
VotingEnsemble	0.04199	

$BNB		
Algorithm name	NRMSE	Ensemble Weight
StandardScalerWrapper, LightGBM	0.02872	0.18182
MaxAbsScaler, LightGBM	0.03093	0.09091
RobustScaler, DecisionTree	0.07109	0.09091
MinMaxScaler, DecisionTree	0.04438	0.18182
StandardScalerWrapper, LightGBM	0.07133	0.18182
AutoArima	0.0241	0.18182
ExponentialSmoothing	0.02765	0.09091
VotingEnsemble	0.01853	

Figure 4.1 Voting Ensemble and the Ensemble Weight results in NRMSE

4.2.2 Forecasting results

The price forecast results of each crypto currency are compared with the actual price data and shown in figure 4.2, 4.3 and 4.4. With no new data to correct the prediction, degradation in accuracy can be observed. Furthermore, the forecasting results are relatively steady compared to the actual price. In figure 4.5 the forecasting errors are quantified by using NRMSE and average percentage errors.

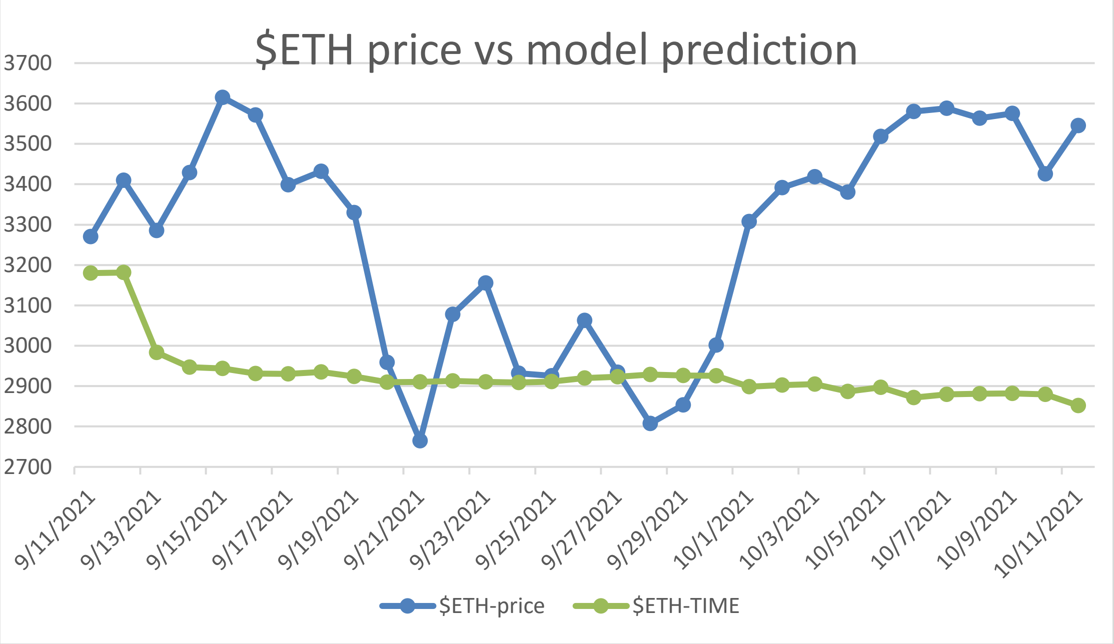


Figure 4.2 $ETH price vs model prediction

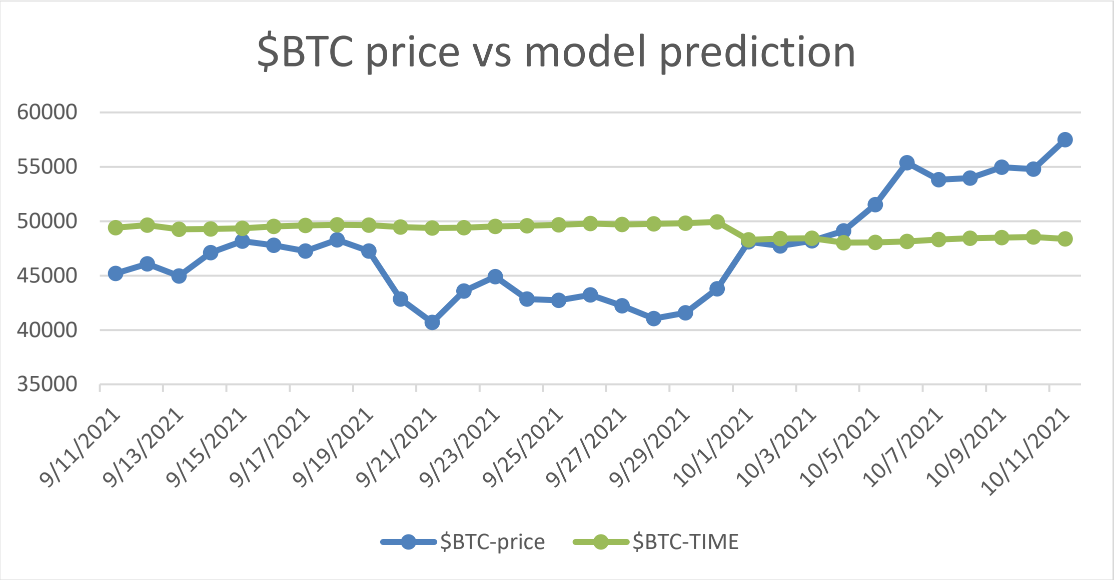


Figure 4.3 $BTC price vs model prediction

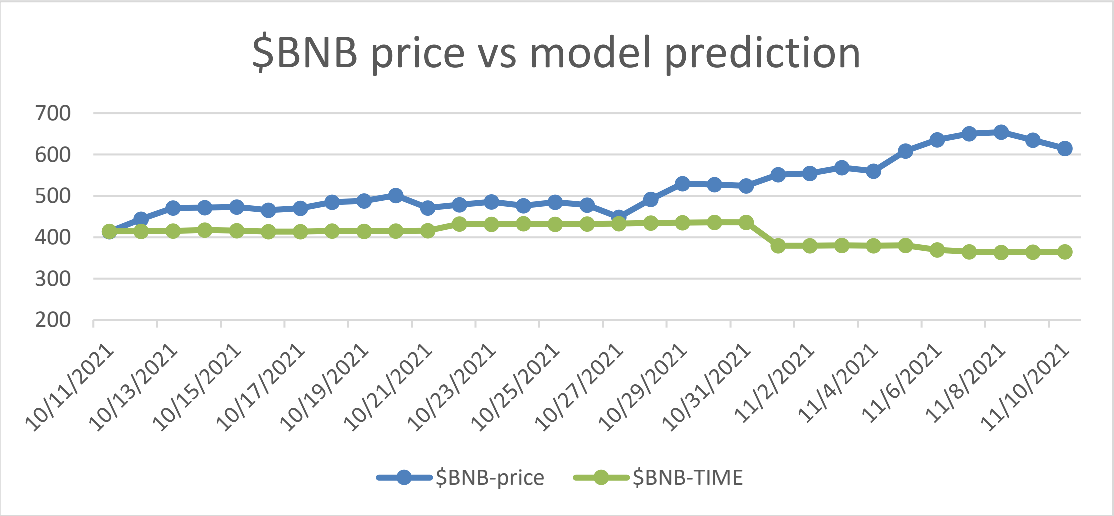
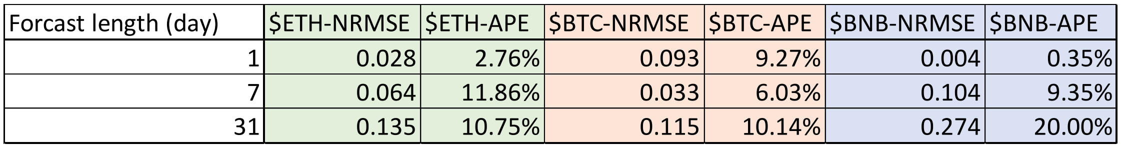


Figure 4.4 $BNB price vs model prediction



Forcast length (day)	$ETH-NRMSE	$ETH-APE	$BTC-NRMSE	$BTC-APE	$BNB-NRMSE	$BNB-APE
1	0.028	2.76%	0.093	9.27%	0.004	0.35%
7	0.064	11.86%	0.033	6.03%	0.104	9.35%
31	0.135	10.75%	0.115	10.14%	0.274	20.00%

Figure 4.5 NRMSE and average percentage error (APE) of the models with different forecast length

CHAPTER V

RESEARCH FINDINGS, RECOMMENDATIONS AND FUTURE STUDY

5.1 Findings and recommendations

The regression models, time series models, with the actual price comparison are shown in figure 5.1, 5.2, and 5.3. Furthermore, the NRMSE and the average percentage error results with three forecasting lengths are shown in figure 5.4. Several useful information can be extracted by examining the graphical and quantitative results, which assists constructing better experiments in the future.

In Figure 5.1 presenting $ETH data, the ensemble regression model result has been around the mid-range of monthly high and low. On the contrary, time series model result has underpredicted the price for the majority of the forecasting period. In Figure 5.2 presenting $BTC data, there is clear resemblance between the ensemble regression model and the ensemble time series model with time series model consistently outputting higher results. The similarity between the models can be a result of having similar ML algorithms in the finial ensemble models. Since during the training phase of time series analysis, the AutoML tool also tested traditional regression models as part of the recommendation system. In Figure 5.3 presenting $BNB data, both models underpredict the value when the $BNB price is on the upward trend. Furthermore, similar to the $BTC data, both models show similar prediction movement with time series model outputting higher results. Gathering from Figure 5.4, in both $BTC and $BNB forecast in the early time the regression models perform better with smaller NRMSE and average percentage error. However, as forecasting length grows, time series analysis shown better predictions.

Both models are useful in predicting price in the early stage, and the predictability degrades as the forecasting length extends. The forecast results can be improved by supplying new CC price after each forecasting period. In addition, long term analysis is not useful during volatile trading period since the forecast outputs are mostly steady when constant correction from the actual data is not available.

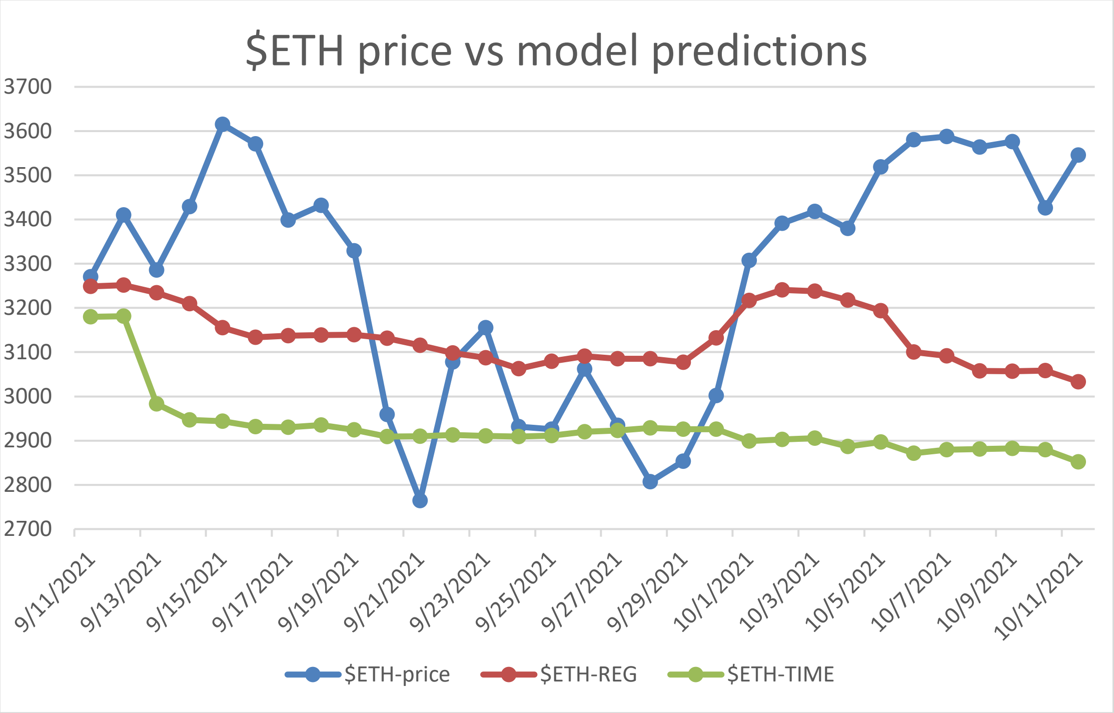


Figure 5.1 $ETH price vs models prediction

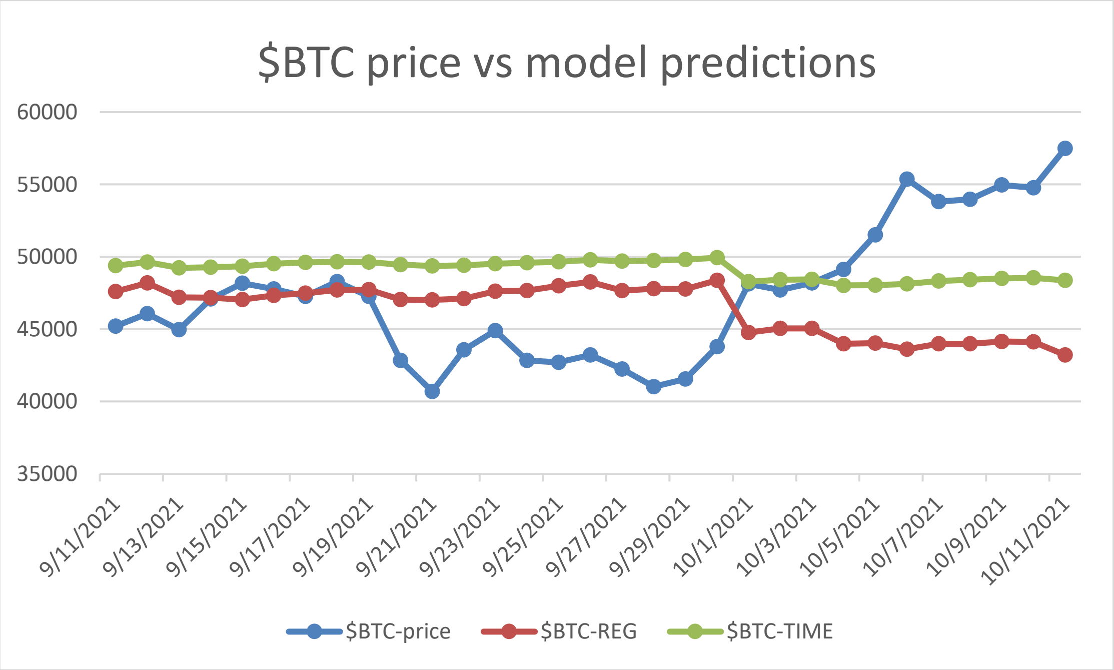


Figure 5.2 $BTC price vs models prediction

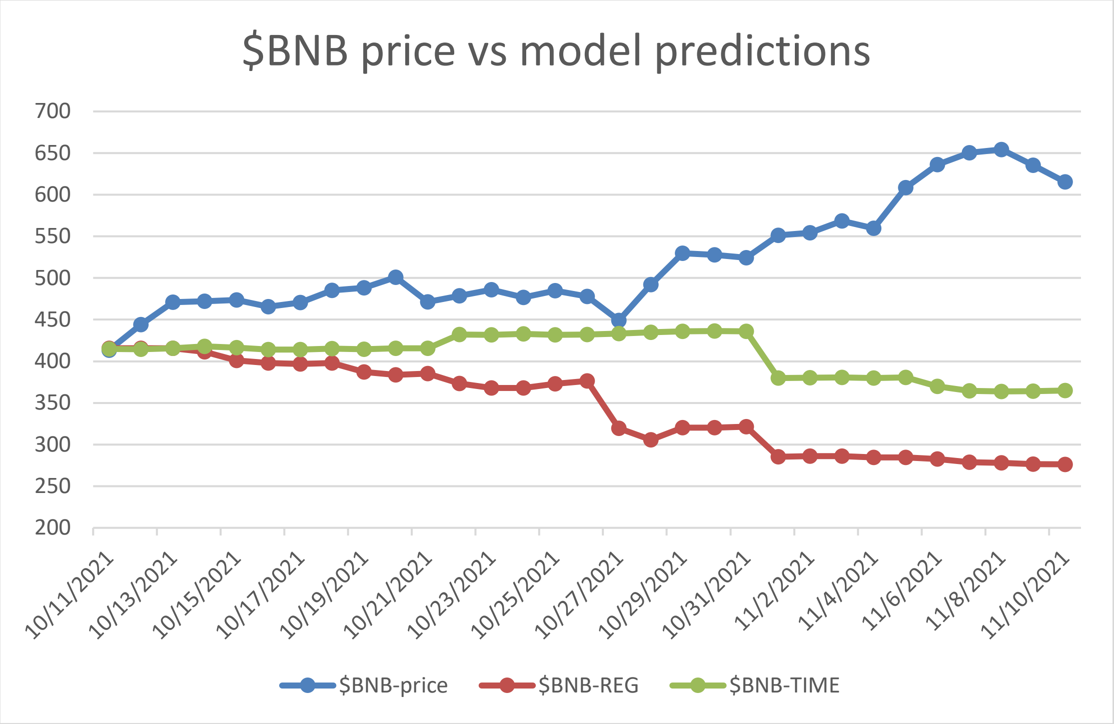
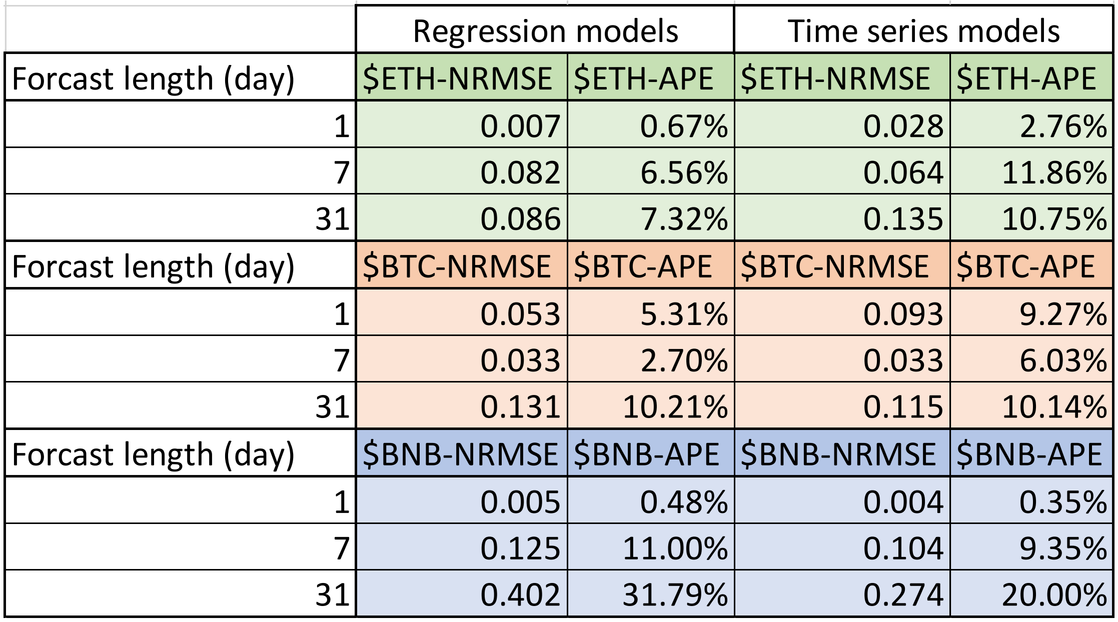


Figure 5.3 $BNB price vs models prediction



	Regression models		Time series models	
Forcast length (day)	$ETH-NRMSE	$ETH-APE	$ETH-NRMSE	$ETH-APE
1	0.007	0.67%	0.028	2.76%
7	0.082	6.56%	0.064	11.86%
31	0.086	7.32%	0.135	10.75%
Forcast length (day)	$BTC-NRMSE	$BTC-APE	$BTC-NRMSE	$BTC-APE
1	0.053	5.31%	0.093	9.27%
7	0.033	2.70%	0.033	6.03%
31	0.131	10.21%	0.115	10.14%
Forcast length (day)	$BNB-NRMSE	$BNB-APE	$BNB-NRMSE	$BNB-APE
1	0.005	0.48%	0.004	0.35%
7	0.125	11.00%	0.104	9.35%
31	0.402	31.79%	0.274	20.00%

Figure 5.4 NRMSE and average percentage error (APE) of regression and time series models with different forecast length

5.2 Future study

The Azure Machine Learning Studio has proven to produce reasonable forecasting results while saving tremendous efforts from implementing individual algorithms to find the best performing models. To better examine the potential of the AutoML, future studies can supply additional data to enhance the forecasting capability. In the case of crypto currency forecasting, supplemental data such as the aggregates trades, sentiment index, volume, trade order flow data, et cetera. Furthermore, beside price prediction, classification analysis can also be useful to forecast the trend of the CC price. Moreover, future study can enhance the prediction accuracy by automatically supply new data to the models for corrections.

In conclusion, Azure Machine Learning Studio is a great tool to simplify basic ML process and quickly discover suitable models to work with. However, it is important to understand that like-kind services are not a one-stop shop for proficient machine learning models. To construct an adequate model requires substantial domain knowledge and understanding of the ML algorithms. With combination of both the services and knowledge of the users can greatly increase the efficiency of the workflow.