As a Software Engineer, I have been exposed to different approaches of project management as such Software Development Life Cycle SDLC (Waterfall), Agile (Scum, Kanban) and Data-Driven (Data Science Life Cycle - DSLC). Standard project management techniques Waterfall or Agile do not fit well with data-driven projects (AI, ML, Data Science). Standard practices are geared towards application development along with stakeholder participation. Whereas on data-driven projects, it is the learning from data analysis & ML models that have to be infused into the application, thus demanding a different approach. DSLC as a methodology, it includes descriptions of the typical phases of a project, the tasks involved with each phase, and an explanation of the relationships between these tasks. As a process model, CRISP-DM (CRoss Industry Standard Process for Data Mining) provides an overview of the data mining life cycle.

Here I will discuss one of those projects (a trading system) which uses the Data-Drive approach.

1. Business understanding

The User(trader) must determine those financial instruments that they want to trade. The more familiar the user with an instrument, the better. Depending on the strategy, some instruments are more suitable than others. The platform should offer applications programming interface (API) trading. This allows users to interact with the online broker programmatically. That means users can stream market data and make orders with programming languages like Python.

2. Data understanding

The user pulls historical data via the Broker API. Brokers typically provide price and volume data. If necessary, the user needs to identify additional sources for fundamental data. Data understanding is a key workflow in any data project. It allows analysts to gain a deeper understanding of the underlying data. Data understanding includes.

• Data visualization

• Data inconsistencies (detecting missing or corrupted data)

3. Data preparation

This step/phase includes data manipulation and choice of data (starting point)

• Data manipulation (Cleaning, formatting, aggregating, and reshaping the data, and also storing of data to local or cloud repository)

• Starting point

The next three steps go hand in hand. Depending on the strategy, these steps are executed simultaneously or one after another. The user starts with a rough idea of what a profitable strategy could look like. Then, the strategy is defined with a programming language.

4. Modelling

The user must test the performance of the strategy on the data at hand. This is called backtesting. The performance includes risk and return/profit metrics. Those strategies are optimized to fit the underlying data. ML strategies are a good example. ML models should fit the sample data. The only drawback here is that many fitted strategies tend to overfit. Overfitted strategies seem to be profitable on the data at sample data (“in-sample”), but they fail to generate profits in live (“out-sample”). Forward testing, also known as out-sample backtesting, is a helpful tool to identify overfitting. The user tests the strategy on new data that the strategy has not seen before. When overfitting could be an issue, it’s common practice to split the dataset. The user defines and optimizes the strategy on the training set, then forward and tests the strategy on the test set. This step should only continue if the strategy performs well after trading costs and that performance is confirmed by forward testing.

5. Evaluation

Once the data analysis is complete, an implementation algorithm is written which is different from the backtesting algorithm. it’s important to test the implementation in a simulated environment. Many brokers offer practice accounts that allow paper trading. There is zero risk of losing money in a paper trading session.

6. Deployment

Once the algorithm is in production, it receives real market data and books live trade orders in the market. An in-production algorithm can produce profits and losses. There is no guarantee that an in-production strategy will generate profits, and many strategies deteriorate over time. Thus, it’s important to closely monitor performance. Algorithmic trading sessions can last for many hours. Technical issues can harm and/or stop trading sessions, which can lead to high losses. Thus, it’s best practice to deploy trading algorithms on cloud servers.

Hi Henry,

The implementation process which, you describe is Data-Driven approach. Incrementing the functionality by Unit test scenarios and trying to analyse data. Sorry to hear that you have to deal with messy database, in the past there was a mythology SSADM for the normalization process of relational data. Nowadays I work with NoSQL database.

Hi Hetal,

The process you describe is based on Data-Driven approach Data Science Life Cycle (DSLC) mythology, if it is DM or not DM approach, it depends if you have used data mining techniques in analysing your data.