AKDENIZ UNIVERSITY
FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING

KZ ENGINE

(Trading Signal Engine for Movement direction change with Decision Tress and Sentiment Analysis)

Uğur AKYEL

GRADUATION SENIOR PROJECT REPORT

SUPERVISOR: Asist.Prof.Dr. Ogr. Alper ÖZCAN

JUNE 2023

ANTALYA

ABSTRACT

KZ ENGINE

Uğur AKYEL

Graduation Senior Project **in Computer Engineering**

Supervisor: Dr. Alper ÖZCAN

May 2023; ---- 32 Pages

The objective of this project is to use various techniques such as ***Decision Trees, Sentiment Analysis with Natural Language Processing, and Japanese Candlestick Art*** to forecast the structure of the next candle for popular cryptocurrencies like Bitcoin, Ethereum, Binance coin, Ripple, and Doge coin. The predictions will be made using a binary format and will aim to determine whether the price movement of the cryptocurrencies in the market will increase or decrease within the next 1 hour and next 1 day based on the candle structure.

In addition to these techniques, the research also involves studying other technical analysis and indicator structures, as well as incorporating *Twitter data and sentiment analysis scores into the model.*

The research resulted in the development of an application with a backend data and AI modelling pipeline, as well as a frontend for users to view the coin signal results and how many tweets were evaluated.

However, it is important to note that predicting the movement of cryptocurrencies can be challenging due to their volatility and the many external factors that may impact their value. Therefore, while this project may provide useful insights, investors should exercise caution and consider all relevant information before making investment decisions.

The motivation behind this project is that *when people decide to buy a coin, generally the price of that coin in the stock market starts to fall, and when all indicators say "sell," the price goes up, or when they say "buy," the price goes down. This is called the point of financial saturation.* ***My goal is to create an artificial intelligence-based system that can prevent this from happening, by detecting these two points and presenting them to users through various methods.*** Additionally, I aim to turn this system into an application and meet with customers. By evaluating this process with artificial intelligence and applications, I want to create an application that will help people make long-term profits and reduce trading stress by detecting price trends.

Financial saturation is a well-known phenomenon in the world of trading and investing, and it can be difficult for individuals to navigate. With the advancements in technology and the increasing popularity of artificial intelligence, there is an opportunity to create a system that can help individuals make better decisions when it comes to trading. By detecting the points of financial saturation, individuals can avoid buying or selling at the wrong time and potentially lose money. This project aims to develop such a system using various methods and algorithms, including the use of machine learning and neural networks. Additionally, by creating an application that can be used by customers, this system can be accessible to a wider audience, ultimately helping more people make profitable trades and achieve their financial goals.

The first step of this project is to develop a machine learning workflow to predict cryptocurrency prices. The workflow will begin with data pre-processing using multiple APIs, including ***Tradingview, Binance, Yahoo Finance, and CTXC. The API*** structures will be examined through various trials.

Next, technical analysis libraries under Python will be utilized to conduct technical analysis and create new features. Once the necessary indicators and features are identified, technical analysis strategies that have proven to be effective in the finance market will be labelled. The addition of Japanese candlestick recognition to these features is a desired outcome.

To utilize the Japanese Candlestick Art feature, the TA-LIB library will be employed as it is the only library that provides this particular candlestick structure. Furthermore, it provides programming language support for C and C++.

The goal of this project is to develop profitable strategies for accurately predicting cryptocurrency prices using a data pipeline that generates data frames for both deep learning modelling and technical analysis with the help of ***the pandas library***.

Natural language processing operations will be conducted using Twitter APIs to gather related data, and the impact of new features and people's thoughts and sentiments on cryptocurrency prices will be monitored through data mining and its APIs. The optimization and effects of the model will be observed with the process added to the data pipeline.

Sentiment analysis is an important aspect of this project as it helps to understand the emotions and attitudes of people towards a particular cryptocurrency. This information can be used to predict the future trends of the market and make informed trading decisions.

Decision trees are widely used in machine learning for their ability to classify data based on certain conditions. They are a popular algorithm that can be used for both classification and regression tasks, and they are particularly useful for analysing complex data sets. One of the main advantages of decision trees is their interpretability, which means that they can be easily understood and visualized by humans. This makes them a useful tool for data analysis and decision-making.

Sentiment analysis is another important application of machine learning that involves the use of algorithms to analyse and understand human language. It involves analysing text data, such as social media posts, customer reviews, and news articles, to identify the sentiment or emotional tone of the language. Sentiment analysis is a valuable tool for businesses and organizations as it allows them to understand the opinions and attitudes of their customers and stakeholders.

Xgboost is a popular machine learning algorithm that is widely used in the field of data science for its ability to provide highly accurate predictions. It is a type of decision tree that uses a gradient boosting algorithm to create a model that can accurately predict the outcome of a particular event or situation*. Xgboost can be used for a wide range of applications, including forecasting the next candle in cryptocurrency markets*.

In summary, decision trees are a powerful machine learning algorithm that can be used for a variety of applications, including sentiment analysis and predicting the next candle in cryptocurrency markets. Xgboost is a specific type of decision tree that is commonly used for its ability to provide highly accurate predictions.

The challenges you have mentioned for your app are indeed significant ones. Obtaining good and effective data from Twitter can be a challenge, especially when it comes to sentiment analysis. Many people tend to post tweets with positive sentiment, regardless of whether they understand the underlying price action of the cryptocurrency markets. This makes it difficult to accurately gauge sentiment and use it to inform trading decisions.

In addition to obtaining good data, the choice of features and indicators used for modelling or Decision Trees structure is also crucial. It's important to try out different combinations of features and indicators to determine which ones are most effective for predicting cryptocurrency price movements. This process can be time-consuming and require significant computational resources, but it is essential for building an accurate and effective model.

Overall, the challenges you have identified highlight the importance of careful data selection, pre-processing, and feature engineering when developing a machine learning model for cryptocurrency price prediction. With a robust and well-designed model, however, it is possible to achieve high levels of accuracy and profitability in this exciting and rapidly evolving field.

In the final step, we will show you the challenges we have encountered and evaluated. To ensure that this is not just a research project, we will create an artificial intelligence pipeline on Python with clean architecture and solid principles, as well as a backend. We will track and save our data and models with PostgreSQL and signals. Afterwards, we will create a web interface with ReactJS to allow everyone to benefit from this information. We will also discuss the difficulties we encountered in monitoring data using various deploy and sockets. We will show you other machine learning methods we have tried and the reasons why we chose the techniques we are currently using.

When it comes to Twitter, we will discuss the problems we encountered and solved when pulling data and fitting it into the model. We will also talk about the

importance of data pre-processing in natural language processing and how we tackled the challenge of sentiment analysis. It is essential to preprocess the data before feeding it into the model to eliminate noise, inaccuracies, and bias.

Furthermore, we will emphasize the significance of the solid principles and clean architecture in the development of the backend and frontend. These principles ensure maintainability, scalability, and testability of the codebase. We will also show the importance of the web interface in providing a user-friendly way to interact with the model's predictions and the benefits of using PostgreSQL in storing and managing data.

Overall, we are excited to showcase the results of our project and the challenges we have overcome to develop a robust and reliable system for predicting cryptocurrency prices.

KEYWORDS: Decision Trees, Sentiment Analysis, Data Mining, Data Preprocessing, API, Binance, Bitcoin, Ethereum, Doge, Ripple, CTXC, Deep Learning, Indicators, Xgboost, LigthGBM, LSTM, Machine Learning Workflow, NLP, Pandas, Python, Sentiment Analysis, Technical Analysis, Twitter, Tradingview, Clean Architecture, SOLID principles, Reactjs

FOREWORD

I would like to thank my teacher Alper OZCAN, who constantly pushes me to work even during the holidays and always shares information with the progress of the project.

Besides, I would like to express my gratitude to my mother and father, who were always there for me, and to my brother, lawyer Kursat, who always stayed in the same house and listened to my ideas with the project and gave me different perspectives, and to my sister Elif, whom I always felt by my side and with whom we had programming conversations.

TABLE OF CONTENTS

ABSTRACT....................i
FOREWORD....................v
TABLE OF CONTENTS....................vii
SYMBOLS AND ABBREVIATIONS....................vii
LIST OF FIGURES....................viiiiii
1. INTRODUCTION....................9
2. LITERATURE REVIEW....................11
2.1.Requirements....................11
2.2. Motivation....................11
2.3 Research Problem ….………………………………………………………………………...…………..12
3. PROJECT IMPLEMENTATION....................13
3.1. Data pipeline....................13
3.2. Twitter Sentiment Analysis....................14
3.3. Trading Strategies for Feature Extraction....................16
3.4. KZ Score/Index Indicator....................18
3.5. Featured Binary Matrix....................19
3.6. Xgboost for Classification Task....................20
3.7. Alternative Solution with LSTM and LightGBM....................22
4. MODEL RESULTS AND COMPARISONS....................25
4.1. New Approach only Cryptocurrencies....................25
5. PROJECT STRUCTURE....................27
5.1. Backend Structure....................27
5.2. Frontend Structure....................27
6. CONCLUSIONS....................30
7. REFERENCES....................31
CURRICULUM VITAE

SYMBOLS AND ABBREVIATIONS

AI	: Artificial Intelligence
API	: Application Programming Interface
ARIMA:	Autoregressive Moving Average
DL	: Deep Learning
LSTM	: Long-short Term Memory
MA	: Moving Averages
ML	: Machine Learning
NLP	: Natural Language Process
RNN	: Recurrent Neural Network
DCT	: Decision Trees
Boosting	: Gradient Boosting

LIST OF FIGURES

Figure 1.1 Main structure for logic of project 10

Figure 3.1 Indicator Data Matrix 14

Figure 3.2 Flow Diagram for Sentiment Analysis 15

Figure 3.3 Twitter Sentiment Analysis and BTC price 16

Figure 3.4 Sentiment score, KZ score 16

Figure 3.5 Cross-over strategy results 17

Figure 3.6 Cross-over strategy result ……………………………………………………...……..………18

Figure 3.7 KZ Score 19

Figure 3.8 Featured Binary Matrix 19

Figure 3.9 Feature Importance on Xgboost 20

Figure 3.10 Xgboost schema and formulas 21

Figure 3.11 Loss plot for Xgboost model train 22

Figure 3.12 Bitcoin 2022 prediction Daily return with Xgboost 22

Figure 3.13 SASA 2022 prediction Daily return with Xgboost 24

Figure 3.14 AKBANK 2022 prediction Daily return with Xgboost 25

Figure 4.1 Bitcoin 1 hour model backtest result 26

Figure 4.2 Bitcoin 1 hour model backtest result 26

Figure 4.3 Last Accuracy Scores 27

Figure 5.1 Clean Architecture for KZEngine 27

Figure 5.2 Main screen 28

Figure 5.3 Diagram Machine Learning Workflow 29

Figure 5.4 Signal Card 29

1. INTRODUCTION

This project involves obtaining OHLC and volume data, extracting technical indicators, and obtaining Japanese candlestick values. Twitter data is also fetched with a specific hashtag for sentiment analysis, which is a challenging part of the project. Data normalization is done before modeling, and innovative technical analysis indicators are used. The last step involves merging all data into one matrix for forecasting the next candle direction. The project also provides information on trading and strategies. There are four proposed apps for this project: a live tracking system for sentiment analysis, a trading strategy backtesting app, a filtering system for coins, and a next candle direction forecasting app using DL and Xgboost algorithms.

Our main strategy and path for this project. *Obtain OHLC(open, low, high, close price) and volume data. Then getting various technical indicators value from that. Obtain Japanese Candlestick format* with bearish or bull values. These are being features to matrix for our model.

The other side *fetch the Tweets with specific hashtag about related cryptocurrencies or asset/stock*. Then the creating data pipeline for obtain a good format and extract values for sentiment analysis. This is the one challenging part of this project because of including Big Data structure and need knowledge about it. For example I have fetched daily 150000 tweets about Bitcoin. But not only bitcoin I fetched most of the coins or assets tweets.

Before modelling we should some normalization about our data matrix. So we have to find strategies related to each indicators with related to financial concept. For example the Cross-over strategy for EMA’s or band range strategy for RSI. Then labelled it with 0 or 1 for obtaining some good binary matrix data. Also I have found some innovation technical analysis indicator I can show them at the later parts. That name is the KZ Score/Index indicators. It is related to all the indicators and their correlation values.

Last step we have merged all the data to one unique matrix and so we have obtained feature extracted data for forecasting the next candle. Technical analysis labels, Some good financial buy and sell strategies, Tweets sentiment score mean’s value for daily and hourly(that label name is the “compound_score” later we changed the column name is “twitter_sentiment_scores”). Add to our Japan candlestick values to this matrix. Also we should add our KZ score values with a new column.

At now, we have ready to construct our Neural Networks and Xgboost model for forecasting, train and evaluation. I will explain and show results for each steps later. But don’t forget this. This project not only related to forecasting next candle. That provides enormous information about the trading and strategies.

We also create a filtering for all coins. Testing a lot of cross-over strategies and get humans feeling about price action with sentiment analysis. Actually this project should use more than one app if you can use your brain.

1- Twitter Sentiment Analysis Score Live tracking system (Because I haven't noticed yet any similar app.)
2- Trading strategies backtesting and historical result value app
3- Filtering for coins for reducing number of target
4- Next candle direction forecasting with Xgboost algorithms and Sentiment analysis
5- Creating an Application with backend and frontend using Clean Architecture

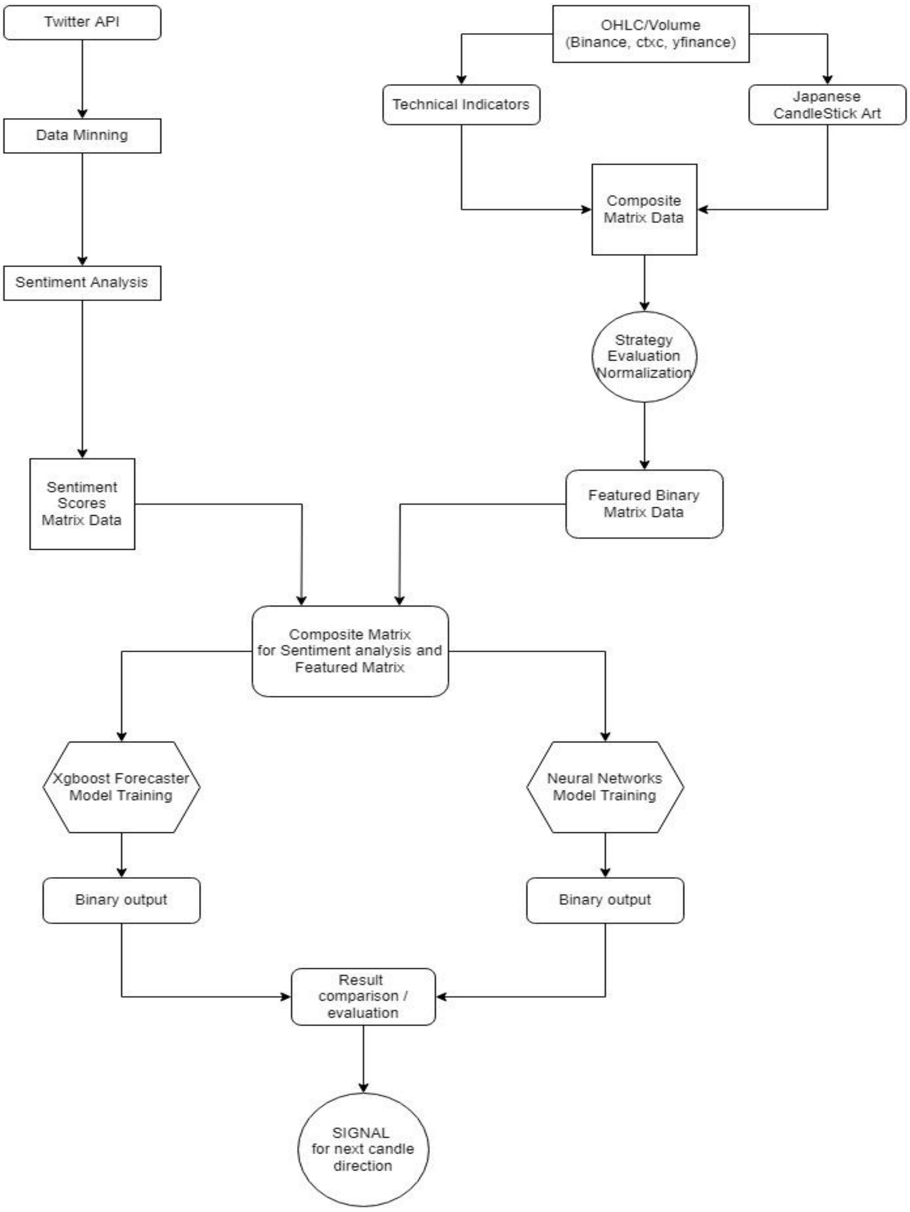

Twitter API
Data Minning
Sentiment Analysis
Sentiment Scores Matrix Data
OHLC/Volume (Binance, ctxc, yfinance)
Technical Indicators
Japanese CandleStick Art
Composite Matrix Data
Strategy Evaluation Normalization
Featured Binary Matrix Data
Composite Matrix for Sentiment analysis and Featured Matrix
Xgboost Forecaster Model Training
Neural Networks Model Training
Binary output
Binary output
Result comparison / evaluation
SIGNAL for next candle direction

Fig 1.1 Main structure for logic of project

2. LITERATURE REVIEW

2.1 Requirements

To extract and organize data, we will be using the Python programming language, along with the ta-lib and pandas_ta libraries. The ta-lib library can sometimes be difficult to install, so we will also include a module that contains the pandas_ta library as an extra. In addition, we will be accessing the Twitter API, for which we have obtained academic access. However, the API keys only provide us with data from the past week. To address this limitation and increase the reliability of our model, we conducted data mining on approximately six months of data for the five coins we initially specified.

To compare the scores and usage patterns of our LSTM and Neural Networks models, we will be using the Xgboost library and PyTorch. After this, we will use Flask to create a backend with a clean architecture. We are not planning to create a user module at this stage, as we want the application to be freely accessible to everyone. For the database, we will be using the open-source PostgreSQL. Finally, we plan to design a user interface using Reactjs to make the application accessible to users. All of this will be accomplished through approximately 8-9 months of work.

In summary, our project involves obtaining OHLC and volume data, extracting various technical indicators, obtaining Japanese candlestick data, and then combining all of this data into a unique matrix for forecasting the next candle. We also aim to extract data for sentiment analysis from Twitter using specific hashtags related to cryptocurrencies or assets/stocks. Our project will use Python for data extraction and pipeline creation, ta-lib and pandas_ta libraries for technical indicators and candlestick data, Twitter API for sentiment analysis, Xgboost and PyTorch for comparing model scores, Flask for creating a backend, PostgreSQL for the database, and Reactjs for designing the user interface. Our goal is to create a freely accessible application that provides trading information and strategies.

2.2 Motivation

The motivation behind this project is that when people decide to buy a coin, generally the price of that coin in the stock market starts to fall, and when all indicators say "sell," the price goes up, or when they say "buy," the price goes down. This is called the point of financial saturation. My goal is to create an artificial intelligence-based system that can prevent this from happening, by detecting these two points and presenting them to users through various methods. Additionally, I aim to turn this system into an application and meet with customers. By evaluating this process with artificial intelligence and applications, I want to create an application that will help people make long-term profits and reduce trading stress by detecting price trends.

Financial saturation is a well-known phenomenon in the world of trading and investing, and it can be difficult for individuals to navigate. With the advancements in technology and the increasing popularity of artificial intelligence, there is an opportunity to create a system that can help individuals make better decisions when it comes to trading. By detecting the points of financial saturation, individuals can avoid buying or selling at the wrong time and potentially lose money. This project aims to develop such a system using various methods and algorithms, including the use of machine learning and neural networks. Additionally, by creating an application that can be used by customers, this system can be accessible to a wider audience, ultimately helping more people make profitable trades and achieve their financial goals.

2.2 Research Problems

To complete this project, we have faced many problems for a long time. In the next sections, I will explain how we solved these problems with examples. However, in this section, I will talk about the problems that needed to be solved and what they were.

One of the problems we faced was obtaining the Twitter API key, which took about two weeks to receive through the academic access application. We also needed to figure out how to collect tweets, what format to use to store them, which library to use for sentiment analysis, how to establish an object-oriented structure, and how to deal with discrepancies that arose during the combination of indicator data structures. Additionally, we had to determine how to organize hourly and daily sentiment scores for incoming tweets and why the resulting structure's average score deviated from -1. We also considered whether normalization was necessary for this data.

Furthermore, we needed to figure out how to calculate the characteristics and names of Japan candlesticks for each day according to Japan candlestick art and how to add this data to our main model matrix. We also had to determine how many days of indicator and data, which indicators would be included in the matrix, and how the effects of the day range interval would be observed during calculations. Additionally, we had to determine how to represent all these indicators' strategies or new strategies in binary format for the artificial intelligence model.

We also needed to figure out how to combine Twitter sentiment analysis and other technical analysis data with Japan candlestick art data into a single matrix. We considered how xgboost's results would change with this combination, where to store the results, which features would have the most significant impact on the model, whether to perform feature extraction, and how to handle real-time Twitter data during the live tracking phase. We considered whether to use a previously trained model or retrain the model every hour, and whether the RAM and CPU would be sufficient. We also analyzed the advantages and disadvantages of using a neural network for the application.

Finally, we needed to determine the backend structure's domain and adapters, how to establish relationships between the backend and database within an ORM structure, and what challenges may arise when implementing the Binance API for live tracking via a WebSocket. We also needed to ensure that the frontend and backend of the application could function continuously.

3. PROJECT IMPLEMENTATION

3.1 Data Pipeline

Obtain OHLC(open, low, high, close price) and volume data. Then getting various technical indicators value from that. Obtain Japanese Candlestick format with bearish or bull values. This is the ne matrix for our model.

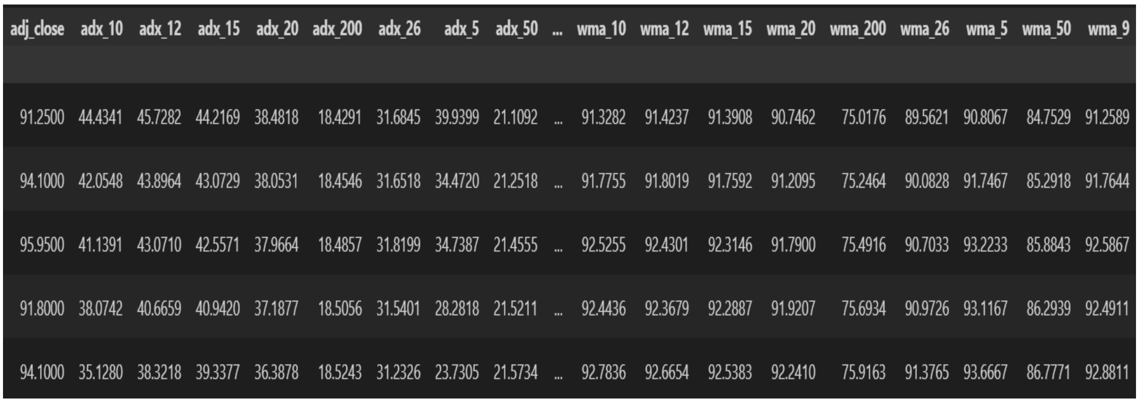
For obtain this OHLC data format we use more than python library. First one is the yfinance library most of the people use them but sometimes volume and some values come with NaN or ‘0’ value. That is a problem for our dataset and we drop this values at this pipeline process. The other library is ctcx that is also good but sometimes slower than the others because it uses mostly instantly scraping from web and this process is get long time. The other important library is related to cryptocurrencies that is Binance library help for the Binance API. Actually this is the fast and good way also it supports the stream functionality and we can easily make another trading operation but for using this we can obtain api keys from Binance. Our project support for this 3 library you can choice either of them.

Then we should calculate technical analysis indicator values. What is this indicators name: *'ad', 'adj_close', 'adx_10', 'adx_12', 'adx_15', 'adx_20', 'adx_200', 'adx_26', 'adx_5', 'adx_50', 'adx_9', 'atrr_10', 'atrr_12', 'atrr_15', 'atrr_20', 'atrr_200', 'atrr_26', 'atrr_5', 'atrr_50', 'atrr_9', 'candle_label', 'candlestick_pattern', 'cci_10', 'cci_12', 'cci_15', 'cci_20', 'cci_200', 'cci_26', 'cci_5', 'cci_50', 'cci_9', 'close', 'cmo_10', 'cmo_12', 'cmo_15', 'cmo_20', 'cmo_200', 'cmo_26', 'cmo_5', 'cmo_50', 'cmo_9', 'daily_return', 'dema_10', 'dema_12', 'dema_15', 'dema_20', 'dema_200', 'dema_26', 'dema_5', 'dema_50', 'dema_9', 'dmn_10', 'dmn_12', 'dmn_15', 'dmn_20', 'dmn_200', 'dmn_26', 'dmn_5', 'dmn_50', 'dmn_9', 'dmp_10', 'dmp_12', 'dmp_15', 'dmp_20', 'dmp_200', 'dmp_26', 'dmp_5', 'dmp_50', 'dmp_9', 'ema_10', 'ema_12', 'ema_15', 'ema_20', 'ema_200', 'ema_26', 'ema_5', 'ema_50', 'ema_9', 'fishert', 'fisherts', 'high', 'ich_kline', 'ich_tline', 'kama_10', 'kama_12', 'kama_15', 'kama_20', 'kama_200', 'kama_26', 'kama_5', 'kama_50', 'kama_9', 'log_return', 'low', 'lowband_10', 'lowband_12', 'lowband_15', 'lowband_20', 'lowband_200', 'lowband_26', 'lowband_5', 'lowband_50', 'lowband_9', 'macd', 'macdhist', 'macdsignal', 'mfi_10', 'mfi_12', 'mfi_15', 'mfi_20', 'mfi_200', 'mfi_26', 'mfi_5', 'mfi_50', 'mfi_9', 'midband_10', 'midband_12', 'midband_15', 'midband_20', 'midband_200', 'midband_26', 'midband_5', 'midband_50', 'midband_9', 'natr_10', 'natr_12', 'natr_15', 'natr_20', 'natr_200', 'natr_26', 'natr_5', 'natr_50', 'natr_9', 'obv', 'open', 'roc_10', 'roc_12', 'roc_15', 'roc_20', 'roc_200', 'roc_26', 'roc_5', 'roc_50', 'roc_9', 'rsi_10', 'rsi_12', 'rsi_15', 'rsi_20', 'rsi_200', 'rsi_26', 'rsi_5', 'rsi_50', 'rsi_9', 'sma_10', 'sma_12', 'sma_15', 'sma_20', 'sma_200', 'sma_26', 'sma_5', 'sma_50', 'sma_9', 'stoch_d', 'stoch_k', 't3_10', 't3_12', 't3_15', 't3_20', 't3_200', 't3_26', 't3_5', 't3_50', 't3_9', 'tema_10', 'tema_12', 'tema_15', 'tema_20', 'tema_200', 'tema_26', 'tema_5', 'tema_50', 'tema_9', 'trima_10', 'trima_12', 'trima_15', 'trima_20', 'trima_200', 'trima_26', 'trima_5', 'trima_50', 'trima_9', 'upband_10', 'upband_12', 'upband_15', 'upband_20', 'upband_200', 'upband_26', 'upband_5', 'upband_50', 'upband_9', 'volume', 'willr_10', 'willr_12', 'willr_15', 'willr_20', 'willr_200', 'willr_26', 'willr_5', 'willr_50', 'willr_9', 'wma_10', 'wma_12', 'wma_15', 'wma_20', 'wma_200', 'wma_26', 'wma_5',* *'wma_50', 'wma_9'*

Additionally we want to use candlestick values and calculation. For all this we choice the use TA-LIB library. But TA-LIB installation sometime faced more difficult problem. For the other users or developer also I add pandas_ta library one exception between this libraries are the pandas_ta don’t have a candlestick calculation. Yes it has but very restricted. I prefer and suggest TA-LIB installation.

Japan Candlestick is a type of price chart that shows **the opening, closing, high and low price points for each given period**. It was invented by Japanese rice merchants centuries ago, and popularised among Western traders by a broker called Steve Nison in the 1990s. Today, Japanese candlestick charts are the most popular way to quickly analyse price action, particularly with technical traders. They offer much more information visually than traditional line charts, showing a market's highest point, lowest point, opening price and closing price at a glance.

High Low Open Close (HLOC) charts display a similar level of detail to candlesticks – but traders tend to favour the latter, finding them easier to analyse quickly than HLOC. It's worth trying out both and seeing which works best for you.



adj_close	adx_10	adx_12	adx_15	adx_20	adx_200	adx_26	adx_5	adx_50	...	wma_10	wma_12	wma_15	wma_20	wma_200	wma_26	wma_5	wma_50	wma_9
91.2500	44.4341	45.7282	44.2169	38.4818	18.4291	31.6845	39.9399	21.1092	...	91.3282	91.4237	91.3908	90.7462	75.0176	89.5621	90.8067	84.7529	91.2589
94.1000	42.0548	43.8964	43.0729	38.0531	18.4546	31.6518	34.4720	21.2518	...	91.7755	91.8019	91.7592	91.2095	75.2464	90.0828	91.7467	85.2918	91.7644
95.9500	41.1391	43.0710	42.5571	37.9664	18.4857	31.8199	34.7387	21.4555	...	92.5255	92.4301	92.3146	91.7900	75.4916	90.7033	93.2233	85.8843	92.5867
91.8000	38.0742	40.6659	40.9420	37.1877	18.5056	31.5401	28.2818	21.5211	...	92.4436	92.3679	92.2887	91.9207	75.6934	90.9726	93.1167	86.2939	92.4911
94.1000	35.1280	38.3218	39.3377	36.3878	18.5243	31.2326	23.7305	21.5734	...	92.7836	92.6654	92.5383	92.2410	75.9163	91.3765	93.6667	86.7771	92.8811

Fig 3.1 Indicator Data Matrix

At this time we don’t discuss the strategies return for profit or loss our purposes is the obtain a meaningful matrix dataset with time-series format for forecasting the next candle.

3.2 Twitter Sentiment Analysis

This part is very challenges for me. Firstly we start use the python’s Tweepy library for fetch the tweets with specific hashtag. But before this we should obtain twitter API keys. So you must apply the twitter developer platform but this is actually need wage monthly 200 dollar. If you are a researcher for academic purposes for project you can get a financial aid for this API keys. I try to 3 time and have gotten a elementary level api keys. This is an restricted api for getting tweets monthly you can fetch 4 million tweets in the first step this was sufficient for this app.

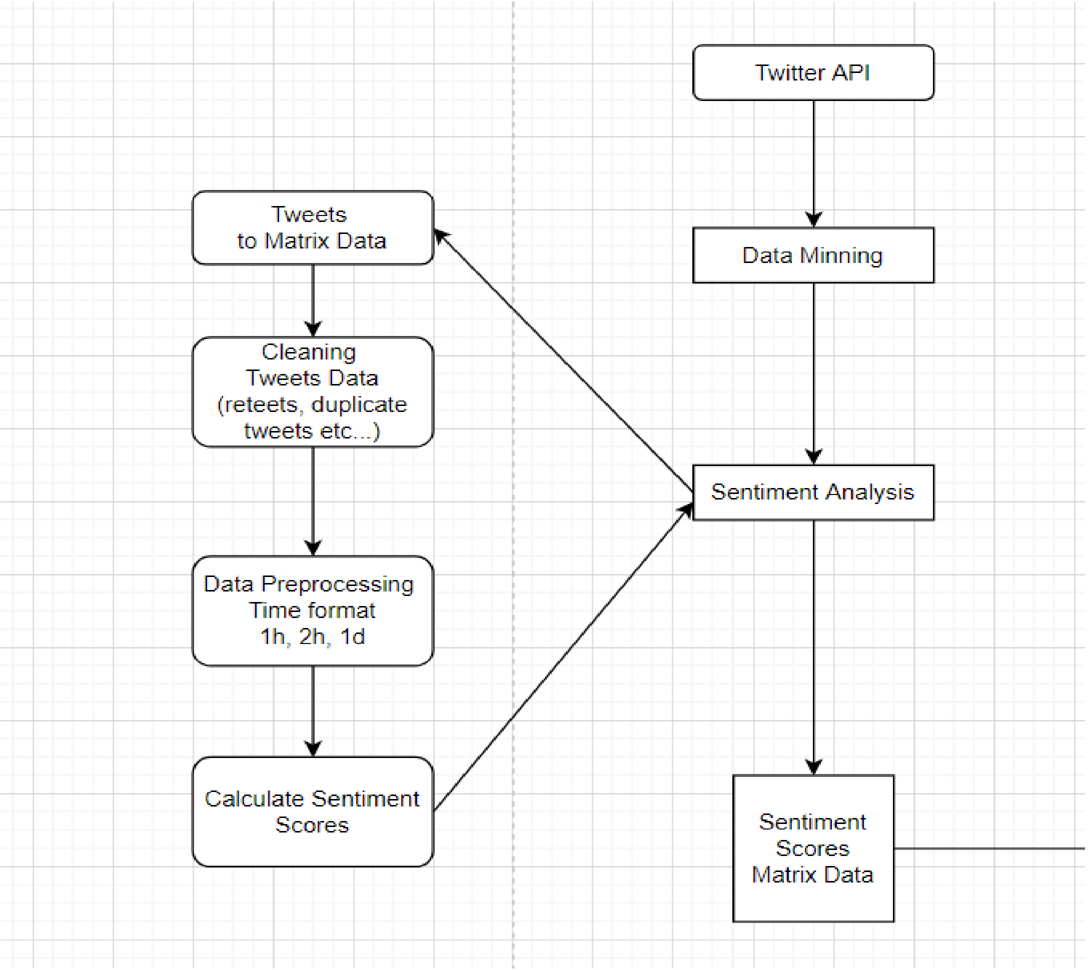

Twitter API
Tweets to Matrix Data
Data Minning
Cleaning Tweets Data (reteets, duplicate tweets etc...)
Sentiment Analysis
Data Preprocessing Time format 1h, 2h, 1d
Calculate Sentiment Scores
Sentiment Scores Matrix Data

Fig 3.2 Flow Diagram for Sentiment Analysis

You should get all the tweets and keep securely then make a pipeline for obtain sentiment polarity scores. For this purposes we use NLTK and VADER sentiment analysis library and also polarity scores. It gives a sentiment with [-1,1] range. Negative, neutral and positive scores. This job some sensitive pipeline process because for one coins we have daily 100 thousands tweets and nearly in 2 months we obtain 4 GB tweets. I want to save and edited this process from this point begin to only obtain twitter sentiment scores and save them to .csv file for using later into a model. I created TwitterSentimentAnalyser() and TwitterCollection() class for this. And daily I fetched tweets for coins. But this is not enough for our model so I searched the bitcoin tweets archive for modelling the historical behavior.

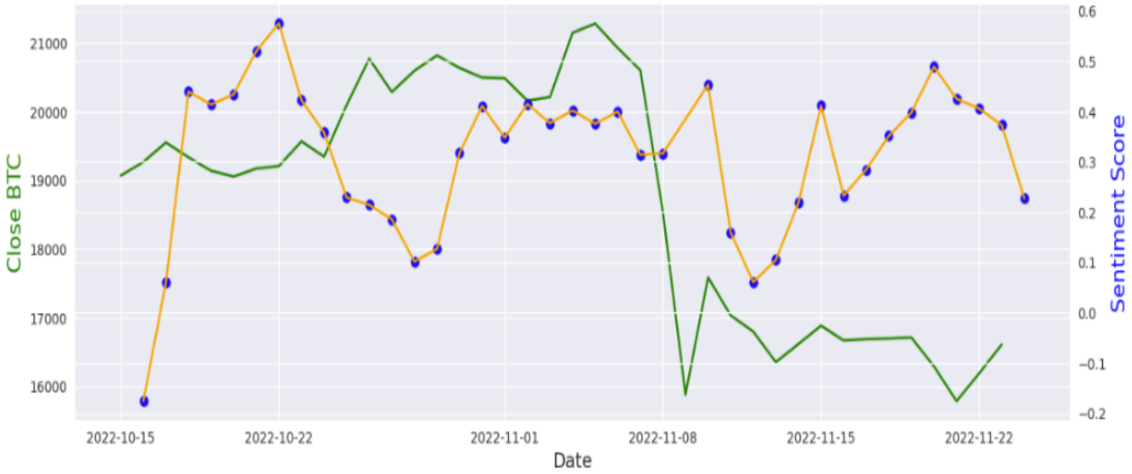


Fig 3.3 Twiter sentiment analysis and BTC price

Perhaps firstly I should use this archive tweets but after 6-8 months I will use my mining tweets for actual price action sentiments for human behavior. As you can see mostly our sentiment scores between 0.1 and 0.4 range. The reason why human usually feel positive and want to feel good thing, obtain profits with their trades. So you can see how this people affect the social media for price action. The other interesting is the tweets numbers. For example I fetched 2 million tweets with related to #bitcoin but the pipeline process and some cleaning also extracted the duplicates and retweets this remaining to 38 thousands tweets. I think the other affect is the twitter bots. Because most of the people or company uses this bots. Maybe in this time Elon Musk cleans this bots. We don't know yet.

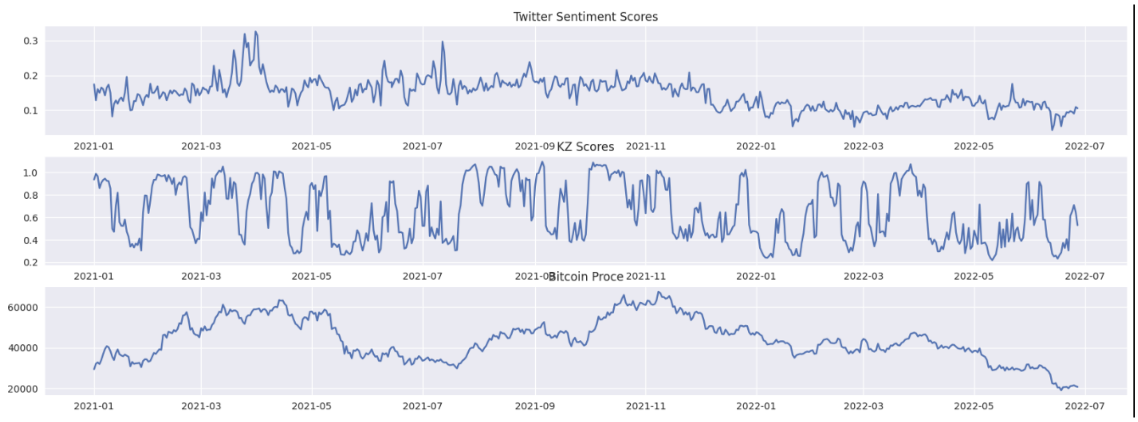
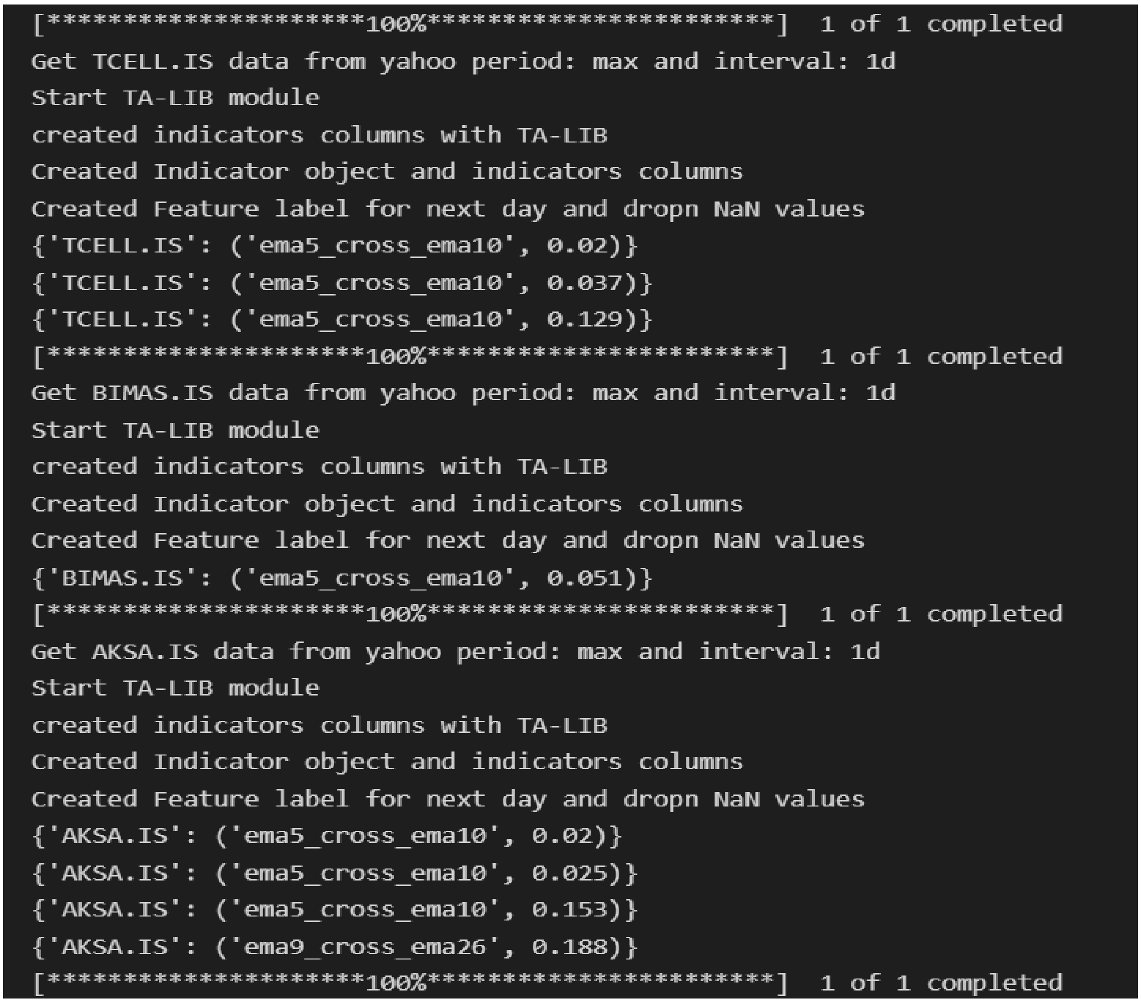


Fig 3.4 Sentiment score, KZ score

3.3 Trading Strategies for Feature Extraction

Forecasting for the next candle is seems to be simple process but if you are look into in deep dive you can see this is not accurate or simple task. Because all the indicators value and some information about it mention and correlation with prices history. If you want to a supervised learning task and want to obtain a today or before 3 days ago you can success with very good result for your model. But if you want to the next day, you will fail. I heard about it most of the times but I want to see it and I studied about it.

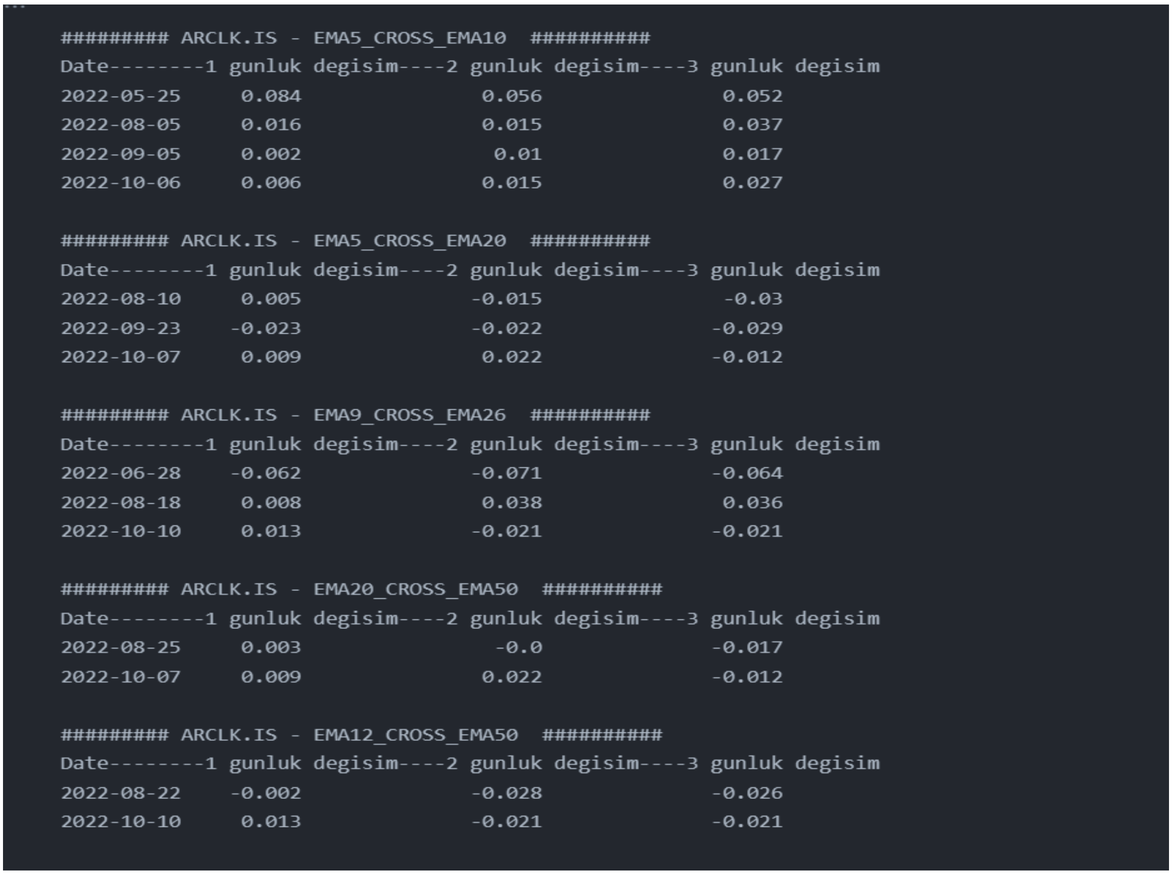


```
[*********************100%***********************]  1 of 1 completed
Get TCELL.IS data from yahoo period: max and interval: 1d
Start TA-LIB module
created indicators columns with TA-LIB
Created Indicator object and indicators columns
Created Feature label for next day and dropn NaN values
{'TCELL.IS': ('ema5_cross_ema10', 0.02)}
{'TCELL.IS': ('ema5_cross_ema10', 0.037)}
{'TCELL.IS': ('ema5_cross_ema10', 0.129)}
[*********************100%***********************]  1 of 1 completed
Get BIMAS.IS data from yahoo period: max and interval: 1d
Start TA-LIB module
created indicators columns with TA-LIB
Created Indicator object and indicators columns
Created Feature label for next day and dropn NaN values
{'BIMAS.IS': ('ema5_cross_ema10', 0.051)}
[*********************100%***********************]  1 of 1 completed
Get AKSA.IS data from yahoo period: max and interval: 1d
Start TA-LIB module
created indicators columns with TA-LIB
Created Indicator object and indicators columns
Created Feature label for next day and dropn NaN values
{'AKSA.IS': ('ema5_cross_ema10', 0.02)}
{'AKSA.IS': ('ema5_cross_ema10', 0.025)}
{'AKSA.IS': ('ema5_cross_ema10', 0.153)}
{'AKSA.IS': ('ema9_cross_ema26', 0.188)}
[*********************100%***********************]  1 of 1 completed
```

Fig 3.5 Cross-over strategy results

We have all the indicators values and we want to create a binary feature matrix with the trading strategies. And this features values mostly 1 or 0. How? We learn trading strategies. Some strategies are cross-over EMA’s, SMA’s, band range strategies, MACD, RSI, ADX all the indicators actually have a strategies.

Another app idea in my mind is to launch an app with this trade distance and let people see which strategy has been more effective on which coins in the past and in which trend times. As an example, the results of a study I have done in ema cross-over strategies are shown in the table below. I tried to show and find out how much profit was made in 3 days with which strategy even at a short distance.



```
######### ARCLK.IS - EMA5_CROSS_EMA10  ##########
Date--------1 gunluk degisim----2 gunluk degisim----3 gunluk degisim
2022-05-25     0.084              0.056              0.052
2022-08-05     0.016              0.015              0.037
2022-09-05     0.002               0.01              0.017
2022-10-06     0.006              0.015              0.027

######### ARCLK.IS - EMA5_CROSS_EMA20  ##########
Date--------1 gunluk degisim----2 gunluk degisim----3 gunluk degisim
2022-08-10     0.005             -0.015              -0.03
2022-09-23    -0.023             -0.022             -0.029
2022-10-07     0.009              0.022             -0.012

######### ARCLK.IS - EMA9_CROSS_EMA26  ##########
Date--------1 gunluk degisim----2 gunluk degisim----3 gunluk degisim
2022-06-28    -0.062             -0.071             -0.064
2022-08-18     0.008              0.038              0.036
2022-10-10     0.013             -0.021             -0.021

######### ARCLK.IS - EMA20_CROSS_EMA50  ##########
Date--------1 gunluk degisim----2 gunluk degisim----3 gunluk degisim
2022-08-25     0.003               -0.0             -0.017
2022-10-07     0.009              0.022             -0.012

######### ARCLK.IS - EMA12_CROSS_EMA50  ##########
Date--------1 gunluk degisim----2 gunluk degisim----3 gunluk degisim
2022-08-22    -0.002             -0.028             -0.026
2022-10-10     0.013             -0.021             -0.021
```

Fig 3.6 Cross-over strategy result

But our main purpose is not to explain the strategies one by one at the moment. here is to use all these strategies to obtain a meaningful and normalized matrix data and use it in our model. For example, in the RSI strategy, we go to a classification process by saying buy at 30, sell at 70, and say the range of 30-70 and the other ranges as 0. like this, we go to another classification process by saying 1 when EMA 10 cuts above EMA 40, and 0 when it falls below. We can make the matrix we have obtained more valuable by classifying only certain and known classical strategies, as well as complicated abs that have been tried before and used by traders. These strategies sometimes include 6-7 or 8 patterns. Some algorithmic traders try to find a pattern series between all strategies and price action, trying to reach the result almost like solving a puzzle.

3.4 KZ Score/Index Indicator

Always ask myself why and when all the people say buy coin but then price decrease, Sell the coin then price up. So I innovate a new Indicator with combination 23 indicators and also some strategies most popular. then all this indicators converted to binary matrix. at the last step I sum up all rows for obtain some knowledge about them. I named it the KZ_INDEX/SCORE. This indicator shows the all the indicators says strong buy signal and the bottom point say that all the indicators strong sell. and you can see actually this 2 things works opposite direction. and you can define buy/sell points. but you cannot implement easily this indicator because it has very complex calculation.

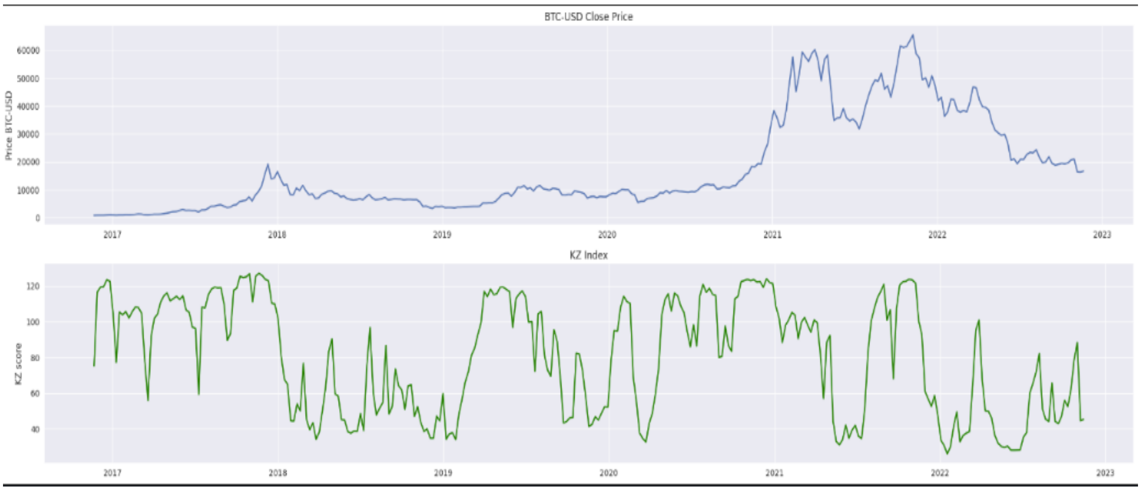
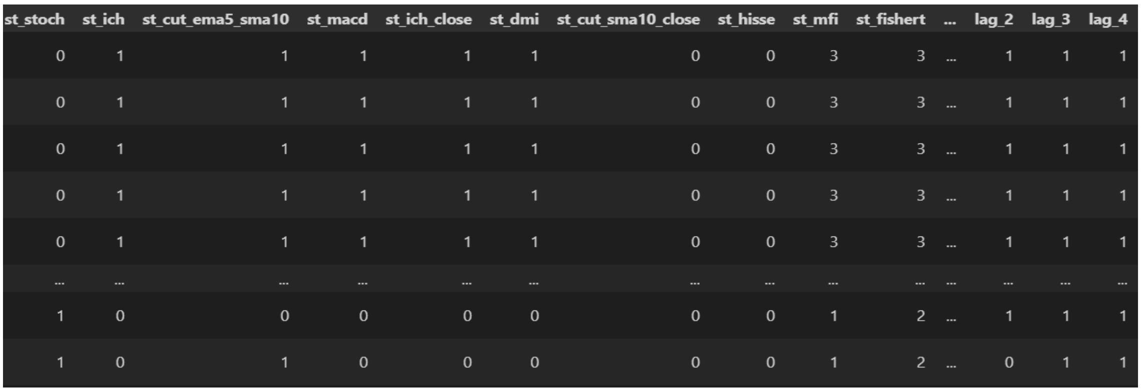


Fig 3.7 KZ Score

This indicator shows the saturation point of all indicators and the oversold points show people the regions that all strategies call sell. It can give a very good feeling and vision for the model as well as it can be used for buying points and selling points. In fact, I am one of those people who defend the thesis that indicators alone are not enough, and thus I am experiencing the happiness of bringing an idea to life that was in my mind before.

3.5 Featured Binary Matrix

The purpose of this stage is to make our data to be created for the model with the data obtained so far suitable for use and to combine all the data. But there is still a problem that we need to hang. Because we basically had two ideas to solve the problem, the first was twitter data and the belief that people's thoughts affect the price, and the other point was to create buying and selling strategies in the opposite direction at the points where all the indicators say buy and sell. but we have a matrix with around 200 labels and twitter sentiment scores are only available on one label.



st_stoch	st_ich	st_cut_ema5_sma10	st_macd	st_ich_close	st_dmi	st_cut_sma10_close	st_hisse	st_mfi	st_fishert	...	lag_2	lag_3	lag_4
0	1	1	1	1	1	0	0	3	3	...	1	1	1
0	1	1	1	1	1	0	0	3	3	...	1	1	1
0	1	1	1	1	1	0	0	3	3	...	1	1	1
0	1	1	1	1	1	0	0	3	3	...	1	1	1
0	1	1	1	1	1	0	0	3	3	...	1	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1	0	0	0	0	0	0	0	1	2	...	1	1	1
1	0	1	0	0	0	0	0	1	2	...	0	1	1

Fig 3.8 Featured Binary Matrix

Although I thought of increasing these labels with a poly features process at first, I could get this number data, which was stuck between 0 and 1, to the maximum of the 3rd power. More, the data becomes meaningless. but instead of normalizing this data with 0 v1 we will insert it into the model as it is and I will show how this increases the effect on the model. In fact, it is obtained by shifting the meaning level over the sentiment scores, since all the logical master data consists of 1s and 0s. We will do the same on the KZ index.

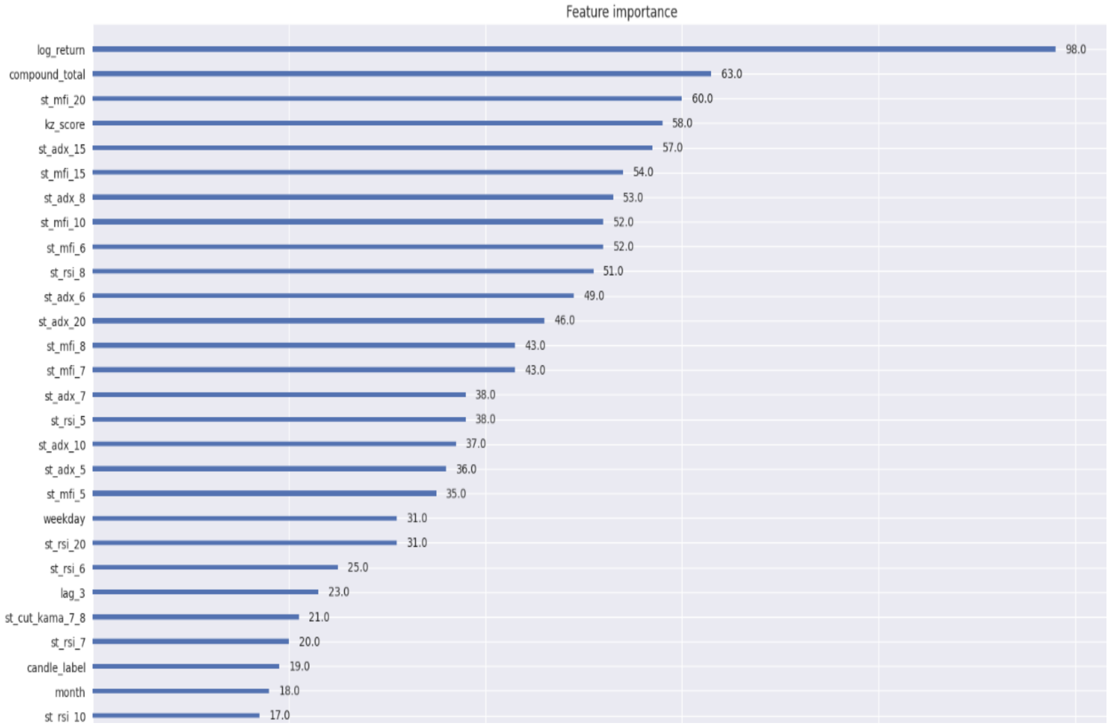


Fig 3.9 Feature Importance on Xgboost

3.6 Xgboost for Classification Task

"XGBoost is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models.

When using gradient boosting for regression, the weak learners are regression trees, and each regression tree maps an input data point to one of its leafs that contains a continuous score. XGBoost minimizes a regularized (L1 and L2) objective function that combines a convex loss function (based on the difference between the predicted and target outputs) and a penalty term for model complexity (in other words, the regression tree functions). The training proceeds iteratively, adding new trees that predict the residuals or errors of prior trees that are then combined with previous trees to make the final prediction. It's called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models."

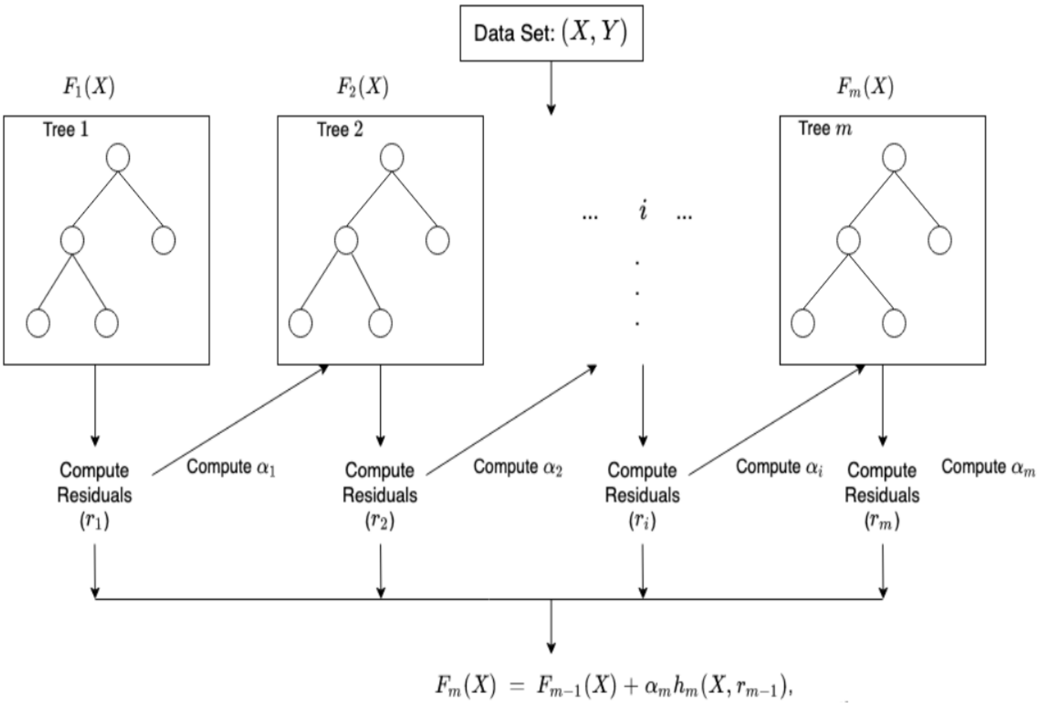

Data Set: (X, Y)
$F_1(X)$
Tree 1
$F_2(X)$
Tree 2
... i ...
$F_m(X)$
Tree m
Compute Residuals (r_1)
Compute α_1
Compute Residuals (r_2)
Compute α_2
Compute Residuals (r_i)
Compute α_i
Compute Residuals (r_m)
Compute α_m
$F_m(X) = F_{m-1}(X) + \alpha_m h_m(X, r_{m-1}),$

Fig 3.10 Xgboost schema and formulas

We uses gridSearchCV and cross-validation from Sci-kit learn library for obtain best parameters our Xgboost model. Approximation is n_estimation assign to 150, max_depth to 1, eta is to 0.3 for gradient boosting. This parameters seem to be very low if you are familiar with Xgboost hyperparameter operation. But this have some reason because of historical entropy. If you are trying to predict 2 days ago, this accuracy score nearly 75-80 percentage you have good score but trying to next candle or day your score nearly 55 percentage this seems to be very low but I can say this field and financial forecasting with advanced or Neural Networks method is very complicated and tends to be failure operation. So you can accept your assumption and you must do backtest, then decide which strategy using for your trade.

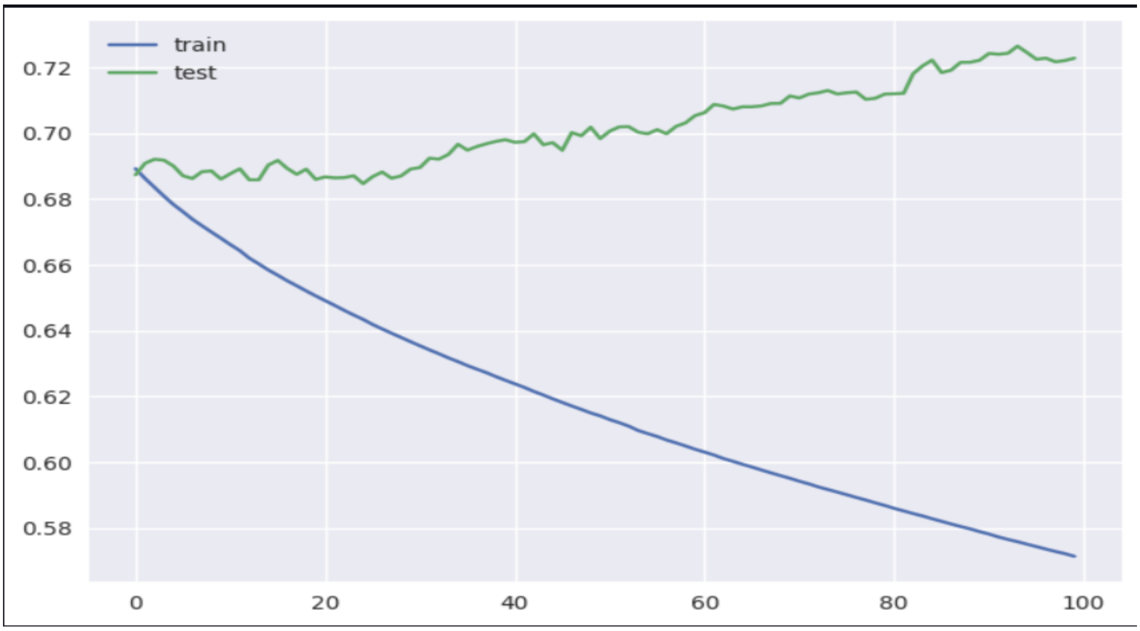


Fig 3.11 Loss plot for Xgboost model train

3.7 Alternative Solution with Xgboost, LSTM and LightGBM

I want to improve myself in price prediction and Data Science, which is the most well-known and curious area in the field of data science, where there are problems in getting results, because I wonder why I can't get clear results even though it seems that a giant industry in which so many people are based entirely on numerical data. Last year, I examined LSTM and RNN structures in the same project for the literature survey part. Here, in order to solve the vanishing gradient and exploding gradient problems, which are the errors that occur in creating prediction models of sequential data in RNN studies, I learned how to minimize the effect of data overload and data loss on modeling with input and forget gates by placing two extra gates in LSTM. In the early summer, I experimented with it on alkaline price data.

When we started working with Alper on the project, he suggested working with Xgboost, which has become a trend in the last period, and LightGBM, which is supported by microsoft. Before I started working on these and models, I started to develop the data preparation process with python. Although many APIs and institutions offer finance data free of charge, I started to see the effect of this missing and little data given in the modeling studies that followed, on the margin of error results.

In the beginning, during my studies and researches on bitcoin, I was getting less good results on other papers such as BIST, the Istanbul stock exchange, in technical review and indicator results. I learned that one reason for this is that Bitcoin data and price movements are not stable. I learned from the volatile market that ARIMA models, which have given good results in the short and medium-term, will not give good results on Bitcoin and crypto. As a way to prevent this, I found that not only technical and price data, but also finding other impressive new features and a feature extraction operation were important. Because, in most of my studies, I had a hard time extracting more than 57 percent of R-squared data from modeling on Bitcoin. However, after the same methods and hyperparameter tunings, for example, I was able to get results up to 69 percent in the SASA index on BIST.

At first, I pulled the data from yahoo and created a featured dataframe with all the necessary indicators. In addition, I added the candle structures to the data with the python Ta-lib wrapper library. Here, I labeled each candle structure as bull and bear and according to this, we decided to make a binary classification as 1 and 0 in the prediction label. In addition to this, I created a new label by categorizing through an indicator strategy studied on the Istanbul stock market. However, I have brought the strategies of important indicators such as mean reversion, MA crossing, Ichmoku cloud, Stochastic rsi into a new modeling data table in which I have just created the buy/sell signaling strategies and normalized the price data together with the other data. For all these normalization and strategy determination processes, I had to do a long bior research on indicators and their mathematical ranges of use during trading and normally. The libraries I used primarily for this were pandas_ta and Ta-lib.

Later on, I will see the backtested servers to create features related to these strategies. For this, I examined and experimented with various python kituphans such as pandas_ta, vectortz, and backtesting. During these trials, I want to write them with a whole class by working on my own methods and developing the Bacttests the way I want.

I will get the results with different backtest methods for each strategy and see which indicator is more successful on which assets.

Then I plan to create a modeling by converting them to 0 and 1 columns. Although I did research on LSTM last year primarily, I will do model trials with Xgboost and LigthGBM based on my instructor's recommendations. I will use GridSearchCV deep learning modeling to ensure their hyperparameter tuning and best optimization. The model that I will use as a comparison at first will be random-walk 1 day. Because I learned that the best approach and margin of error in finance theory is 1 day difference on price data. I will try to predict another day by minimizing the margin of error with the error functions I will do.

When these studies are finished, in order to improve the results, I will use the data I have taken from twitter and telegram channels to use Sentiment analysis, and the daily twitter data taken with the VADER sentiment analysis library in the NLP application, and the patients will be scored as positive and negative, and I will include them in my data production as a separate feature and column in the modeling.

For this, I applied to get the Twitter API and I was able to get it after a lot of effort. In order to facilitate data extraction and data mining, I will try to facilitate this with a twitter_collection class and bring it into an object oriented structure. The same operations will be valid for data coming from telegram channels. Twitter provides us with a period of about 1 week, as these APIs allow data to be withdrawn in certain and short periods. It is necessary to draw tweets of the data to be studied continuously and at regular intervals. I'll create a build that will automate this or just do it when modeling times come.

I will investigate how deep learning methods can be registered and deployed in the project. I can use the Amazon EC2 system from the research I did last year. but I would like to talk about some of the tests I made before these, and the results of Xgboost and GridSearchCV, as well as SASA and Binance on the Bist index, and Bitcoi on it.

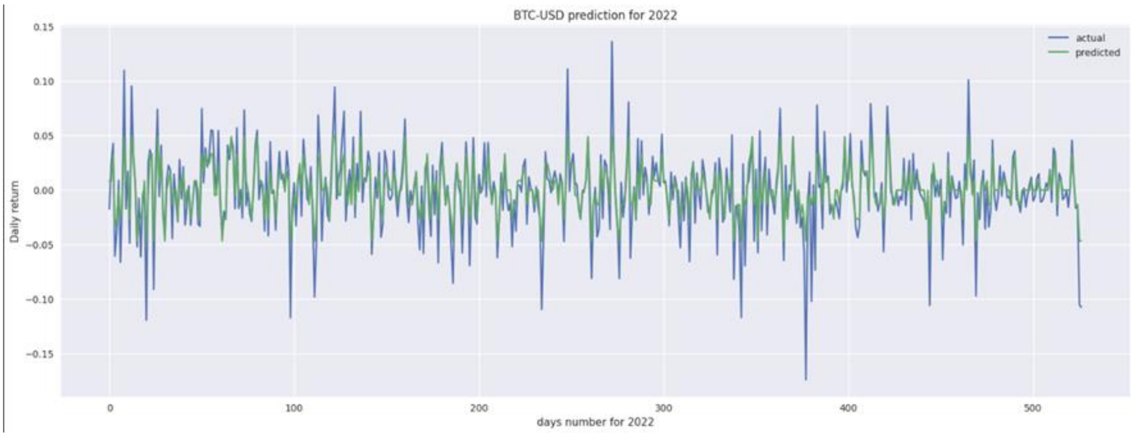
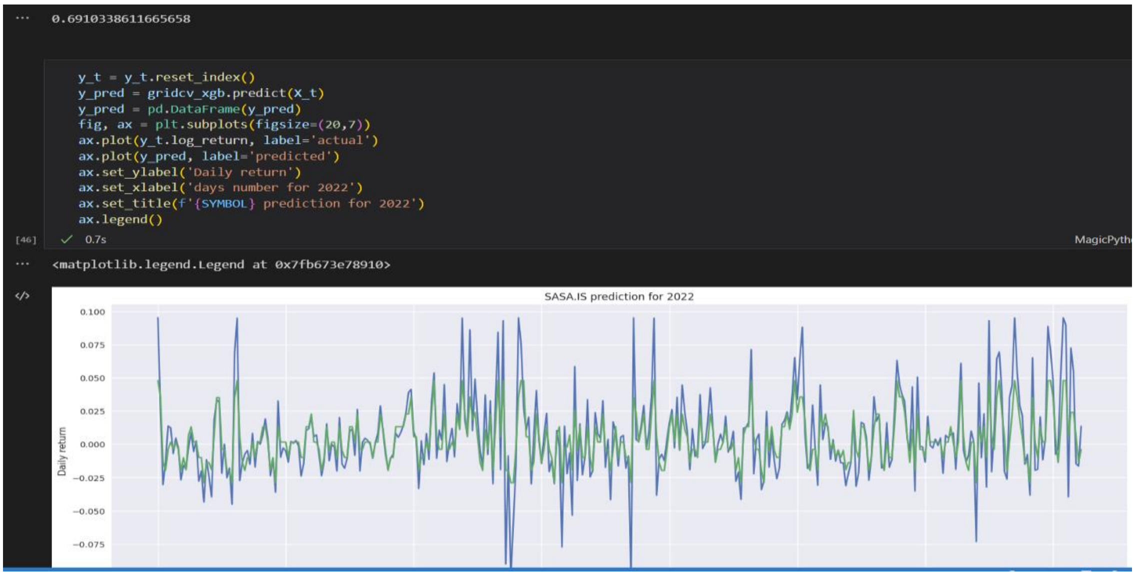


Figure 3.12 Bitcoin 2022 prediction Daily return with Xgboost

When we try to predict the next candle on an hourly or daily basis, we see that the compatibility ratios of R2, that is, the predictions and the results in the data, do not exceed 57 percent. One reason for this is the extremely volatile movements of the Bitcoin price, and another reason is that the distribution of the binary feature that we use as the result label is about fifty percent. Because in these cases, the prediction results are not good and you need to set a threshold. However, this 50 percent distribution prevents this.



```
0.6910338611665658
```

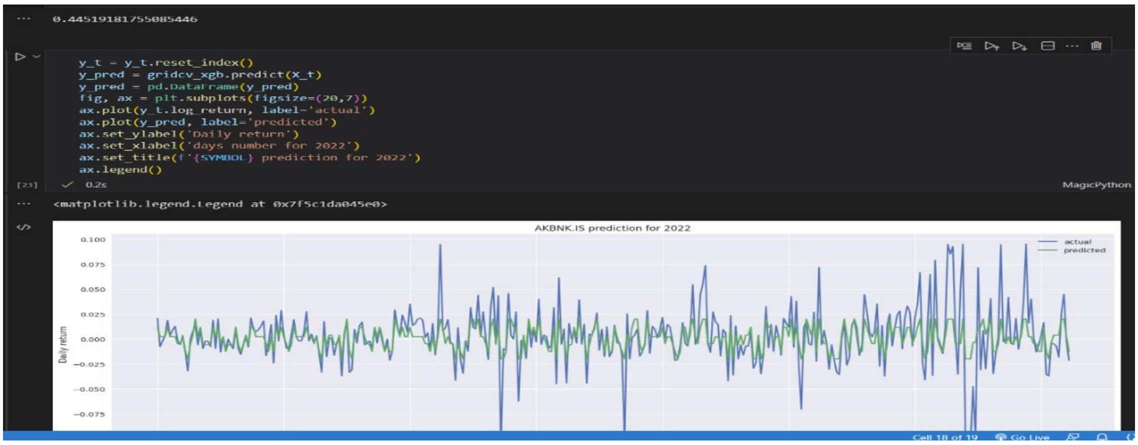
```
y_t = y_t.reset_index()
y_pred = gridcv_xgb.predict(X_t)
y_pred = pd.DataFrame(y_pred)
fig, ax = plt.subplots(figsize=(20,7))
ax.plot(y_t.log_return, label='actual')
ax.plot(y_pred, label='predicted')
ax.set_ylabel('Daily return')
ax.set_xlabel('days number for 2022')
ax.set_title(f'{SYMBOL} prediction for 2022')
ax.legend()
```

```
<matplotlib.legend.Legend at 0x7fb673e78910>
```

Figure 3.13 SASA 2022 prediction Daily return with Xgboost

When the same study is done on SASA, the overall rate rises to around 69 percent, which corresponds to a very good rate. I will try to present the studies and results related to this in a more detailed and tabular form in the next report, but before that, I need to adapt these processes to the project in an easy-to-operate manner.

In our recent studies, we decided to do it on the Bank papers on the Bist. One of the reasons for this is that the stagnation of these bank papers, instead of Binance and Bist papers, which are extremely volatile, is very suitable and favorable for this research in the last period. Another example work I have done with Xgboost is on AKBANK data.



```
0.44519181755085446
```

```
y_t = y_t.reset_index()
y_pred = gridcv_xgb.predict(X_t)
y_pred = pd.DataFrame(y_pred)
fig, ax = plt.subplots(figsize=(20,7))
ax.plot(y_t.log_return, label='actual')
ax.plot(y_pred, label='predicted')
ax.set_ylabel('Daily return')
ax.set_xlabel('days number for 2022')
ax.set_title(f'{SYMBOL} prediction for 2022')
ax.legend()
```

```
<matplotlib.legend.Legend at 0x7f5c1da045e0>
```

Figure 3.14 AKBANK 2022 prediction Daily return with Xgboost

4. MODEL RESULTS AND COMPARISONS

4.1 New Approach only Cryptocurrencies

After some work on the Istanbul stock exchange, especially when extracting tweets, since the translation process is also applied, approximately 50,000 tweets are extracted every hour. As the translation process takes a long time and we do not have enough RAM, we decided to use this process in the current program structure and only run the engine on the specified coins, such as BTC, DOGE, ETH, XRP, and BNB. Among these coins, we observed that good results were obtained in some coins with 1-hour and 2-hour durations, while insufficient results were obtained in some coins due to the variability of the model results and accuracy scores. In the last period, we will share the results we obtained in a short time and hourly basis and in the time range where we previously did data mining with you.

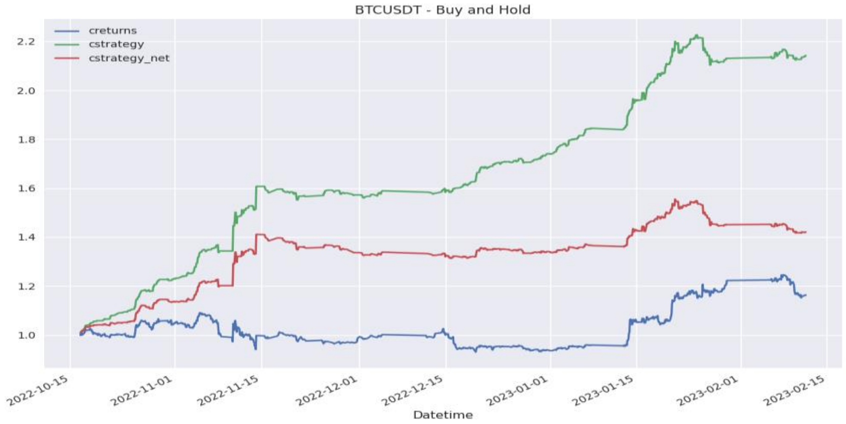


Fig 4.1 Bitcoin 1 hour model backtest result

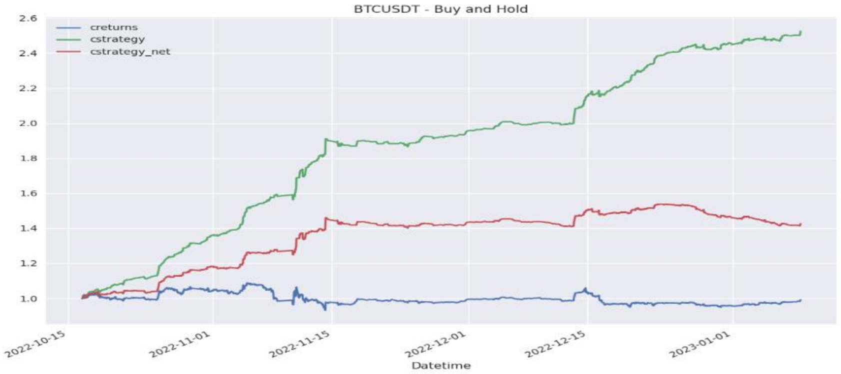
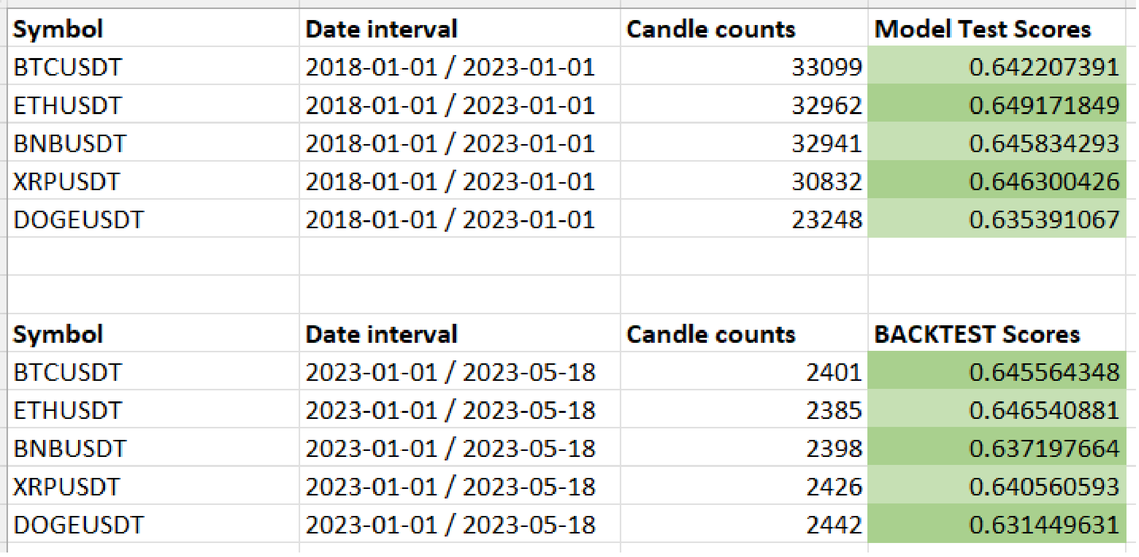


Fig 4.2 Bitcoin 2 hour model backtest result

We can see the results of the model we obtained from the three-month data set, which was processed at a 1-hour interval, specifically for Bitcoin. The blue line shows the net return in the three-month period, while the green line shows the result obtained from the model and the red line shows the return obtained by deducting the trade fees. As you can see, there is a significant difference between the model result and the result after deducting the trade fees. The main reason for this is the increase in transaction fees due to the large number of buy-sell transactions made on an hourly basis. With our strategy, sentiment analysis, and decision trees, we can achieve approximately 40% profit instead of losing money on a three-month interval. The other plot shows the backtest result of our model created with a two-hour data layer, and we see that this model has returned the same level of profit, approximately 40%, over a five-month period.



Symbol	Date interval	Candle counts	Model Test Scores
BTCUSDT	2018-01-01 / 2023-01-01	33099	0.642207391
ETHUSDT	2018-01-01 / 2023-01-01	32962	0.649171849
BNBUSDT	2018-01-01 / 2023-01-01	32941	0.645834293
XRPUSDT	2018-01-01 / 2023-01-01	30832	0.646300426
DOGEUSDT	2018-01-01 / 2023-01-01	23248	0.635391067

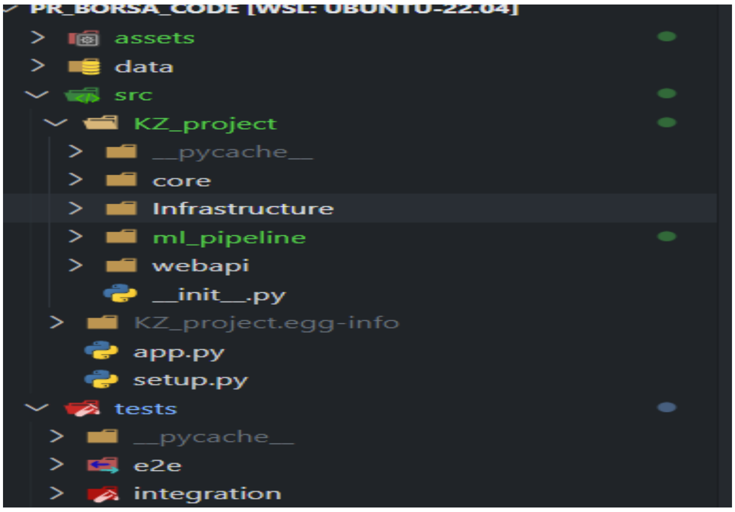
Symbol	Date interval	Candle counts	BACKTEST Scores
BTCUSDT	2023-01-01 / 2023-05-18	2401	0.645564348
ETHUSDT	2023-01-01 / 2023-05-18	2385	0.646540881
BNBUSDT	2023-01-01 / 2023-05-18	2398	0.637197664
XRPUSDT	2023-01-01 / 2023-05-18	2426	0.640560593
DOGEUSDT	2023-01-01 / 2023-05-18	2442	0.631449631

Fig 4.3 Last Accuracy Score

5. PROJECT OVERVIEW

5.1 Backend Structure

As the first fundamental step, we placed the signal tracker forecaster model and our assets into our domain and placed the interfaces here. Afterwards, we created the most important component, the machine learning pipeline I. We planned and integrated the execution of data preprocessing and digging stages into the structure. To fetch and process data from external service APIs, we wrote necessary external service extensions into the infrastructure module. After obtaining sentiment scores from Twitter data, we merged them with the matrix we created. We attempted to find the best parameters for XGBoost to achieve optimal efficiency using grid search cross-validation, and we compared the accuracy scores. We constructed a solid foundation based on clean architecture and SOLID principles.

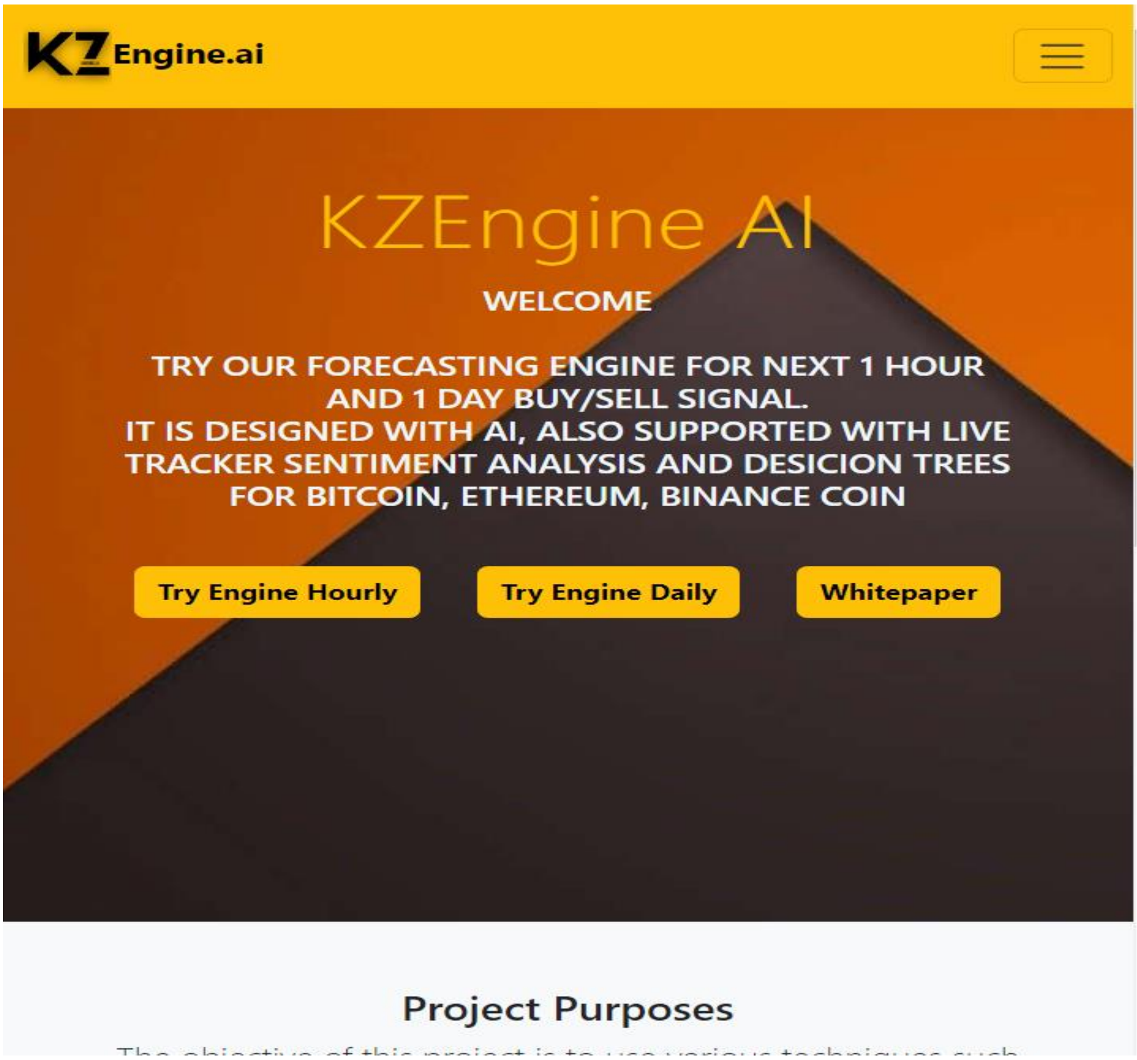


```
PR_BORSA_CODE [WSL: UBUNTU-22.04]
  > assets
  > data
  v src
    v KZ_project
      > __pycache__
      > core
      > Infrastructure
      > ml_pipeline
      > webapi
        __init__.py
    > KZ_project.egg-info
      app.py
      setup.py
  v tests
    > __pycache__
    > e2e
    > integration
```

Fig 5.1 Clean Architecture for KZEngine

5.2 Frontend Structure

I aimed to design a user-friendly interface for the frontend that allows the user to fully understand the project. Firstly, I wanted to explain the pipeline of the project and the algorithm used within it. On a separate page, called the "Digger" page, I presented the project's results, accuracy scores, and how the next hour and day will be closed with a candlestick pattern to the user. The specified diagram clearly illustrates the machine learning workflow.



KZEngine AI

WELCOME

TRY OUR FORECASTING ENGINE FOR NEXT 1 HOUR AND 1 DAY BUY/SELL SIGNAL.
IT IS DESIGNED WITH AI, ALSO SUPPORTED WITH LIVE TRACKER SENTIMENT ANALYSIS AND DESICION TREES FOR BITCOIN, ETHEREUM, BINANCE COIN

Project Purposes

The objective of this project is to use various techniques such

Fig 5.2 Main screen

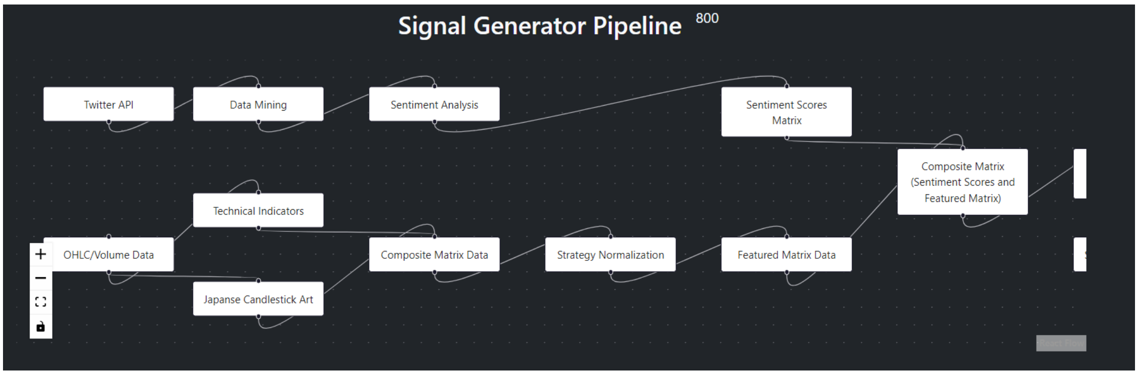
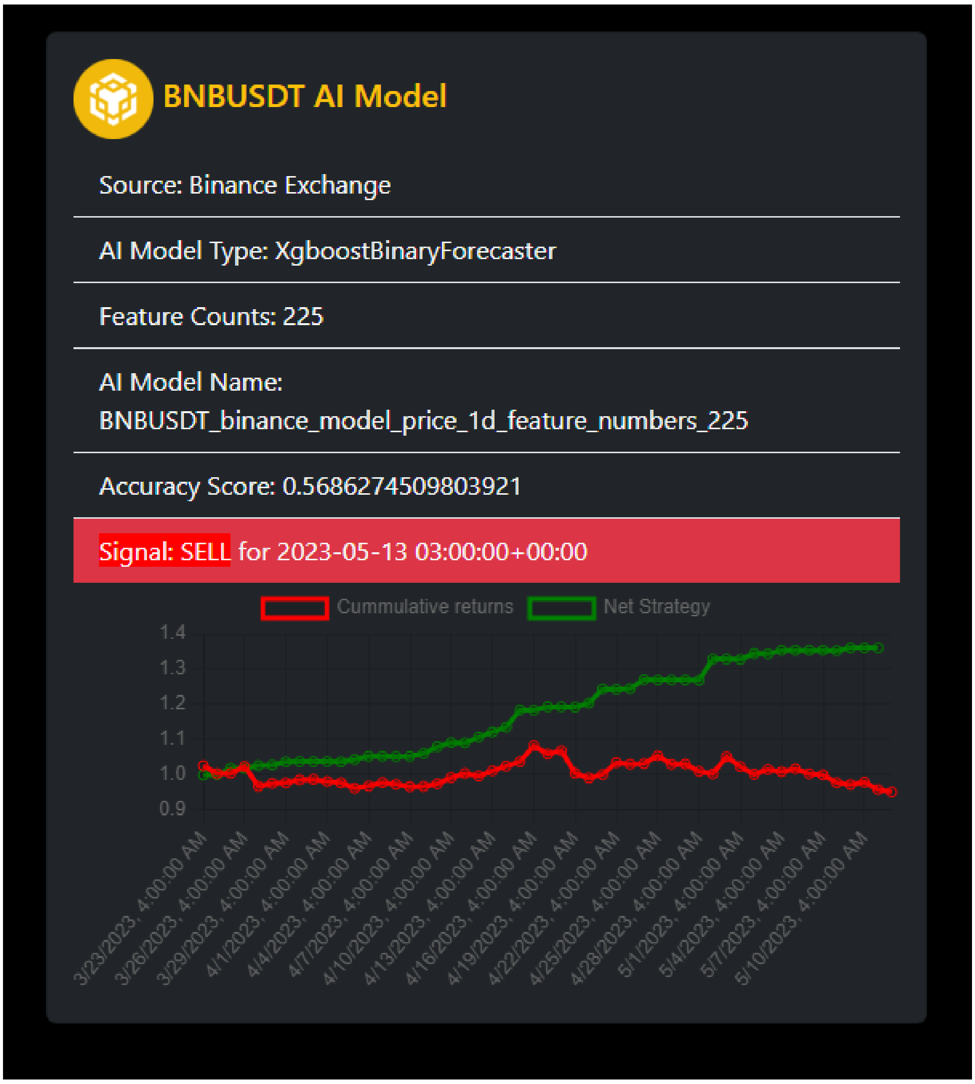


Fig 5.3 Diagram Machine Learning Workflow



BNBUSDT AI Model

Source: Binance Exchange

AI Model Type: XgboostBinaryForecaster

Feature Counts: 225

AI Model Name: BNBUSDT_binance_model_price_1d_feature_numbers_225

Accuracy Score: 0.5686274509803921

Signal: SELL for 2023-05-13 03:00:00+00:00

Fig 5.4 Signal Card

6. CONCLUSIONS

As a result, although some positive results are obtained on Bitcoin in daily data, negative results are obtained on both Bist and Cryptocurrencies, especially in hourly data. In addition to this, work is still going on the data pipeline process that I created. Necessary studies as NLP and sentiment analysis will still be added. Although I am thinking of converting the application to an application or establishing a platform in the future, I will work on price data and technical analysis to get good results before that. This project is versatile and includes many stages. When it comes to the backend, it is of course necessary to write clean code to ensure that this complex structure continues to function in a regular manner.

7. REFERENCES

Mladen Victor Wickerhauser - Introducing Financial Mathematics_ Theory, Binomial Models, and Applications (Chapman and Hall_CRC Financial Mathematics Series)-Chapman and Hall_CRC (2022)

Qian Yao - Blockchain-based New Financial Infrastructures_ Theory, Practice and Regulation-Springer (2022)

Yves Hilpisch - Artificial Intelligence in Finance_ A Python-Based Guide-O'Reilly Media (2020)

Graf, R.J. and Rowland, G.G. 1987. Effect of plant density on yield and components of yield of faba bean. *Canadian J. Plant Science*, 676 (1): 1-10.

Şengör, A.M.C. 1991. Plate tectonics and orogenic research after 25 years: Synopsis of a Tethyan perspective. *Tectonophysics*, 187(1): 315-330.

Yves Hilpisch - Python for Finance_ Mastering Data-Driven Finance Book-O'Reilly (2018)

FinAI Workshop in Brno will take place at Masaryk University, Faculty of Economics and Administration (Lipova 41a) on October 21-22, 2022.

AlgoHive: https://medium.com/hackernoon/how-were-using-machine-learning-and-trading-bots-to-predict-crypto-prices-5e8bdd5a9f2f

CURRICULUM VITAE

Ugur AKYEL was born in Bolu in 1988. He studied high school in Hatay and Ankara. I started Computer Engineering at Akdeniz University in 2019.