ÖZYEĞİN UNIVERSITY
FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE

CS 402
2021 Spring

SENIOR PROJECT REPORT

Crypto-currency Price Prediction Using News and Social Networks Data

By
Barış Karaer
Ertan Ayanlar

Supervised By
Emre Sefer

Declaration of Own Work Statement/ (Plagiarism Statement)

Hereby I confirm that all this work is original and my own. I have clearly referenced/listed all sources as appropriate and given the sources of all pictures, data etc. that are not my own. I have not made any use of the essay(s) or other work of any other student(s) either past or present, at this or any other educational institution. I also declare that this project has not previously been submitted for assessment in any other course, degree or qualification at this or any other educational institution.

Student Name and Surname:	Barış Karaer, Ertan Ayanlar
Signature:	<Signatures of the group members >
Place, Date:	Özyeğin University, 31.05.2021

Abstract

Crypto currencies are digital assets that are created to work as a medium of exchange where each record of the ownership of these coins are stored on a digitalized ledger that uses strong cryptography. These crypto currencies are mostly not controlled by anyone (decentralized control) as opposed to central banking systems. Since it is decentralized, the biggest drivers of volatility in the crypto market are because of the speculation of the investors. These speculations and the volume of attention the crypto market gets, heavily moves the price of crypto currencies. The goal of our project is to design and implement a machine learning model which achieves good price predictions for Bitcoin, the crypto currency that was first created by Satoshi Nakamoto in a white paper which is called “Bitcoin: A Peer-to-Peer Electronic Cash system”, and Ethereum, a decentralized open source blockchain with smart contract functionality that was developed by Vitalik Buterin. The predictions are made using Twitter sentiment, Google Trends data. The Google Trends values are collected as weekly points and Twitter sentiment analysis is collected as hourly data. These values are also preprocessed and used in various ways to help the time series model make better predictions. Different models are created in order to show the significance of shifting data and choosing different features. This project uses different machine learning methods to predict the prices such as Linear Regression, Time Series predictions. In detail, LSTM and XGBOOST is used as time series machine learning models. In addition to the train test splitting technique, different cross validation techniques such as rolling forecast origin and sliding window are used to tune our hyperparameters and in the end, to get the best prediction results out of the time series machine learning models. In depth analysis using metrics and comparisons are made between predicting Ethereum and Bitcoin prices. Then, hourly predictions are made using Twitter sentiment analysis using Linear Regression and Time Series models.

Contents

I. Introduction

Crypto Currency Price Prediction Using Social Networks and News Data is a new project for us, but many academics have implemented predictions on the future prices of Bitcoin. Our approach is to collect data from many different social media / news channels and estimate the market speculation and then, use these features to predict accurate Bitcoin and Ethereum prices using different machine learning algorithms. First, we decided to use different time series models to compare the accuracies, scores and make assumptions on which of them are better to predict the future prices. In our case, we have used Linear Regression and 2 time series algorithms, LSTM and XGBOOST. The data, that is given as independent and dependent variables to the machine learning algorithms, are collected by us. The independent variables are Google Trends, Twitter Sentiment analysis scores. Various news, social media data are collected from the LunarCrush [5] website. The purpose of using these time series models and the selection of data will be explained in detail in the Solution Approach section, the issues we faced while tackling different problems will be explained in this section as well.

For predicting with Google Trends, the dependent variables are weekly closing prices for BTC and ETH. For predicting with the LunarCrush data, the dependent variables are daily closing prices for these cryptocurrencies. These datasets are further explained in the Results and Discussion section. The data that was used, and the code repository can be found on our GitHub repository [9] which is provided in the References. Each model is created with default hyperparameters to predict before tuning. After predicting with appropriate hyperparameters, we have used different cross validation techniques like rolling forecast origin and sliding window to tune our hyperparameters. All these models are analyzed with metrics like Root Mean Squared Error and Mean Absolute Error.

This report gives detailed information about Crypto Currency Price Prediction Using News and Social Networks Data and the outline for the following section is as follows: Second chapter is about the techniques and tools that were used while implementing this project. Third chapter will analyze the problem and explain our solution. Solution Approach section will explain our approach for predicting future prices. The Results and Discussion section will show detailed graphs, metrics, comparisons of the resulting predictions. In the Related Work section, the alternative ways of approaching this kind of a problem will be discussed. Finally, in the Conclusion and Future Work section, our conclusions about our findings will be explained and the future work that can be done about this subject will be discussed.

II. Background

Various kinds of tools and techniques are used for this project and the main ones are as follows:

- **Python**: We have used the Python programming language to use the machine learning libraries and make predictions.
- **Visualizing Libraries**: We have used many visualizing libraries while using Python. These libraries are, MatPlotLib for plotting data, SeaBorn for visualizing the data, Pandas for manipulating data and preprocessing it.
- **SciKit Learn**: This is a library we used to create different machine learning models for our data.
- **Keras and XGBOOST**: These libraries will be used for importing the LSTM and XGBOOST machine learning time series models.
- **Tweepy**: Tweepy library is used to provide access to the Twitter API and handle rate-limiting limitations of Twitter API.
- **Various Open Source API's**: We have used many API's to collect data from social media. The CoinMarketCap API is used to collect the prices of crypto currencies. The Twitter API is used to collect tweets from the Twitter application about selected crypto currencies.
- **LunarCrush**: This website is a tool where anyone can browse and see the graphs of social media and news sentiment corresponding to various cryptocurrencies.
- **Jupyter Notebook**: The Jupyter Notebook tool is used for code implementations. A visualization can be seen at Appendix-3.
- **IntelliJ PyCharm**: This IDE is used in the development of the data collection module. A visualization can be seen at Appendix-2.
- **Google Trends**: This was used to access the popularity of some keywords that are searched throughout the internet. For Bitcoin, the keyword “Bitcoin” was used to collect the popularity data.
- **Azure Sentiment Analysis**: This tool was used to predict if a certain Tweet is a positive tweet about a certain crypto. If the outcome of this analysis is negative or positive, it labels our data accordingly.
- **Azure Language Detection**: This tool was used to detect the languages of fetched tweets.

III. Problem Statement

Crypto Currency Price Prediction Using Social Networks and News Data is a new project. The main goal of our senior project is to create a good machine learning model that predicts the future prices of a selected crypto currency, Bitcoin and Ethereum. This project is aimed to create different machine learning models and lastly acquire minimum error metrics while making this prediction. This project was offered by the faculty member, Dr. Emre Sefer. Mr Sefer will be our supervisor throughout the whole project.

The volatility of the crypto currency market is widely known but the speculation of the investors and people all around the world majorly steers the prices of these currencies. Our main goal is to find data that has good correlation with the corresponding cryptocurrency and predict the future price of these values based on these datasets.

Assumptions

Our assumption is that after analyzing the data from the social networks, past prices, trends in the web and the sentiment of the tweets that are sent in the Twitter platform, we can make an accurate prediction about the future prices of these currencies. Although the sentiment of the market is very volatile and hard to document, we assume that the data we collect from Google Trends, Twitter, LunarCrush accurately describes the sentiment of the market conditions.

Constraints

Our constraint for this project is that we will be using different types of machine learning techniques to make predictions, these techniques are called Linear Regression and Time Series (LSTM and XGBOOST). Furthermore, there are some constraints on the data that we collected. The BTC and ETH closing prices are fixed to certain dates. Some past years are not used because they are not necessary. Similarly, the google trends data is collected based on the collected prices. Because of this, average weekly closing prices are calculated for ETH and BTC. Furthermore, sentiment data that was collected from the LunarCrush website is only 2 years at total, which was enough for us to make a prediction. The Tweets that are collected from the Twitter application will be for the past 2 months, it is nearly impossible to collect all the Tweets about BTC or ETH for the past 5 years. This way, whether good or bad we can make deductions for the market sentiment. But essentially, the data we acquired from LunarCrush was far superior to the data we collected from the Twitter API.

IV. Solution Approach

DESIGN DECISIONS

Since our project is separated into two modules, one that is responsible for data collection and other that is responsible for price prediction, they should be examined separately.

Part 1: Data Collection Module

Data collection module is responsible for collecting the required data to feed the machine learning model that is used for predicting the Bitcoin price. Tweets from Twitter, past Bitcoin and Ethereum price data, Google Trends data, news and social media data for a specific cryptocurrency is collected through this module. This module is coded in Python.

In our first iteration, we decided to collect the tweet ourselves with the help of "Tweepy" module and run sentiment analysis on them using Azure's Sentiment Analysis Module. The "Tweepy" module is used on several papers [10]. We filter the tweets based on the language ('EN') and a date interval. We do not have to specify a language since we are using Azure's Text Analytics' language detection endpoint to figure out the language of the tweet that is needed for the sentiment analysis later. We run sentiment analysis over the collected tweets using the Microsoft Azure Cloud services' Text Analytics API, by sending a POST request. To achieve this, first we need to determine the language of each tweet via Azure's Language Detection service. Tweets that are currently stored as an array needs to be converted to a format that Azure's Language Detection endpoint can understand. We convert by placing the tweet array in a Python dictionary with key 'data' and pass this dictionary to a function that organizes the contents in a format Azure's Language Detection endpoint understand. Now that we have the request body of the Azure's Language Detection endpoint ready, we send a POST request to that endpoint and get a JSON object as response containing the detected languages of the tweets and their corresponding id's. The request body for Azure's Sentiment Analysis endpoint is prepared in the same fashion by placing tweets array under a 'documents' key of a dictionary. A POST request is made to the sentiment analysis endpoint, and we get a JSON object as response containing each tweets sentiment score. A mean sentiment score is calculated and used on our price prediction module. However, because of the size limitations of POST requests and Azure's Text Analytics endpoint, we are not able to send all tweets for analysis at once. To overcome this, we are splitting the calls to this endpoint in separate requests containing the half of the total tweets. Sadly, Twitter API only allows its users to only access past 7 days historical data at most and because of this limitation we decided to collect 1000 tweets daily starting from January 2021 and store them as CSV. Past prices of Bitcoin are collected using the CoinMarketCap website and stored in a CSV file. Google Trends data is downloaded directly from Google's official website.

After we collected 2 months of tweet data and ran sentiment analysis on them, we decided to go with another approach to obtain our Twitter data and sentiment analysis since we figured out that 1000 tweets daily were not enough to come up with a 'correct' sentiment score to determining the overall sentiment of the day and the Azure's Free Tier services were not enough to process all this data. At first, we replaced Azure's Sentiment Analysis with VaderSentiment, an open-source sentiment analysis library but the end results were not encouraging still because of the low volume of tweets we were able to collect from Twitter. Then we decided to use datasets from Kaggle containing Tweets regarding Bitcoin. First dataset we used contains sentiment scores of the tweets but the values of the sentiment scores are very small values like 0.09 and it did not have a correlation with the closing price of the Bitcoin over those dates. So, we decided to recalculate the sentiment scores using VaderSentiment. The first data set contains 17.7 million Tweets [4] over the time period of 01/08/2017 and 21/01/2019. It is also an hourly dataset, reporting tweet stats for that hourly period like total volume of negative, positive, neutral tweets and the highest price Bitcoin had during the period and the last price during the period. On this dataset we found the reported total volume of positive tweets on the dataset had the best correlation with the closing prices of Bitcoin and we decided to use it as feature both for our hourly and daily price prediction analysis. We tried another data set from Kaggle, which only contained Tweets and ran sentiment analysis over those 16 million tweets [6] with VaderSentiment again. This time correlation of our sentiment analysis scores were low, around 0.30 so we decided not to use this data set. Our final and most recent approach was to discover how popular cryptocurrency analytics websites like LunarCrush [5] work, and we extracted their analytics data and decided to use it in our Price Prediction module. LunarCrush is a website targeting social media analysis and its impacts on the cryptocurrencies. We found out that they retrieve the data for the charts on their website by sending REST requests to their API and we analyzed their API endpoints and managed to come up with a request that let us retrieve 720 days of data containing price data and social scores for the social platforms Twitter, Reddit, and overall news. For Twitter, the retrieved data had a 'tweets' field reporting the volume of the tweets for that day and 5 different sentiment scores for that day under 'tweet_sentiment1', 'tweet_sentiment2' and so on. Other social data contained in the response of this API request were number of Reddit posts, Reddit comments, social score and contributors and a 'news' field reflecting count of the news for the requested cryptocurrency on that day. We found high correlation of these features with the closing prices of Bitcoin and Ethereum, so we used this data for our Twitter feature on LSTM and XGBOOST models.

Part 2: Price Prediction Module

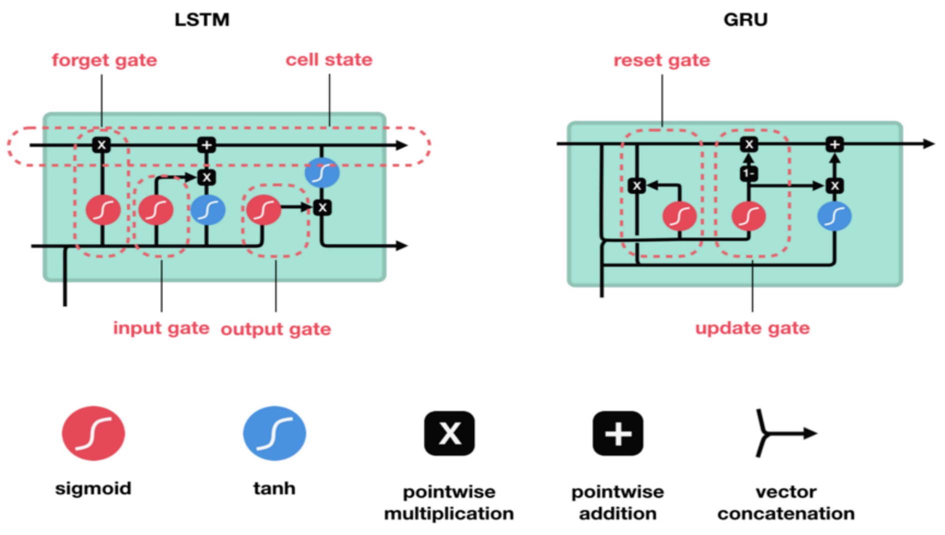
The Price Prediction Module is responsible for predicting the Bitcoin and Ethereum price with different machine learning models. We have used Linear Regression, LSTM and XGBOOST machine learning time series models to predict the prices. The closing prices of these cryptocurrencies are collected from the CoinMarketCap website. We can separate the price prediction to 2 subparts. One subpart uses Google Trend values as independent values and predicts based on a mixture of these values and shifted weekly prices. Second subpart uses LunarCrush Social Media and News dataset mentioned earlier and predicts based on these different features.

In the first subpart, the Google Trends data gives only specific scores to weeks and it does not give daily scores, because of this we have decided to calculate the weekly average closing prices of Bitcoin and Ethereum to use in our dataset. After the predictions for the Linear Regression are made and different metrics are calculated, LSTM time series model is used to make predictions for the Bitcoin and Ethereum closing prices. Different hyper parameters are used to create different results and find the most optimal one. After these predictions, Mean Absolute Error and Root Mean Squared Error is used to find the most optimal model and compare between different features. Same process is applied to a different time series model called XGBOOST for Bitcoin and Ethereum. Different graphs and plots are given to further compare the results of these models.

Time Series Prediction with LSTM

We chose the Time Series machine learning algorithm LSTM (Long-short term memory) to predict prices of Bitcoin and Ethereum. This artificial RNN (Recurrent Neural Network) architecture is used to process entire sequences of data and make predictions. LSTM networks are widely known to be suited for classifying, processing and making predictions based on time series data, so this kind of a model is very well suited for predicting stock market prices or crypto currency prices where speculations are always occurring in the market and the data is based on time. Again, the LSTM model is quite useful when sentiment data are used for predicting. The works of Vo, A. [7] showed us that when sentiment data are used as features for crypto prices, the predictions of the LSTM model would be very accurate to the real closing prices.

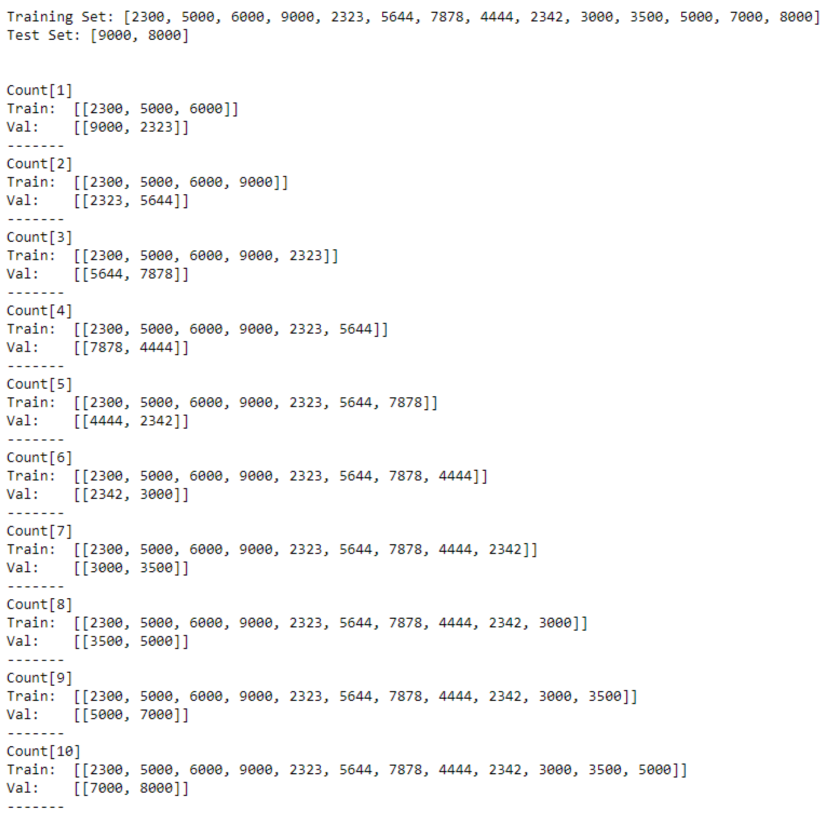
Although RNN's are very useful, it suffers from short term memory. If a data is in a very long sequence, then it will have a very hard time carrying information from earlier time steps to later ones. So, predicting over long periods of time would not end in an accurate prediction. On the other hand, LSTM was created as a solution to this short-term memory problem. The internal gates of the LSTMs can regulate the flow of information. These gates allow the model to learn which data in the sequences are important so it can keep or throw away the unnecessary ones. A simple representation of LSTM and GRU which is like an LSTM can be seen below.



Time Series Prediction with XGBOOST

As an alternative to LSTM, we chose the Time Series machine learning algorithm XGBOOST to predict prices of Bitcoin and Ethereum. This machine learning algorithm is an efficient implementation of stochastic gradient boosting for classification and regression problems. XGBOOST is also known to be used for time series forecasting, so this kind of model is well suited for predicting stock market prices or crypto currency prices where speculations are always occurring in the market and the data is based on time.

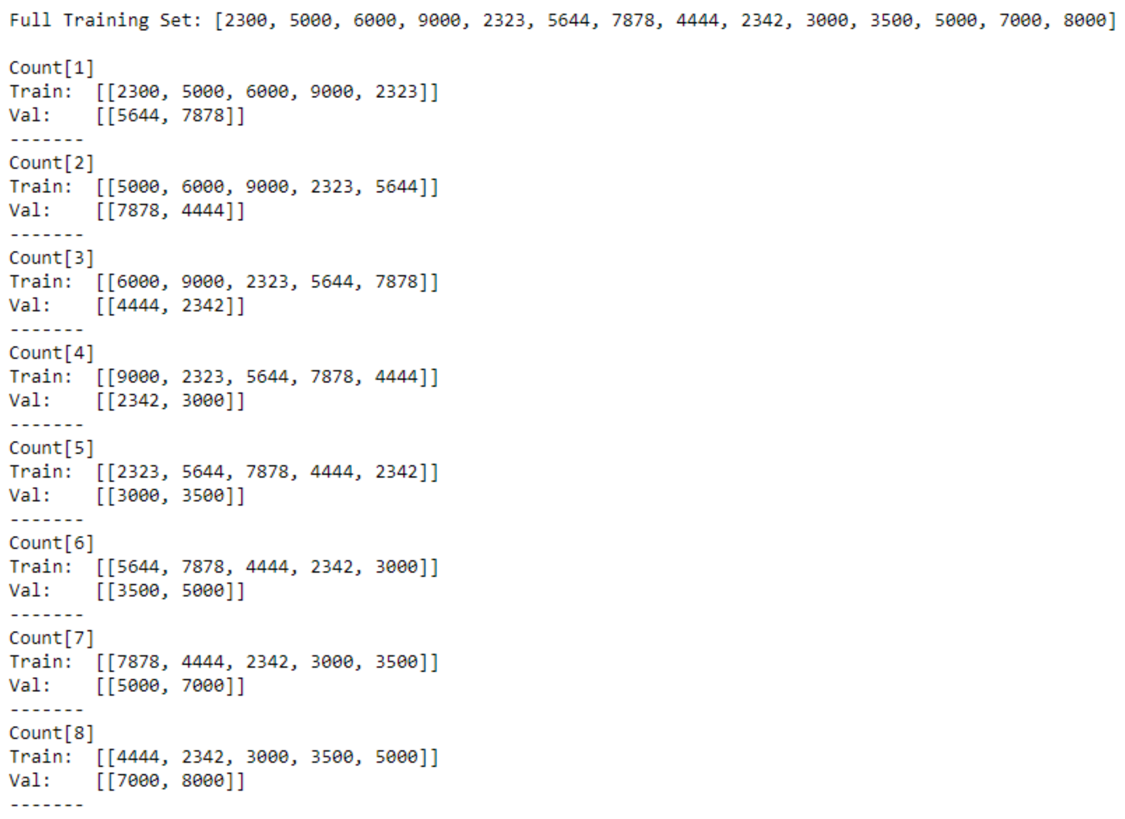
Additionally, 2 different cross validation techniques are used. These techniques are implemented from scratch in Python, which the code repository can be found in the References part of this paper. First one is rolling forecast origin and the second one is sliding window. These techniques are used to find the right hyperparameters for LSTM or XGBOOST models. Different combination of parameters is used for the time series models. The results will be explained in detail in the Results and Discussion part of this paper. To briefly explain the cross-validation techniques, rolling forecast origin is an evaluation technique that updates the forecast origin successively and the forecasts are produced from each origin. The important parameters for the rolling origin function are minimum train size and horizon which affects our train and validation set. These parameters of the rolling forecast origin are fixed to make the comparison between different models easier. The minimum train size is set as 10 and the horizon is set as 2. As an example, Figure 1 shows the training set, test set and the train-validation pair that is created from the training and test set.



```
Training Set: [2300, 5000, 6000, 9000, 2323, 5644, 7878, 4444, 2342, 3000, 3500, 5000, 7000, 8000]
Test Set: [9000, 8000]


Count[1]
Train:  [[2300, 5000, 6000]]
Val:    [[9000, 2323]]
-------
Count[2]
Train:  [[2300, 5000, 6000, 9000]]
Val:    [[2323, 5644]]
-------
Count[3]
Train:  [[2300, 5000, 6000, 9000, 2323]]
Val:    [[5644, 7878]]
-------
Count[4]
Train:  [[2300, 5000, 6000, 9000, 2323, 5644]]
Val:    [[7878, 4444]]
-------
Count[5]
Train:  [[2300, 5000, 6000, 9000, 2323, 5644, 7878]]
Val:    [[4444, 2342]]
-------
Count[6]
Train:  [[2300, 5000, 6000, 9000, 2323, 5644, 7878, 4444]]
Val:    [[2342, 3000]]
-------
Count[7]
Train:  [[2300, 5000, 6000, 9000, 2323, 5644, 7878, 4444, 2342]]
Val:    [[3000, 3500]]
-------
Count[8]
Train:  [[2300, 5000, 6000, 9000, 2323, 5644, 7878, 4444, 2342, 3000]]
Val:    [[3500, 5000]]
-------
Count[9]
Train:  [[2300, 5000, 6000, 9000, 2323, 5644, 7878, 4444, 2342, 3000, 3500]]
Val:    [[5000, 7000]]
-------
Count[10]
Train:  [[2300, 5000, 6000, 9000, 2323, 5644, 7878, 4444, 2342, 3000, 3500, 5000]]
Val:    [[7000, 8000]]
-------
```

Fig. 1. Rolling forecast origin training set, test set, train-validation pair

The sliding window is a very well-known time series cross validation technique where a dataset is split into several pairs and these pairs are created by the sequences taken at a successive equally spaced points in time. Important parameters that affect our train-validation pairs are window size and horizon. These parameters are fixed just like rolling forecast to compare our results between different models. The window size is set as 5 and the horizon is set as 2 for all our models. Figure 2 explains the sliding window technique briefly.



```
Full Training Set: [2300, 5000, 6000, 9000, 2323, 5644, 7878, 4444, 2342, 3000, 3500, 5000, 7000, 8000]

Count[1]
Train:  [[2300, 5000, 6000, 9000, 2323]]
Val:    [[5644, 7878]]
-------
Count[2]
Train:  [[5000, 6000, 9000, 2323, 5644]]
Val:    [[7878, 4444]]
-------
Count[3]
Train:  [[6000, 9000, 2323, 5644, 7878]]
Val:    [[4444, 2342]]
-------
Count[4]
Train:  [[9000, 2323, 5644, 7878, 4444]]
Val:    [[2342, 3000]]
-------
Count[5]
Train:  [[2323, 5644, 7878, 4444, 2342]]
Val:    [[3000, 3500]]
-------
Count[6]
Train:  [[5644, 7878, 4444, 2342, 3000]]
Val:    [[3500, 5000]]
-------
Count[7]
Train:  [[7878, 4444, 2342, 3000, 3500]]
Val:    [[5000, 7000]]
-------
Count[8]
Train:  [[4444, 2342, 3000, 3500, 5000]]
Val:    [[7000, 8000]]
-------
```

Fig. 2. Sliding window training set, train-validation pair

LSTM and XGBOOST are known to be overfitting while training the dataset. These 2 cross validation techniques are used to prevent the overfitting problem that may occur while predicting the Bitcoin and Ethereum prices using these time series models. The second subpart uses the LunarCrush website to collect Social Media Sentiment and News data to predict prices of BTC and ETH using the 2 time series machine learning models mentioned earlier.

Usage of Tools and Techniques

All the tools and libraries that are mentioned on the Background section were used. Our code base is fully in Python. Our project is separated in two modules, the data collection module that is used for retrieving past price of bitcoin and the tweets on the #bitcoin hashtag. Tweepy module is used for retrieving the tweets. Azure’s Text Analytics service is used for sentiment analysis and language detection of the tweets. Tensorflow, Scikit Learn and Keras library are used for price prediction through machine learning. LunarCrush and Kaggle websites are used to collect data for our time series models.

Technical, Operational and Financial Feasibility

Financial Feasibility

Our implementation only requires a mid-tier computer and an internet connection to fetch and process the required data and machine learning operations. We do not need to use cloud computing to obtain our data, and there was no need for any expensive equipment to implement this project.

Technical Feasibility

Our implementation depends on the Tweepy module for twitter data collection, Coinbase API for past prices of the BTC, ETH and Google Trends website for the trends data. LunarCrush website is needed to collect the data of Social Media Sentiment and News about the corresponding cryptocurrency. Tweepy module can be easily replaced, and Twitter’s API can be directly used, another provider can be found for past prices of Bitcoin and for replacing Google Trends, other analytics tools can be used. There are many other Social Media analysis tools for cryptocurrencies, but we found LunarCrush to be the most useful one.

Engineering Standards

Modularization approach is used so that our implementation is divided into two modules; data collection and price prediction module. Processed data is stored as CSV.

Knowledge and Skill Set

We benefited from various courses from Özyeğin University. Main and the most important courses are as follows:

- CS 454 - Introduction to Machine Learning
- CS 452 - Data Science
- CS 320 - Software Engineering
- CS 321 - Programming Languages
- CS 202 - Database Management
- CS 201 - Data Structure
- CS 102 - Object-Oriented Programming

All Computer Science courses helped us to learn another subject, but the key lessons are Data Science and Machine learning courses we take at Özyeğin University.

V. Results and Discussion

In our first iteration, Tweepy library was used in order to access Twitter API and fetch tweets containing a hashtag, #bitcoin in our case. This library allowed us to fetch large numbers of tweets since we configured it to wait when we are rate-limited by Twitter and then continue its job to fetch the tweets, this way we did not have to manage the rate limitations of the Twitter API. On these tweets, Microsoft Azure's Cognitive Services, the Text Analytics service is used to analyze the positiveness of the tweets (sentiment analysis) to get a general idea about what people on Twitter think about the target cryptocurrencies. This way, the speculation of the market is analyzed. Then the tweets are formatted and converted to JSON for Azure Text Analytics service to understand, languages for these tweets are detected by the Azure Language Endpoint, then the formatted tweets are combined with the language data and sent to the Azure Sentiment Analysis endpoint. By doing so, we acquire a list of tweets, each tweets sentiment score and a mean sentiment score of all the tweets that are fetched for the specified time range and then we output them to an CSV file. Below some sample data can be examined.

Mean Sentiment Score between dates 2020-12-01 - 2020-12-09 with data consisting of 9000 tweets: 0.549054

Positive sentiment scores:

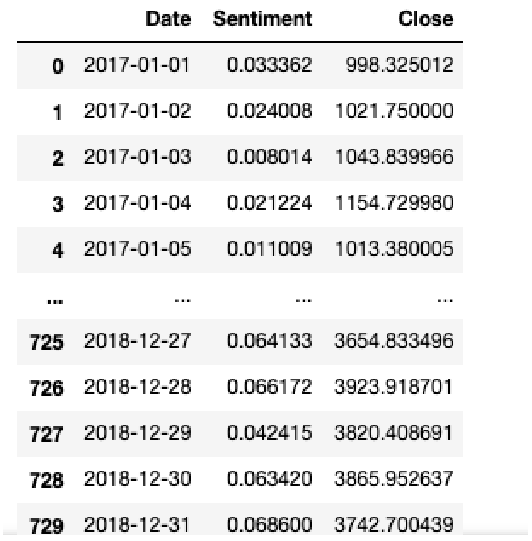
id	date	text	score
1.34 E+18	2020-12-08-23:31:38	The year is 2020 the month is December and you could be earning triple interest paid in BitcoinEarn X3 in BTC	0.999962568
1.34 E+18	2020-12-08-23:49:47	Happy to see this tall green line for btc Lets keep the buy volume comingcrypto btc bitcoin omg link ren	0.999139607
1.34 E+18	2020-12-08-23:07:33	buy the dip What a blessing to buy bitcoin on discount right before christmas	0.994126081

Negative sentiment scores:

1.34 E+18	2020-12-08-22:28:46	Sometimes Bitcoin price movement make me tired	0.092111319
1.34 E+18	2020-12-08-23:22:30	Sell all Bitcoin now dip is coming	0.091784805

Sadly, Twitter API limits fetching of historical data (tweets) are older than 7 days. To overcome this, we have started fetching 1000 tweets each day from Twitter and storing them as CSV since January 2021. After we had enough data collected, we ran sentiment analysis on them using VaderSentiment library, however the results were not good enough to use as a feature because we were only able to collect a very low volume of tweets when compared with the total volume of tweets regarding Bitcoin, potentially tens of thousands every day.

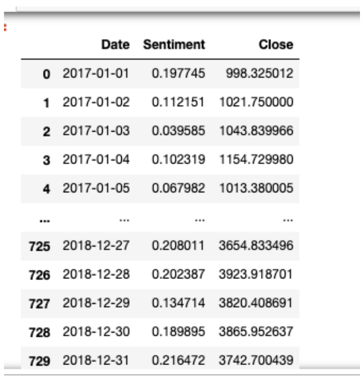
Then we decided to continue with open source Kaggle datasets “Bitcoin tweets – 16M tweets” by Kaggle user alaix14 and “Bitcoin 17.7 million Tweets and price” [4] by Jaime Badiola. The first data set “Bitcoin tweets – 16M tweets” [6] did not include sentiment analysis scores of the tweets so we ran sentiment analysis on them using VaderSentiment library, for tweets ranged in 2017-01-01 to 2018-21-31. We also converted the dataset to a daily format since our predictions were targeted at daily price analysis. After we calculated sentiment scores for each tweet, we split the dataset into two different datasets, one containing tweets with neutral sentiments eliminated and other with included, to measure if tweets with neutral sentiment scores lowers the sentiment score for that day. In figure below sample sentiment analysis results with neutral sentiments included can be seen.



	Date	Sentiment	Close
0	2017-01-01	0.033362	998.325012
1	2017-01-02	0.024008	1021.750000
2	2017-01-03	0.008014	1043.839966
3	2017-01-04	0.021224	1154.729980
4	2017-01-05	0.011009	1013.380005
...	...	...	...
725	2018-12-27	0.064133	3654.833496
726	2018-12-28	0.066172	3923.918701
727	2018-12-29	0.042415	3820.408691
728	2018-12-30	0.063420	3865.952637
729	2018-12-31	0.068600	3742.700439

Fig. 1. Sample Sentiment Analysis Results

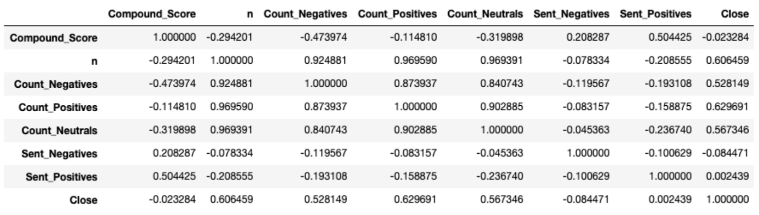
Correlation score for sentiments and closing price of dataset with neutral sentiments included is as mentioned before lower than we expected, 0.31. Figure below contains sample sentiment analysis results for the dataset with neutral sentiments eliminated. When compared with the dataset including neutral sentiments, we can see that sentiment score average of a day is much higher.



	Date	Sentiment	Close
0	2017-01-01	0.197745	998.325012
1	2017-01-02	0.112151	1021.750000
2	2017-01-03	0.039585	1043.839966
3	2017-01-04	0.102319	1154.729980
4	2017-01-05	0.067982	1013.380005
...	...	...	...
725	2018-12-27	0.208011	3654.833496
726	2018-12-28	0.202387	3923.918701
727	2018-12-29	0.134714	3820.408691
728	2018-12-30	0.189895	3865.952637
729	2018-12-31	0.216472	3742.700439

Fig. 2 Sample Sentiment Analysis with neutral sentiments eliminated

As expected, correlation score for sentiments and closing price of dataset with neutral sentiments eliminated is higher than the dataset with neutral sentiments included, 0.33. Thus, we decided to use this dataset instead. The second Kaggle dataset, “Bitcoin 17.7 million Tweets and price”, did not contain tweets since sentiment analysis was already ran on the tweets, it contained compound sentiment scores, count of negative, positive, and neutral tweets for tweets collected on each hour. On this dataset, we found that Count of Positive tweets had the highest correlation score with the closing price of Bitcoin, resulting in a correlation score of 0.629.



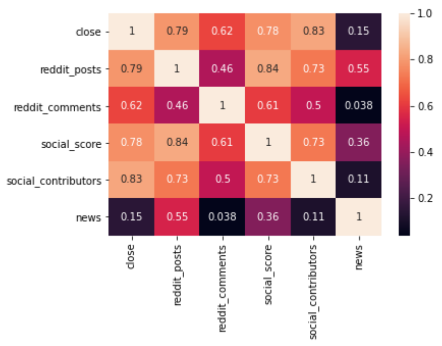
	Compound_Score	n	Count_Negatives	Count_Positives	Count_Neutrals	Sent_Negatives	Sent_Positives	Close
Compound_Score	1.000000	-0.294201	-0.473974	-0.114810	-0.319898	0.208287	0.504425	-0.023284
n	-0.294201	1.000000	0.924881	0.969590	0.969391	-0.078334	-0.208555	0.606459
Count_Negatives	-0.473974	0.924881	1.000000	0.873937	0.840743	-0.119567	-0.193108	0.528149
Count_Positives	-0.114810	0.969590	0.873937	1.000000	0.902885	-0.083157	-0.158875	0.629691
Count_Neutrals	-0.319898	0.969391	0.840743	0.902885	1.000000	-0.045363	-0.236740	0.567346
Sent_Negatives	0.208287	-0.078334	-0.119567	-0.083157	-0.045363	1.000000	-0.100629	-0.084471
Sent_Positives	0.504425	-0.208555	-0.193108	-0.158875	-0.236740	-0.100629	1.000000	0.002439
Close	-0.023284	0.606459	0.528149	0.629691	0.567346	-0.084471	0.002439	1.000000

Fig. 3 17.7 million Tweets Dataset

Although these Kaggle datasets correlation score were better than our previous analyses we researched for better datasets used by popular analytics websites that can be found online, like LunarCrush. LunarCrush's dataset were collected through their API, by modifying API request parameters we were able to get social sentiment data of 2 years, from current day to past 720 days. We achieved extremely good correlation results from this data, some features hitting up to 84 % correlation score for Bitcoin and 94 % for Ethereum. We decided to use this dataset on our Price Prediction library to support Twitter and other social sentiment scores.

Bitcoin Closing Price Prediction Using LunarCrush Data with LSTM and XGBOOST

The dataset of LunarCrush starts from 25-05-2019 and ends at 12-05-2021. We have used different social media features to predict the prices of Bitcoin. First graph shows news and some social media features, second graph shows the Twitter volume and sentiment features we retrieved from LunarCrush. When we inspect the correlation of these features, we see very high values such as 83 % for social contributors, and 78 % for reddit posts. In detail, the features for social media and news are reddit posts, reddit comments, social score, social contributors, and news. The features for Twitter are the Twitter volume and different sentiment analysis results.



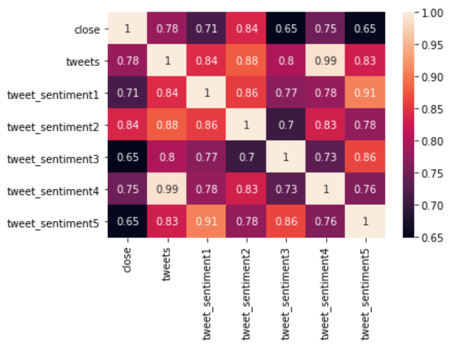


Fig. 4 and 5 BTC Correlation values for LunarCrush Dataset

We have implemented 2 BTC predictions with the LunarCrush dataset. One where we use only the social media features and shifted prices and for the other one, we use strictly only Twitter features and shifted prices. There are at total 610 train values and 80 test values that was split from the data frames mentioned earlier. The graphs of the LSTM predictions can be found below. The predictions of both datasets are quite similar. The RMSE value of Twitter dataset is 2108.44. The MAE value is 1622.68. The RMSE value of News dataset is 2133.08 and the MAE value is 1755.41.

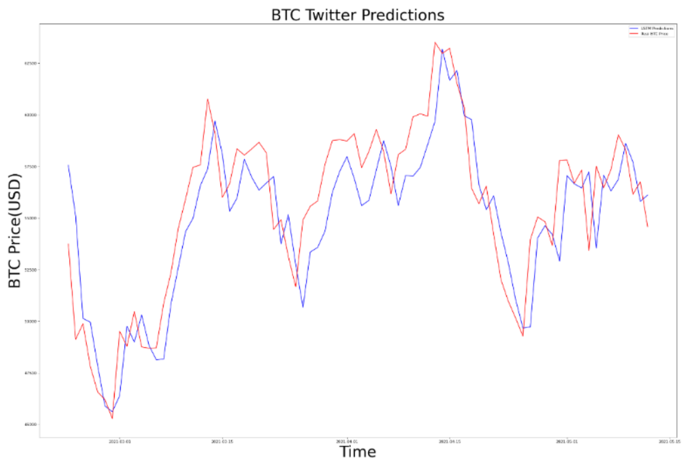


Fig. 6 BTC LSTM LunarCrush Twitter Dataset Prediction Graph

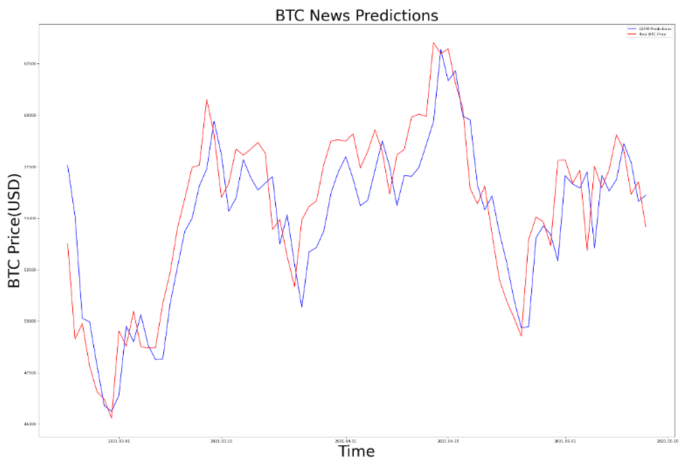


Fig. 7 BTC LSTM LunarCrush News Dataset Prediction Graph

The XGBOOST predictions for BTC can be seen below. First, we use only the social media features and shifted prices and for the other one, we use strictly only twitter features and shifted prices. There are at total 610 train values and 80 test values that was split from the data frames mentioned earlier. The RMSE value of Twitter dataset is 18559.93. The MAE value is 17605.28. The RMSE value of News dataset is 18838.85 and the MAE value is 17647.51. We can see that the XGBOOST predictions are similar to each other as well, but the LSTM predictions are much superior compared to XGBOOST.

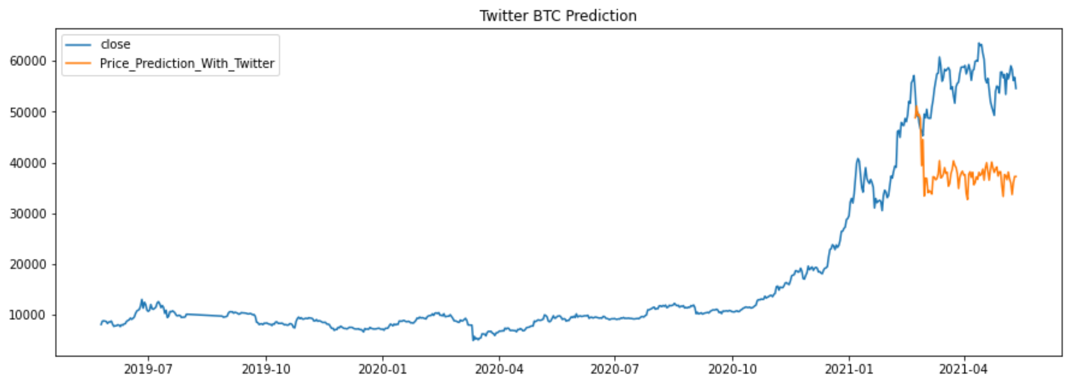


Fig. 8 BTC XGBOOST LunarCrush Twitter Dataset Prediction Graph

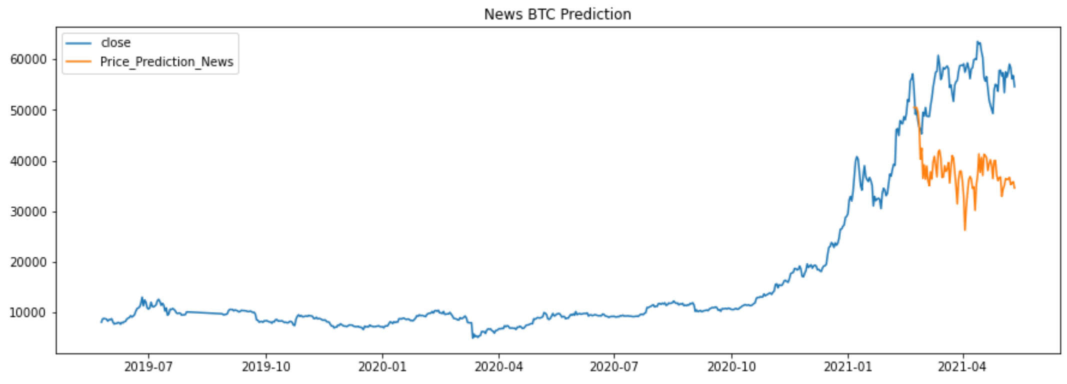
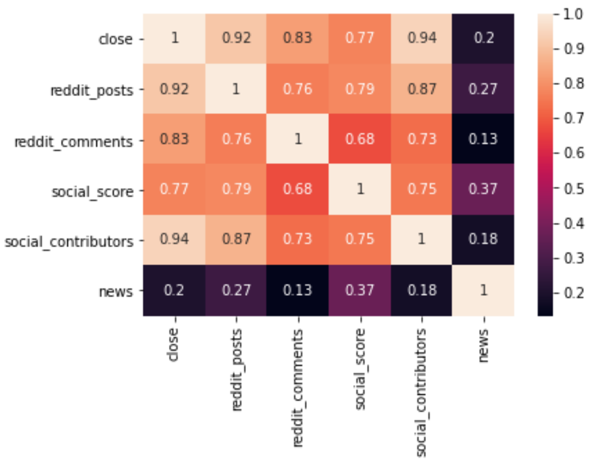


Fig. 9 BTC XGBOOST LunarCrush News Dataset Prediction Graph

Ethereum Closing Price Prediction Using LunarCrush Data with LSTM and XGBOOST

The dataset of LunarCrush starts from 25-05-2019 and ends at 12-05-2021. We have used different social media features to predict the prices of Bitcoin. First graph shows news and some social media features, second graph shows the Twitter volume and sentiment features we retrieved from LunarCrush. When we inspect the correlation of these features, we see very high values such as 94 % for social contributors, and 92 % for tweets. In detail, the features for social media and news are reddit posts, reddit comments, social score, social contributors, and news. The features for Twitter are the Twitter volume and different sentiment analysis results. Compared to BTC we can see that the correlations for ETH are much higher which means the ETH closing prices are highly affected by the sentiment of the market.



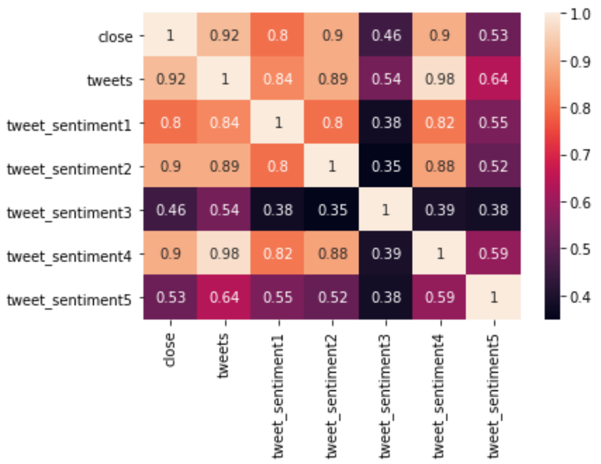


Fig. 10 and 11 ETH Correlation values for LunarCrush Dataset

Similar to BTC, we implemented 2 ETH predictions with the LunarCrush dataset. The train and test length stay the same, but the dataset changes for the values corresponding to the cryptocurrency ETH. The graphs for the predictions using LSTM can be found below. The predictions of both datasets are quite similar. The RMSE value of Twitter dataset is 182.22. The MAE value is 116.34. The RMSE value of News dataset is 179.37 and the MAE value is 118.6.

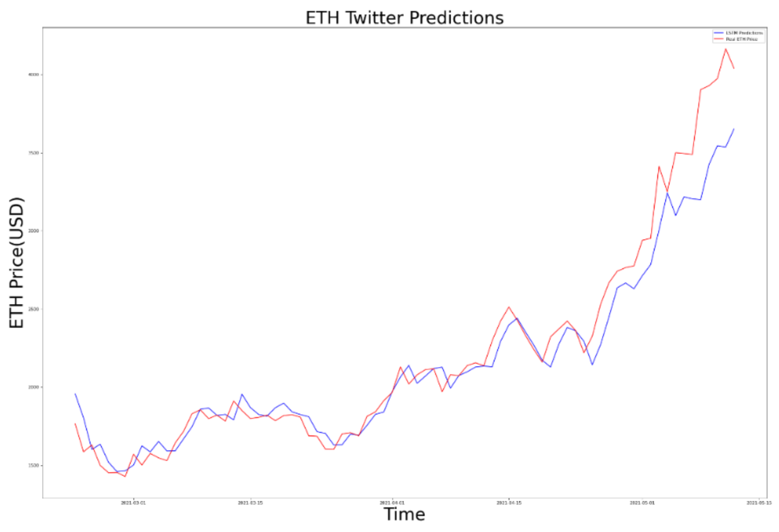


Fig. 12 ETH LSTM LunarCrush Twitter Dataset Prediction Graph

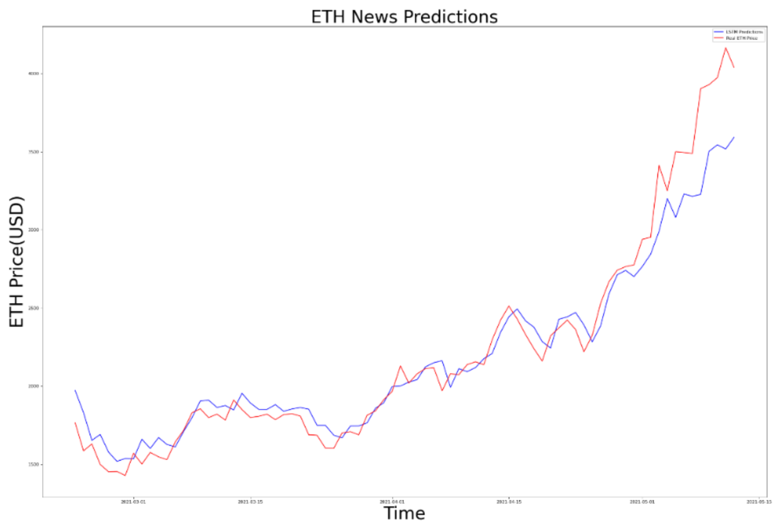


Fig. 13 ETH LSTM LunarCrush News Dataset Prediction Graph

The XGBOOST predictions for ETH can be seen below. First, we use only the social media features and shifted prices and for the other one, we use strictly only twitter features and shifted prices. There are at total 610 train values and 80 test values that was split from the data frames mentioned earlier. The RMSE value of Twitter dataset is 786.32. The MAE value is 493.22. The RMSE value of News dataset is 821.34 and the MAE value is 535.48. Again, we can see that the XGBOOST predictions are similar to each other as well, but the LSTM predictions are much superior compared to XGBOOST. Also, surprisingly, the twitter sentiment features provided us with a slightly better prediction result compared to the news features.

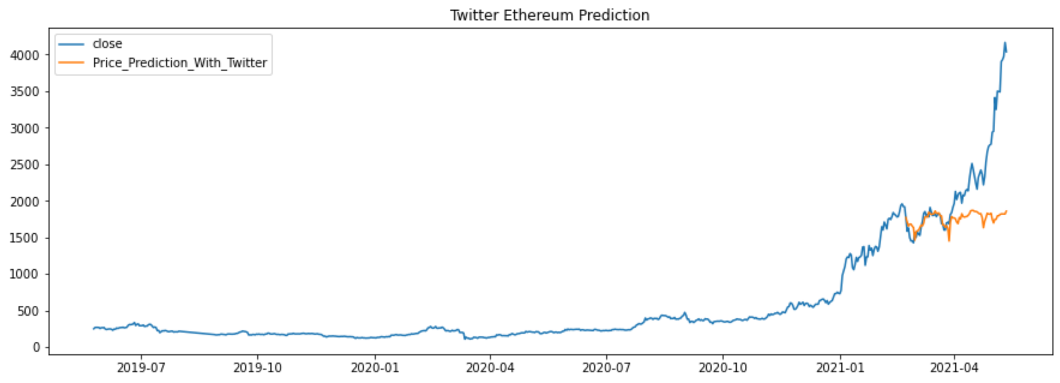


Fig. 14 ETH XGBOOST LunarCrush Twitter Dataset Prediction Graph

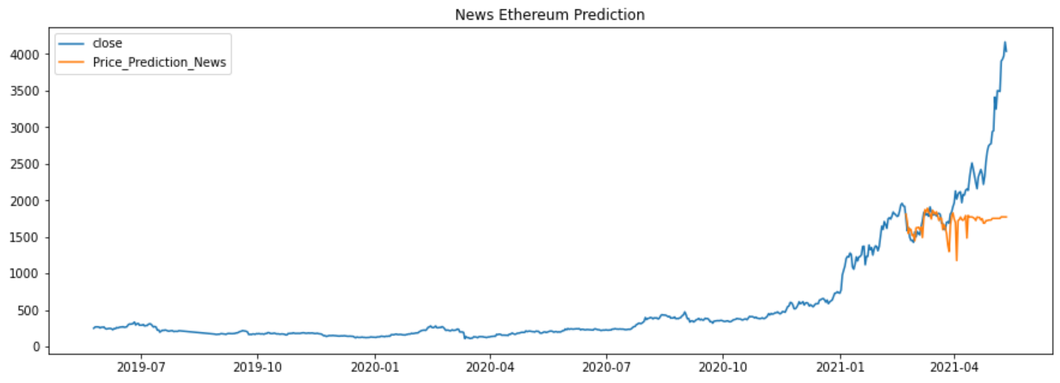


Fig. 15 ETH XGBOOST LunarCrush News Dataset Prediction Graph

Prediction with Linear Regression using Google Trends Data

We have used the Linear Regression machine learning algorithm. Linear Regression is a Linear Model that assumes a linear relationship between the independent variables (x) and a single output variable (dependent variable y). In this case the independent variable is the google trends data and the dependent variable is the closing price of Bitcoin. We have already split the data into training and test sets. We have used metrics to analyze this model and here are the plots and graphs implemented to further analyze this prediction. The root mean squared error that is calculated is 3009.65. The absolute mean squared error is 2725.53. The visualization can be also seen in Appendix-5.

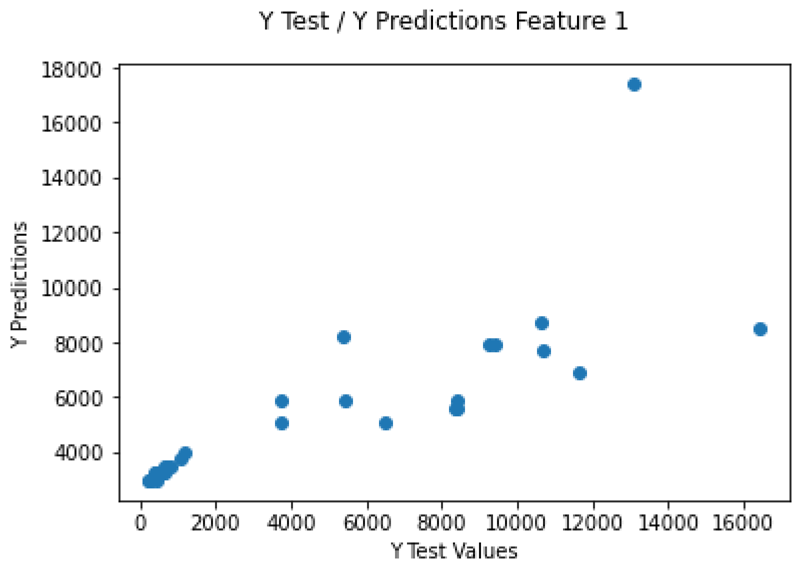
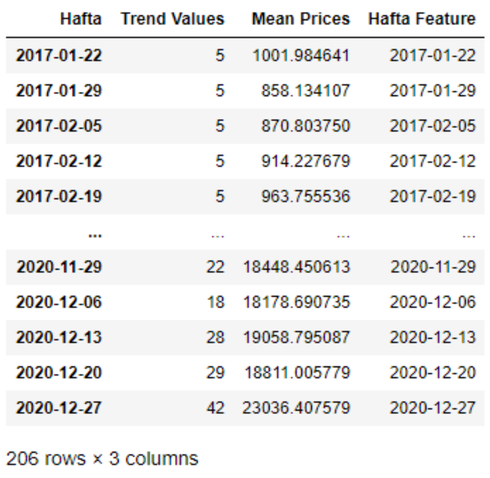


Fig. 16 Linear Regression Prediction Graph

Bitcoin Closing Price Prediction with LSTM

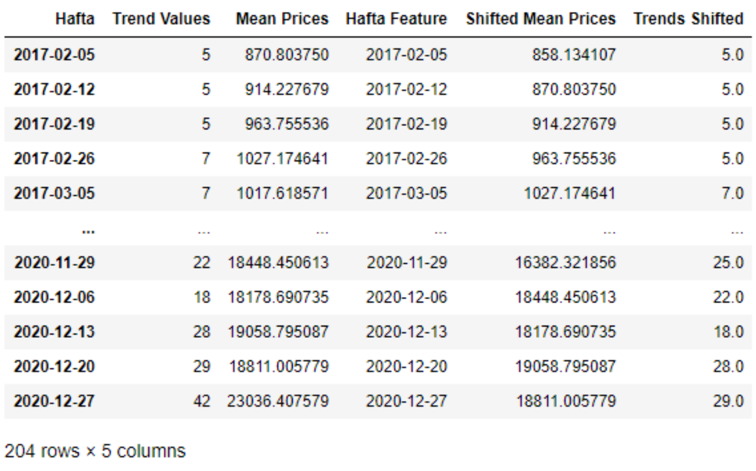
In this project, we acquired the data of the closing prices of Bitcoin between the beginning of 2017 and the end of 2020. Specific dates can be seen in Figure 17 and 18. Keras library is used to implement LSTM. Using the train test splitting technique, we have split the data into 126 weeks of train data and 80 weeks of test data. Different features are used to train the model. The month, year, day of month, Google Trend values, shifted price values and shifted Trend values are used as independent variables for the LSTM model. These independent variables for Bitcoin can be seen below:



Hafta	Trend Values	Mean Prices	Hafta Feature
2017-01-22	5	1001.984641	2017-01-22
2017-01-29	5	858.134107	2017-01-29
2017-02-05	5	870.803750	2017-02-05
2017-02-12	5	914.227679	2017-02-12
2017-02-19	5	963.755536	2017-02-19
...	...	...	...
2020-11-29	22	18448.450613	2020-11-29
2020-12-06	18	18178.690735	2020-12-06
2020-12-13	28	19058.795087	2020-12-13
2020-12-20	29	18811.005779	2020-12-20
2020-12-27	42	23036.407579	2020-12-27

206 rows × 3 columns

Fig. 17. Bitcoin Dataset

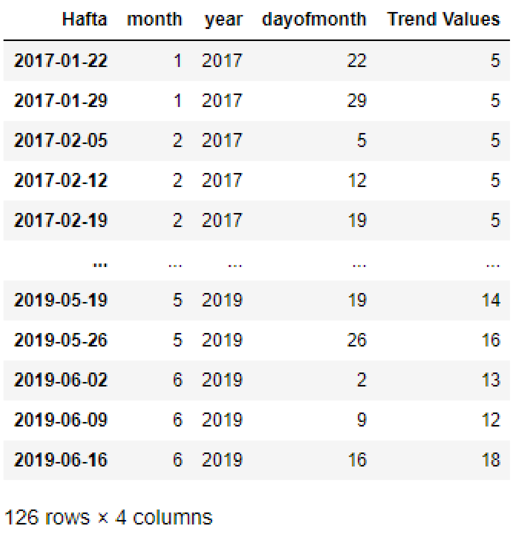


Hafta	Trend Values	Mean Prices	Hafta Feature	Shifted Mean Prices	Trends Shifted
2017-02-05	5	870.803750	2017-02-05	858.134107	5.0
2017-02-12	5	914.227679	2017-02-12	870.803750	5.0
2017-02-19	5	963.755536	2017-02-19	914.227679	5.0
2017-02-26	7	1027.174641	2017-02-26	963.755536	5.0
2017-03-05	7	1017.618571	2017-03-05	1027.174641	7.0
...	...	...	...	...	...
2020-11-29	22	18448.450613	2020-11-29	16382.321856	25.0
2020-12-06	18	18178.690735	2020-12-06	18448.450613	22.0
2020-12-13	28	19058.795087	2020-12-13	18178.690735	18.0
2020-12-20	29	18811.005779	2020-12-20	19058.795087	28.0
2020-12-27	42	23036.407579	2020-12-27	18811.005779	29.0

204 rows × 5 columns

Fig. 18. Bitcoin Dataset with shifted values

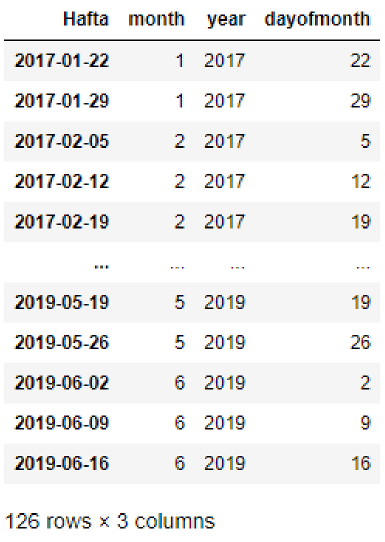
The default hyperparameter we used for prediction is 50 units for the LSTM, 1 Dense unit, batch size of 5 and 100 epochs. These values for the hyperparameter were used to predict Bitcoin before hyperparameter tuning with cross validation. We will first analyze the prediction of Bitcoin prices using month, year, day of month and Google Trend values. This dataset can be seen in Figure 19. Then, we will analyze the predictions of Bitcoin prices by omitting the Trend values and using only month, year, day of month as features. This specific dataset can be seen in Figure 20. After analyzing the scenarios mentioned earlier, we will be showing the predictions of data that were shifted. At total 126 rows (weeks) are used for training and 80 rows (weeks) are used for testing our model. The dependent values of the model are the corresponding average weekly price of Bitcoin.



Hafta	month	year	dayofmonth	Trend Values
2017-01-22	1	2017	22	5
2017-01-29	1	2017	29	5
2017-02-05	2	2017	5	5
2017-02-12	2	2017	12	5
2017-02-19	2	2017	19	5
...	...	...	...	...
2019-05-19	5	2019	19	14
2019-05-26	5	2019	26	16
2019-06-02	6	2019	2	13
2019-06-09	6	2019	9	12
2019-06-16	6	2019	16	18

126 rows × 4 columns

Fig. 19. The Training dataset for month, year, day of month and trend features



Hafta	month	year	dayofmonth
2017-01-22	1	2017	22
2017-01-29	1	2017	29
2017-02-05	2	2017	5
2017-02-12	2	2017	12
2017-02-19	2	2017	19
...	...	...	...
2019-05-19	5	2019	19
2019-05-26	5	2019	26
2019-06-02	6	2019	2
2019-06-09	6	2019	9
2019-06-16	6	2019	16

126 rows × 3 columns

Fig. 20. The training dataset for only month, year and day of month features

After feeding our model with the independent variables mentioned earlier, we predict the prices based on the test data and show it in 2 graphs. Figure 22 is the prediction using only date values, Figure 21 is the prediction using date and Google Trends values. The blue line is predicted, and the red line is real weekly prices of Bitcoin.

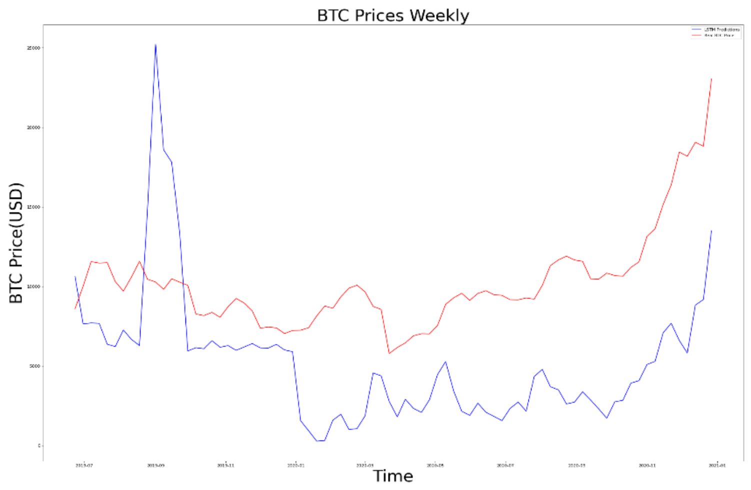


Fig. 21. The predictions for date and trend values

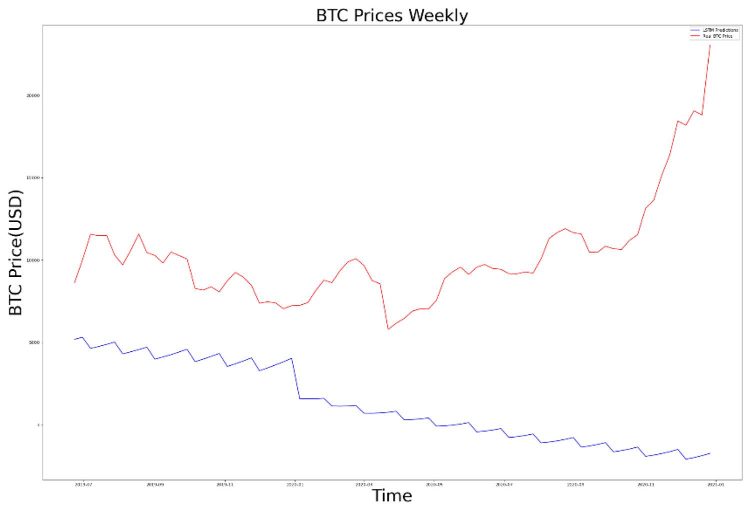
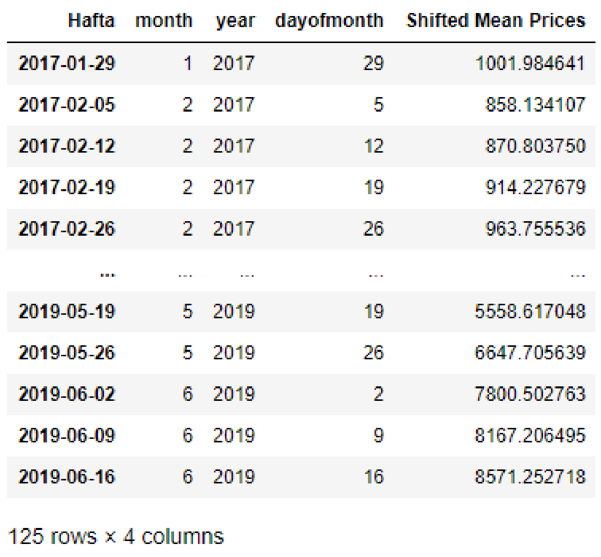


Fig. 22. The predictions for only date values

Clearly the values of Google Trends have significantly enhanced our model, despite the sudden spike of 2019-09 which is caused by the sudden increase of Trends value. We have used Mean Absolute error and Root Mean Squared error to further analyze and conclude our predictions. When we used only dates as features, the RMSE result is 10362.77, the MAE result is 9224.89. When dates and trend values are used as features, the RMSE result is 6555.80, the MAE result is 5843.18. Although LSTM models are prone to overfitting, this can be prevented by tuning hyperparameters and analyzing the model. Because of this, we have implemented 2 different cross validation techniques to find the best hyperparameter. The detailed cross validation explanations will be explained after analyzing the shifted independent values prediction.

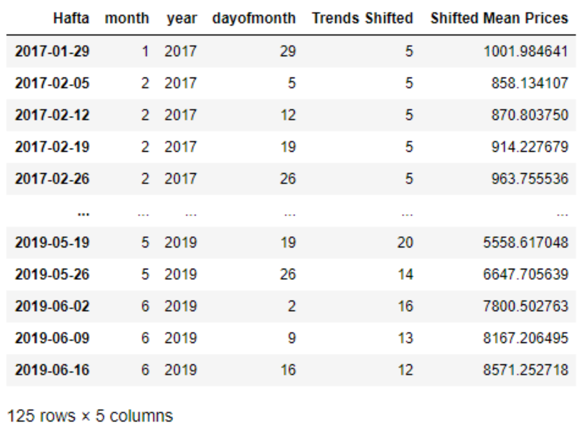
The values of prices and trend values are shifted to the right for 1 day, so our model can get the price and trend values from the day before and predict the corresponding row appropriately. After making these changes, 2 different train test splits are created. Both uses the year, month, day of month values. As an extra, the first one uses shifted trend values, the second one uses shifted trend values and shifted weekly prices of Bitcoin. Here are the data frames to summarize:



Hafta	month	year	dayofmonth	Shifted Mean Prices
2017-01-29	1	2017	29	1001.984641
2017-02-05	2	2017	5	858.134107
2017-02-12	2	2017	12	870.803750
2017-02-19	2	2017	19	914.227679
2017-02-26	2	2017	26	963.755536
...	...	...	...	...
2019-05-19	5	2019	19	5558.617048
2019-05-26	5	2019	26	6647.705639
2019-06-02	6	2019	2	7800.502763
2019-06-09	6	2019	9	8167.206495
2019-06-16	6	2019	16	8571.252718

125 rows × 4 columns

Fig. 23. The dataset for shifted mean price and date values



Hafta	month	year	dayofmonth	Trends Shifted	Shifted Mean Prices
2017-01-29	1	2017	29	5	1001.984641
2017-02-05	2	2017	5	5	858.134107
2017-02-12	2	2017	12	5	870.803750
2017-02-19	2	2017	19	5	914.227679
2017-02-26	2	2017	26	5	963.755536
...	...	...	...	...	...
2019-05-19	5	2019	19	20	5558.617048
2019-05-26	5	2019	26	14	6647.705639
2019-06-02	6	2019	2	16	7800.502763
2019-06-09	6	2019	9	13	8167.206495
2019-06-16	6	2019	16	12	8571.252718

125 rows × 5 columns

Fig. 24. The dataset for shifted mean price, shifted trends and date values

After feeding our model with the independent variables mentioned earlier, we predict the prices based on the test data and show it in 2 graphs. Figure 25 is the prediction using shifted price values, Figure 26 is the prediction using shifted price and shifted Google Trends values. The blue line is predicted, and the red line is real weekly prices of Bitcoin.

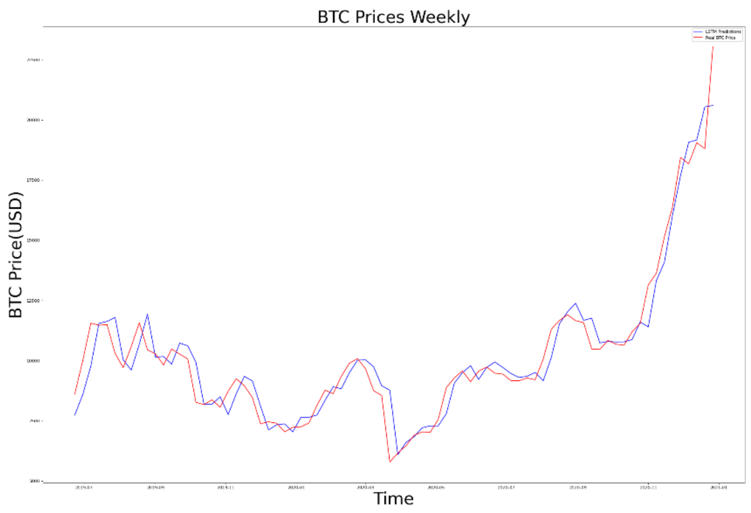


Fig. 25. The prediction graph for shifted price and date values

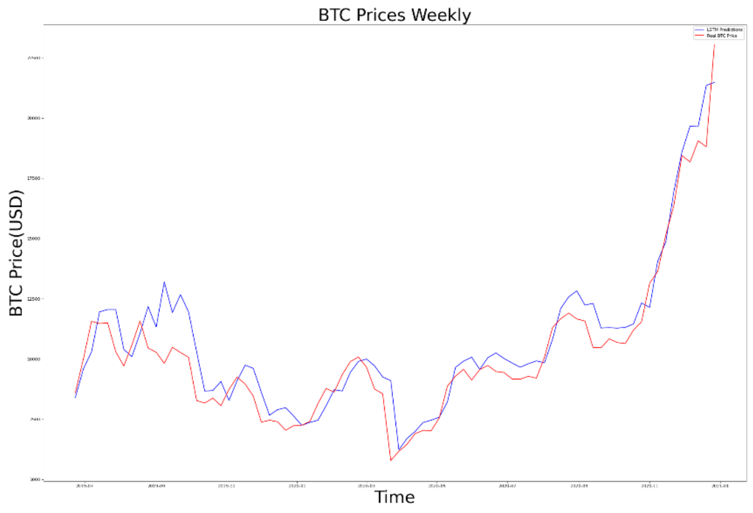
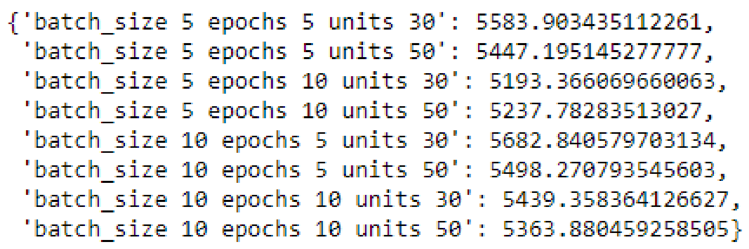


Fig. 26. The prediction graph for shifted price, date and trends values

Unfortunately, the shifted Google Trends values did not affect the performance of our model when shifted average prices are used. When we used only shifted average prices as features, the RMSE result is 819.62, the MAE result is 587.59. When shifted average prices and shifted trend values are used as features, the RMSE result is 1024.85, the MAE result is 767.12.

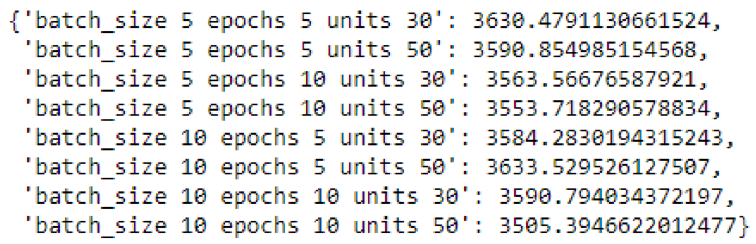
Bitcoin Cross Validation with LSTM

2 different cross validation techniques are implemented to find the right hyperparameters for the LSTM model. These techniques are called rolling forecast origin and sliding window, they are explained in depth in the Solution Approach part of this paper. A combination of 5-10 batch sizes, 5-10 epochs, 30-50 units are used as hyperparameters to calculate the root mean squared errors and find the best hyperparameter among them. These combinations are used for all the implementations of the cross validation. More combinations for the hyperparameters could be used but because of the long runtime of the LSTM models we used 8 combinations at total. The rolling forecast origin results when date and trend features are used, can be seen in Figure 27 and the sliding window implementation results can be seen in Figure 28. The best hyperparameters that was found for rolling forecast origin are 5 for the batch size, 10 for the epochs and 30 for the units. The RMSE for this hyperparameter is 5193.36. The best hyperparameters found for sliding window are 10 for the batch size, 10 for the epochs and 50 for the units. The RMSE for this hyperparameter is 3505.



```
{'batch_size 5 epochs 5 units 30': 5583.903435112261,
 'batch_size 5 epochs 5 units 50': 5447.195145277777,
 'batch_size 5 epochs 10 units 30': 5193.366069660063,
 'batch_size 5 epochs 10 units 50': 5237.78283513027,
 'batch_size 10 epochs 5 units 30': 5682.840579703134,
 'batch_size 10 epochs 5 units 50': 5498.27079354560З,
 'batch_size 10 epochs 10 units 30': 5439.358364126627,
 'batch_size 10 epochs 10 units 50': 5363.880459258505}
```

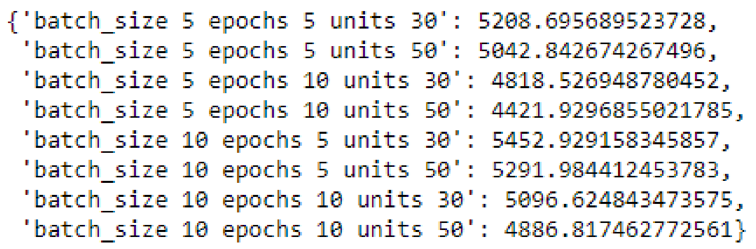
Fig. 27. Results for Bitcoin, rolling forecast origin, using date and trends features



```
{'batch_size 5 epochs 5 units 30': 3630.4791130661524,
 'batch_size 5 epochs 5 units 50': 3590.854985154568,
 'batch_size 5 epochs 10 units 30': 3563.56676587921,
 'batch_size 5 epochs 10 units 50': 3553.718290578834,
 'batch_size 10 epochs 5 units 30': 3584.2830194315243,
 'batch_size 10 epochs 5 units 50': 3633.529526127507,
 'batch_size 10 epochs 10 units 30': 3590.794034372197,
 'batch_size 10 epochs 10 units 50': 3505.3946622012477}
```

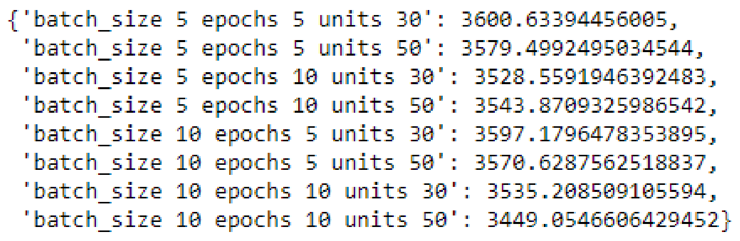
Fig. 28. Results for Bitcoin, sliding window, using date and trends features

Previously, the results for features of date and trend values are shown. Then, the same implementation is calculated for the features of date, shifted mean prices, and shifted trends prices. The rolling forecast origin results can be seen in Figure 29 and the sliding window implementation results can be seen in Figure 30. The best hyperparameters that was found for rolling forecast origin are 5 for the batch size, 10 for the epochs and 50 for the units. The RMSE for this hyperparameter is 4421.92. The best hyperparameters found for sliding window are 10 for the batch size, 10 for the epochs and 50 for the units. The RMSE for this hyperparameter is 3449.05.



```
{'batch_size 5 epochs 5 units 30': 5208.695689523728,
 'batch_size 5 epochs 5 units 50': 5042.842674267496,
 'batch_size 5 epochs 10 units 30': 4818.526948780452,
 'batch_size 5 epochs 10 units 50': 4421.9296855021785,
 'batch_size 10 epochs 5 units 30': 5452.929158345857,
 'batch_size 10 epochs 5 units 50': 5291.984412453783,
 'batch_size 10 epochs 10 units 30': 5096.624843473575,
 'batch_size 10 epochs 10 units 50': 4886.817462772561}
```

Fig. 29. Results for Bitcoin, rolling forecast origin, using date, shifted trends and shifted mean price features

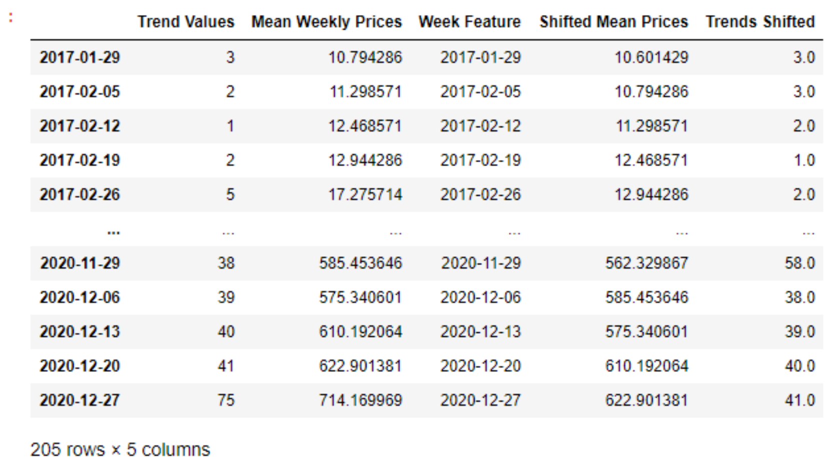


```
{'batch_size 5 epochs 5 units 30': 3600.63394456005,
 'batch_size 5 epochs 5 units 50': 3579.4992495034544,
 'batch_size 5 epochs 10 units 30': 3528.5591946392483,
 'batch_size 5 epochs 10 units 50': 3543.8709325986542,
 'batch_size 10 epochs 5 units 30': 3597.1796478353895,
 'batch_size 10 epochs 5 units 50': 3570.6287562518837,
 'batch_size 10 epochs 10 units 30': 3535.208509105594,
 'batch_size 10 epochs 10 units 50': 3449.0546606429452}
```

Fig. 30. Results for Bitcoin, sliding window, using date, shifted trends and shifted mean price features

Ethereum Closing Price Prediction with LSTM

We acquired the data of the closing prices of Ethereum between the beginning of 2017 and the end of 2020 same as Bitcoin. The libraries that were used are as same as the Bitcoin implementation. Using the train test splitting technique, we have split the data into 126 weeks of train data and 80 weeks of test data. Different features are used to train the model. The month, year, day of month, Google Trend values, shifted price values and shifted Trend values are used as independent variables for the LSTM model. These independent variables for Ethereum can be seen below.



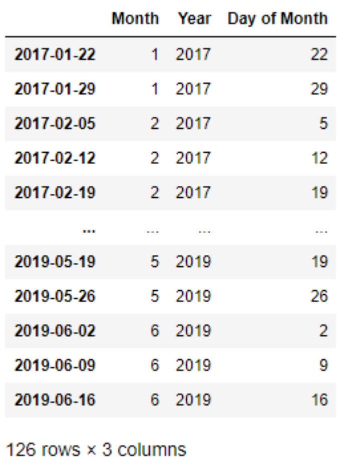
	Trend Values	Mean Weekly Prices	Week Feature	Shifted Mean Prices	Trends Shifted
2017-01-29	3	10.794286	2017-01-29	10.601429	3.0
2017-02-05	2	11.298571	2017-02-05	10.794286	3.0
2017-02-12	1	12.468571	2017-02-12	11.298571	2.0
2017-02-19	2	12.944286	2017-02-19	12.468571	1.0
2017-02-26	5	17.275714	2017-02-26	12.944286	2.0
...	...	...	...	...	...
2020-11-29	38	585.453646	2020-11-29	562.329867	58.0
2020-12-06	39	575.340601	2020-12-06	585.453646	38.0
2020-12-13	40	610.192064	2020-12-13	575.340601	39.0
2020-12-20	41	622.901381	2020-12-20	610.192064	40.0
2020-12-27	75	714.169969	2020-12-27	622.901381	41.0

205 rows × 5 columns

Fig. 31. The dataset used for Ethereum price prediction

The default hyperparameter we used for prediction is 50 units for the LSTM, 1 Dense unit, batch size of 5 and 100 epochs. These hyperparameters that were used are the same as Bitcoin to compare the results. These values for the hyperparameter were used to predict Ethereum before hyperparameter tuning with cross validation. We will first analyze the prediction of Ethereum prices using month, year, day of month and Google Trend values. Then, we will analyze the predictions of Ethereum prices by omitting the Trend values and using only month, year, day of month as features. After analyzing the scenarios mentioned earlier, we will be showing the predictions of data that were shifted.

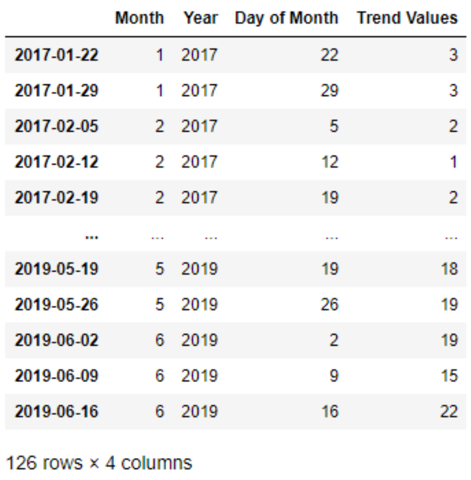
At total, 126 rows (weeks) are used for training and 80 rows (weeks) are used for testing our model. The dependent values of the model are the corresponding average weekly price of Ethereum. Here are the features used for both scenarios:



	Month	Year	Day of Month
2017-01-22	1	2017	22
2017-01-29	1	2017	29
2017-02-05	2	2017	5
2017-02-12	2	2017	12
2017-02-19	2	2017	19
...	...	...	...
2019-05-19	5	2019	19
2019-05-26	5	2019	26
2019-06-02	6	2019	2
2019-06-09	6	2019	9
2019-06-16	6	2019	16

126 rows × 3 columns

Fig. 32. The training dataset for month, year and day of month



	Month	Year	Day of Month	Trend Values
2017-01-22	1	2017	22	3
2017-01-29	1	2017	29	3
2017-02-05	2	2017	5	2
2017-02-12	2	2017	12	1
2017-02-19	2	2017	19	2
...	...	...	...	...
2019-05-19	5	2019	19	18
2019-05-26	5	2019	26	19
2019-06-02	6	2019	2	19
2019-06-09	6	2019	9	15
2019-06-16	6	2019	16	22

126 rows × 4 columns

Fig. 33. The training dataset for month, year, day of month and trend values

After feeding our model with the independent variables mentioned earlier, we predict the prices based on the test data and show it in 2 graphs. Figure 34 is the prediction using only date values, Figure 35 is the prediction using date and Google Trends values. The blue line is predicted, and the red line is real weekly prices of Ethereum.

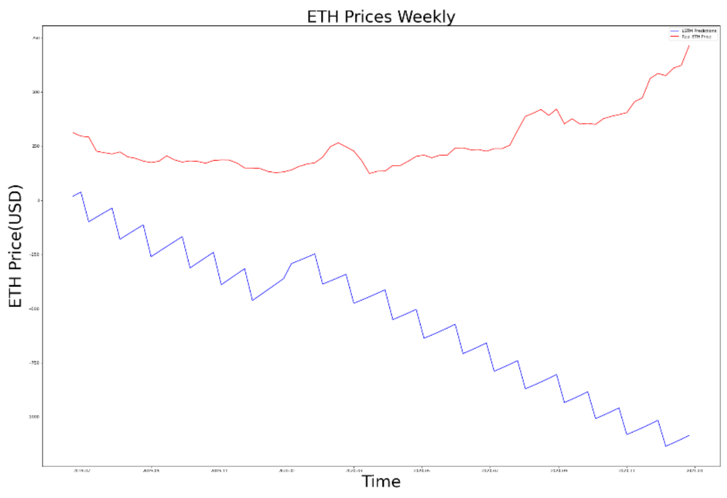


Fig. 34. The ETH prediction graph for using only date values

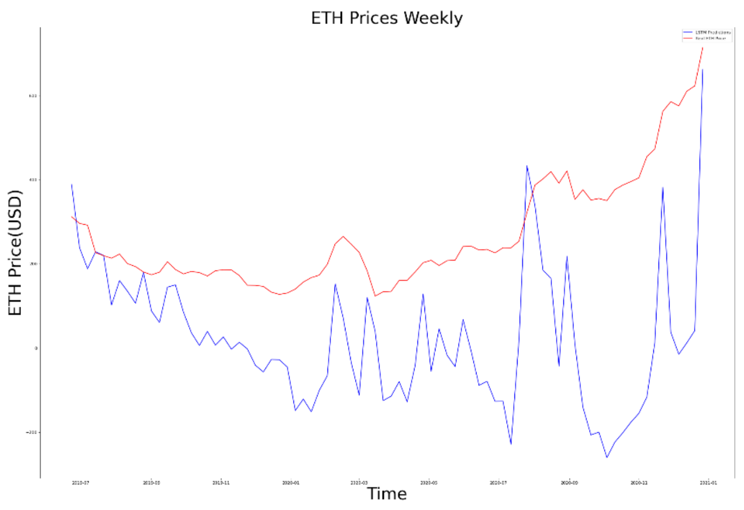
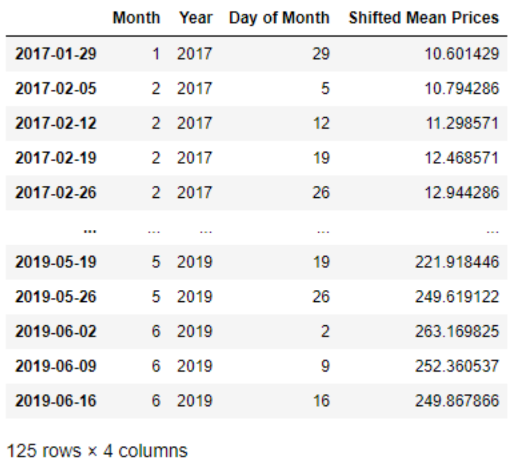


Fig. 35. The ETH prediction graph for using date and trends values

The values of Google Trends have significantly enhanced our model, Figure 34 only uses the date features which did not help our model at all. Even though Figure 35 does not make good predictions, it can be seen that the Trend values have helped our model significantly. We have used Mean Absolute Error and Root Mean Squared Error to further analyze and conclude our predictions. When we used only dates as features, the RMSE result is 921.50, the MAE result is 813.75. When dates and trend values are used as features, the RMSE result is 305.51, the MAE result is 250.91. Same as Bitcoin, we have implemented 2 different cross validation techniques to find the best hyperparameter. The detailed cross validation explanations will be explained after analyzing the shifted independent values prediction.

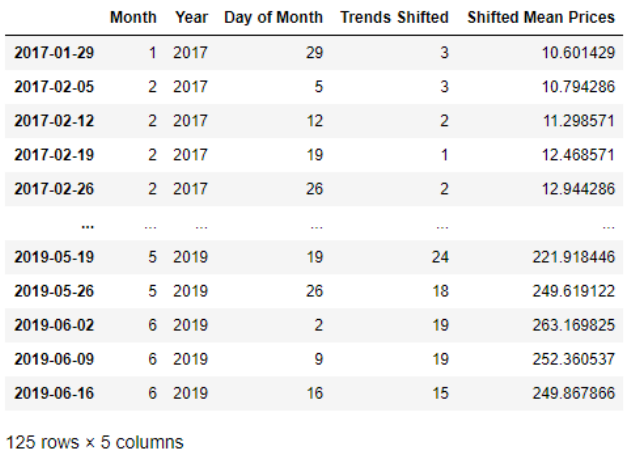
The values of prices and trend values are shifted to the right for 1 day, so our model can get the price and trend values from the day before and predict the corresponding row appropriately. After making these changes, 2 different train test splits are created. Both uses the year, month, day of month values. As an extra, the first one uses shifted trend values, the second one uses shifted trend values and shifted weekly prices of Ethereum. Figure 36 and 37 shows the datasets that were used for our model.



	Month	Year	Day of Month	Shifted Mean Prices
2017-01-29	1	2017	29	10.601429
2017-02-05	2	2017	5	10.794286
2017-02-12	2	2017	12	11.298571
2017-02-19	2	2017	19	12.468571
2017-02-26	2	2017	26	12.944286
...	...	...	...	...
2019-05-19	5	2019	19	221.918446
2019-05-26	5	2019	26	249.619122
2019-06-02	6	2019	2	263.169825
2019-06-09	6	2019	9	252.360537
2019-06-16	6	2019	16	249.867866

125 rows × 4 columns

Fig. 36. The dataset for shifted mean price and date values



	Month	Year	Day of Month	Trends Shifted	Shifted Mean Prices
2017-01-29	1	2017	29	3	10.601429
2017-02-05	2	2017	5	3	10.794286
2017-02-12	2	2017	12	2	11.298571
2017-02-19	2	2017	19	1	12.468571
2017-02-26	2	2017	26	2	12.944286
...	...	...	...	...	...
2019-05-19	5	2019	19	24	221.918446
2019-05-26	5	2019	26	18	249.619122
2019-06-02	6	2019	2	19	263.169825
2019-06-09	6	2019	9	19	252.360537
2019-06-16	6	2019	16	15	249.867866

125 rows × 5 columns

Fig. 37. The dataset for shifted mean price, shifted trends and date values

After feeding our model with the independent variables mentioned earlier, we predict the prices based on the test data and show it in 2 graphs. Figure 38 shows the prediction using shifted price values, Figure 39 shows the prediction using shifted price and shifted Google Trends values. The blue line is predicted, and the red line is real weekly prices of Ethereum.

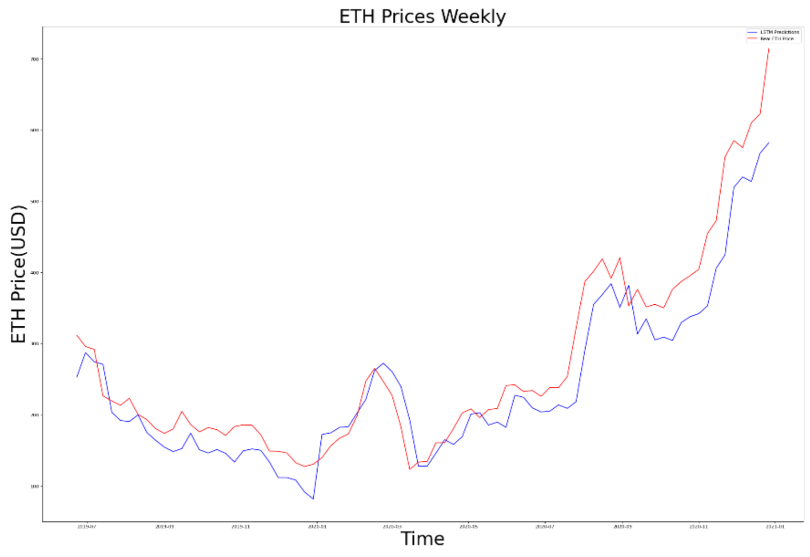


Fig. 38. The dataset for shifted mean price and date values

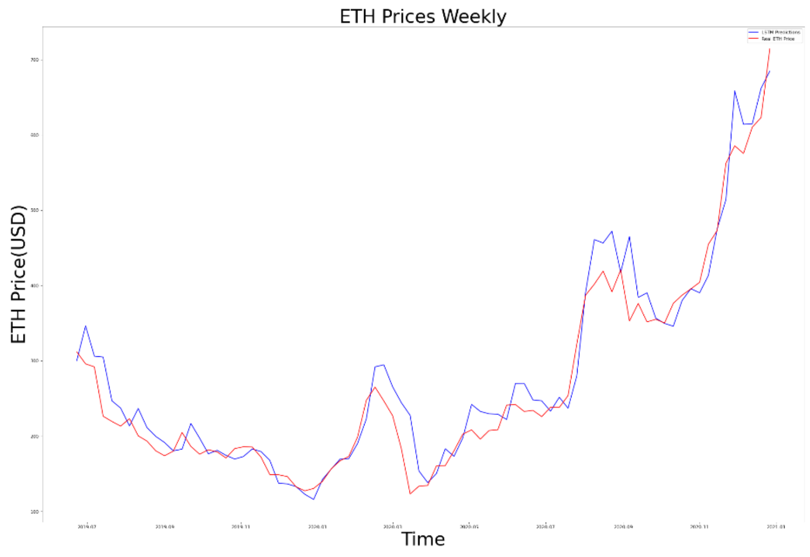


Fig. 39. The dataset for shifted mean price and date values

We calculated the Root Mean Squared Error and Mean Absolute Error of these graphs. When we used only shifted average prices as features, the RMSE result is 46.57, the MAE result is 37.17. When shifted average prices and shifted trend values are used as features, the RMSE result is 32.71, the MAE result is 23.25. If we observe both graphs, they look quite similar but after analyzing the metrics we concluded that when we used both shifted Trend and shifted price values, they have helped our model to predict better prices compared to using only shifted price values. We can see the enhancement in these Ethereum predictions where we use the shifted trend values as an extra feature for the LSTM model.

Ethereum Cross Validation with LSTM

Same techniques as Bitcoin Cross Validation with LSTM are used, they are explained in depth in the Solution Approach part of this paper. A combination of 5-10 batch sizes, 5-10 epochs, 30-50 units are used as hyperparameters to calculate the Root Mean Squared Errors and find the best hyperparameter among them. These combinations are used for all the implementations of the cross validation. More combinations for the hyperparameters could be used but because of the long runtime of the LSTM models we used 8 combinations at total. The rolling forecast origin results when date and trend features are used, can be seen in Figure 40 and the sliding window implementation results can be seen in Figure 41. The best hyperparameters that was found for rolling forecast origin are 5 for the batch size, 10 for the epochs and 50 for the units. The RMSE for this hyperparameter is 46.07. The best hyperparameters found for sliding window are 5 for the batch size, 10 for the epochs and 50 for the units. The RMSE for this hyperparameter is 946.24.

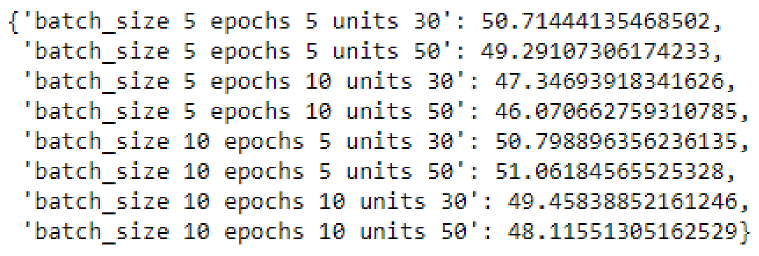
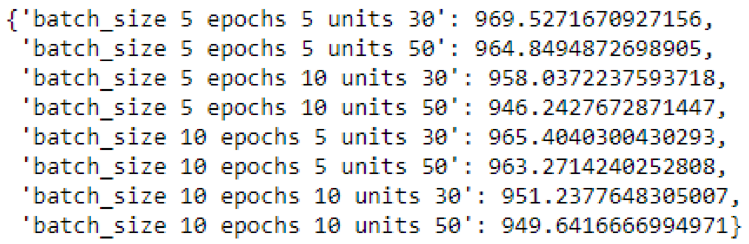

```
{'batch_size 5 epochs 5 units 30': 50.71444135468502,
 'batch_size 5 epochs 5 units 50': 49.29107306174233,
 'batch_size 5 epochs 10 units 30': 47.34693918341626,
 'batch_size 5 epochs 10 units 50': 46.070662759310785,
 'batch_size 10 epochs 5 units 30': 50.798896356236135,
 'batch_size 10 epochs 5 units 50': 51.06184565525328,
 'batch_size 10 epochs 10 units 30': 49.45838852161246,
 'batch_size 10 epochs 10 units 50': 48.11551305162529}
```

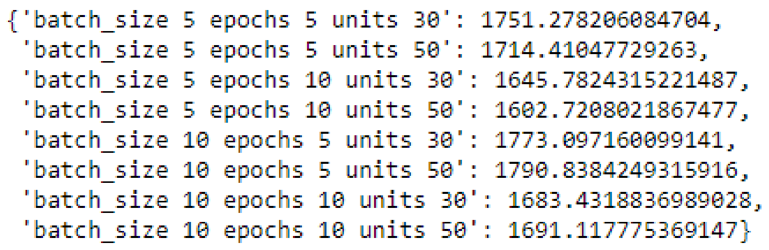
Fig. 40. Ethereum, rolling forecast origin results using date and trends features



```
{'batch_size 5 epochs 5 units 30': 969.5271670927156,
 'batch_size 5 epochs 5 units 50': 964.8494872698905,
 'batch_size 5 epochs 10 units 30': 958.0372237593718,
 'batch_size 5 epochs 10 units 50': 946.2427672871447,
 'batch_size 10 epochs 5 units 30': 965.4040300430293,
 'batch_size 10 epochs 5 units 50': 963.2714240252808,
 'batch_size 10 epochs 10 units 30': 951.2377648305007,
 'batch_size 10 epochs 10 units 50': 949.6416666994971}
```

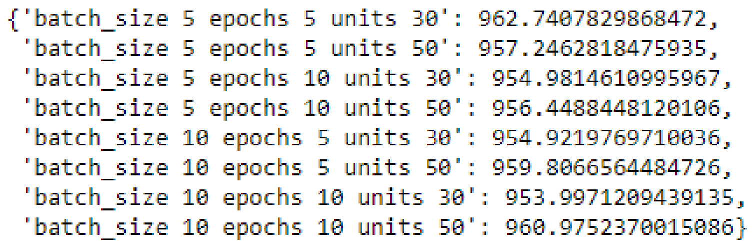
Fig. 41. Ethereum, sliding window results using date and trends features

Previously, the results for features of date and trend values are shown. Then, the same implementation is calculated for the features of date, shifted mean prices, and shifted trends prices. The rolling forecast origin results can be seen in Figure 42 and the sliding window implementation results can be seen in Figure 43. The best hyperparameters that was found for rolling forecast origin are 5 for the batch size, 10 for the epochs and 50 for the units. The RMSE for this hyperparameter is 1602.72. The best hyperparameters found for sliding window are 10 for the batch size, 10 for the epochs and 30 for the units. The RMSE for this hyperparameter is 953.99.



```
{'batch_size 5 epochs 5 units 30': 1751.278206084704,
 'batch_size 5 epochs 5 units 50': 1714.41047729263,
 'batch_size 5 epochs 10 units 30': 1645.7824315221487,
 'batch_size 5 epochs 10 units 50': 1602.7208021867477,
 'batch_size 10 epochs 5 units 30': 1773.097160099141,
 'batch_size 10 epochs 5 units 50': 1790.8384249315916,
 'batch_size 10 epochs 10 units 30': 1683.4318836989028,
 'batch_size 10 epochs 10 units 50': 1691.117775369147}
```

Fig. 42. Ethereum, rolling forecast origin results using date, shifted prices and shifted trends features



```
{'batch_size 5 epochs 5 units 30': 962.7407829868472,
 'batch_size 5 epochs 5 units 50': 957.2462818475935,
 'batch_size 5 epochs 10 units 30': 954.9814610995967,
 'batch_size 5 epochs 10 units 50': 956.4488448120106,
 'batch_size 10 epochs 5 units 30': 954.9219769710036,
 'batch_size 10 epochs 5 units 50': 959.8066564484726,
 'batch_size 10 epochs 10 units 30': 953.9971209439135,
 'batch_size 10 epochs 10 units 50': 960.9752370015086}
```

Fig. 43. Ethereum, sliding window results using date, shifted prices and shifted trends features

Bitcoin Closing Price Prediction with XGBOOST

The data that is used in the XGBOOST implementation is the same as the Bitcoin LSTM implementation. Also, the train test splitting ratio (126 weeks of train data and 80 weeks of test data) and different features used for XGBOOST is the same explained as the LSTM model mentioned earlier. The dates of the closing prices are between 2017 and 2020. The xgboost library for Python is used for using the time series model. The default hyperparameter is 10 for the minimum child weight, 0.3 for the Col sample by tree, 0.1 learning rate, 5 for the max depth, 10 for the alpha and 100 for the estimators. Again, the dataset that will be used for predicting is the same as the Bitcoin LSTM dataset used before. The dependent values of the model are the corresponding average weekly price of Bitcoin. After feeding our model with the independent variables we predict the prices based on the test data and show it in 2 graphs. Figure 44 is the prediction using date and trends values, Figure 45 is the prediction using only date values. The blue line is predicted, and the orange line is real weekly prices of Bitcoin.

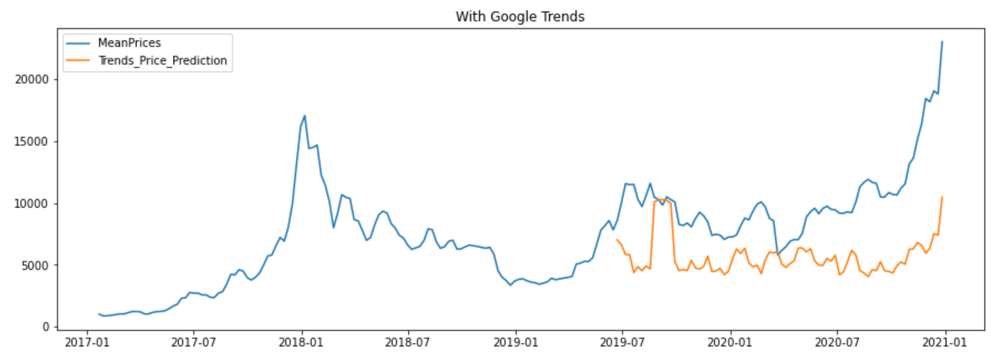


Fig. 44. BTC prediction with XGBOOST using trend and date values

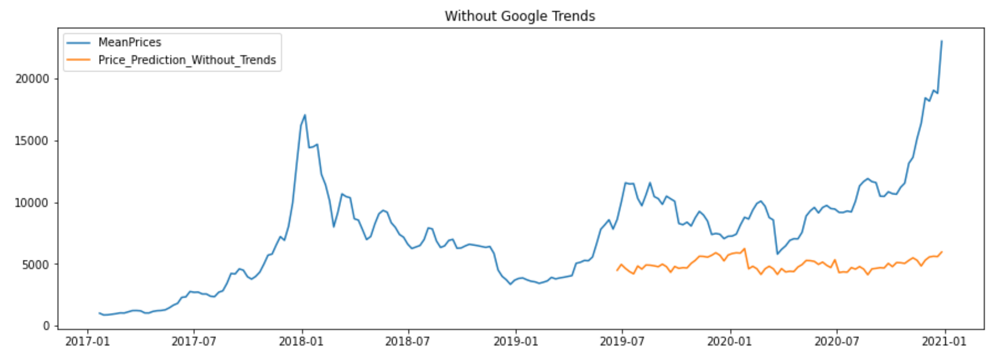


Fig. 45. BTC prediction with XGBOOST using only date values

The values of Google Trends have significantly enhanced our model. The sudden spike of 2019-09 is caused by the sudden increase of Trends value. We have used MAE and RMSE to further analyze and conclude our predictions. When we used only dates as features, the RMSE result is 6052.07, the MAE result is 5256.78. When dates and trend values are used as features, the RMSE result is 5371.73, the MAE result is 4544.24. When we compare these error results it is very clear that the trend values affect the predictions positively. We have implemented 2 different cross validation techniques for XGBOOST as well. The detailed cross validation explanations will be explained after analyzing the shifted independent values prediction.

The values of prices and trend values are shifted to the right for 1 day, so our model can get the price and trend values from the day before and predict the corresponding row appropriately. The train test split ratio and the dataset are the same as the one used in LSTM. 2 train, test dataset is created, and both uses the year, month, day of month values. As an extra, the first one uses shifted trend values, the second one uses shifted trend values and shifted weekly prices of Bitcoin. We predict the prices based on the test data and show it in 2 graphs. Figure 46 is the prediction using shifted trends values, Figure 47 is the prediction using shifted price and shifted Google Trends values. The blue line is predicted, and the orange line is the real weekly prices of Bitcoin.

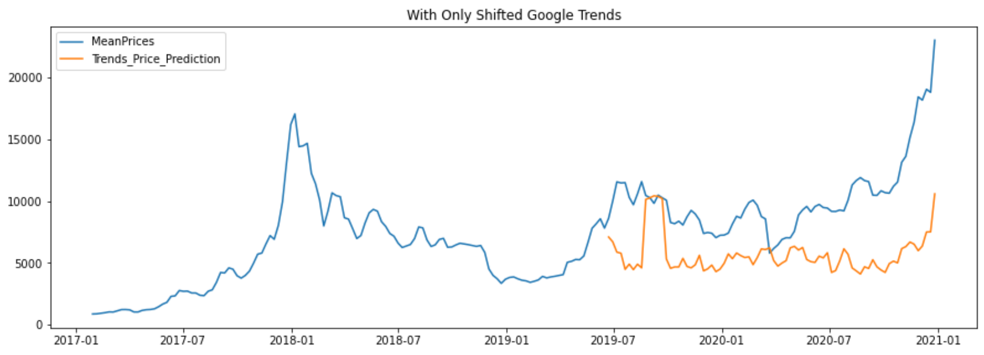


Fig. 46. BTC prediction with XGBOOST using dates and shifted trends values

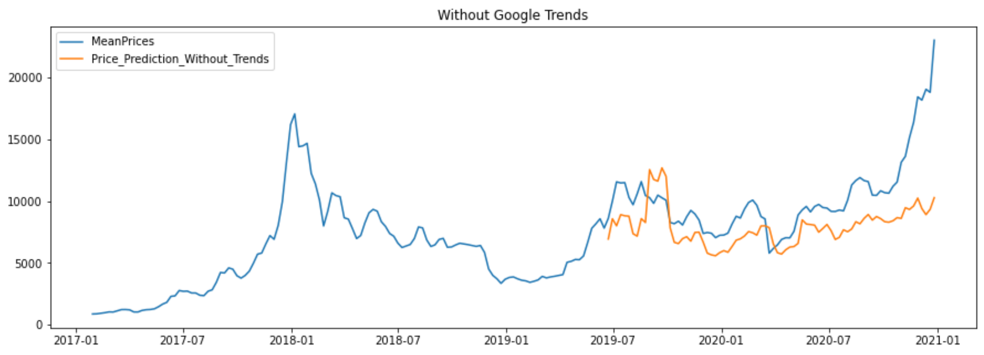


Fig. 47. BTC prediction with XGBOOST using dates, shifted trends and shifted prices values

We used the same metrics that were used for LSTM. When we used dates and shifted trends as features, the RMSE result is 5347.77, the MAE result is 4532.11. When dates, shifted trends and shifted average prices are used as features, the RMSE result is 3381.60, the MAE result is 2559.94. If we observe both graphs, the predictions are quite different from each other. We can see the enhancement in these Bitcoin predictions when shifted prices are added as a feature.

Bitcoin Cross Validation with XGBOOST

The techniques explained in the Solution Approach part of this paper are used for this part. A combination of 0.20-0.25-0.30 learning rates, 3-4-5-6-8 max depths, 5-7 minimum child weights, 0.3-0.4 gamma, 0.5-0.7 Col sample by tree are used as hyperparameters to calculate the Root Mean Squared Errors and find the best hyperparameter among them. These combinations are used for all the implementations of the XGBOOST cross validation. So, at total 120 different models are created and their errors are calculated. The rolling forecast origin results when date and trend features are used, can be seen in Appendix-6 and the sliding window implementation results can be seen in Appendix-7. The best hyperparameters that was found for rolling forecast origin are 0.2 for the learning rate, 4 for the max depth, 5 for the minimum child weight, 0.3 for the gamma and 0.5 for the Col sample by tree. The RMSE for this hyperparameter is 4411.12. The best hyperparameters found for sliding window are 0.25 for the learning rate, 3 for the max depth, 5 for the minimum child weight, 0.3 for the gamma and 0.5 for the Col sample by tree. The RMSE for this hyperparameter is 3322.24.

Ethereum Closing Price Prediction with XGBOOST

The data that is used in the XGBOOST implementation is the same as the Ethereum LSTM implementation. Also, the train test splitting ratio (126 weeks of train data and 80 weeks of test data) and different features used for XGBOOST is the same explained as the LSTM model mentioned earlier. The dates of the closing prices are between 2017 and 2020. The default hyperparameters are the same as the Bitcoin closing price prediction model to compare both models. Again, the dataset that will be used for predicting is the same as the Ethereum LSTM dataset used before. The dependent values of the model are the corresponding average weekly price of ETH. After feeding our model with the independent variables we predict the prices based on the test data and show it in 2 graphs. Figure 48 is the prediction using date and trends values, figure 49 is the prediction using only date values. The blue line is predicted, and the orange line is real weekly prices of ETH.

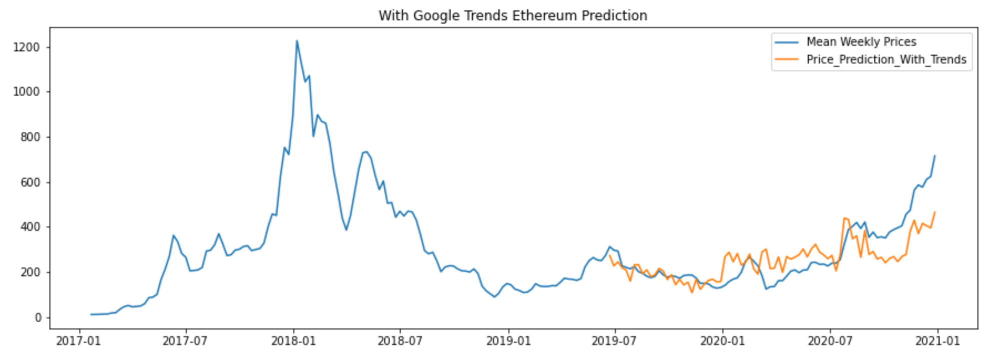


Fig. 48. ETH prediction with XGBOOST using dates, trends values

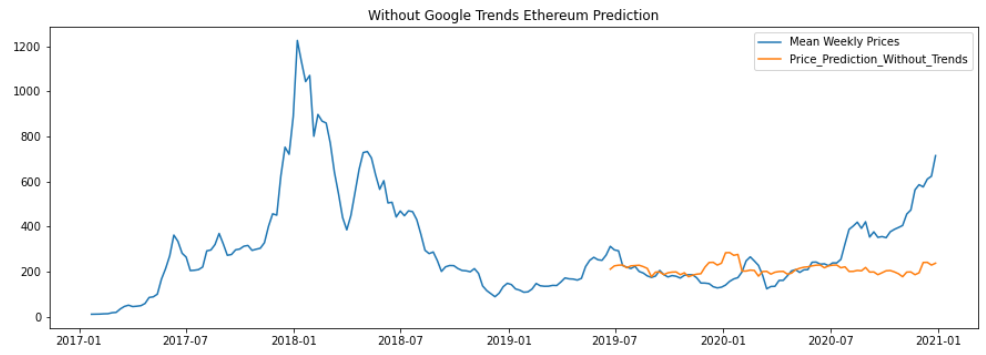


Fig. 49. ETH prediction with XGBOOST using only dates values

As it can be seen from Figure 48 values of Google Trends have significantly enhanced our model, to further prove that there is an enhancement, we have used Mean Absolute Error and Root Mean Squared Error to further analyze and conclude our predictions. When we used only dates as features, the RMSE result is 146.89, the MAE result is 96.32. When dates and trend values are used as features, the RMSE result is 89.44, the MAE result is 68.87. When we compare these error results it is very clear that the trend values affect the predictions positively. We have implemented 2 different cross validation techniques for XGBOOST as well. The detailed cross validation explanations will be explained after analyzing the shifted independent values prediction.

Now the shifted features will be used to predict the ETH prices. The train test split ratio and the dataset are the same as the one used in ETH LSTM. 2 train, test dataset is created, and both uses the year, month, day of month values. As an extra, the first one uses shifted trend values, the second one uses shifted trend values and shifted weekly prices of Bitcoin. We predict the prices based on the test data and show it in 2 graphs. Figure 50 is the prediction using shifted trends values, figure 51 is the prediction using shifted price and shifted Google Trends values. The blue line is predicted, and the orange line is the real weekly prices of ETH.

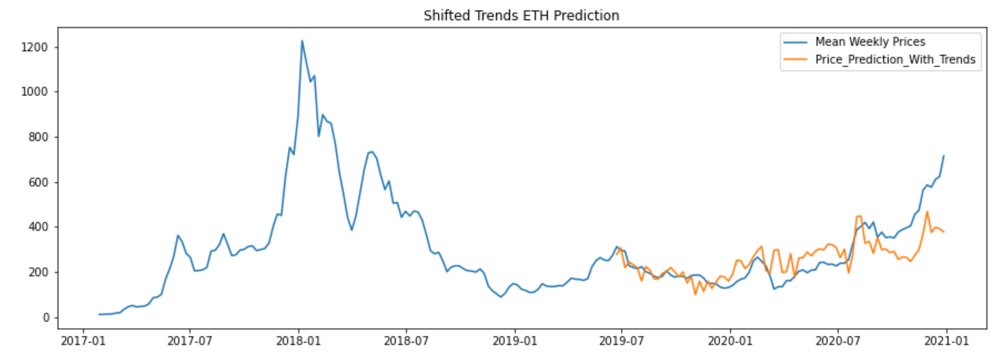


Fig. 50. ETH prediction with XGBOOST using dates and shifted trend values

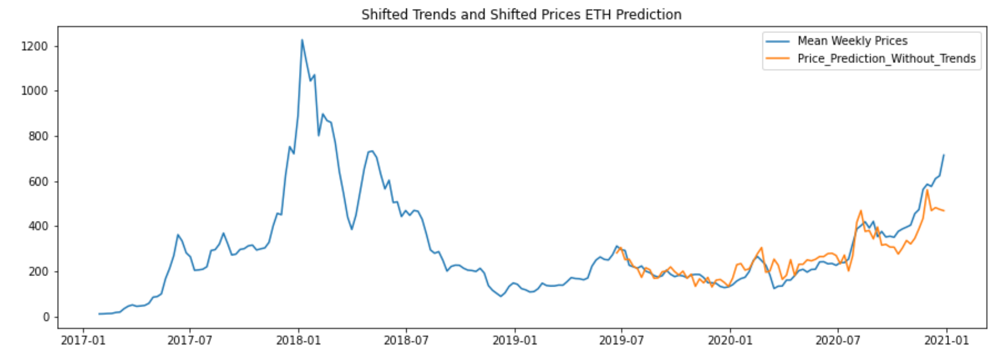


Fig. 51. ETH prediction with XGBOOST using dates, shifted trend and shifted prices values

We used the same metrics that were used for LSTM. When we used shifted dates and shifted trends as features, the RMSE result is 93.10, the MAE result is 68.67. When dates, shifted trends and shifted average prices are used as features, the RMSE result is 59.9, the MAE result is 43.67. If we observe both graphs, the predictions are similar but after analyzing the error results we concluded that shifted prices show a significant enhancement to our predictions.

Ethereum Cross Validation with XGBOOST

The techniques explained in the Solution Approach part of this paper are used for this part. The combinations of hyperparameters that were used are the same as the BTC Cross Validation implementation. At total 120 different models are created and their errors are calculated. The rolling forecast origin results when date and trend features are used, can be seen in Appendix-8 and the sliding window implementation results can be seen in Appendix-9. The best hyperparameters that was found for rolling forecast origin are 0.3 for the learning rate, 3 for the max depth, 7 for the minimum child weight, 0.3 for the gamma and 0.5 for the Col sample by tree. The RMSE for this hyperparameter is 680.56. The best hyperparameters found for sliding window are 0.2 for the learning rate, 3 for the max depth, 5 for the minimum child weight, 0.3 for the gamma and 0.5 for the Col sample by tree. The RMSE for this hyperparameter is 605.92.

After applying the same techniques to the dates, shifted trends, shifted prices values, the best hyperparameters for rolling forecast origin were found as 0.2 for the learning rate, 6 for the max depth, 5 for the minimum child weight, 0.3 for the gamma and 0.7 for the Col sample by tree. The RMSE for this hyperparameter is 541.12. The best hyperparameters for sliding window were found as 0.2 for the learning rate, 3 for the max depth, 5 for the minimum child weight, 0.3 for the gamma and 0.5 for the Col sample by tree. The RMSE for this hyperparameter is 603.37.

VI. Related Work

Related Work section will be divided into two sections, Related Tools and Related Techniques. In the first part we will explain various data collection and machine learning tools that could be alternatively used. Then, for the second part we will briefly explain different techniques that could be used to achieve different outcomes from the time series models.

Related Tools

There are numerous ways of data collecting and predicting with time series models. First, we will discuss data collection alternatives. During our research, we found that there were many websites regarding social and news analytics for cryptocurrencies. For example, our alternative to LunarCrush was DataPool, a similar cryptocurrency analytics website. Since we found far more data for distinct cryptocurrencies compared to DataPool, we decided to collect data from LunarCrush. Also, similar websites to Google Trends could be used like Semrush, Serpstat or SE Ranking. Some are more expensive and has a lot more options/data than others, but we used Google Trends because it was a lot easier to use compared to the alternatives. We have used the Tweepy library on our data collection module for accessing Twitter API. Instead of using this library, we could directly access the Twitter API by sending POST/GET requests. However, we would have to implement several methods for various endpoints of Twitter API and that would slow down the development process. Also, Tweepy library has built-in support for handling rate-limits of Twitter API which has a great usage in our project since we must pull large amounts of tweets from Twitter. For language detection of the tweets and sentiment analysis we have used Microsoft Azure's Text Analytics services. Instead of using this service for language detection we could have used TextBlob or Polyglot library. For more reliable language detection we have decided to go with Azure and decided to benefit from its sentiment analysis service as well thus only dealing with one provider on text analytics operations. It is worth noting that we have used Azure's free plan that has limitations while other options we have mentioned are free to use although being less capable.

Secondly, we have used the Python programming language for all our project. We could have used other languages like R for machine learning purposes, but we were extensively familiar with Python, and collecting data from the web was very useful when we used this language.

Related Techniques

There are many different techniques of achieving the task of time series forecasting. First, our goal was to predict closing prices using Linear Regression, LSTM and XGBOOST. Alternatively, Multi-Layer Perceptron's, ARIMA or CNN's (Convolutional Neural Networks) can be applied to time series forecasting. Additionally, we have used the top 2 cross validation techniques that is being used in these kinds of projects (rolling forecast origin, sliding window). There could be other ways of executing the cross validation, but we were only aware of these 2 methods. It goes without saying that the parameters of these cross-validation techniques heavily affect the outcome of the best hyperparameter and score output. The parameters of these methods could be changed to get a superior hyperparameter for the time series model. Lastly, different libraries for the machine learning models could be used if the goal is to predict with other models like ARIMA or MLP's.

VII. Conclusion and Future Work

In this section, we would like to conclude our work by providing a summary of our results and we would like to explain potential future work that can be done. There could be many enhancements to our project. For example, applying these machine learning models to other cryptocurrencies, trying entirely different time series models, collecting more and more data as time progresses and making predictions when cryptocurrency market matures, possibly after several decades. Furthermore, different data mining, cleaning methods and time series machine learning models could be used in the future. All in all, there are limitless ways these predictions could be executed but essentially, all the requirements that were mentioned on our proposal are successfully completed.

Impacts of the project

There are numerous economic impacts of our project. The prices and social media, news data of Bitcoin and Ethereum are collected as input. These data are open source and can be freely used by everyone. After collecting these data, the correlation of these data with the prices are documented and we used these data to predict the weekly/closing prices. Additionally, several metrics like Root Mean Squared Error and Mean Absolute Error are calculated after making these predictions by using Linear Regression and different time series models like XGBOOST, LSTM. 2 different cross validation techniques called rolling forecast origin and sliding window are used to calculate the best hyperparameter among many hyperparameters that were prepared. By shifting several features, using social media and news data, these outputs were acquired. These predictions can be used to further understand the capabilities of different time series machine learning models and the way they react to independent variables. Also, the methods that were used in our project to predict cryptocurrencies could potentially be used for predicting stock prices. The dominance of the LSTM models compared to XGBOOST, Linear Regression is documented in our findings, these outputs could potentially be proved by using stock prices as well. There are no social impacts we can think of, but these outputs are very beneficial for investors and people who are new to the market. New investors could use our outputs to further gain insight on the market sentiment and the predictions of machine learning models.

Economic and Ethical Issues

There are some economic risks, issues that come with our work. Investors should always invest their money while assessing their own risk. These cryptocurrencies are very volatile, and they can change negatively, positively in a small amount of time. Investors should research these cryptocurrencies and invest their money accordingly.

Acknowledgements

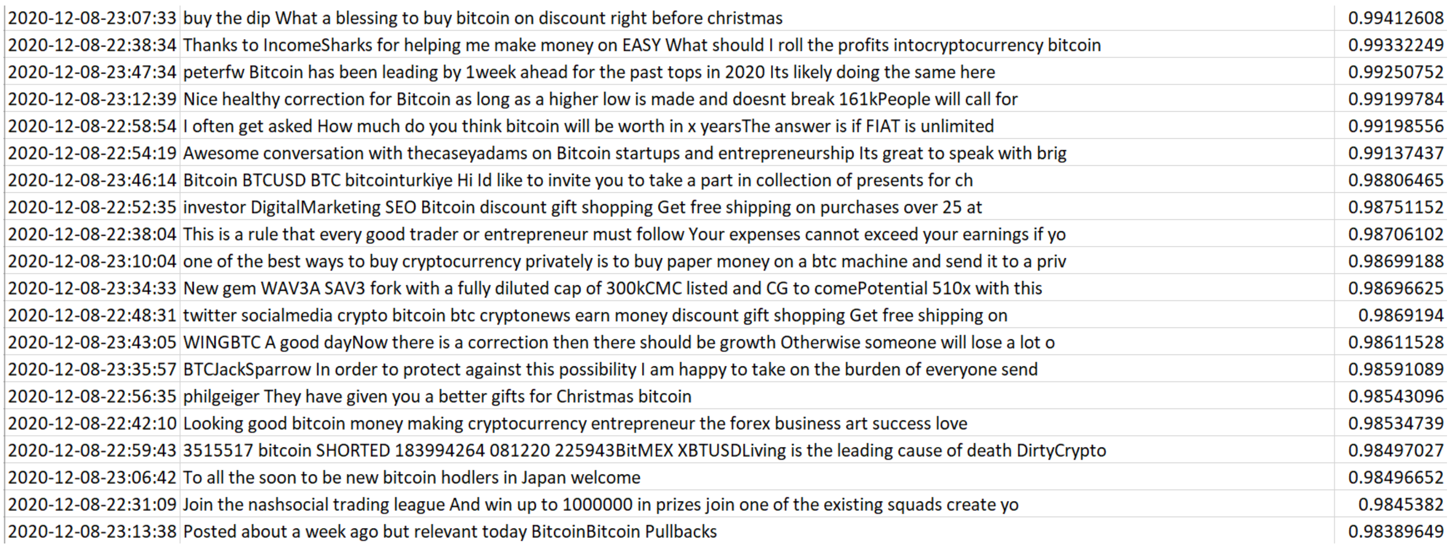
We would like to thank Dr. Emre SEFER for guiding us through this project and providing valuable feedback.

References

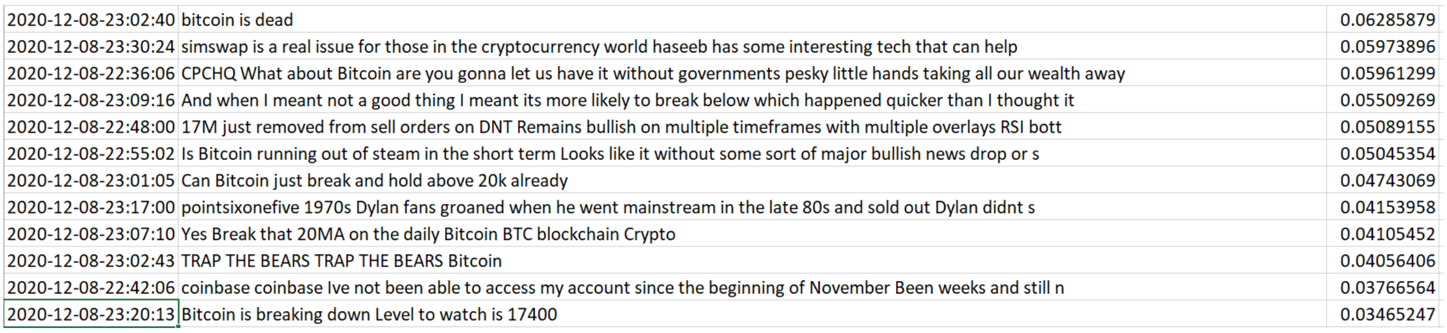
1. Wright, C., 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. *SSRN Electronic Journal*
2. Medium. 2021. *Illustrated Guide to LSTM's and GRU's: A step by step explanation*. [online] Available at: <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21> [Accessed 1 June 2021].
3. Scholar.smu.edu.2021.[online]Available at: <https://scholar.smu.edu/cgi/viewcontent.cgi?article=1039&context=datasciencereview> [Accessed 1 June 2021].
4. Kaggle.com. 2021. *Bitcoin 17.7 million Tweets and price*. [online] Available at: <https://www.kaggle.com/jaimebadiola/bitcoin-tweets-and-price> [Accessed 1 June 2021].
5. LunarCRUSH. 2021. *LunarCRUSH | Social Media Analytics for Cryptocurrencies*. [online] Available at: <https://lunarcrush.com> [Accessed 1 June 2021].
6. Kaggle.com. 2021. *Bitcoin tweets - 16M tweets*. [online] Available at: <https://www.kaggle.com/alaix14/bitcoin-tweets-20160101-to-20190329> [Accessed 1 June 2021].
7. Vo, A., 2019. Sentiment Analysis of News for Effective Cryptocurrency Price Prediction. International Journal of Knowledge Engineering, 5(2), pp.47-52.
8. CHOI, H. and VARIAN, H., 2012. Predicting the Present with Google Trends. *Economic Record*, 88, pp.2-9.
9. GitHub. 2021. *bariskaraer/Predicting-Crypto-Currency-Using-Social-Networks-Data*. [online] Available at: <https://github.com/bariskaraer/Predicting-Crypto-Currency-Using-Social-Networks-Data> [Accessed 1 June 2021].
10. Abraham, J., Higdon, D., Nelson, J. and Ibarra, J., 2018. Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis. *SMU Data Science Review*, 1(3), p.21.

Appendix

1) Sample Sentiment Analysis Results

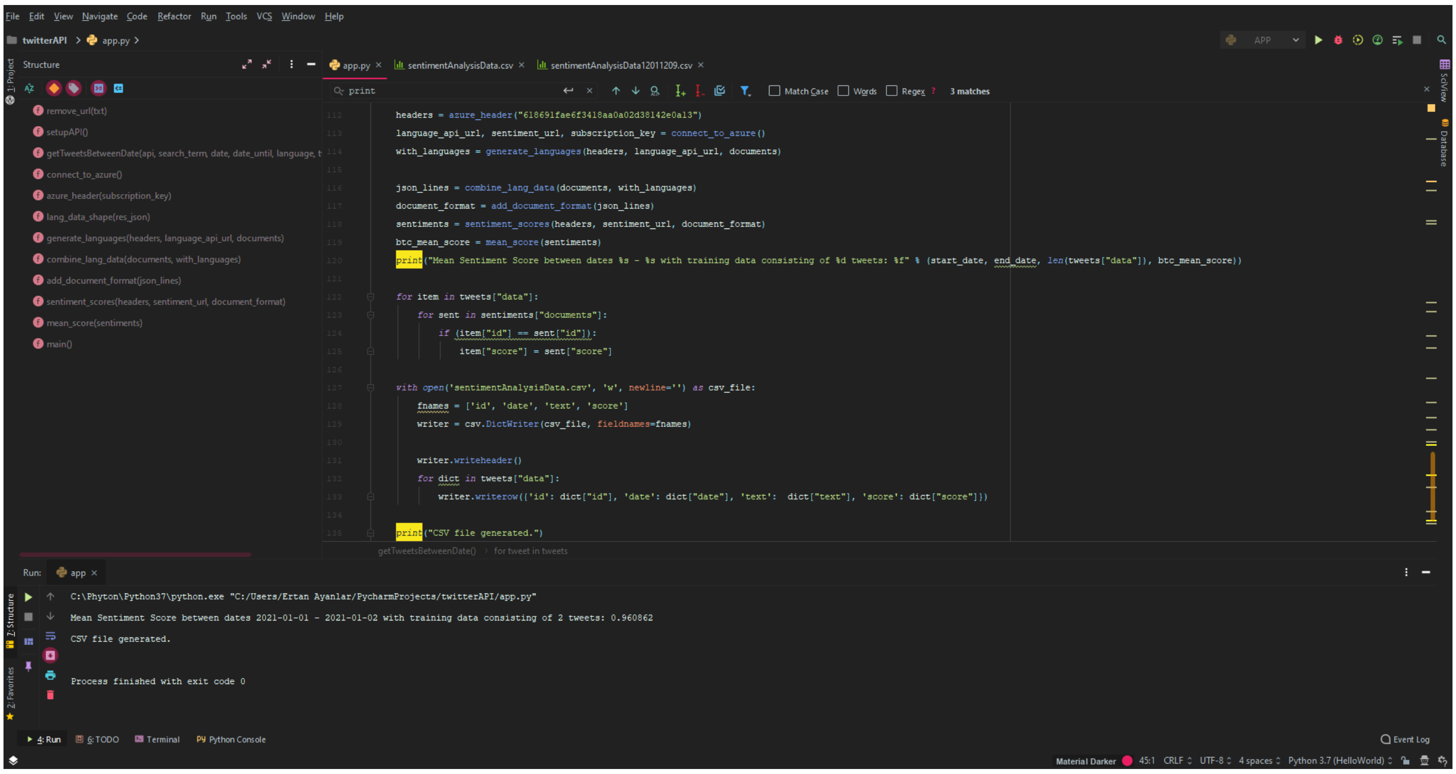


2020-12-08-23:07:33	buy the dip What a blessing to buy bitcoin on discount right before christmas	0.99412608
2020-12-08-22:38:34	Thanks to IncomeSharks for helping me make money on EASY What should I roll the profits intocryptocurrency bitcoin	0.99332249
2020-12-08-23:47:34	peterfw Bitcoin has been leading by 1week ahead for the past tops in 2020 Its likely doing the same here	0.99250752
2020-12-08-23:12:39	Nice healthy correction for Bitcoin as long as a higher low is made and doesnt break 161kPeople will call for	0.99199784
2020-12-08-22:58:54	I often get asked How much do you think bitcoin will be worth in x yearsThe answer is if FIAT is unlimited	0.99198556
2020-12-08-22:54:19	Awesome conversation with thecaseyadams on Bitcoin startups and entrepreneurship Its great to speak with brig	0.99137437
2020-12-08-23:46:14	Bitcoin BTCUSD BTC bitcointurkiye Hi Id like to invite you to take a part in collection of presents for ch	0.98806465
2020-12-08-22:52:35	investor DigitalMarketing SEO Bitcoin discount gift shopping Get free shipping on purchases over 25 at	0.98751152
2020-12-08-22:38:04	This is a rule that every good trader or entrepreneur must follow Your expenses cannot exceed your earnings if yo	0.98706102
2020-12-08-23:10:04	one of the best ways to buy cryptocurrency privately is to buy paper money on a btc machine and send it to a priv	0.98699188
2020-12-08-23:34:33	New gem WAV3A SAV3 fork with a fully diluted cap of 300kCMC listed and CG to comePotential 510x with this	0.98696625
2020-12-08-22:48:31	twitter socialmedia crypto bitcoin btc cryptonews earn money discount gift shopping Get free shipping on	0.9869194
2020-12-08-23:43:05	WINGBTC A good dayNow there is a correction then there should be growth Otherwise someone will lose a lot o	0.98611528
2020-12-08-23:35:57	BTCJackSparrow In order to protect against this possibility I am happy to take on the burden of everyone send	0.98591089
2020-12-08-22:56:35	philgeiger They have given you a better gifts for Christmas bitcoin	0.98543096
2020-12-08-22:42:10	Looking good bitcoin money making cryptocurrency entrepreneur the forex business art success love	0.98534739
2020-12-08-22:59:43	3515517 bitcoin SHORTED 183994264 081220 225943BitMEX XBTUSDLiving is the leading cause of death DirtyCrypto	0.98497027
2020-12-08-23:06:42	To all the soon to be new bitcoin hodlers in Japan welcome	0.98496652
2020-12-08-22:31:09	Join the nashsocial trading league And win up to 1000000 in prizes join one of the existing squads create yo	0.9845382
2020-12-08-23:13:38	Posted about a week ago but relevant today BitcoinBitcoin Pullbacks	0.98389649



2020-12-08-23:02:40	bitcoin is dead	0.06285879
2020-12-08-23:30:24	simswap is a real issue for those in the cryptocurrency world haseeb has some interesting tech that can help	0.05973896
2020-12-08-22:36:06	CPCHQ What about Bitcoin are you gonna let us have it without governments pesky little hands taking all our wealth away	0.05961299
2020-12-08-23:09:16	And when I meant not a good thing I meant its more likely to break below which happened quicker than I thought it	0.05509269
2020-12-08-22:48:00	17M just removed from sell orders on DNT Remains bullish on multiple timeframes with multiple overlays RSI bott	0.05089155
2020-12-08-22:55:02	Is Bitcoin running out of steam in the short term Looks like it without some sort of major bullish news drop or s	0.05045354
2020-12-08-23:01:05	Can Bitcoin just break and hold above 20k already	0.04743069
2020-12-08-23:17:00	pointsixonefive 1970s Dylan fans groaned when he went mainstream in the late 80s and sold out Dylan didnt s	0.04153958
2020-12-08-23:07:10	Yes Break that 20MA on the daily Bitcoin BTC blockchain Crypto	0.04105452
2020-12-08-23:02:43	TRAP THE BEARS TRAP THE BEARS Bitcoin	0.04056406
2020-12-08-22:42:06	coinbase coinbase Ive not been able to access my account since the beginning of November Been weeks and still n	0.03766564
2020-12-08-23:20:13	Bitcoin is breaking down Level to watch is 17400	0.03465247

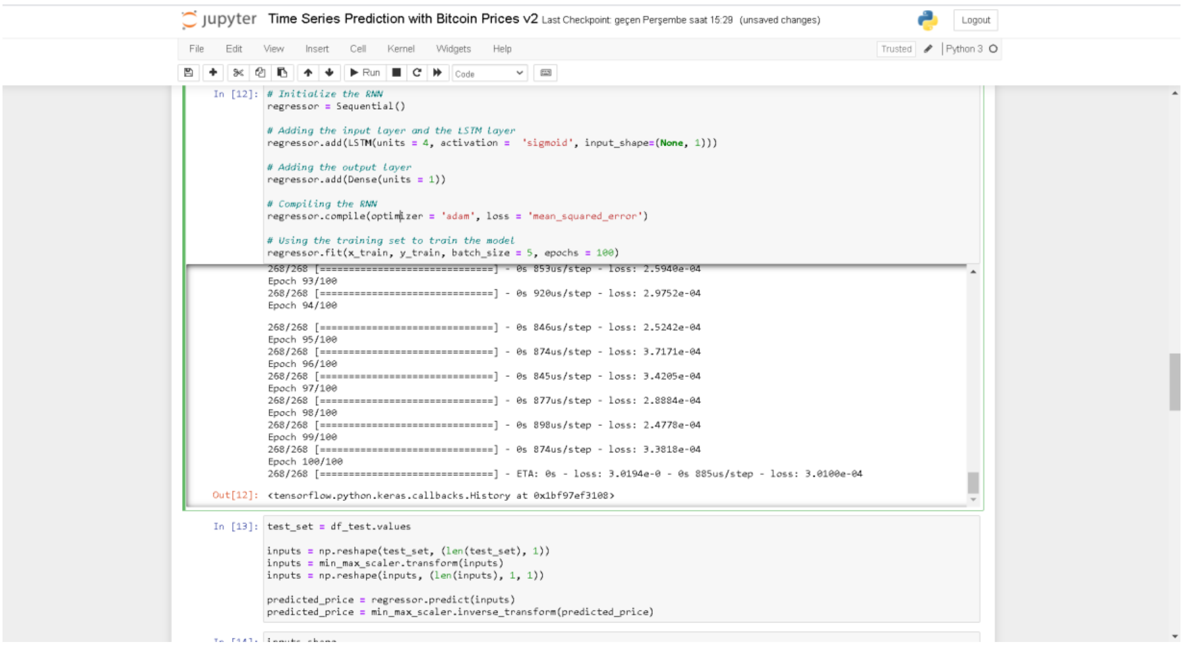
2) Data Collection Module IntelliJ PyCharm Working Environment


File Edit View Navigate Code Refactor Run Tools VCS Window Help
twitterAPI > app.py >
Structure
remove_url(txt)
setupAPI()
getTweetsBetweenDate(api, search_term, date, date_until, language, t
connect_to_azure()
azure_header(subscription_key)
lang_data_shape(res_json)
generate_languages(headers, language_api_url, documents)
combine_lang_data(documents, with_languages)
add_document_format(json_lines)
sentiment_scores(headers, sentiment_url, document_format)
mean_score(sentiments)
main()
app.py
sentimentAnalysisData.csv
sentimentAnalysisData12011209.csv
print
Match Case
Words
Regex
3 matches
getTweetsBetweenDate() > for tweet in tweets
Run: app
4: Run
6: TODO
Terminal
Python Console
Event Log
Material Darker 45:1 CRLF UTF-8 4 spaces Python 3.7 (HelloWorld)

```python
    headers = azure_header("618691fae6f3418aa0a02d38142e0a13")
    language_api_url, sentiment_url, subscription_key = connect_to_azure()
    with_languages = generate_languages(headers, language_api_url, documents)

    json_lines = combine_lang_data(documents, with_languages)
    document_format = add_document_format(json_lines)
    sentiments = sentiment_scores(headers, sentiment_url, document_format)
    btc_mean_score = mean_score(sentiments)
    print("Mean Sentiment Score between dates %s - %s with training data consisting of %d tweets: %f" % (start_date, end_date, len(tweets["data"]), btc_mean_score))

    for item in tweets["data"]:
        for sent in sentiments["documents"]:
            if (item["id"] == sent["id"]):
                item["score"] = sent["score"]

    with open('sentimentAnalysisData.csv', 'w', newline='') as csv_file:
        fnames = ['id', 'date', 'text', 'score']
        writer = csv.DictWriter(csv_file, fieldnames=fnames)

        writer.writeheader()
        for dict in tweets["data"]:
            writer.writerow({'id': dict["id"], 'date': dict["date"], 'text':  dict["text"], 'score': dict["score"]})

    print("CSV file generated.")
```

```
C:\Phyton\Python37\python.exe "C:/Users/Ertan Ayanlar/PycharmProjects/twitterAPI/app.py"
Mean Sentiment Score between dates 2021-01-01 - 2021-01-02 with training data consisting of 2 tweets: 0.960862
CSV file generated.

Process finished with exit code 0
```

3) Price Prediction Module Jupyter Notebook Working Environment

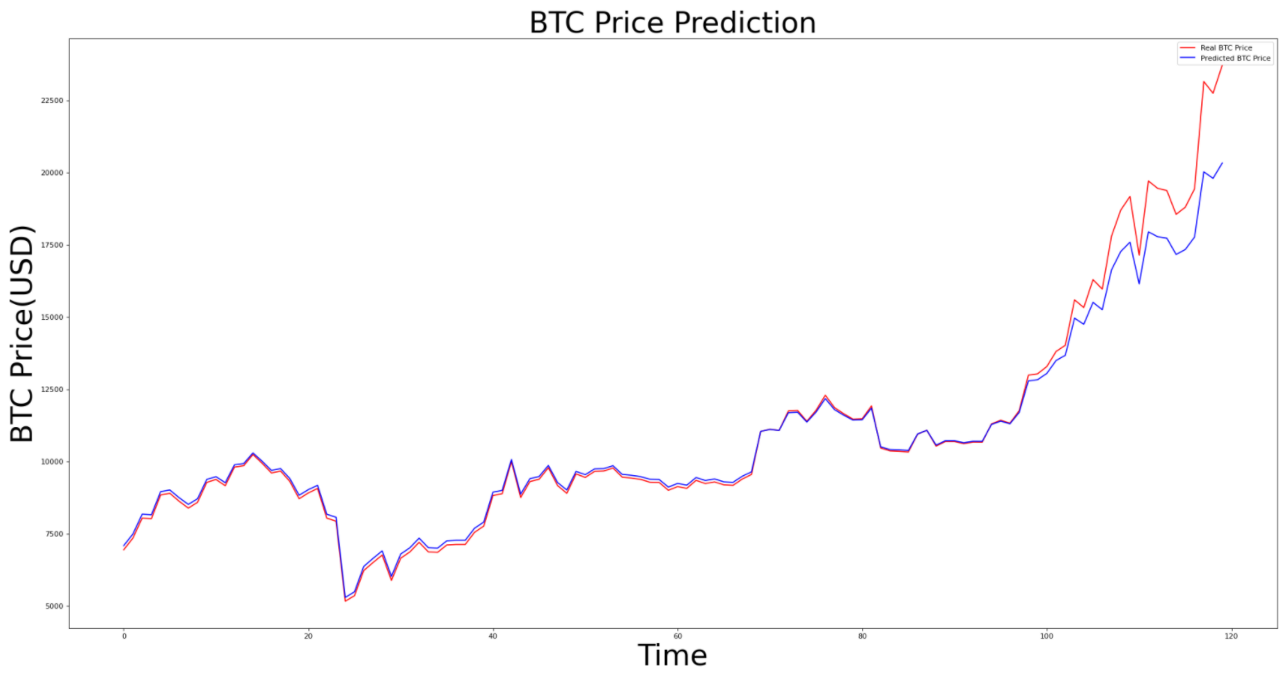

jupyter
Time Series Prediction with Bitcoin Prices v2 Last Checkpoint: geçen Perşembe saat 15:29 (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
In [12]: # Initialize the RNN
         regressor = Sequential()

         # Adding the input layer and the LSTM layer
         regressor.add(LSTM(units = 4, activation =  'sigmoid', input_shape=(None, 1)))

         # Adding the output layer
         regressor.add(Dense(units = 1))

         # Compiling the RNN
         regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')

         # Using the training set to train the model
         regressor.fit(x_train, y_train, batch_size = 5, epochs = 100)
```

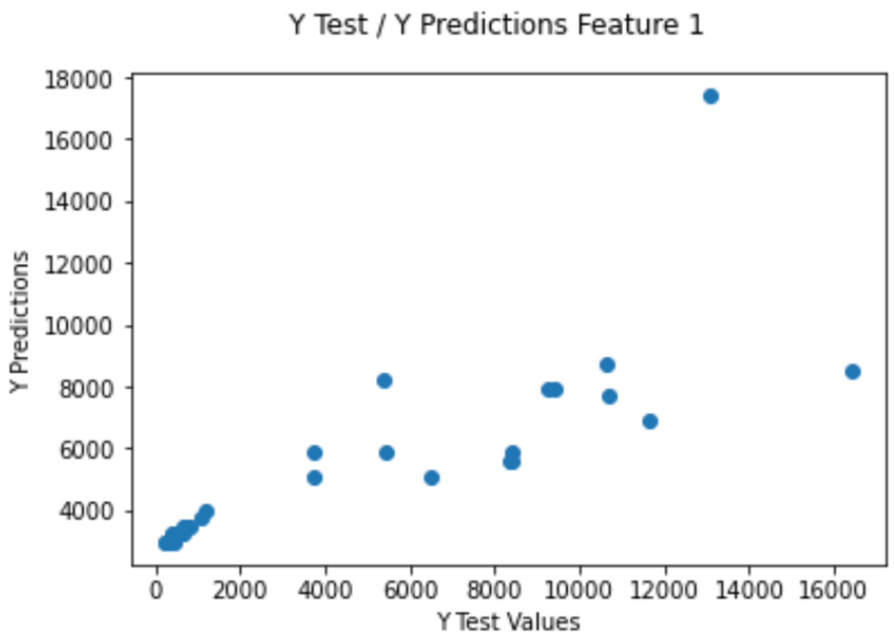
```
268/268 [==============================] - 0s 853us/step - loss: 2.5940e-04
Epoch 93/100
268/268 [==============================] - 0s 920us/step - loss: 2.9752e-04
Epoch 94/100

268/268 [==============================] - 0s 846us/step - loss: 2.5242e-04
Epoch 95/100
268/268 [==============================] - 0s 874us/step - loss: 3.7171e-04
Epoch 96/100
268/268 [==============================] - 0s 845us/step - loss: 3.4205e-04
Epoch 97/100
268/268 [==============================] - 0s 877us/step - loss: 2.8884e-04
Epoch 98/100
268/268 [==============================] - 0s 898us/step - loss: 2.4778e-04
Epoch 99/100
268/268 [==============================] - 0s 874us/step - loss: 3.3818e-04
Epoch 100/100
268/268 [==============================] - ETA: 0s - loss: 3.0194e-0 - 0s 885us/step - loss: 3.0100e-04

Out[12]: <tensorflow.python.keras.callbacks.History at 0x1bf97ef3108>
```

```
In [13]: test_set = df_test.values

         inputs = np.reshape(test_set, (len(test_set), 1))
         inputs = min_max_scaler.transform(inputs)
         inputs = np.reshape(inputs, (len(inputs), 1, 1))

         predicted_price = regressor.predict(inputs)
         predicted_price = min_max_scaler.inverse_transform(predicted_price)
```

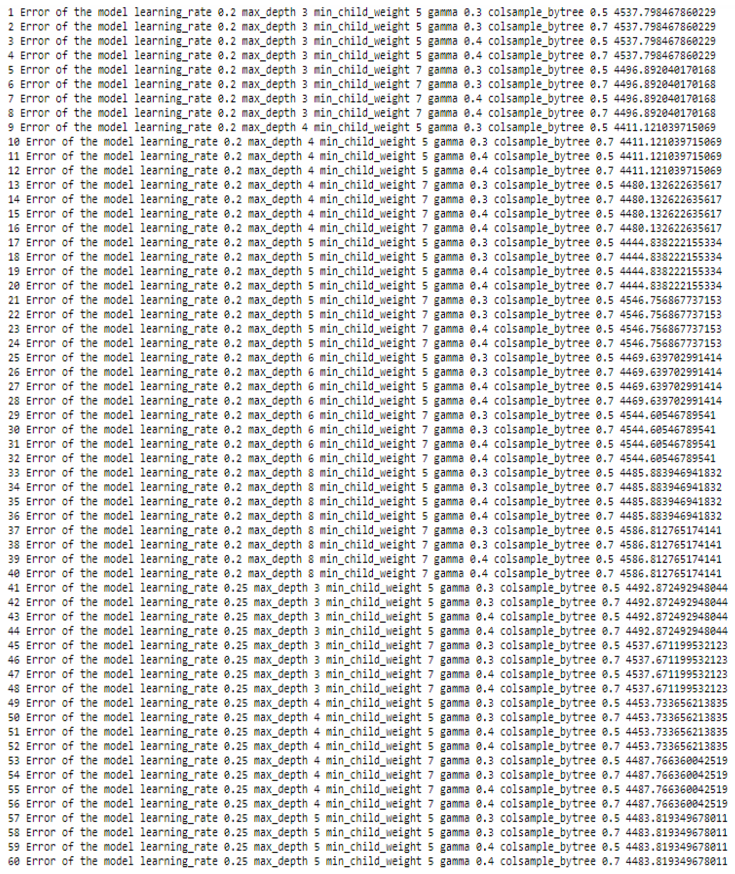
4) Price Prediction Module Time Series LSTM Prediction Graph



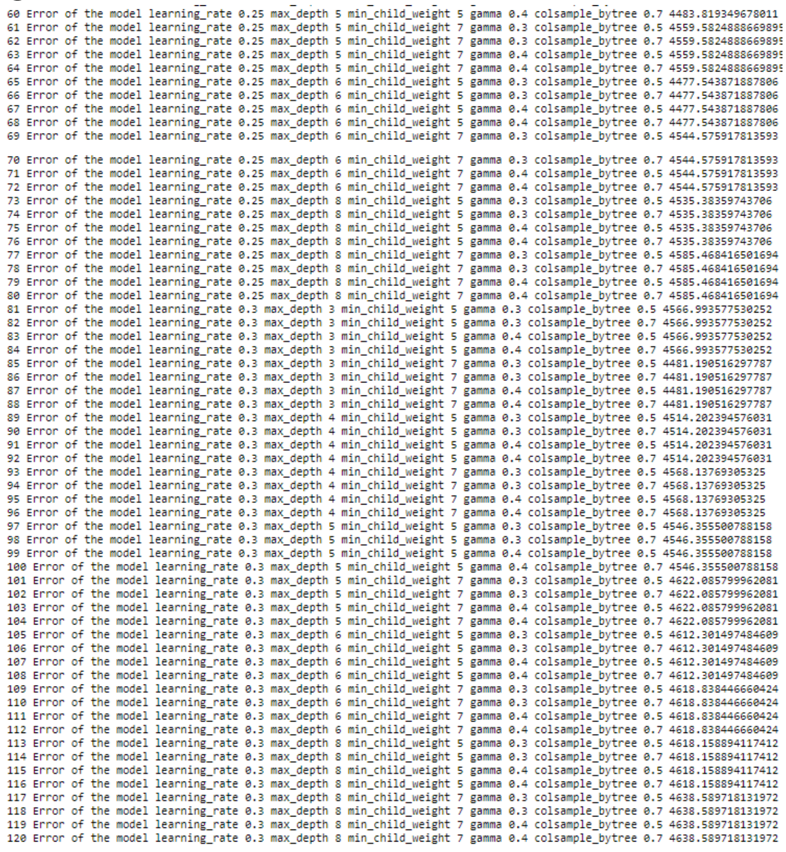
5) Price Prediction ModuleLinear Regression with Google Trends Data Prediction Graph



6) Bitcoin Rolling Forecast Cross Validation Results using XGBOOST

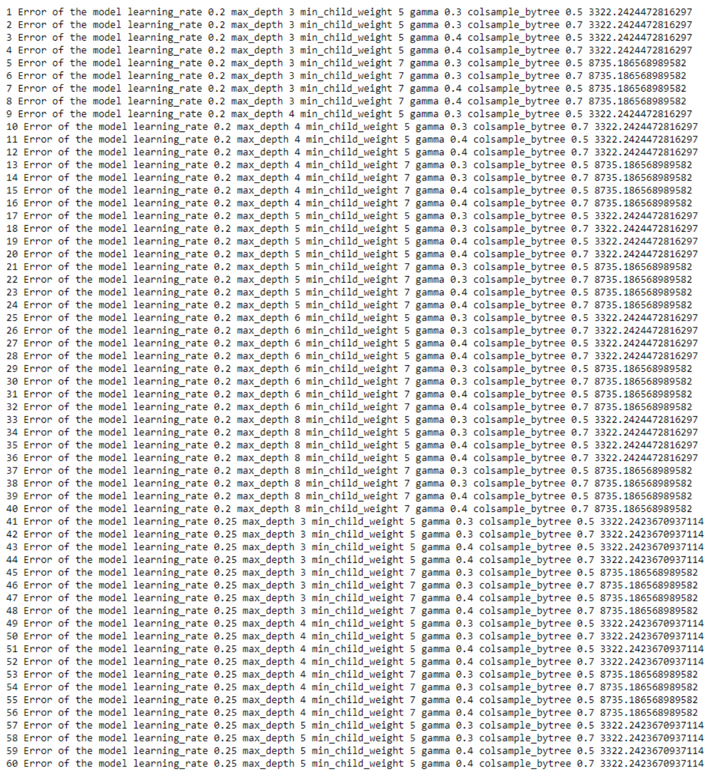


```
1 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 4537.798467860229
2 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 4537.798467860229
3 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 4537.798467860229
4 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 4537.798467860229
5 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 4496.892040170168
6 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 4496.892040170168
7 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 4496.892040170168
8 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 4496.892040170168
9 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 4411.121039715069
10 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 4411.121039715069
11 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 4411.121039715069
12 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 4411.121039715069
13 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 4480.132622635617
14 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 4480.132622635617
15 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 4480.132622635617
16 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 4480.132622635617
17 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 4444.838222155334
18 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 4444.838222155334
19 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 4444.838222155334
20 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 4444.838222155334
21 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 4546.756867737153
22 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 4546.756867737153
23 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 4546.756867737153
24 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 4546.756867737153
25 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 4469.639702991414
26 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 4469.639702991414
27 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 4469.639702991414
28 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 4469.639702991414
29 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 4544.60546789541
30 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 4544.60546789541
31 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 4544.60546789541
32 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 4544.60546789541
33 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 4485.883946941832
34 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 4485.883946941832
35 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 4485.883946941832
36 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 4485.883946941832
37 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 4586.812765174141
38 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 4586.812765174141
39 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 4586.812765174141
40 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 4586.812765174141
41 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 4492.872492948044
42 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 4492.872492948044
43 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 4492.872492948044
44 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 4492.872492948044
45 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 4537.671199532123
46 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 4537.671199532123
47 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 4537.671199532123
48 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 4537.671199532123
49 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 4453.733656213835
50 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 4453.733656213835
51 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 4453.733656213835
52 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 4453.733656213835
53 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 4487.766360042519
54 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 4487.766360042519
55 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 4487.766360042519
56 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 4487.766360042519
57 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 4483.819349678011
58 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 4483.819349678011
59 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 4483.819349678011
60 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 4483.819349678011
```

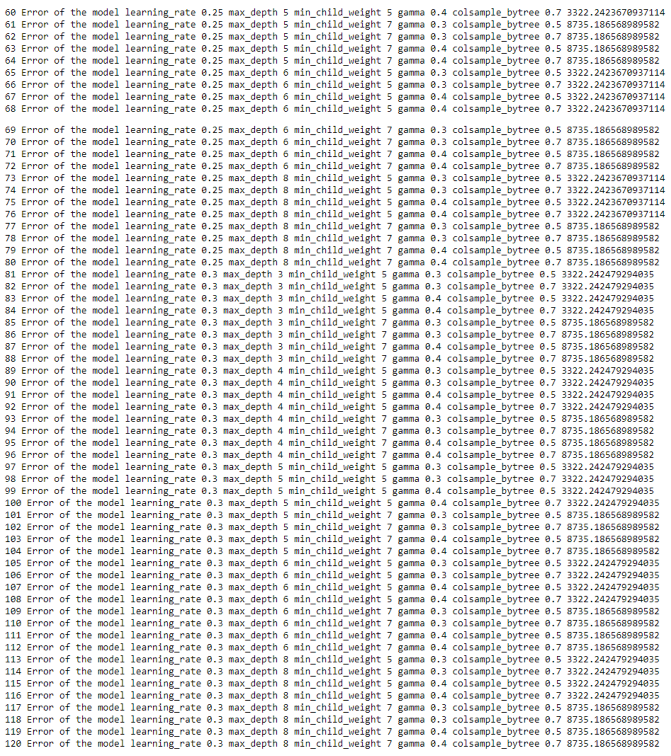


```
60 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 4483.819349678011
61 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 4559.5824888669895
62 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 4559.5824888669895
63 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 4559.5824888669895
64 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 4559.5824888669895
65 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 4477.543871887806
66 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 4477.543871887806
67 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 4477.543871887806
68 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 4477.543871887806
69 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 4544.575917813593

70 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 4544.575917813593
71 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 4544.575917813593
72 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 4544.575917813593
73 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 4535.38359743706
74 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 4535.38359743706
75 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 4535.38359743706
76 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 4535.38359743706
77 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 4585.468416501694
78 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 4585.468416501694
79 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 4585.468416501694
80 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 4585.468416501694
81 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 4566.993577530252
82 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 4566.993577530252
83 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 4566.993577530252
84 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 4566.993577530252
85 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 4481.190516297787
86 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 4481.190516297787
87 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 4481.190516297787
88 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 4481.190516297787
89 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 4514.202394576031
90 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 4514.202394576031
91 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 4514.202394576031
92 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 4514.202394576031
93 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 4568.13769305325
94 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 4568.13769305325
95 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 4568.13769305325
96 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 4568.13769305325
97 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 4546.355500788158
98 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 4546.355500788158
99 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 4546.355500788158
100 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 4546.355500788158
101 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 4622.085799962081
102 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 4622.085799962081
103 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 4622.085799962081
104 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 4622.085799962081
105 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 4612.301497484609
106 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 4612.301497484609
107 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 4612.301497484609
108 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 4612.301497484609
109 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 4618.838446660424
110 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 4618.838446660424
111 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 4618.838446660424
112 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 4618.838446660424
113 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 4618.158894117412
114 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 4618.158894117412
115 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 4618.158894117412
116 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 4618.158894117412
117 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 4638.589718131972
118 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 4638.589718131972
119 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 4638.589718131972
120 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 4638.589718131972
```

7) Bitcoin Sliding Window Cross Validation Results using XGBOOST

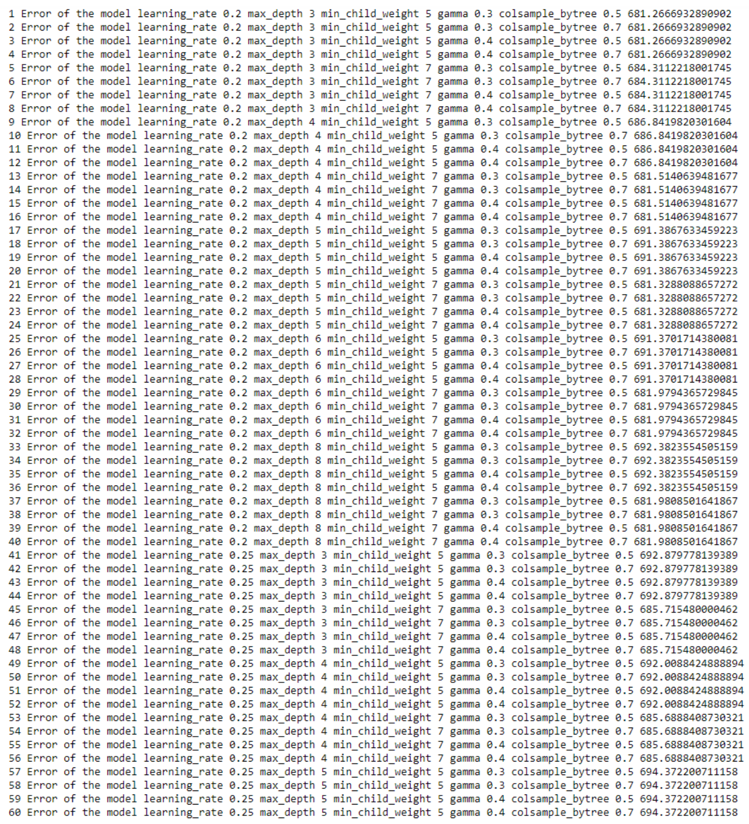


```
1 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 3322.2424472816297
2 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 3322.2424472816297
3 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 3322.2424472816297
4 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 3322.2424472816297
5 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 8735.186568989582
6 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 8735.186568989582
7 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 8735.186568989582
8 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 8735.186568989582
9 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 3322.2424472816297
10 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 3322.2424472816297
11 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 3322.2424472816297
12 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 3322.2424472816297
13 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 8735.186568989582
14 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 8735.186568989582
15 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 8735.186568989582
16 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 8735.186568989582
17 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 3322.2424472816297
18 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 3322.2424472816297
19 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 3322.2424472816297
20 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 3322.2424472816297
21 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 8735.186568989582
22 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 8735.186568989582
23 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 8735.186568989582
24 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 8735.186568989582
25 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 3322.2424472816297
26 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 3322.2424472816297
27 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 3322.2424472816297
28 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 3322.2424472816297
29 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 8735.186568989582
30 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 8735.186568989582
31 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 8735.186568989582
32 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 8735.186568989582
33 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 3322.2424472816297
34 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 3322.2424472816297
35 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 3322.2424472816297
36 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 3322.2424472816297
37 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 8735.186568989582
38 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 8735.186568989582
39 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 8735.186568989582
40 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 8735.186568989582
41 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 3322.2423670937114
42 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 3322.2423670937114
43 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 3322.2423670937114
44 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 3322.2423670937114
45 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 8735.186568989582
46 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 8735.186568989582
47 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 8735.186568989582
48 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 8735.186568989582
49 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 3322.2423670937114
50 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 3322.2423670937114
51 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 3322.2423670937114
52 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 3322.2423670937114
53 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 8735.186568989582
54 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 8735.186568989582
55 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 8735.186568989582
56 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 8735.186568989582
57 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 3322.2423670937114
58 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 3322.2423670937114
59 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 3322.2423670937114
60 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 3322.2423670937114
```

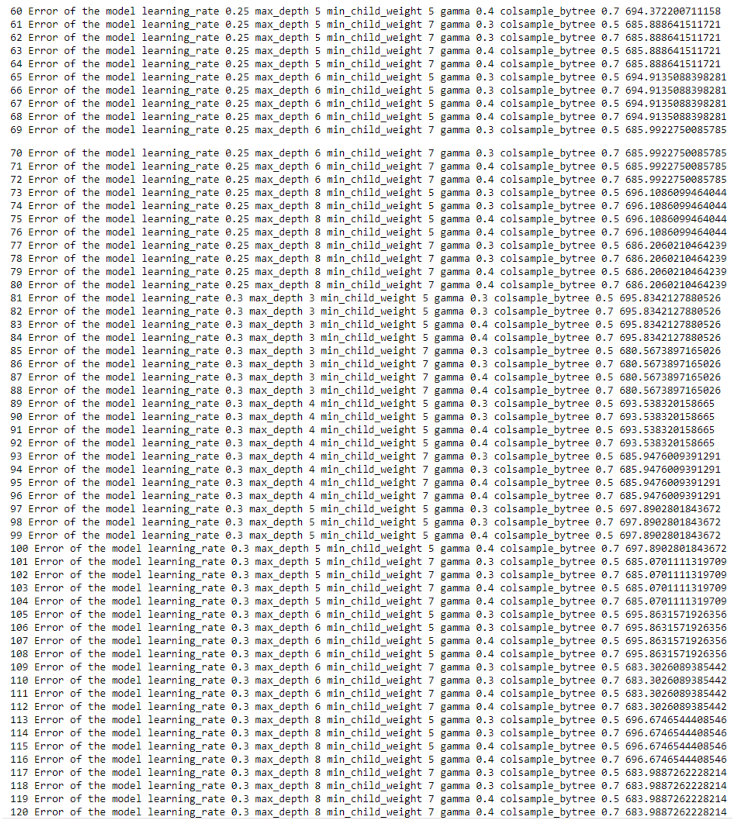


```
60 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 3322.2423670937114
61 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 8735.186568989582
62 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 8735.186568989582
63 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 8735.186568989582
64 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 8735.186568989582
65 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 3322.2423670937114
66 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 3322.2423670937114
67 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 3322.2423670937114
68 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 3322.2423670937114

69 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 8735.186568989582
70 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 8735.186568989582
71 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 8735.186568989582
72 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 8735.186568989582
73 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 3322.2423670937114
74 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 3322.2423670937114
75 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 3322.2423670937114
76 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 3322.2423670937114
77 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 8735.186568989582
78 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 8735.186568989582
79 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 8735.186568989582
80 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 8735.186568989582
81 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 3322.242479294035
82 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 3322.242479294035
83 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 3322.242479294035
84 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 3322.242479294035
85 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 8735.186568989582
86 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 8735.186568989582
87 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 8735.186568989582
88 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 8735.186568989582
89 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 3322.242479294035
90 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 3322.242479294035
91 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 3322.242479294035
92 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 3322.242479294035
93 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 8735.186568989582
94 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 8735.186568989582
95 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 8735.186568989582
96 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 8735.186568989582
97 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 3322.242479294035
98 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 3322.242479294035
99 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 3322.242479294035
100 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 3322.242479294035
101 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 8735.186568989582
102 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 8735.186568989582
103 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 8735.186568989582
104 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 8735.186568989582
105 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 3322.242479294035
106 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 3322.242479294035
107 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 3322.242479294035
108 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 3322.242479294035
109 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 8735.186568989582
110 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 8735.186568989582
111 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 8735.186568989582
112 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 8735.186568989582
113 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 3322.242479294035
114 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 3322.242479294035
115 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 3322.242479294035
116 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 3322.242479294035
117 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 8735.186568989582
118 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 8735.186568989582
119 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 8735.186568989582
120 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 8735.186568989582
```

8) Ethereum Rolling Forecast Origin Cross Validation Results using XGBOOST

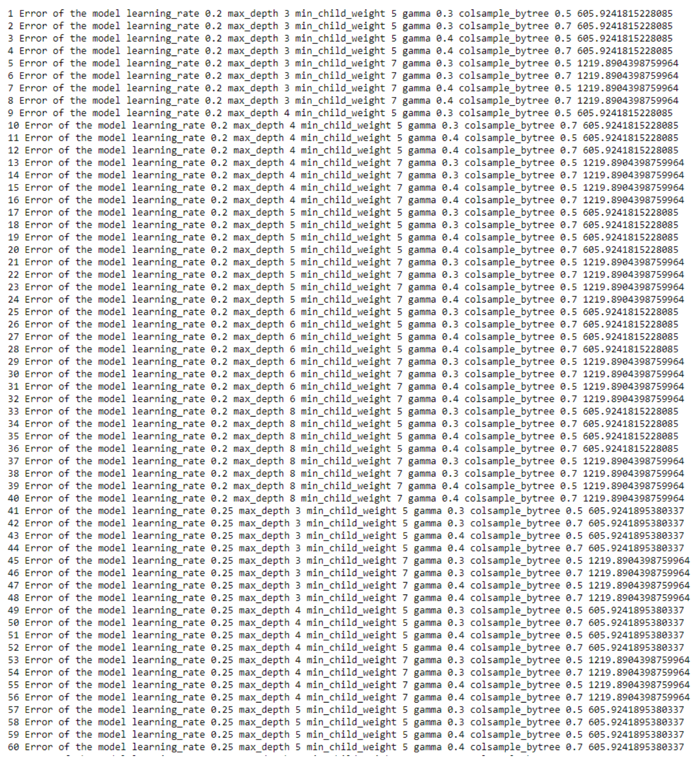


```
1 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 681.266693289092
2 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 681.266693289092
3 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 681.266693289092
4 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 681.266693289092
5 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 684.3112218001745
6 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 684.3112218001745
7 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 684.3112218001745
8 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 684.3112218001745
9 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 686.8419820301604
10 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 686.8419820301604
11 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 686.8419820301604
12 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 686.8419820301604
13 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 681.5140639481677
14 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 681.5140639481677
15 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 681.5140639481677
16 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 681.5140639481677
17 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 691.3867633459223
18 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 691.3867633459223
19 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 691.3867633459223
20 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 691.3867633459223
21 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 681.3288088657272
22 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 681.3288088657272
23 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 681.3288088657272
24 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 681.3288088657272
25 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 691.3701714380081
26 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 691.3701714380081
27 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 691.3701714380081
28 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 691.3701714380081
29 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 681.9794365729845
30 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 681.9794365729845
31 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 681.9794365729845
32 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 681.9794365729845
33 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 692.3823554505159
34 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 692.3823554505159
35 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 692.3823554505159
36 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 692.3823554505159
37 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 681.9808501641867
38 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 681.9808501641867
39 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 681.9808501641867
40 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 681.9808501641867
41 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 692.879778139389
42 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 692.879778139389
43 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 692.879778139389
44 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 692.879778139389
45 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 685.715480000462
46 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 685.715480000462
47 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 685.715480000462
48 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 685.715480000462
49 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 692.0088424888894
50 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 692.0088424888894
51 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 692.0088424888894
52 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 692.0088424888894
53 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 685.6888408730321
54 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 685.6888408730321
55 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 685.6888408730321
56 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 685.6888408730321
57 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 694.372200711158
58 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 694.372200711158
59 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 694.372200711158
60 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 694.372200711158
```

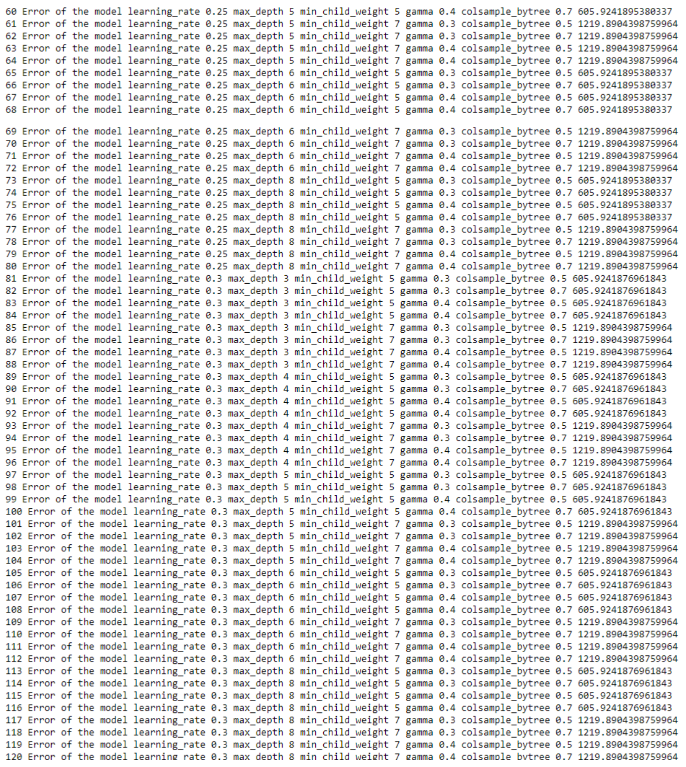


```
60 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 694.372200711158
61 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 685.888641511721
62 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 685.888641511721
63 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 685.888641511721
64 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 685.888641511721
65 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 694.9135088398281
66 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 694.9135088398281
67 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 694.9135088398281
68 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 694.9135088398281
69 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 685.9922750085785

70 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 685.9922750085785
71 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 685.9922750085785
72 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 685.9922750085785
73 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 696.1086099464044
74 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 696.1086099464044
75 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 696.1086099464044
76 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 696.1086099464044
77 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 686.2060210464239
78 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 686.2060210464239
79 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 686.2060210464239
80 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 686.2060210464239
81 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 695.8342127880526
82 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 695.8342127880526
83 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 695.8342127880526
84 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 695.8342127880526
85 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 680.5673897165026
86 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 680.5673897165026
87 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 680.5673897165026
88 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 680.5673897165026
89 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 693.538320158665
90 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 693.538320158665
91 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 693.538320158665
92 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 693.538320158665
93 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 685.9476009391291
94 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 685.9476009391291
95 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 685.9476009391291
96 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 685.9476009391291
97 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 697.8902801843672
98 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 697.8902801843672
99 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 697.8902801843672
100 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 697.8902801843672
101 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 685.0701111319709
102 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 685.0701111319709
103 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 685.0701111319709
104 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 685.0701111319709
105 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 695.8631571926356
106 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 695.8631571926356
107 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 695.8631571926356
108 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 695.8631571926356
109 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 683.3026089385442
110 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 683.3026089385442
111 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 683.3026089385442
112 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 683.3026089385442
113 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 696.6746544408546
114 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 696.6746544408546
115 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 696.6746544408546
116 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 696.6746544408546
117 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 683.9887262228214
118 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 683.9887262228214
119 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 683.9887262228214
120 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 683.9887262228214
```

9) Ethereum Sliding Window Cross Validation Results using XGBOOST



```
1 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 605.9241815228085
2 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 605.9241815228085
3 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 605.9241815228085
4 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 605.9241815228085
5 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 1219.8904398759964
6 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 1219.8904398759964
7 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 1219.8904398759964
8 Error of the model learning_rate 0.2 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 1219.8904398759964
9 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 605.9241815228085
10 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 605.9241815228085
11 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 605.9241815228085
12 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 605.9241815228085
13 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 1219.8904398759964
14 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 1219.8904398759964
15 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 1219.8904398759964
16 Error of the model learning_rate 0.2 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 1219.8904398759964
17 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 605.9241815228085
18 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 605.9241815228085
19 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 605.9241815228085
20 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 605.9241815228085
21 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 1219.8904398759964
22 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 1219.8904398759964
23 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 1219.8904398759964
24 Error of the model learning_rate 0.2 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 1219.8904398759964
25 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 605.9241815228085
26 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 605.9241815228085
27 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 605.9241815228085
28 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 605.9241815228085
29 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 1219.8904398759964
30 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 1219.8904398759964
31 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 1219.8904398759964
32 Error of the model learning_rate 0.2 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 1219.8904398759964
33 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 605.9241815228085
34 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 605.9241815228085
35 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 605.9241815228085
36 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 605.9241815228085
37 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 1219.8904398759964
38 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 1219.8904398759964
39 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 1219.8904398759964
40 Error of the model learning_rate 0.2 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 1219.8904398759964
41 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 605.9241895380337
42 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 605.9241895380337
43 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 605.9241895380337
44 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 605.9241895380337
45 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 1219.8904398759964
46 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 1219.8904398759964
47 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 1219.8904398759964
48 Error of the model learning_rate 0.25 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 1219.8904398759964
49 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 605.9241895380337
50 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 605.9241895380337
51 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 605.9241895380337
52 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 605.9241895380337
53 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 1219.8904398759964
54 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 1219.8904398759964
55 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 1219.8904398759964
56 Error of the model learning_rate 0.25 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 1219.8904398759964
57 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 605.9241895380337
58 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 605.9241895380337
59 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 605.9241895380337
60 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 605.9241895380337
```



```
60 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 605.9241895380337
61 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 1219.8904398759964
62 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 1219.8904398759964
63 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 1219.8904398759964
64 Error of the model learning_rate 0.25 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 1219.8904398759964
65 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 605.9241895380337
66 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 605.9241895380337
67 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 605.9241895380337
68 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 605.9241895380337

69 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 1219.8904398759964
70 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 1219.8904398759964
71 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 1219.8904398759964
72 Error of the model learning_rate 0.25 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 1219.8904398759964
73 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 605.9241895380337
74 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 605.9241895380337
75 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 605.9241895380337
76 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 605.9241895380337
77 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 1219.8904398759964
78 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 1219.8904398759964
79 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 1219.8904398759964
80 Error of the model learning_rate 0.25 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 1219.8904398759964
81 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 605.9241876961843
82 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 605.9241876961843
83 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 605.9241876961843
84 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 605.9241876961843
85 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 1219.8904398759964
86 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 1219.8904398759964
87 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 1219.8904398759964
88 Error of the model learning_rate 0.3 max_depth 3 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 1219.8904398759964
89 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 605.9241876961843
90 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 605.9241876961843
91 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 605.9241876961843
92 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 605.9241876961843
93 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 1219.8904398759964
94 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 1219.8904398759964
95 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 1219.8904398759964
96 Error of the model learning_rate 0.3 max_depth 4 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 1219.8904398759964
97 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 605.9241876961843
98 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 605.9241876961843
99 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 605.9241876961843
100 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 605.9241876961843
101 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 1219.8904398759964
102 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 1219.8904398759964
103 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 1219.8904398759964
104 Error of the model learning_rate 0.3 max_depth 5 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 1219.8904398759964
105 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 605.9241876961843
106 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 605.9241876961843
107 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 605.9241876961843
108 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 605.9241876961843
109 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 1219.8904398759964
110 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 1219.8904398759964
111 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 1219.8904398759964
112 Error of the model learning_rate 0.3 max_depth 6 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 1219.8904398759964
113 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.5 605.9241876961843
114 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 5 gamma 0.3 colsample_bytree 0.7 605.9241876961843
115 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.5 605.9241876961843
116 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 5 gamma 0.4 colsample_bytree 0.7 605.9241876961843
117 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.5 1219.8904398759964
118 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 7 gamma 0.3 colsample_bytree 0.7 1219.8904398759964
119 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.5 1219.8904398759964
120 Error of the model learning_rate 0.3 max_depth 8 min_child_weight 7 gamma 0.4 colsample_bytree 0.7 1219.8904398759964
```