Financial Data Forecaster
Detailed Project Plan
Group FYP21020

Kwok Zi Xing 3035603813
Lam Zi Hao 3035549146
Project Background

In recent times, artificial intelligence has an increased usage in trading, this is due to the nature of computers which are capable of processing huge amounts of computations in nanoseconds and able to discover patterns in huge datasets which are unnoticeable by humans. Thus, numerous strategies have been developed with the help of machine learning and natural language processing to aid traders in forecasting financial data and making trading decisions.

Vanilla projects focus on forecasting a single stock’s future price based on its own historical price. In this project, we take a different approach by building a model that first identifies correlated pairs of stocks through price action and industry, then forecast the stock’s future price with another highly correlated stock’s historical price.

The idea behind the project stemmed from spotting similar chart formations within the same sector. With a feature available on vBroker in Figure 1 that states similar chart patterns, we pondered upon the technology and accuracy behind this feature on the app. Thus, we came up with the idea of this project.

![Figure 1. Similar Candles feature on the vBroker trading app.](image-url)
Project Objective

The main objective of this project is to forecast future stock prices of one stock based on another’s stock historical price based upon verified correlated pairs of stocks identified using unsupervised machine learning. However, as we are still in the early stages of the project, we are actively brainstorming new ideas that may complement our current project plan.

Scope

The scope of study within this project is limited to the Hong Kong stock market and the financial security under study are active Hong Kong stocks.

Methodology

1. Data Collection
   To acquire data for our project, we will be utilizing web scraping technologies such as BeautifulSoup and Selenium to gather data not retrievable by the free Python API yfinance. Web scraping will be scraped from the HKEX website for active stock tickers and ETF tickers trading in the Hong Kong exchange currently. For daily stock prices data, we will use yfinance and pandas-datareader since yfinance covers stocks within the Hong Kong stock market and is sufficiently quick. To scale up our operations, we may have to consider using commercial data sources such as rapidAPI for quicker data collection speeds.

2. Data Preprocessing and Ratio Calculation
   We will filter ETF tickers from our list of available tickers in the Hong Kong market to acquire the list of all actively traded stocks in the Hong Kong exchange. Then, we will call the yfinance API to collect historical price data and company industry for all actively traded stocks in Hong Kong.

   After collecting the historical prices of all stocks in the Hong Kong market, we will implement an algorithm that calculates the price ratio for every peak and trough created by the stock’s price. We are utilizing a 10% threshold for our ratio calculation, that means if the price of a stock rises 10% from its recent low or vice versa, we will store and identify the price as a data point. To standardize our ratios, we will begin calculation from the stock’s historical low price. After gathering the ratios required, we will utilize standard data preprocessing techniques such as Feature Extraction, Standardization and Principal Component Analysis to prepare fitting our data into clustering methods. This process ensures that our data is compressed and standardized into a more ideal format for meaningful information to be extracted from the data during the training stage and to save computation time.
3. Stock clustering
   We will experiment with various clustering methods such as DBScan, Optics and K-means clustering to identify the best method for our use-case. The training data will be fed into the clustering models and the results will be visualized.

4. Forecasting prices
   Based upon similar historical prices ratios and company industry sectors, we will identify correlated pairs of stocks from our clustering model. Therefore, our model will be able to generate a list of correlated stock pairs. By transforming this list of correlated stock pairs into a 2D matrix with the first column and row the historical prices, we will be able to cross-predict stocks from ratios of other stocks and test its accuracy through the backtesting library.

5. Backtesting
   We will be utilizing Python backtesting libraries at Pythonrepo and online to identify suitable backtesting libraries to facilitate the backtesting process of our clustering model’s performance and predicted prices. We are yet to settle upon a finalised backtesting library to execute our backtesting process and we are still exploring. If required, we will modify the libraries available online to cater to our backtesting needs.

Deliverables & Schedule

   a. Scrape.py (HKEX and ETF data)

   a. Prices.csv (Historical prices for all stocks in HK)
   b. Ratio.py (calculating ratios for all stocks)
   c. Preprocessing.py (Feature Extraction, Standardization and Principal Component Analysis)

   a. Cluster.py (Various clustering methods for visualisation)

   a. Predict.py (Script to predict price)

   a. Backtest.py (Backtesting process)
Feasibility Assessment

To identify the feasibility of this project, we have conducted manual backtesting upon the strategy proposed. We have identified two shares within the Hong Kong stock market to be within the same cluster and have conducted price forecasting over two months. The criteria used to identify these two stocks as within the same clusters are that both are in China’s technology sector. The shares are Xiaomi (1810.HK) and Meitu (1357.HK).

Figure 1. Historical Price of Meitu (1357.HK) with historical low at $1.32 on 2nd December 2019.
Assuming both stocks began from the historical lowest, from figure 1 and figure 2, the historical low of Xiaomi is $8.28 whilst Meitu being $1.32. Since Xiaomi prices are formed before Meitu, we will then use Xiaomi’s price to forecast Meitu’s price. In the first rally, Xiaomi rallied to $14.00 from $8.28. By utilising ratios and multiplying Xiaomi’s price ratio with Meitu’s historical price, we obtain \( \left( \frac{14.00}{8.28} \right) \times 1.32 = 2.23 \). This provided us with a forecast of $2.23 whilst in fact, Meitu rallied up to $2.27 which deviated from the actual price of 1.2%.

Furthermore, Xiaomi fell down to $9.20 from $14.00. Through the same method, \( \left( \frac{9.20}{14.00} \right) \times 1.32 = 1.49 \) which Meitu was projected to fall to $1.49 but it fell to $1.37 in reality. This gives us a predicted-to-actual price deviation of 8.75%.

In subsequent rallies, Xiaomi rose to $35.90 from $8.28 without breaking through any recent low structures and temporal peaks forming at $17.50 and $26.95. That gives us a price estimates calculations of \( \left( \frac{17.50}{8.28} \right) \times 1.32 = 2.11, \) \( \left( \frac{35.90}{8.28} \right) \times 1.32 = 4.30 \) and \( \left( \frac{35.90}{8.28} \right) \times 1.32 = 5.72 \) without breaking the structure. However, Meitu rose to $4.50 and subsequently fell back down to $1.41. This meant the initial two targets were met in a short period of time with $2.11 and $4.30 being achieved. The predicted price of $4.30 deviated from the actual price of Meitu of $4.5 by 4.44%. 

Figure 2. Historical Price of Xiaomi (1810.HK) with historical low at $8.28 on 2nd September 2019.
Conclusion

Although the criteria set for clustering stocks together within this project may be simple, we are working on incorporating more features into the clustering as it is the core of our project. We believe that leveraging upon Machine Learning to crunch large datasets in the right direction will provide us with an edge in the world of stocks.