
UCLA x Optimas Applied Business Project

Machine learning to identify orthogonal alphas in Chinese stock market

Project Coach:

Luke Lu

Group Members:

Anshruta Thakur Xinyi Liu

Jiaxin Chen Yi Lu

Project Goal

- Predict CSI 800 stock returns through machine learning models and build trading strategies.
- Identify orthogonal alphas with asig-fsig strategy.

Contents

- Data Preprocessing
- Random Forest Model
- XGBoost Model
- Model Comparison and Conclusion

Data Preprocessing

Data Preprocessing and Dimensionality Reduction

Original Dataset

- CSI 800 stock information (2008-01-31 to 2022-12-30)
- Company information, 2043 features, and current monthly returns

Data Preprocessing Steps

- Drop features with more than 70% missing data
- Standardize the data ~ ensure uniformity and facilitate comparison across different features
- Impute missing values ~ with the average values of each feature

Feature Categorization

- Group similar features into categories for easier analysis. Example categories include market growth metrics, free cash flow (FCF) metrics, etc.

Data Preprocessing Steps

Principal Component Analysis (PCA):

- Conduct PCA *within each category* to reduce dimensionality
- PCA with 5 and 10 components to explore different levels of dimensionality reduction

Final Datasets:

- **Standardized Features:** Original dataset after preprocessing steps.
- **PCA (5 Components):** Reduced-dimensional datasets *within each category*, capturing major variation using *five principal components*.
- **PCA (10 Components):** A more detailed representation of reduced-dimensional datasets *within each category* by incorporating *ten principal components*.

Next Steps: Predictive Modeling and Strategy Evaluation

1. Feature Filtering with IC Scores:

- Utilize IC scores to filter relevant features

2. Predictive Models:

- Test two models: Random Forest and XGBoost

3. Strategy Backtesting & Evaluate Alpha:

- Backtest trading strategies; Calculate alpha to measure the strategy's excess return compared to the benchmark (CSI 300)

4. Conclusion:

- Determine model and strategy providing orthogonal alphas

Random Forest

Random Forest - Contents

- Brief introduction to random forest
- Data preprocessing & Feature selection
- Build the model
- Backtesting
- Investigate alphas
- Comparison and conclusion

Brief introduction to random forest

- Introduced by Leo Breiman (2001)
 - Breiman, L. (2001). Random forests. *Machine learning*, 45, 5-32.
- Ensemble method, Extension of bagging
- Combines the opinions of many “trees” (individual models) to make better predictions, creating a more robust and accurate overall model
- Good at dealing with a large number of features

Random Forests

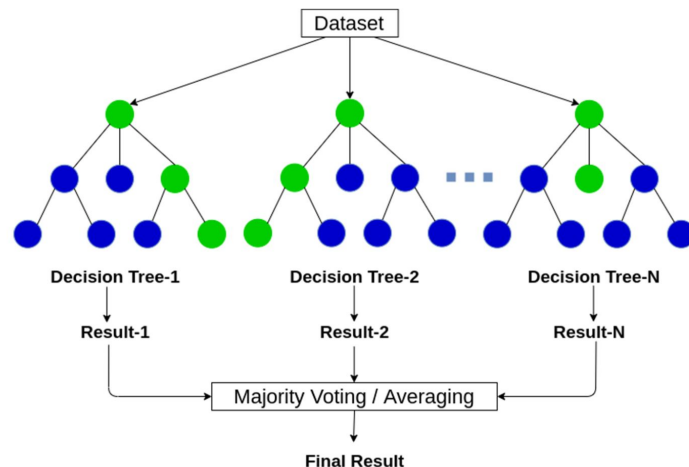
LEO BREIMAN

Statistics Department, University of California, Berkeley, CA 94720

Editor: Robert E. Schapire

Abstract. Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges a.s. to a limit as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. Using a random selection of features to split each node yields error rates that compare favorably to Adaboost (Y. Freund & R. Schapire, *Machine Learning: Proceedings of the Thirteenth International conference*, **, 148–156), but are more robust with respect to noise. Internal estimates monitor error, strength, and correlation and these are used to show the response to increasing the number of features used in the splitting. Internal estimates are also used to measure variable importance. These ideas are also applicable to regression.

Keywords: classification, regression, ensemble

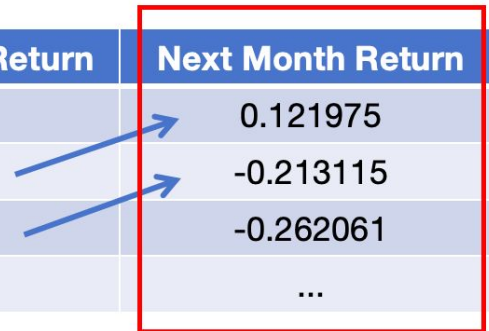


Data preprocessing & Feature selection

- **Data preprocessing**

- Dataset: CSI 800 stocks with monthly returns
- Time period: 2008-01-31 to 2022-11-30
- PCA processing (Mentioned above)
- Replacing the NA values with mean values
- Shift next month returns to the previous month

Date	Symbol	Features	Current Month Return	Next Month Return
2009-01-23	000767.SZ	...	-0.162885	0.121975
2009-02-27	000767.SZ	...	0.121975	-0.213115
2009-03-31	000767.SZ	...	-0.213115	-0.262061
...



Data processing & Feature selection

- Feature selection

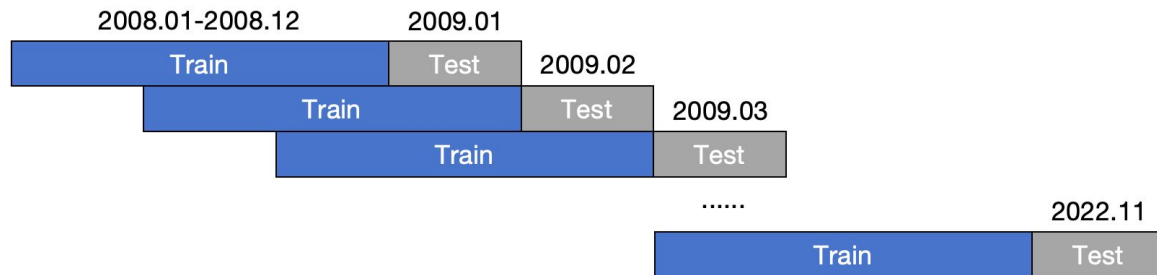
	Threshold	Original Number of Features	Selected Number of Features
Standardized Features Dataset	Mean absolute IC score > 0.1	1186	356
PCA 5 Components Dataset	Mean absolute IC score > 0.05	320	37
PCA 10 Components Dataset	Mean absolute IC score > 0.05	483	237

Build the model

- **Optimizing parameters:** Using RandomizedSearchCV()
- **The model:** sklearn.ensemble.RandomForestRegressor()
- **Parameters:**
 - n_estimator = 100
 - Max_depth = 3
 - Min_samples_split = 5
 - Min_samples_leaf = 2

Backtesting

- **Backtesting Period:** 2009-01-31 to 2022-11-30, monthly basis
- **Rolling window**



- **Layered backtesting**
 - Sort stocks into 5 layers by predicted returns (descending order).
 - The monthly return for each layer is the averaged actual returns of all stocks in that layer.
 - For each layer, observe performance metrics
 - e.g. Cumulative returns, Sharpe ratio, Max drawdown, etc.

Backtesting - Results

- **Standardized data - Cumulative returns**



- Layer 1: Stocks with highest returns
- Layer 5: Stocks with lowest returns

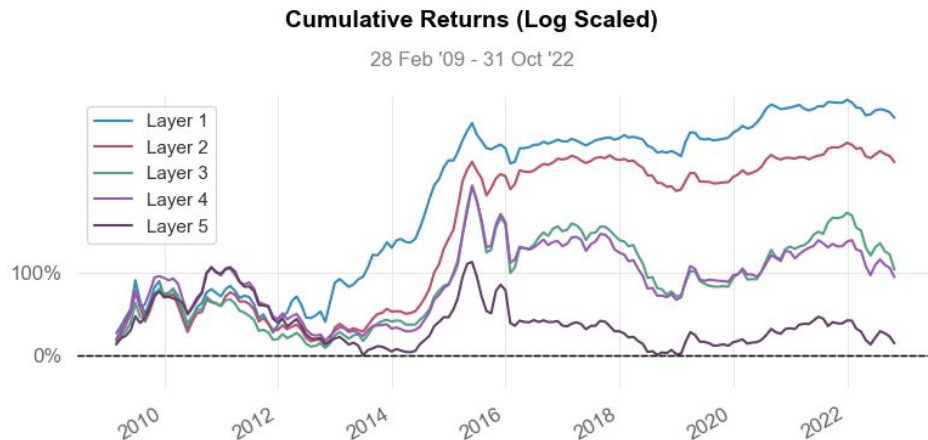
Backtesting - Results

- Standardized data - Performance metrics

	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Average Annual Return	7.2%	5.14%	6.59%	6.23%	3.06%
Cumulative Return	164.84%	101.62%	144.32%	133.12%	52.48%
Sharpe	0.39	0.32	0.38	0.37	0.25
Max Drawdown	-53.97%	-50.8%	-51.03%	-46.5%	-62.1%
Volatility (ann.)	27.4%	27.13%	25.47%	25.83%	26.9%

Backtesting - Results

- **PCA 5 components - Cumulative returns**



- Layer 1: Stocks with highest returns
- Layer 5: Stocks with lowest returns

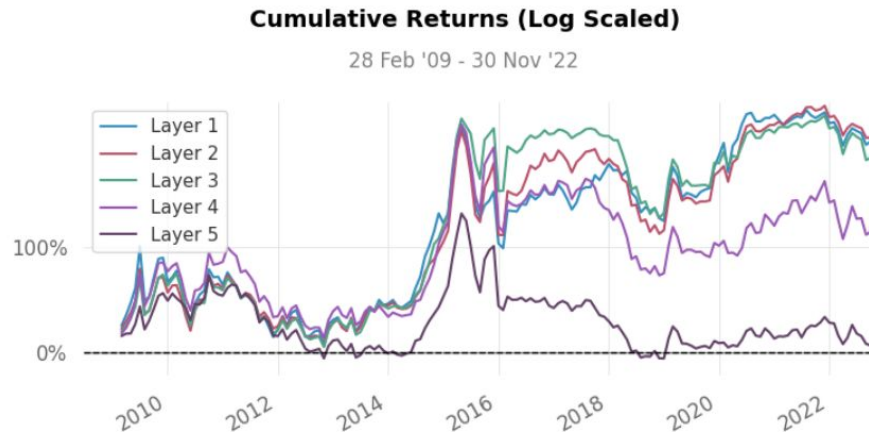
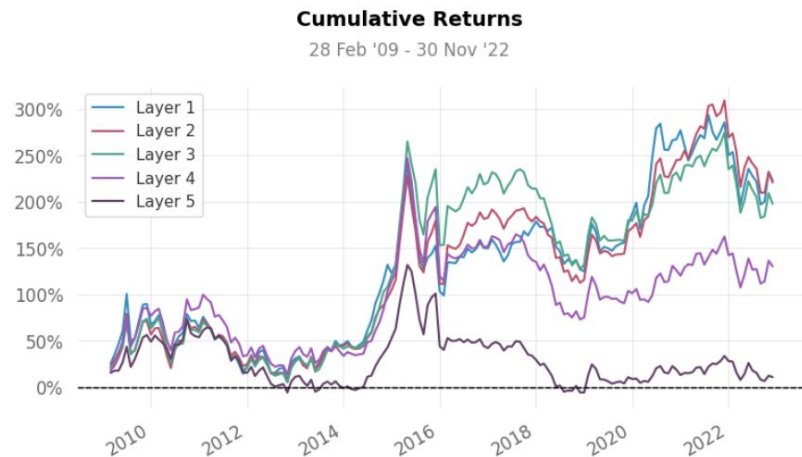
Backtesting - Results

- PCA 5 components - Performance metrics

	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Average Annual Return	14.98%	10.51%	5.19%	4.85%	0.95%
Cumulative Return	606.24%	305.14%	103.05%	94.07%	14.08%
Sharpe	0.73	0.6	0.36	0.35	0.14
Max Drawdown	-39.3%	-38.4%	-45.04%	-46.63%	-52.82%
Volatility (ann.)	23.08%	20.53%	19.22%	18.99%	18.74%

Backtesting - Results

- **PCA 10 components - Cumulative returns**



- Layer 1: Stocks with highest returns
- Layer 5: Stocks with lowest returns

Backtesting - Results

- PCA 10 components - Performance metrics

	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Average Annual Return	8.74%	8.68%	8.1%	6.14%	0.74%
Cumulative Return	223.09%	220.75%	197.66%	130.39%	10.9%
Sharpe	0.46	0.47	0.45	0.37	0.15
Max Drawdown	-46.81%	-40.99%	-39.83%	-50.13%	-59.33%
Volatility (ann.)	25.58%	24.4%	23.55%	23.86%	22.89%

Investigate Alphas

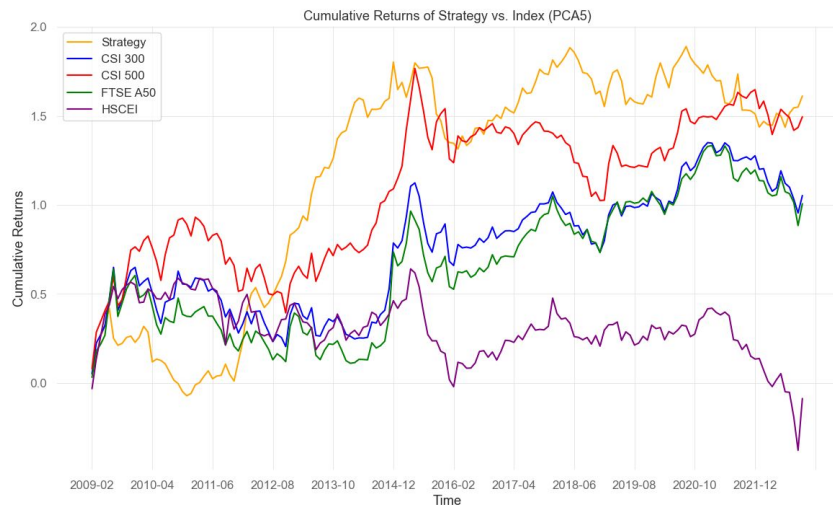
- **Long & Short strategy:** Long the top 20 stocks in Layer 1 and short the bottom 20 stocks in Layer 5.
- Compare the cumulative strategy returns with CSI 300/CSI 500/ FTSE A50/HSCEI index.
- Alpha is calculated with *CSI 300 as the benchmark*.
- **Alpha:**
 - CSI 800 Dataset
 - Standardized Features: 0.06
 - **PCA 5: 0.13**
 - PCA 10: 0.04
 - asig-fsig Dataset (Long top 20 stocks)
 - Alpha = -0.02

Pearson's Correlation Coefficient

- **Data to compare: *asig-fsig strategy***
 - Resample the data into monthly basis.
 - In each month, long the top 20 stocks and short CSI 300 to obtain returns.
 - Calculate the correlation coefficient between *returns from asig-fsig strategy* and alphas *from machine learning strategies*.
- **Results**
 - Standardized data vs. asig-fsig strategy: 0.072
 - PCA 5 components vs. asig-fsig strategy: -0.045
 - **PCA 10 components vs. asig-fsig strategy: -0.03**

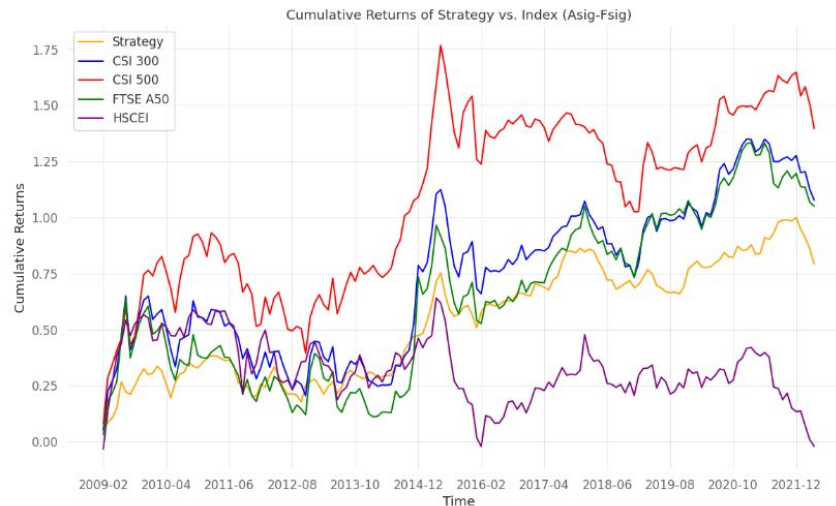
Strategy Returns vs. Index Returns

● PCA 5



- CAGR: 6.35%
- Sharpe: 0.5
- Max Drawdown: -43.38%

● Asig-Fsig



- CAGR: 3.42%
- Sharpe: 0.45
- Max drawdown: -22.57%

Conclusion on random forest model

	Number of Features Selected	Alpha	Correlation Coefficient with asig-fsig strategy
Standardized Features Dataset	356	0.06	0.072
PCA 5 Components Dataset	37	0.13	-0.045
PCA 10 Components Dataset	237	0.04	-0.03

- PCA 5 components data: Best performance based on strategy metrics.
- PCA 10 components data: Lowest Correlation (Orthogonal Alpha)
- Further Testing: Model 2 ~ XGBoost

XGBOOST

Content

- What is xgboost?
- Data preparation
- Build xgboost model
- Backtesting
- What do we learn?

What is xgboost?

- Original Paper: **XGBoost: A Scalable Tree Boosting System** (<https://arxiv.org/pdf/1603.02754v1.pdf>)
- We want to employ the XGBoost (“Extreme Gradient Boosting”) to build an efficient and scalable training of machine learning models to catch the return of stocks.

XGBoost: A Scalable Tree Boosting System

Tianqi Chen
University of Washington
tqchen@cs.washington.edu

Carlos Guestrin
University of Washington
guestrin@cs.washington.edu

ABSTRACT

Tree boosting is a highly effective and widely used machine learning method. In this paper, we describe a scalable end-to-end tree boosting system called XGBoost, which is used widely by data scientists to achieve state-of-the-art results on many machine learning challenges. We propose a novel sparsity-aware algorithm for sparse data and weighted quantile sketch for approximate tree learning. More importantly, we provide insights on cache access patterns, data compression and sharding to build a scalable tree boosting system. By combining these insights, XGBoost scales beyond billions of examples using far fewer resources than existing systems.

Keywords

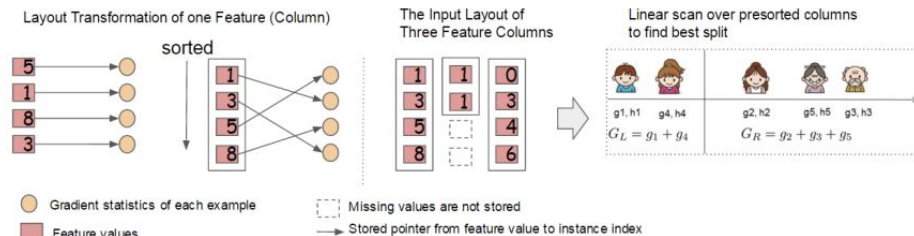
Large-scale Machine Learning

1. INTRODUCTION

Machine learning and data-driven approaches are becoming very important in many areas. Smart spam classifiers protect our email by learning from massive amounts of spam data and user feedback; advertising systems learn to match

problems. Besides being used as a stand-alone predictor, it is also incorporated into real-world production pipelines for ad click through rate prediction [15]. Finally, it is the default choice of ensemble method and is used in challenges such as the Netflix prize [3].

In this paper, we describe XGBoost, a scalable machine learning system for tree boosting. The system is available as an open source package². The impact of the system has been widely recognized in a number of machine learning and data mining challenges. Take the challenges hosted by the machine learning competition site Kaggle for example. Among the 29 challenge winning solutions³ published at Kaggle's blog during 2015, 17 solutions used XGBoost. Among these solutions, eight solely used XGBoost to train the model, while most others combined XGBoost with neural nets in ensembles. For comparison, the second most popular method, deep neural nets, was used in 11 solutions. The success of the system was also witnessed in KDDCup 2015, where XGBoost was used by every winning team in the top-10. Moreover, the winning teams reported that ensemble methods outperform a well-configured XGBoost by only a small amount [1].



Data processing & Feature selection

- Dataset: CSI 800 stocks with monthly returns
 - PCA 5 components dataset
 - Select features with mean absolute IC > 0.03
 - 314 features
 - PCA 10 components dataset
 - Select features with mean absolute IC > 0.03
 - 481 features
 - Standardized features dataset
 - Select features with mean absolute IC > 0.03
 - 1185 features

Data preparation

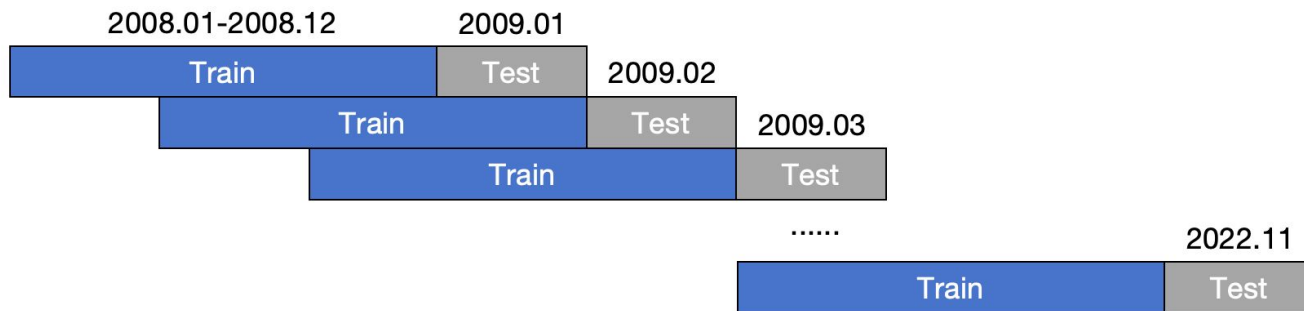
- Pick the dataset from 2008-01-31 to 2022-11-30
- PCA processing (Mentioned above)
- Replacing the Na values with the average value
- Replace the return with next month return
- Convert the dataset into DMatrix (xgb.DMatrix)

Date	Symbol	Next month Return
2009-01-23	000767.SZ	xxxxxx
2009-01-23	600058.SH	xxxxxx
.....		
2021-12-31	xxxxxxx	xxxxxxx

Date	Symbol	Return
2009-02-27	000767.SZ	xxxxxx
2009-02-27	600058.SH	xxxxxx

Build xgboost model

Rolling based: Train on the 12-month data, then predict out-of-sample monthly return



$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k \in \mathcal{F}$$

$$obj(\theta) = \sum_i^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k)$$

K = Number of trees
F = Sets of possible CARTs(space of regression tree)

`xgb.train(params, data, num_round)`

Model parameters

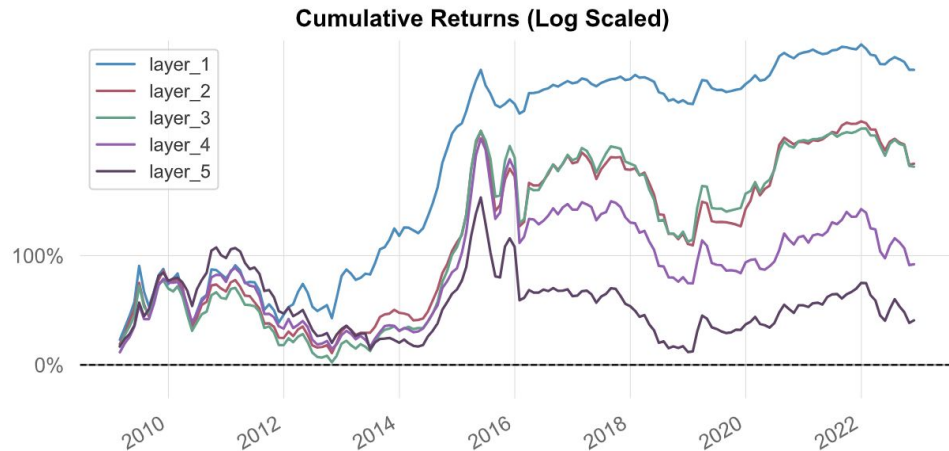
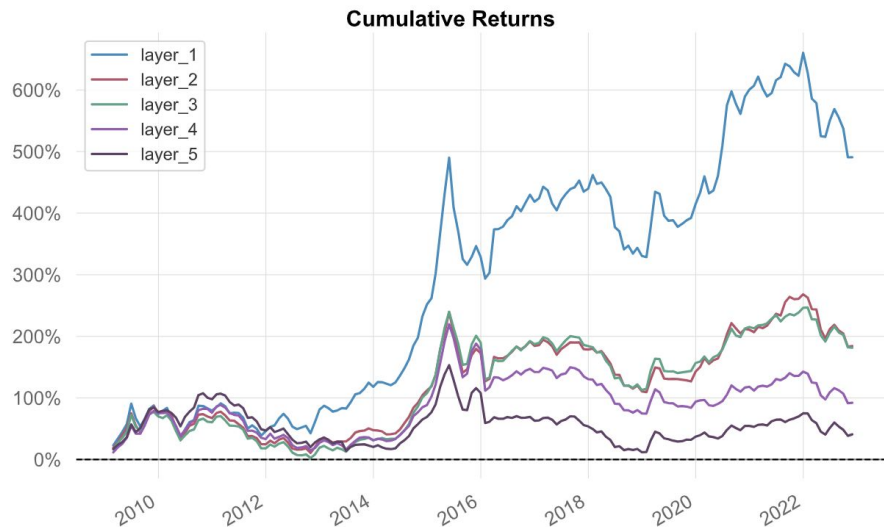
- Package: xgboost
- num_round = 100
- Max_depth = 3
- Learning_rate = 0.1
- Objective = reg:squarederror
- Eval_metric = rmse

Backtesting

- Steps for backtesting
 - Backtesting Period: *2009-02-28 to 2022-11-30*, monthly basis.
 - For each month, group the stocks according to predicted next month return into 5 layers.
 - Each layer contains 160 stocks.
 - Calculate monthly return for each layer by getting equal weighted return of the 160 stocks.
 - Plot cumulative returns, calculate different metrics to compare.

Backtesting - Results

- **PCA 5 components - Cumulative return**



- Layer 1: Stocks with highest predicted returns
- Layer 5: Stocks with lowest predicted returns

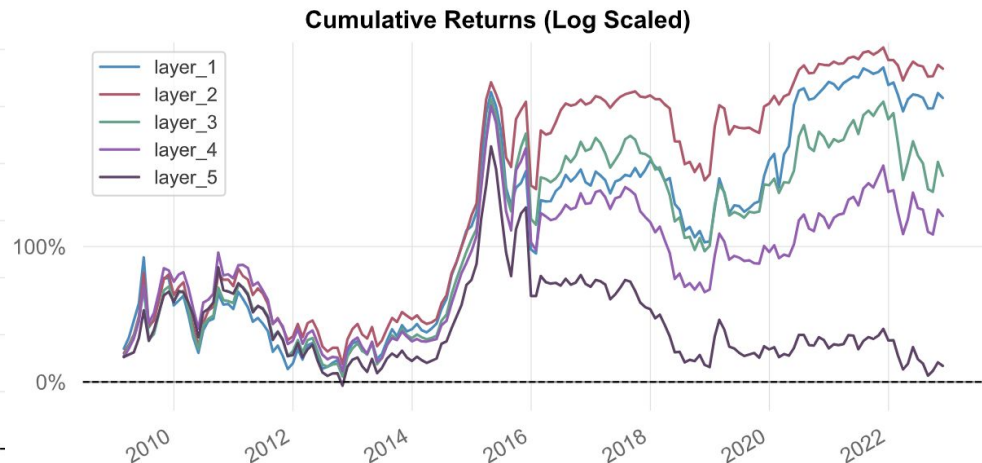
Backtesting - Results

- **PCA 5 components** - Performance metrics

	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Average Annual Return	17.37%	10.92%	10.83%	7.54%	4.96%
Cumulative Return	490.72%	184.24%	181.42%	92.08%	40.81%
Sharpe	0.71	0.48	0.48	0.34	0.22
Max Drawdown	-33.26%	-40.44%	-42.27%	-45.37%	-55.86%
Volatility (ann.)	21.15%	19.63%	19.55%	19.01%	19.15%

Backtesting - Results

- **PCA 10 components - Cumulative return**



- Layer 1: Stocks with highest predicted returns
- Layer 5: Stocks with lowest predicted returns

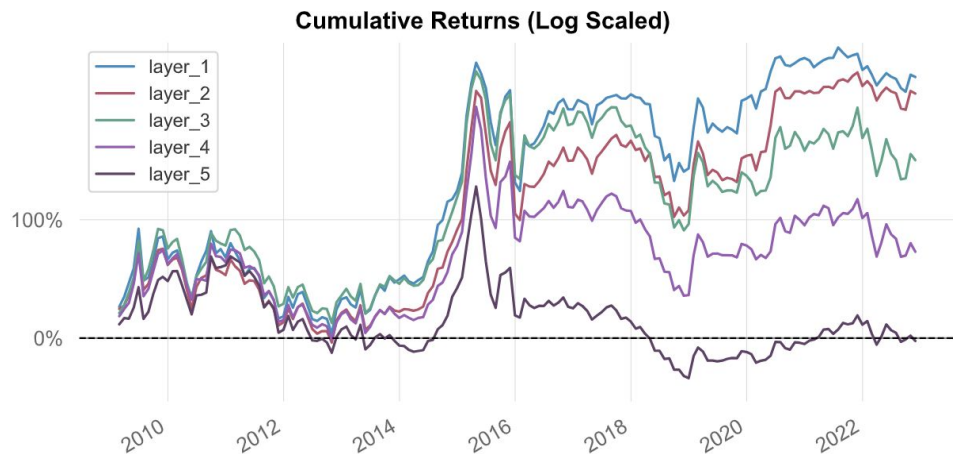
Backtesting - Results

- **PCA 10 components** - Performance metrics

	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Average Annual Return	13.52%	14.79%	10.84%	9.79%	4.19%
Cumulative Return	226.74%	298.67%	152.34%	122.61%	11.85%
Sharpe	0.47	0.54	0.4	0.36	0.15
Max Drawdown	-45.68%	-38.35%	-39.37%	-46.43%	-61.79%
Volatility (ann.)	25.08%	23.76%	23.23%	23.34%	23.78%

Backtesting - Results

- Standardized data - Cumulative return



- Layer 1: Stocks with highest predicted returns
- Layer 5: Stocks with lowest predicted returns

Backtesting - Results

- Standardized data - Performance metrics

	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Average Annual Return	15.05%	13.44%	11.41%	8.47%	3.94%
Cumulative Return	260.21%	217.68%	150.36%	73.03%	-2.36%
Sharpe	0.48	0.45	0.39	0.28	0.13
Max Drawdown	-46.05%	-45.99%	-49.28%	-54.03%	-70.97%
Volatility (ann.)	27.48%	25.69%	25.64%	26.11%	26.89%

Backtesting - Results

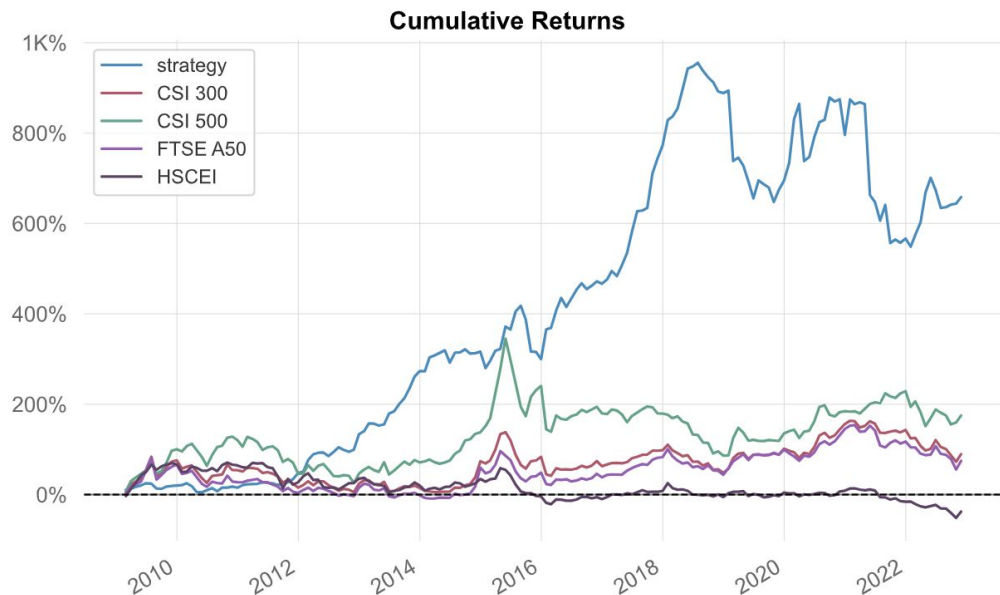
- We also use different IC thresholds to train the model.
 - Larger thresholds: less features, filter more correlated features

Layer 1	PCA5 IC>0.03	PCA5 IC>0.05	PCA10 IC>0.03	PCA10 IC>0.05	Standardized IC>0.03	Standardized IC>0.05
Number of features	314	37	481	237	1185	983
Average Annual Return	17.37%	18.49%	13.52%	15.07%	15.05%	15.05%
Cumulative Return	490.72%	557.47%	226.74%	261.23%	260.21%	261.23%
Sharpe	0.71	0.73	0.47	0.48	0.48	0.48
Max Drawdown	-33.26%	-34.59%	-45.68%	-45.78%	-46.05%	-45.98%
Volatility (ann.)	21.15%	22.0%	25.08%	27.4%	27.48%	27.4%

Backtesting - Strategy

- Long-Short Strategy:
 - Long top 20 stocks and short bottom 20 stocks for each month
 - Compare with CSI 300/CSI 500/ FTSE A50/HSCEI index
 - Calculate alpha compared with CSI 300 as benchmark

PCA5 IC>0.03	Strategy
Alpha	17.81%
Beta	-0.001696
Max drawdown	-38.55%
Volatility	18.68%



Backtesting - Strategy

- For different datasets and different IC thresholds, we compare the alpha for the strategy
 - Smaller IC threshold gives larger strategy alpha

	PCA5 IC>0.03	PCA5 IC>0.05	PCA10 IC>0.03	PCA10 IC>0.05	Standardized IC>0.03	Standardized IC>0.05
Number of features	314	37	481	237	1185	983
Strategy alpha	17.81%	17.36%	14.22%	11.78%	21.38%	14.06%

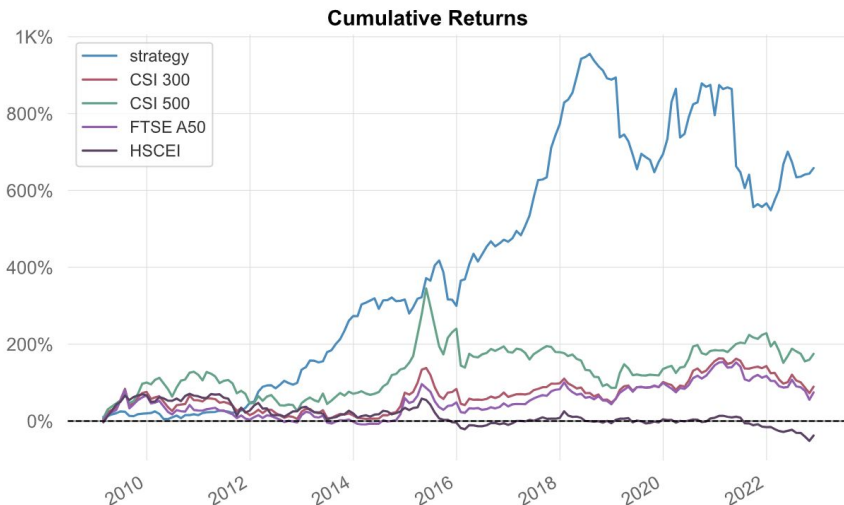
Comparison with another strategy

- Data to compare: **Asig-fsig Dataset**
 - In each month, long the top 20 stocks and short CSI 300 to obtain returns.
 - Calculate the returns from the asig-and-fsig strategy and compare with returns from our long-short strategy
- The *correlation* between our long-short strategy and the asig-and-fsig strategy is **-0.09594**.
- The correlation is small, the two strategies share little similarity

Comparison with another strategy

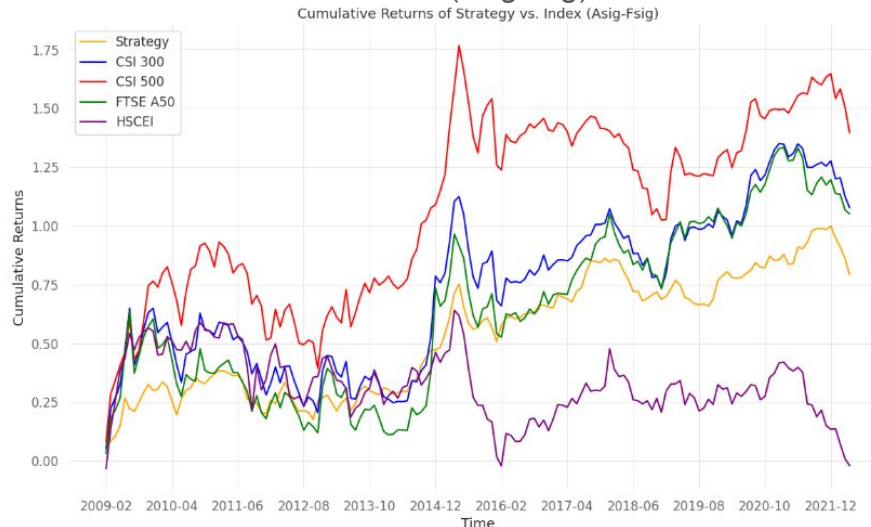
- **PCA 5 (0.03) ~ China A**

Cumulative return (pca5)



- CAGR: 8.04%
- Sharpe: 0.71
- Max drawdown: -38.55%

Cumulative return(asig-fsig)



- CAGR: 3.42%
- Sharpe: 0.45
- Max drawdown: -22.57%

Comparison and Conclusion

Comparison: Random Forest vs XGboost

	Layer 1 average annual return	Strategy average annual return	Strategy alpha	Maximum Drawdown	Correlation with Second Dataset	Running Time
Random Forest	14.98%	9.43%	13%	-39.3%	-0.045	Depends on the number of features
XGboost	17.37%	10.7%	17.81%	-33.26%	-0.0959	5-12 mins

Comparison: PCA dataset vs Standardized dataset

	Characteristics	Strategy performance
PCA components dataset	Further removes the linear correlation across features. Imputing mean for each category.	Better
Standardized dataset	Standardizing each feature in raw data without imputing mean in each category	Worse

Conclusion

- We used two machine learning models, random forest and xgboost, successfully identified orthogonal alphas in Chinese stock market
- We learned that
 - Data preprocessing is important, different methods give different results
 - When building the model, we should always delete the future information in training data, like *Close*, *Prev_Close* and *next month return*
 - Parameters in the model should be optimized and tuned
- Steps to Improve:
 - Rolling IC scores to filter out features
 - Optimize parameters based on a rolling time-period

Thank you!
