# Team: Surviving Bear (markets)

Members: Prasanth Duvvur, Rajesh Paul, Samuel

## Goal

Goal of this project is to develop a system that can identify stocks for value investing, especially during a bear market.

## Idea / Hypothesis

Since there are no absolute ways to predict stock markets, we approach this goal more as a aid for a buyer to do the analysis on shortlisted stocks before making a final decision. The thought is we should be able to identify good value stocks by studying historical values and then applying that to current market. Typically, best values are found during bear markets so the thought is that purchases would be made in a bear market or if sudden market conditions affect certain stocks. Looking through all the stocks manually to identify such stocks is very tedious, so the thought is to leverage machine learning. Once a set of good stocks are identified and ranked, we will need to further analyze manually to work in other factors like future prospects for the company, current events, current market conditions etc. to determine when to buy a stock and at what price.

## High level system design:

Machine Learning module

Data ETL (Python)

Data Sources (Web)

Models

DB

Data Visualization

Data Source: Stock fundamentals data historical (alteast 20 years), current stock price data

Data ETL: This layer is for initial data analysis and loading data into the database. We may need to add some calculated fields.

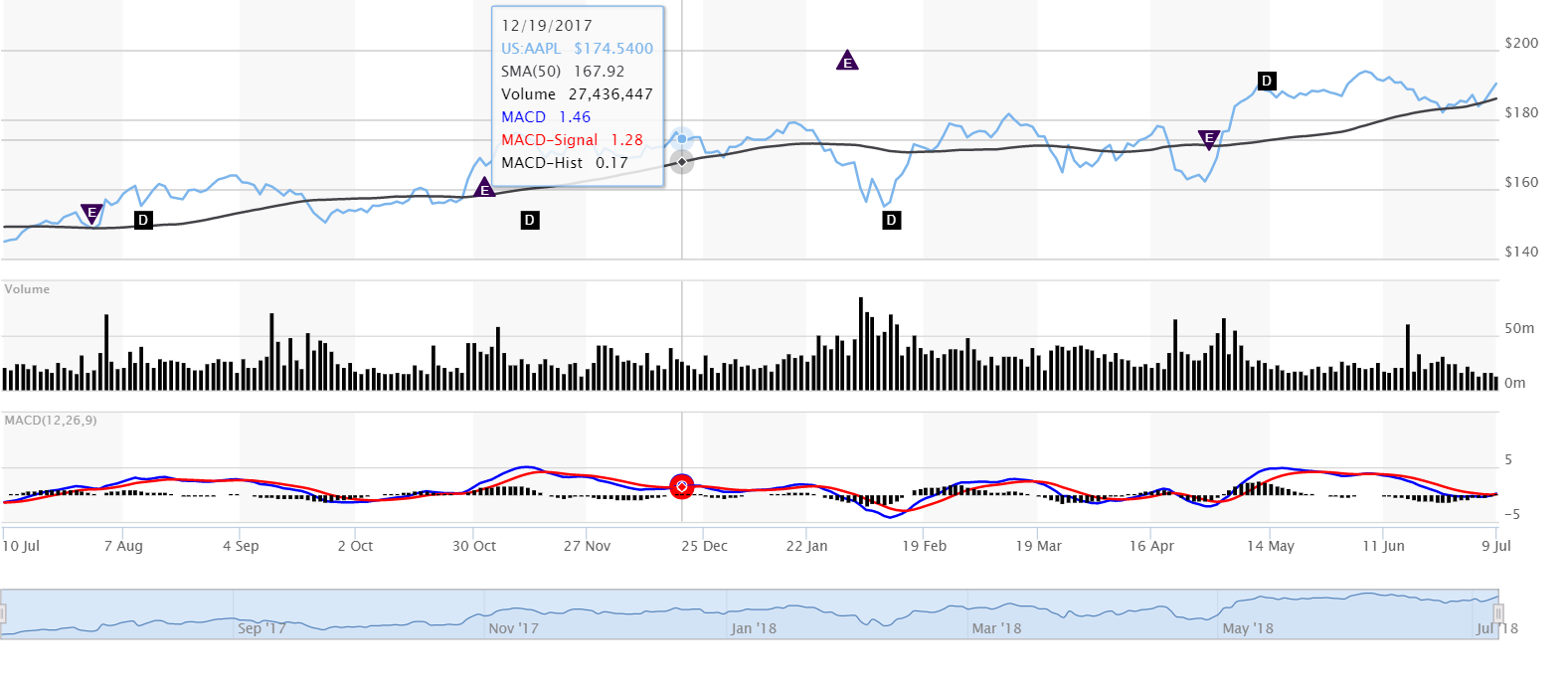
DB: Storage for data

Machine learning layer: This layer will be trained based on the data to create models. These models should then be tested and if satisfactory used for future. Multiple models could be produced and models should be re-trained over time. An ensemble model could be incorporated to bring together the results from the different models.

Visualization layer: This layer will be the front end for the user it will have multiple dashboards

1. Pick stocks by applying ML models. Present the results in an interactive manner. User input could be investment horizon
2. Further analyze selected stocks using dashboard

Here are some other charts that truly inspired us:



This chart shows a lot of possibility of user interaction. We would like to have similar functionality on our charts, allowing the user to see values at different points of time when they mouse over the map.

Steps

Data loading

1. CSV files for each stock downloaded and stored to DB using python and MySQL (webscraping, processing in python, saved to csv, loaded into MySQL, added more caculated columns like returns for different time periods)
2. Identify measures/features we want to use for modeling and the returns we want to use for target

## Data Analysis

### Decision tree analysis

Decision tree is the first model being used. The decision tree will also provide a subset of features/criteria to use for the other models.

The strategy is to compare decision tree model accuracy for different depths and pick the one which is most accurate but has the least depth, i.e, identify the depth after which increase in accuracy is not significant (<5%).

For the first round any row with positive returns is considered as a good investment.

Accuracy score by level of DT and for different time periods, for returns > 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Max depth** | **Quarterly** | **One** | **Three** | **Five** | **Seven** |
| Full |  | 0.607250 | 0.602719 | 0.548338 | 0.395770 |
| 2 |  | 0.864048 | 0.805135 | 0.492447 | 0.267371 |
| 3 |  | 0.851963 | 0.790030 | 0.504531 | 0.262839 |
| 4 |  | 0.84138 | 0.761329 | 0.504532 |  |
| 5 |  |  | 0.761329 | 0.534743 |  |
| 6 |  |  | 0.752265 | 0.534743 |  |

**Quaterly time period:**

Counts for calculated buy for test data: Counter({1.0: 465, 0.0: 164})

Counts for calculated buy for train data: Counter({1.0: 32248, 0.0: 20808})

The best and simplest level seems to be 2.

Factors for level 2: EPS basic, mktcap\_revenue\_value

Overall though accuracy for 1 qtr is lower than other time periods, so should be considered appropriately.

**1 Year time period:**

Counts for calculated buy for test data: Counter({1.0: 572, 0.0: 90})

Counts for calculated buy for train data: Counter({1.0: 19538, 0.0: 11996})

Level 2 and 3 are the best

Level 2: mktcap\_revenue\_value, mktcap\_cash\_value, Free cash flow per share

Level 3: mktcap\_revenue\_value, mktcap\_cash\_value, Free cash flow per share, P/B ratio, EPS basic

**3 Year time period:**

Counts for calculated buy for test data: Counter({1.0: 533, 0.0: 129})

Counts for calculated buy for train data: Counter({1.0: 17443, 0.0: 8679})

Three year results have a significantly higher rate of accuracy, with level 2 being the highest

Level 2 factors: mktcap\_revenue\_value, mktcap\_cash\_value, Equity to assets ratio

**5 year time period:**

Counts for calculated buy for test data: Counter({0.0: 340, 1.0: 322})

Counts for calculated buy for train data: Counter({1.0: 13182, 0.0: 7734})

Level 5 seems to be best

Factors for level 5:

**7 yr time period:**

Counts for calculated buy for test data: Counter({0.0: 501, 1.0: 161})

Counts for calculated buy for train data: Counter({1.0: 10844, 0.0: 5083})

Percentages are below 50 so not worth considering

### Random Forest analysis

Score and feature importance by time period

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | 1 year | 3 year | 5 year | 7 year |
| Score test data | 0.785498 | 0.77039 | 0.495468 | 0.321752 |
| P/B ratio | 0.098436 | 0.10593 |  |  |
| Free\_cash\_flow\_per\_share | 0.101277 | 0.10539 |  |  |
| Equity to assets ratio | 0.102129 | 0.09999 |  |  |
| Current ratio | 0.090440 | 0.08683 |  |  |
| EPS basic | 0.092523 | 0.08559 |  |  |
| ROA | 0.086263 | 0.08429 |  |  |
| ROE | 0.084483 | 0.08079 |  |  |
| Long-term\_debt\_to\_equity | 0.081662 | 0.08038 |  |  |
| P/E ratio | 0.08031 | 0.07998 |  |  |
| cash\_oper\_gt\_earning\_val |  |  |  |  |
| Dividend payout ratio |  |  |  |  |
| mktcap\_cash\_value |  |  |  |  |
| mktcap\_revenue\_value |  |  |  |  |
| marketcap\_bookvalue\_value |  |  |  |  |
| mktcap\_free\_cash\_flow\_value |  |  |  |  |
| entvalue\_earnings\_value |  |  |  |  |
|  |  |  |  |  |

7 year data:

Counts for calculated buy for test data: Counter({0.0: 501, 1.0: 161})

Counts for calculated buy for train data: Counter({1.0: 10844, 0.0: 5083})

5 yr return:

Counts for calculated buy for test data: Counter({0.0: 336, 1.0: 326})

Counts for calculated buy for train data: Counter({1.0: 14088, 0.0: 6828})

**3 yr return:**

Counts for calculated buy for test data: Counter({1.0: 533, 0.0: 129})

Counts for calculated buy for train data: Counter({1.0: 17443, 0.0: 8679})

P5

**1 yr return:**

Counts for calculated buy for test data: Counter({1.0: 572, 0.0: 90})

Counts for calculated buy for train data: Counter({1.0: 19538, 0.0: 11996})

Similar to the Decision Tree analysis, best score seems to be for 3 years time frame, 1 qtr should probably be dropped. The top factors common for 1,3 and 5 yr terms are ROA, P/E ratio, ROE, P/B ratio, EPS basic we should proceed with just these features for the other models

SVC linear

### KNN Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| Parameters | 1 yr | 3 yr | 5 yr |
| 16 | k=11 Acc: 0.751 | k=9 Acc: 0.754 | k=5 Acc: 0.523 |
| Best 5 |  | k=13 Acc: 0.730 |  |

3 yr:

Counts for calculated buy for test data: Counter({1.0: 533, 0.0: 129})

Counts for calculated buy for train data: Counter({1.0: 17443, 0.0: 8679})

5 yr:

Counts for calculated buy for test data: Counter({0.0: 336, 1.0: 326})

Counts for calculated buy for train data: Counter({1.0: 14088, 0.0: 6828})