# Final Year Project Idea

### BitCoin Value Prediction Tool

1. Pull data from web (<https://api.gdax.com>).
2. Calculate average value per time period (5/10/15 min?)

*N.B. too many trades to do this with all trades (often 10,000+ per second, mostly within 1p of each other), take the first few trades of each 1 minute to create a 5-minute average.*

* 1. Isolate extremely large sales/purchases:
     1. Could be someone trying to fix market.
     2. Could be someone who is aware of imminent change in value.

1. Calculate value change between time periods (as a percentage) and store this value in database (Oracle? The currency has a value at a time period)
2. Create user interface (desktop application) to show statistics to user
   1. Most recent change can simply be a value
   2. Draw graph to show value change:
   3. Actual value graph.
   4. Percentage change graph.
   5. Allow user to customise graph to focus on specific time period for each graph.
3. Run algorithms to predict change for next time period:
   1. x time periods of consistent growth/decline (confidence in market growing/falling).
   2. Daily value change trends (local markets opening/closing).
   3. Weekly value change trends (freeing up capital for weekends?)
   4. Monthly value change trends (purchasing after payday? Selling before payday for cash flow?)
   5. Annual value change trends (freeing up capital for seasonal purchases? **“Sell-in-May-and-go-away”**)
4. Advise user to buy or sell based on prediction in 5.
   1. Ethics?
5. Extend 6 to allow user to automate purchasing and selling based on predictions.
   1. User needs to sign liability waiver.
   2. User needs to input min value before sale and max value to purchase at.
6. Expand application to include additional currencies on GDAX.

Use GDAX guides to pull product data (NodeJS/Ruby/Python/PHP).

Store in Oracle DB.

Pull into Matlab, calculate price changes.

Research stock prediction algorithms:

* Regression.
* Use on small data set.
* Compare (what earns most £)

Settle on single algorithm.

GUI for end user.

* Software development, database development, data analysis.
* To be used by: Crypto-Currency Stock Holders (Bitcoin, potentially Ethereum).
* I want to make a system that pulls an average trade value from the GDAX api and posts it to my database, analyses the price change against prior collected data and makes a prediction based on previous data held within the database.
  + Efficiency - program able to perform analysis faster than humans.
  + Accuracy - program able to ensure zero mistakes in calculations (not necessarily that the recommendation is correct, just that the mathematics is).
  + Impartiality - program able to ensure no emotional decision factored into recommendation.
* A desktop system that pulls data from GDAX API to create an average price for a time period, stores it in a database and makes predictions for future changes based on prior price movements. Provides the client the ability to see likely change in currency value and act accordingly (potentially with an automated feature built into the system).
* Method:
  + AGILE planning.
  + GDAX API – Python/PHP to pull from and populate my database.
  + Oracle DB - C# for my own API endpoints.
  + Matlab to analyse the data/create an interface for the user.
* No hardware/software requirements other than those already supplied by university (i.e. space on university oracle server, use of university matlab license, etc.).
* Learning requirements:
  + How to access data through GDAX API, they supply basic help on how to do so in (several languages including Python and PHP).
  + Storing above data in Oracle DB in Python/PHP.
  + Access Oracle DB through Matlab (or C# API endpoints if necessary).
  + Research into stock market prediction algorithms.
  + Matlab - anything that I require that isn't taught in AINT351.
* Risks:

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Probability** | **Impact** | **Course of Action** |
| GDAX API discontinuing | Low | High | Source alternative API endpoint. |
| Oracle server goes offline | Low | High | Wait for Oracle/university to resolve, research and consult with supervisor about alternative data storage methods such as a NoSQL database that could be stored locally. |
| Loss of entire project. | Low | High | Keep numerous backups. |
| Loss of small portions of project. | Medium | Low | Use repository to ensure up to date version accessible. |
| Equipment loss/damage. | Medium | Low | Switch from desktop to laptop/spare laptop or use university equipment such as open access. |
| Difficulty with learning requirements mentioned above | Medium | Medium | Apply more time to learning the specific skills required to complete the task. |
| I lack specific knowledge to complete a segment of the project | High | Medium | Accept that all new projects have a learning process attached to them, allow extra time for parts that haven’t been attempted previously. |