Chiyoung Jeffrey lii

2/9/2019

IST 659 Project Proposal

Stock and Technical Analysis Database

# Business Description

The business described in this document is one that provides stock market and technical data to stock market traders, as well as providing a record of trading strategies. It services the financial sector. The database provided will contain the open, low, high, and close prices of selected stocks, technical indicators of those stocks, and the results of back testing various strategies. Technical traders use a myriad of strategies to trade stocks. Each strategy needs different kind of stock data to identify trends (e.g. Dow theory) and risk analysis (e.g. finding retracements).

Dow theory identifies trend using momentum, price movement, supports and resistances. Dow theory only considers market closing prices. Retracements are temporary reversals of trends. They are often used with other technical indicators, such as the exponential moving average, or Bollinger bands. Retracements are considered low risk areas of activity, so long as other technical indicators corroborate the overall trend.

Because of these strategies, the database will also contain entities with data on three technical indicators (mathematical values derived from stock data used to make decisions on trading stocks). Firstly, the exponential moving average (EMA), a weighted sum of averages over a certain period of days. Secondly, the Moving Average Convergence Divergence (MACD), a momentum indicator that is based on multiple EMA sums. Finally, Bollinger Bands (bbands) that are the values two standard deviations away from the current moving average.

It is also essential that traders keep records of their trades, so they can improve their skills. That is why the database will also include three entities for recommendations, recommendation versions, and back test results.

For financial groups requiring stock market data, the costs and effort to build and maintain a data repository may draw resources away from their core business model. Given a conservative estimate of a small team of five database administrators averaging 100,000-dollar salaries, the cost of maintaining this financial data is already 500,000 dollars a year. This is not even considering hardware costs, location costs, licensing fees, and services. While these costs are trivial for large groups such as Goldman Sachs or Charles Schwab, these costs are potentially business ending for small startups and medium sized financial groups.

On the other hand, the business can service multiple organizations at the same time. The business can generate value by leveraging the talents of the same team for multiple clients, allowing clients to have access to a database for a fraction of the cost.

# Problem Statement

Given an ideal client “A”, a start-up trading firm full of analysts who have no experience with databases. The only way “A” can maintain their data is through files and spreadsheets. However, this is cumbersome to access, because trader Bob may not have access to trader Tom’s spreadsheets. Bob must ask Tom for his spreadsheets on the closing prices of the SPY stock over the last hundred days, but Tom doesn’t have it right now. Tom says he’ll give Bob the spreadsheets tomorrow.

By mistake, Tom deleted some of his spreadsheets, and instead gives Bob some spreadsheets that don’t have the last 14 days of stock market movement. This results in Bob missing a recent uptrend and botching an important trade, resulting in the loss of a million dollars.

Firm “A” is having a classic information management problem and decides to do something about it. They research creating and maintaining a database but realize how expensive it is.

# Proposed Solution

Database administration is expensive. So rather than asking financial groups to start up an IT division that will always be secondary to their business model, our clients should outsource their database administration needs to us. The alternative is to hire on specialists that will never gain the support they require to truly flourish in their chosen specialization, it is a much better idea to simply subcontract out the job of starting, maintaining, and enhancing their database needs for a simple, flat price, with contracts for specialized requirements.

The most basic business function would be to allow all users create, read, update, and delete all tables they have access to, whenever they want, so long as applicable security standards are met. Once they have the stock and technical data, they can then generate a back test using whatever strategy they please. After back testing, the traders can store the result in the database to have a record of the efficiency of the strategy. Using these stored results, traders can slowly improve on their issues.

# Users

The primary users will be traders who depend on stock and technical data to make traders. A trader will be given the ability to read data from certain tables, with limitations based on how often they can be read. Limited ability to update, add, and delete rows would be allowed, but dropping tables would not be allowed. For example, a user should be able to access a given table only one million times a day. Traders that want to refine their strategies would be able to store their back test results in another set of tables.

# Potential Entities and their Attributes

1. Daily\_Data
   1. Daily\_data\_key
   2. Ticker
   3. Day
2. Time\_Series\_Daily
   1. Time\_series\_daily\_key
   2. Daily\_data\_key
   3. Open
   4. Close
   5. High
   6. Low
3. Daily\_technicals
   1. Daily\_technicals\_key
   2. Daily\_data\_key
4. EMA
   1. Ema\_key
   2. Ema
   3. Daily\_Technicals\_key
5. Macd
   1. Macd\_key
   2. Daily\_technicals\_key
   3. Macd\_hist
   4. Macd\_signal
   5. Macd
6. Bbands
   1. Bbands\_key
   2. Daily\_Technicals\_Key
   3. Real\_middle\_band
   4. Real\_upper\_band
   5. Real\_lower\_band
7. Recommendation\_type
   1. Recommendation\_type\_key
   2. Trend
   3. Name
   4. Type
   5. Description
8. Recommendation\_version
   1. Recommendation\_version\_key
   2. Recommendation\_type\_key
   3. Version
9. Backtest\_results
   1. Backtest\_Results\_key
   2. Recommendation\_version\_key
   3. Date\_tested
   4. Test\_begin\_date
   5. Test\_end\_date
   6. Base\_Capital
   7. Biggest\_loss
   8. Total\_gain
   9. Successes
   10. Failures
   11. Fizzles
   12. Total
   13. Percent\_Success
   14. Percent\_not\_failed