Submission Number: 1 **Submission Date: 30/11/21**

Group Number: Group 1, with Professor Ciresi.

Group Members: 2

Full Legal Name	Location (Country)	E-Mail Address	Non-Contributing Member (X)
Tilemachos Kosmetsas	Greece	kosmetsastilemahos@yahoo.com	
Christos Koutkos	Greece	christoskoutkos@msn.com	

Statement of integrity: By typing the names of all group members in the text box below, you confirm that the assignment submitted is original work produced by the group (*excluding any non-contributing members identified with an "X" above*).

Tilemachos Kosmetsas, Christos Koutkos

Use the box below to explain any attempts to reach out to a non-contributing member. Type (N/A) if all members contributed.

** Note, you may be required to provide proof of your outreach to non-contributing members upon request.*

The problem:

When given a time-series, a trader or a financial engineer, has to exploit his/her data to the fullest. Since high-frequency data incorporate a lot of noise, even the simple notion of acknowledging a bull or a bear market (for a human), can become a huge endeavor for the programmer. This definition is not so simple in algorithmic terms. How can one distinguish (with an actual trading bot implementation) a bear from a bull market by identifying patterns and imbalances and how one can decide when to enter the market when there is so much noise in the data? Is there a way to sample the data and create 'imbalance' bars that capture the surge of information at specific moments in a meaningful way?

Our Goal:

The goal of the project is to create a sampling method for BTCUSD(T) tick data. The algorithm must be iterative and dynamic in order to account for future structural changes in volume, number of buy/sell orders and the 'timedelta' between the trades.

Ideally, a successful sampling, should result to a meaningful signal creation. Nevertheless, we expect that randomness, sudden changes to the market structure, external forces, regulation etc might render a portion of these signals insignificant. Thus, in order to proceed with this project, a strong assumption must be made: that information leakage will create buy/sell and volume 'imbalances' prior to price action in most occasions.

By compressing the information of level 1 data, into proper bars, we can compress the data significantly while retaining their most important aspects, and thus, we will be able to use, any ML algorithm needed.

Objectives:

- Organize the data for optimal testing/working with Python. Memory management is crucial.
- Create imbalance bars, run bars (De Prado)
- Create a model in order to cluster the bars into 'long' or 'short' signals.
- Back-testing the algorithm with test data, implementing performance optimizations and repeating.
- Optional: Identifying structural changes in volatility with a signal decomposition approach.
- Summarize conclusions.