

# Capstone Proposal: Stock Recommendation

People who operates in day trade have to analyze the previous days and recognize the stock movement patterns. There is a certain limit to how much attention humans can give to tasks. Opening multiple computer windows to operate on different companies' assets can be effortful.

Whereas, Machine Learning models are capable of recognizing patterns and make decisions faster. Furthermore, it can be scaled to cover more stock options.

The proposal for this capstone is an app which will recommend stock operations to be concluded in the same day.

## Data

The dataset used is the *Huge Stock Price Data: Intraday Minute Bar* by [Möbius](#) which can be found at [kaggle](#) and [github](#).

The data used to train the model will be the minute-by-minute data of stocks. Among many others, the stocks can be:

```
$ tree -L 2 data/stocks
data/stocks/
├── histdata/
│   ├── ETXEUR/
│   ├── GRXEUR/
│   ├── JPXJPY/
│   ├── SPXUSD/
│   └── README.md
```

Here is an example of the EUR\_USD:

```
import pyfinancialdata
data = pyfinancialdata.get(
    provider="oanda",
    instrument="EUR_USD",
    year=2017,
)
data.tail(3)
```

		close	high	low	open	volume
price						
date						
2017-12-29	21:57:00	1.20045	1.20071	1.20004	1.20018	50
		1.20045				

2017-12-29	21:58:00	1.20041	1.20041	1.20041	1.20041	1
		1.20041				
2017-12-29	21:59:00	1.20039	1.20039	1.19970	1.20036	14
		1.20039				

The data will be explored and cleaned if necessary. Then, it will be prepared and transformed to provide the model.

## Algorithms and models

A couple of algorithms will be defined and the models will be trained with different hyperparameters. The models will be compared between themselves according to accuracy and efficiency. The accuracy will be calculated by the root mean square error of the prediction and the real data. The efficiency will be measured by calculating the possible profit by the end of the day.

One model will be chosen to be deployed to production. An api will be developed to access the model and calculate the recommendation. The api will then calculate the amount and whether to buy or sell the stock in order to have profit.

A report will be written structured as the examples given at [udacity/machine-learning](https://udacity.com/course/machine-learning-course/)

## Reference

The materials that will be used as reference are the following:

Course [Unit: Stocks and bonds](#) from Khan Academy

Course [Machine Learning for Trading](#) from Udacity

Book Machine Learning for Algorithmic Trading by Stefan Jansen

Book [Python for Finance](#) by Yves Hilpisch