BerkeleyHaas

Stock Selection using Machine Learning based Classification algorithms

MFE Team 17: Trilok Yadav, Vikrant Dhall, Akshay Gupta, Ritvik Goyal, Rohan Gandhi, Suki Yang

Supervisor: Prof. Kevin Coldiron

Introduction

- Problem Description:
  - Most of the applications of ML to stock markets have focused on predicting returns which suffer from residual errors in regression
  - Traditional factor models have shown to perform poorly over long periods
- Our Approach:
  - Rather than predicting the stock returns, we focus on identifying over/under performing stocks by creating five quantile buckets (+2 to -2)
    - Classify stocks based on combination of fundamental and market factors
  - For classification, comparison of bagging and boosting techniques is done
    - Rolling window regression of 12 months of data to predict 1-month buckets of stocks

Literature Survey

- RU ZHANG, TONG CAO (2018)
- SUN YUTONG, HANQING ZHAO (2015)
- XIAOYUN ZHANG, WANYI CHEN (2019)

Ru Zhang, Tong Cao (2018)

- Objective: AdaBoost to classify Shanghai and Shenzhen 300 stocks
- 68 individual Factors which can be broadly classified as follows:

Valuation	Growth	Financial quality	Gearing
Market Value	Volatility	Share Price	Beta
Turnover	Mood	Market Value	Technology

- Training: Top 30% as strong stocks, Bottom 30% as weak and middle 40% ignored
- Result: Top decile outperforms HS300 while bottom decile underperforms HS300

Zhang and Cao (2018). Multi-factor Stock Selection Model Based on Adaboost. Macrothink Institute, vol. 8(4), pages 119-129

Sun Yutong, Hanqing Zhao (2015)

- Objective: Advanced AdaBoost to classify Shanghai A-share Stocks (excluding suspended stocks)
- Modification: Cost assigned for misclassifying bad stocks as good stocks is higher than that of misclassifying good stocks as bad
- Data Preprocessing:
  - Winsorization to nullify the extreme oscillations
  - Missing values replaced with average values
- Training: Top 50% as strong stock, Bottom 50% as weak
- Result: Better accuracy rate and recall rate as compared to basic Adaboost Algorithm

Yutong and Zhao (2015). Stock Selection Model Based on Advanced AdaBoost Algorithm. 7th International Conference on Modelling, (ICMIC 2015)

Xiaoyun Zhang, Wanyi Chen (2019)

- Objective: XGBoost to classify Shanghai and Shenzhen 300 stocks
- Preprocessing the data before every iteration in order to take into account the newly added data which can change the distribution
- Data Preprocessing:
    - Normalization to eliminate the influence of different indicators dimensions
    - Market value and industry neutralization to prevent concentrated selections
- Results (Advantages over SVM and LR):
    - Superior in the prediction accuracy AUC of the model
    - Highest cumulative returns in back-testing period
    - XGBoost can conveniently count the contribution of each factor to the model, which helps to reduce the workload of investors in indicators selection

Zhang and Chen (2019). Stock Selection Based on Extreme Gradient Boosting. IEEE. 10.23919/2019.8865781

Classification algorithms

- DECISION TREES
- RANDOM FOREST
- ADABOOST CLASSIFIER
- GRADIENT BOOSTING

BerkeleyHaas

ML Decision Tree algorithms

- Use ensemble method of combining weak classifiers
    - Idea 1: Large number of uncorrelated models will outperform individual models
    - Idea 2: Subset feature selection at each node split
- Two methods of model selection:
    - Bagging: random sampling from the dataset with replacement which results in different trees. e.g.: random forest
    - Boosting: sequential learners where subsequent model is created by increasing weight of samples which were previously misclassified. e.g.: adaboost, gradient boosting

Random forest classifier

- Algorithm:
    - Large number of single decision trees which give a prediction for each class
    - Class with most probability is the final output
    - Trees created using sampling with replacement
    - Additionally, each tree chooses different features for prediction
- Issue:
    - Often results in overfitting and has poor out sample performance
    - Also observed in our model for quantile classification (~35% accuracy)

AdaBoost classifier

- Sequential algorithm, combines weak learners, corrects for misclassification
    - Performs better than random forest (~39% accuracy)
- Algorithm:
    - Step 1: Initialize the weight of each data points
    - Step 2: From all the fitted classifiers for each dataset, select the one with lowest classification error
    - Step 3: Calculate each classifier weight, the misclassified data points are upweighted after each iteration
    - We can get the final prediction after adding the weighted sum of predictions after n iterations

Gradient Boosting

- Gradient Boosting is about gradient optimization of loss function rather than weighted voting in AdaBoost
    - Loss function measures how good are model's coefficients at fitting underlying data
    - Directly optimize the boosted model predictions
    - Gradients added to the running training process by fitting the next tree also to these values
    - Shows highest performance (~42% accuracy)
- Hyperparameters for convergence to global minima:
    - Learning rate: the "step size" with which we descend the gradient
    - Shrinkage: reduction of the learning rate

Methodology

- FACTORS
- DATA PROCESSING
- MODEL TRAINING
- PORTFOLIO CONSTRUCTION

Factors

- 30 factors have been used as explanatory variables
- Fundamental factors:
  - Activity: payables, receivables turnover, cash conversion cycle
  - Liquidity: cash ratio, quick ratio, current ratio
  - Valuation: P/E, P/S, P/CF, B/M
  - Leverage: D/E, DSCR, ISCR
- Market factors:
  - Style: Momentum, Size
- Fundamental + Market factors:
  - RoA, Volatility, RoE etc.
- Fama French & Carhart 4 factors:
  - Beta market, beta SMB, beta HML and beta UMD

Data Processing

- Latest list of stocks from S&P500
- Prices - Yahoo Finance; Fundamental Data – WRDS
- Interpolation techniques for missing data:
  - Linear: For each ticker & factor pair, values at both ends of missing range are linearly interpolated; maximum missing range <= 3 months.
  - Trend: For each ticker & factor pair, values are scaled proportional to industry median for that factor; maximum missing range > 3 months.
- Missing values after these techniques are replaced with industry median for that specific factor
- We have data for total ~450 stocks

Model Training

- Returns for each month are divided in 5 different quintiles:
    - {-2,-1,0,1,2}, ~90 stocks in each quintile
- Each training period consists of past 12 months of data. Out-of-sample prediction performed for the 13th month.
- Two different models:
    - No lag : fundamentals of last month are used to predict return
    - Lagged : 3-month lagged fundamentals are used to predict next month's return
- Rolling regression to capture short and mid-term signals between fundamentals and returns.

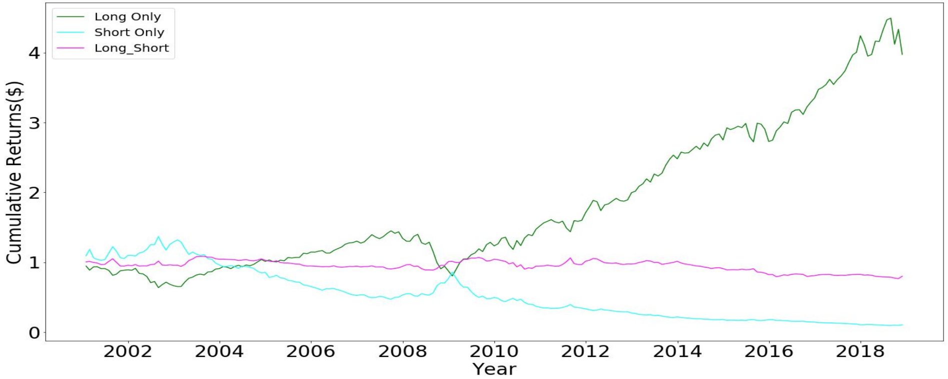
Portfolio Construction

- Filtering stocks:
  - No-rule: Long (Short) position in stocks classified as +2 (-2)
  - Probability: Long (Short) position in top 20 percentile stocks classified as +2 (-2)
- Weighting methods:
  - Long-Only: Equal-weighted & Market Cap-weighted
  - Long-Short: Equal-weighted & Dollar Neutral
- Transaction cost:
  - Reduced returns by abs(w_t - w_{t-1})*15bps (25bps) on long (short)

Results

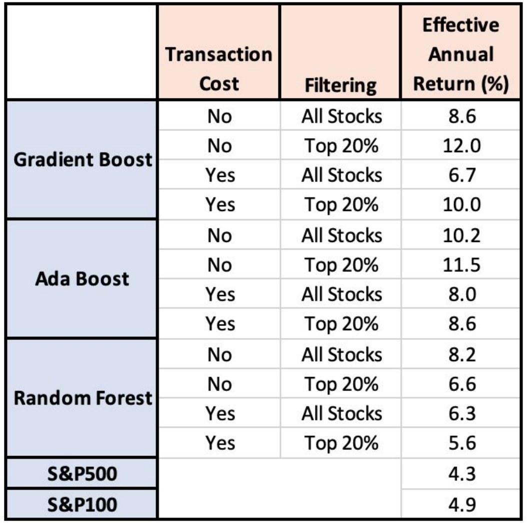
- COMPARISON OF ALGORITHMS
- LONG-ONLY PORTFOLIOS
- RESULTS: GRADIENT BOOSTING
- DRAWDOWN & ACCURACY
- FEATURE IMPORTANCE

Long-Only Portfolios



- Long-only portfolio consistently outperforms the long-short and short only portfolio.
- Stocks in -2 bucket in the training set are not guaranteed to have negative returns.
- Almost equal stocks in long & short bucket results in net reduced returns for long-short.
- Proposed ways to create long-short portfolios:
  - Change training data by tagging only negative returns as -2 (this will make split non-uniform).
  - Short -2 classified stocks if negative returns are ensured (by checking median of training data).

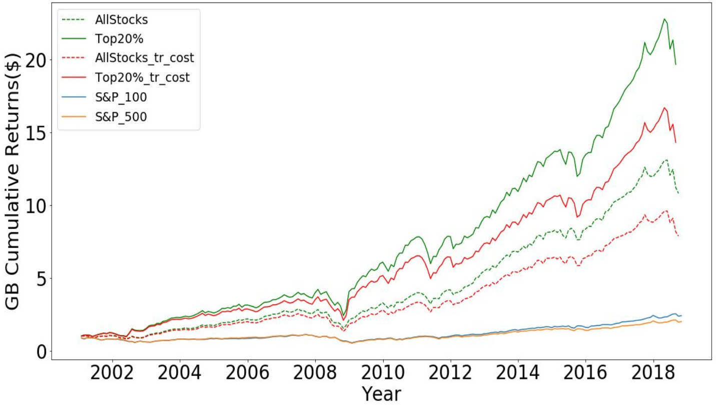
Comparison of Algorithms



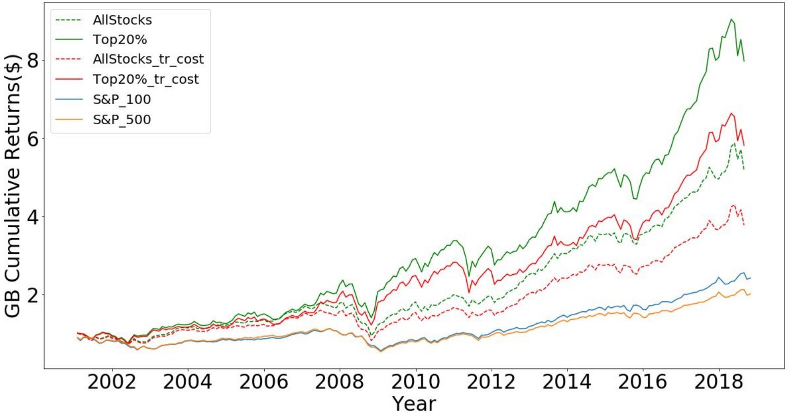
	Transaction Cost	Filtering	Effective Annual Return (%)
Gradient Boost	No	All Stocks	8.6
	No	Top 20%	12.0
	Yes	All Stocks	6.7
	Yes	Top 20%	10.0
Ada Boost	No	All Stocks	10.2
	No	Top 20%	11.5
	Yes	All Stocks	8.0
	Yes	Top 20%	8.6
Random Forest	No	All Stocks	8.2
	No	Top 20%	6.6
	Yes	All Stocks	6.3
	Yes	Top 20%	5.6
S&P500			4.3
S&P100			4.9

- Long-only portfolios using lagged factors (real-world) outperform benchmarks consistently
- Probability portfolios (top 20% stocks) outperform portfolios consisting all stocks
- Most promising returns characteristics are given by Boosting Algorithms

Results Gradient Boost



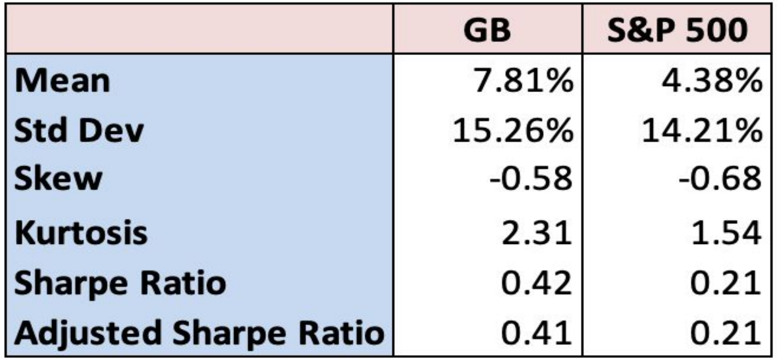
Equally weighted with lagged factors



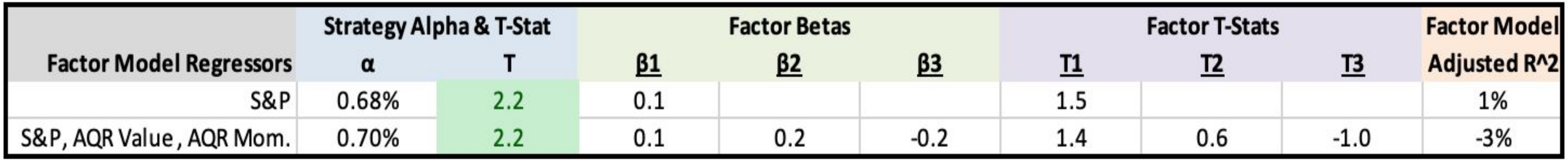
Market Cap with lagged factors

- We focus on the results of Gradient Boost now since it performs the best among all 3 algos
- All results are after transaction costs and using lagged fundamental data.
- Also, taking top 20% of predictions performs better than the portfolio with all predictions

Gradient Boost - No Rule port. (1/3)

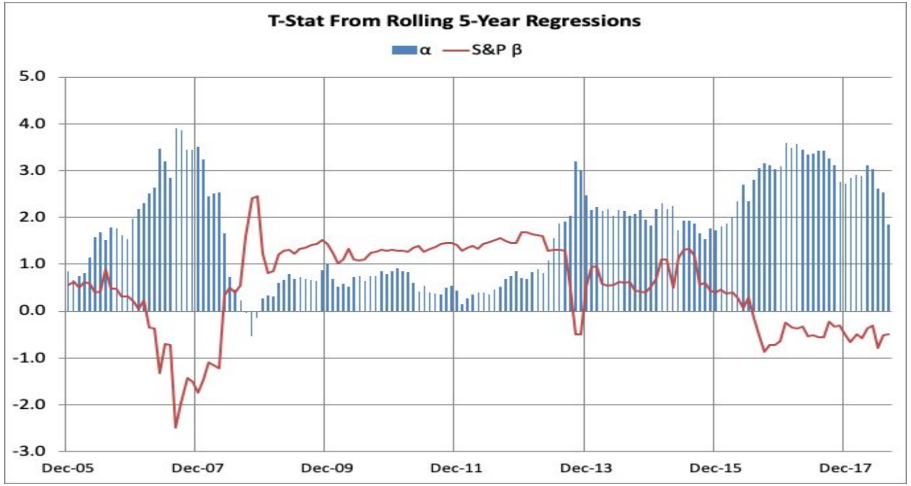


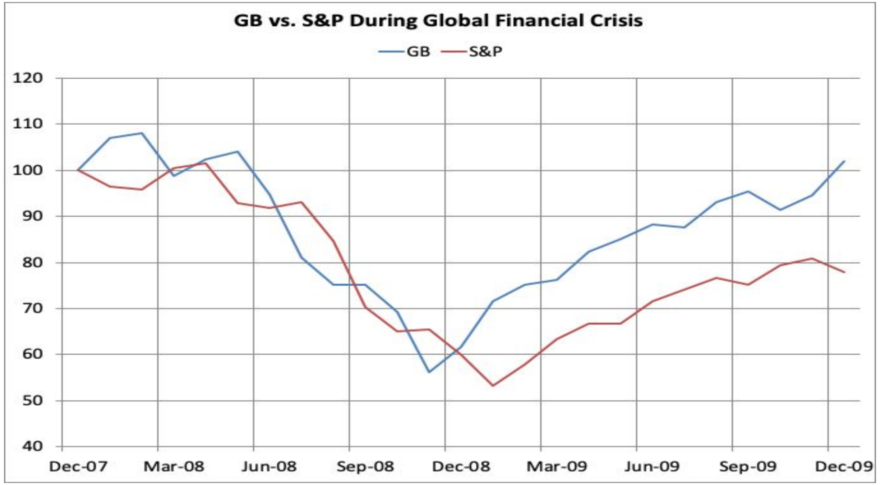
	GB	S&P 500
Mean	7.81%	4.38%
Std Dev	15.26%	14.21%
Skew	-0.58	-0.68
Kurtosis	2.31	1.54
Sharpe Ratio	0.42	0.21
Adjusted Sharpe Ratio	0.41	0.21



	Strategy Alpha & T-Stat		Factor Betas			Factor T-Stats			Factor Model
Factor Model Regressors	α	T	β1	β2	β3	T1	T2	T3	Adjusted R^2
S&P	0.68%	2.2	0.1			1.5			1%
S&P, AQR Value, AQR Mom.	0.70%	2.2	0.1	0.2	-0.2	1.4	0.6	-1.0	-3%

Gradient Boost - No Rule port. (2/3)



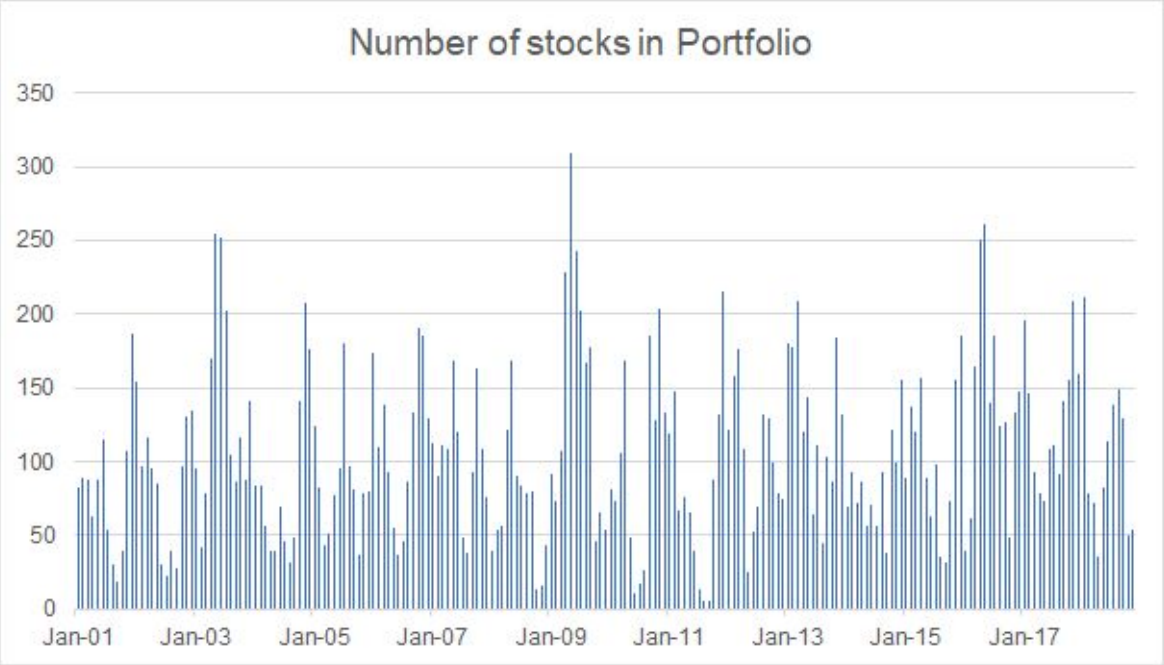


- Alpha is significant in most of the periods except the financial crisis
- Also, market beta is insignificant most of the time
- Beta rises during early phase of GFC, but portfolio recovers earlier and stronger than market.

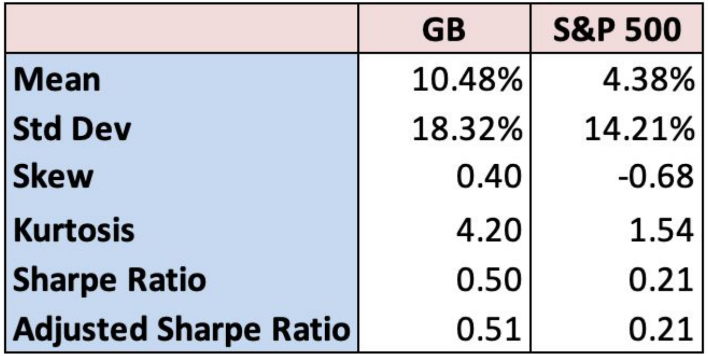
Gradient Boost - No Rule port. (3/3)

- Low realized beta driven by very high turnover.
- Example: 2011
  - September: 16 stocks
  - October: 141 stocks
- Average annual

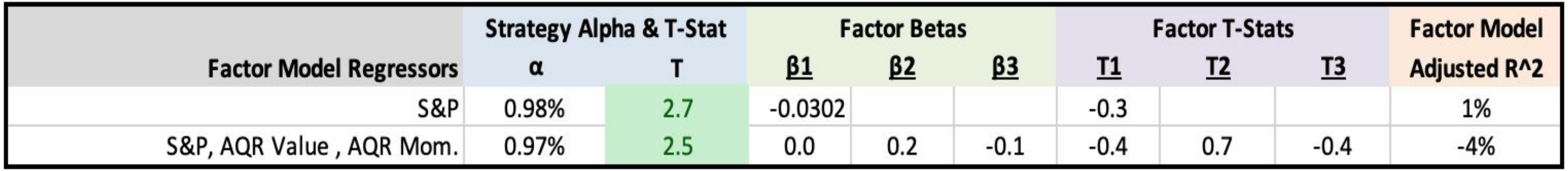
  turnover = 494 stocks
- Adjustment: Apply probability rule to reduce holdings.



Gradient Boost - Top 20% port. (1/3)

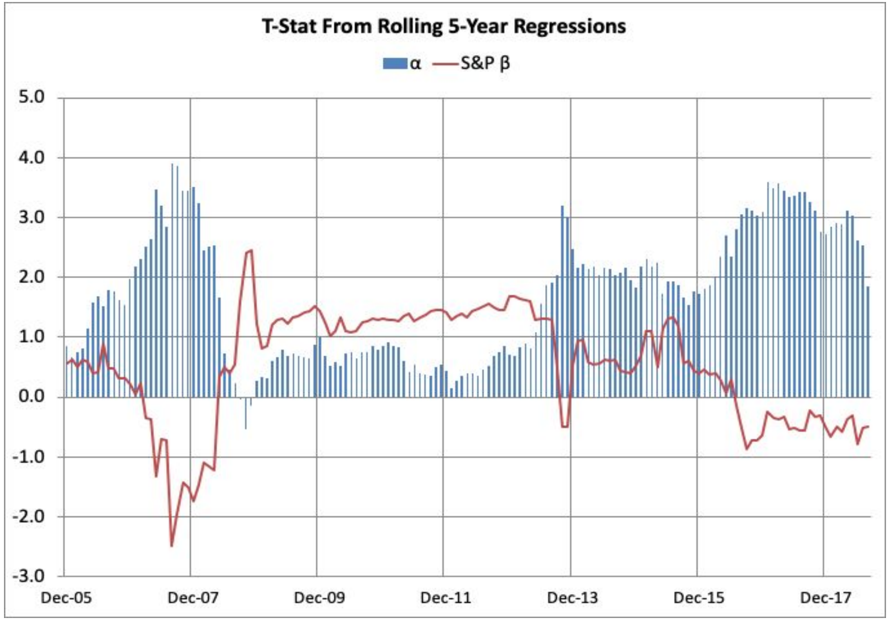


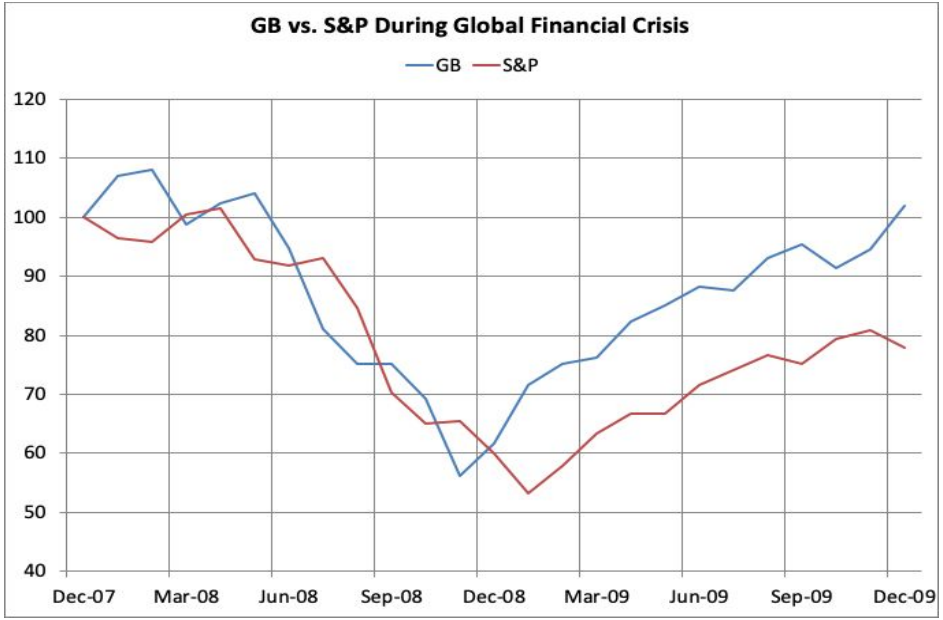
	GB	S&P 500
Mean	10.48%	4.38%
Std Dev	18.32%	14.21%
Skew	0.40	-0.68
Kurtosis	4.20	1.54
Sharpe Ratio	0.50	0.21
Adjusted Sharpe Ratio	0.51	0.21



	Strategy Alpha & T-Stat		Factor Betas			Factor T-Stats			Factor Model
Factor Model Regressors	α	T	β1	β2	β3	T1	T2	T3	Adjusted R^2
S&P	0.98%	2.7	-0.0302			-0.3			1%
S&P, AQR Value , AQR Mom.	0.97%	2.5	0.0	0.2	-0.1	-0.4	0.7	-0.4	-4%

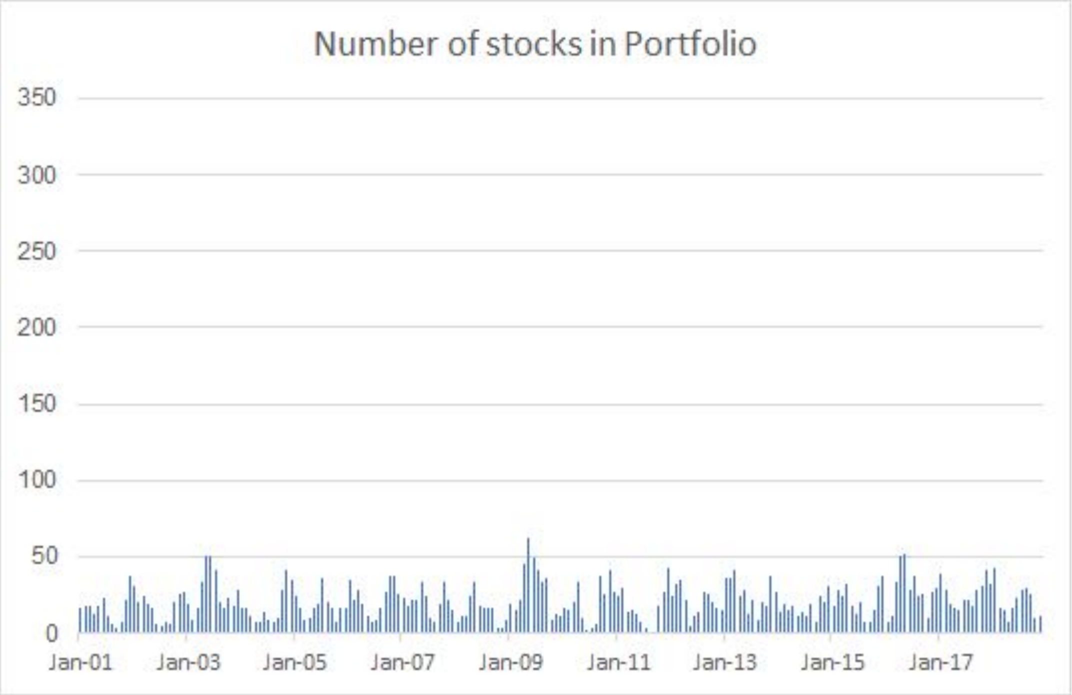
Gradient Boost - Top 20% port. (2/3)





Gradient Boost - Top 20% port. (3/3)

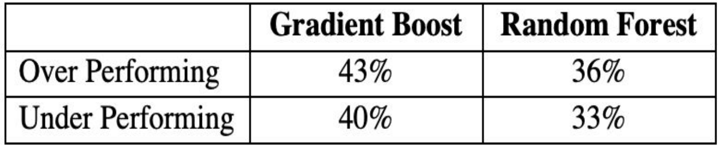
- Probability filter reduces total number stocks and thus total turnover.
- However, # of holdings still varies considerably.
- Example: 2011
  - September: only 1 stock
  - Three months later: 40+ stocks
- Average annual turnover = 98 stocks



BerkeleyHaas

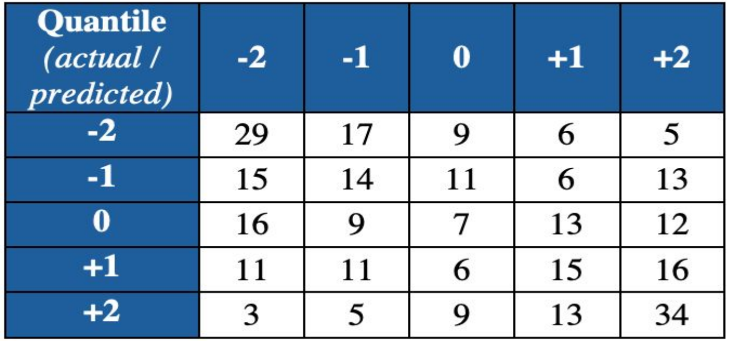
Accuracy

Definition: Number of true +2 stocks divided by total number of stocks predicted as +2.



	Gradient Boost	Random Forest
Over Performing	43%	36%
Under Performing	40%	33%

Average Accuracy of portfolios



Quantile *(actual / predicted)*	-2	-1	0	+1	+2
-2	29	17	9	6	5
-1	15	14	11	6	13
0	16	9	7	13	12
+1	11	11	6	15	16
+2	3	5	9	13	34

Sample confusion matrix giving an accuracy of 34/80 = 42.5%

- Random Forest performs the worst among all 3 algorithms due to overfitting issues.
- Gradient Boost consistently predict over-performing stocks correctly which leads to considerably higher cumulative returns over the period.
- Due to low out-of-sample accuracy, Random Forest predicts lots of underperforming stocks as overperforming which also reduces monthly returns and effect is compounded.

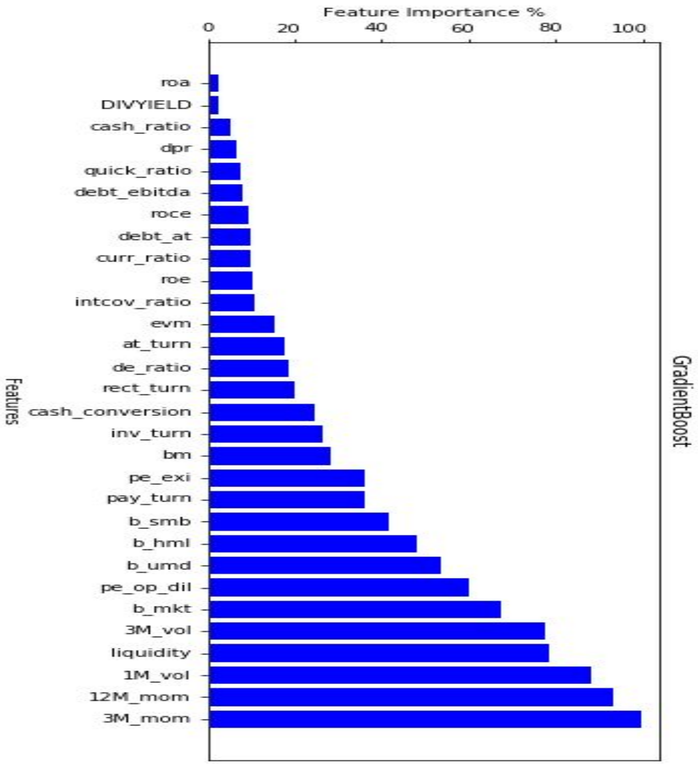
Maximum Drawdown

Algorithm	Portfolio	Max Drawdown
Gradient Boost	**All Stocks**	47%
	Top 20%	46%
Benchmarks	**S&P500**	53%
	S&P100	50%

Average Maximum drawdown of Market cap portfolios

- Even though our strategy is a concentrated and aggressive growth strategy, it’s drawdown are somewhat less than both S&P indices.

Feature Importance



- 12-month and 3-month Momentum are the 2 most important features (for Random Forest & AdaBoost as well).
- Boosting of incorrect classifications gives more importance to other factors such as liquidity, Volatility, Market Beta and momentum, P/E etc. which indicate aggressive growth strategies
- This indicates the overall strategy to be “Aggressive Growth” type.

Conclusions

- A ML based classification model can be a viable alternative to traditional regression-based stock selection models.
- Boosting algorithms perform considerably better than bagging algorithms in the context of stock selection.
- The proposed strategy is an aggressive growth strategy with high turnover.
- It produces higher risk the benchmarks but delivers higher risk-adjusted returns net of transactions costs as well.
- Low beta and high returns suggests strategy could serve as an alternative to hedge funds.

Future Work

- Combining the current model with a hedging model, it would be trained on worst performing months over last 10 years, to improve the performance during downturns and reduce drawdowns.
- Efficiently creating a long-short portfolio to reduce the volatility of the strategy.
- Examine ways to reduce portfolio turnover.

BerkeleyHaas

THANK YOU!