System Requirements Specification

Project Title: Algorithmic Trading Team Name: Dass Team 36 Team Members: * Bhuvanesh Sridharan * Animesh Sinha * Kesav Gadela * Sai Tharun Reddy

Problem Statement

The Project is to compare and contrast existing Algorithms for CryptoCurrency trading and develop our own combinations. We are to follow it up with algorithms for Pump detection in coins. The final stage of the project will be to develop a Machine Learning Application that figures out combinations of these rules to maximize profits. We will Provide Telegram notifications to the users for the same.

System Requirements

IDE and Software Setup

- JetBrains Pycharm project
- GitLab for Version Control
- Python 3.7 with Conda and Pip

Core Library Usage

- Binance (Fetching data on Cryptocurrencies)
- Thalib (For computing the average parameters and other indicators)
- Telegram API for making a Chatbot (BotFather group)

User Profiles

- CryptoCurrency Traders: These are people who are somewhat familiar with finance and cryptocurrencies and are looking for algorithmic advice on when to buy and sell. No computational expertise is expected off them. They shall be sent Telegram Alerts only of when to buy and sell their coins.
- Algorithmic Experts: There is a debugging port to test the profits made and what rules and algorithms failed us when. These are people with both Computational and Financial expertise.

Feature Requirements

- Comparisons of Existing Metrics
- Obtaining the price of Cryptocurrency with time.
- Running the Scripts over a span of 1-2 days and collecting data from the same.
- Applying the 18 given metrics at 3, 5, 15 - minute and marking purchase/sell times.
- Running the above for combination times and contrasting the results.
- Reporting the profits by each of the 18 algorithms on 6 time-combos.
- Preparing a MS Excel as an output for the above.
- Analysing the effect of cost of BitCoin in USD on the performance on these algorithms.
- Plotting profit chart of each algo against BitCoin-USD Exchange rate.
- The CSV file should have 6 columns (Coin Name, Purchase Time, Purchase Value, Sell Time, Sell Value, Profit/Loss)
- Pump Detection
- Reading up on the nature of Pump detection.
- Evaluation of pump based on the volume and price of coins.
- Marking common times of pump in the coins.
- Devicing Algorithms to predict pump.
- Comparing our alert system with existing telegram groups.
- Development of ML Algorithm
- Analyzing Hard-Coded Features from the 18*6 Metrics we have.
- Decide on other features that the ML Model is to be trained on.
- Builing a Convolutional Neural Net to analyze structure in the data.
- Ensure that we make steady profits every day averaged over the week.
- Contrast the performance of this model against rule based and ensure it's better.
- Attempt the Binance interface can be done through WebSockets to increase API limits.
- Keep a cache and not requery the Binance API.
- User Interface End
- There Should be a Telegram group with a ChatBot.
- The ChatBot should send text of when to /buy and /sell.
- There should be added reasoning and graphs with the message.
- The script should be runnable on an online deployed platform.

Since there is no good way to measure the accuracy of these algorithms, we shall attempt our best to make small profits on the run, there is no objective committed scale of measure. There is also no scaling and Deployment speed contraint.

Use Case Diagrams

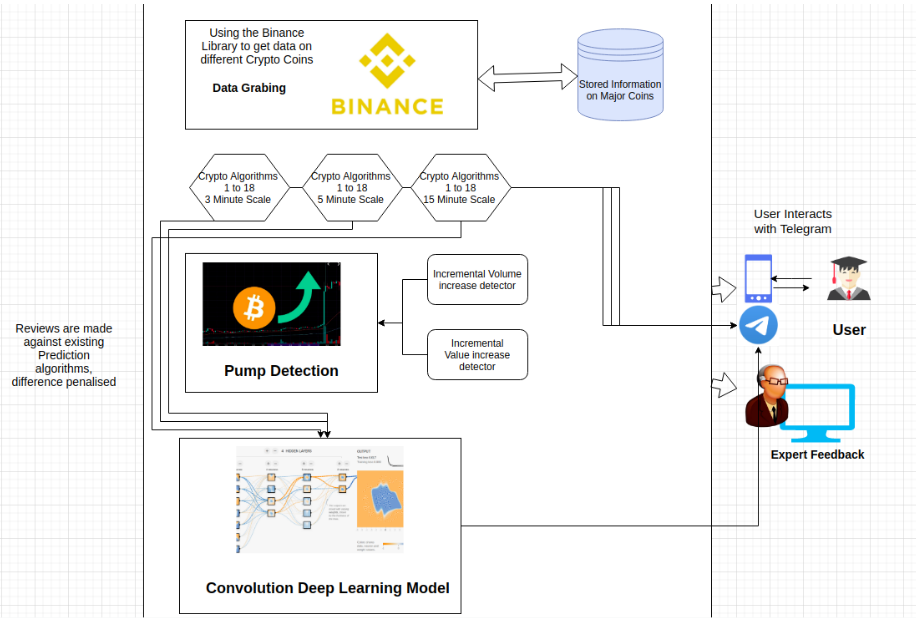
Using the Binance Library to get data on different Crypto Coins
Data Grabing
BINANCE
Stored Information on Major Coins
Crypto Algorithms 1 to 18 3 Minute Scale
Crypto Algorithms 1 to 18 5 Minute Scale
Crypto Algorithms 1 to 18 15 Minute Scale
User Interacts with Telegram
Incremental Volume increase detector
Pump Detection
Incremental Value increase detector
User
Reviews are made against existing Prediction algorithms, difference penalised
Expert Feedback
Convolution Deep Learning Model

Figure 1: UML Design Diagram