Forecasting Stock Returns via Supervised Learning

Machine Learning
Final Project
STAT 479

Samuel Ilic
Jonathan Santoso
Meenmo Kang

Contents

Motivation
Is the Most Popular Financial Theory Even True?

- Efficient Market Hypothesis (**EMH**)
    - Stock prices reflect all publicly available information and beating the market is **NOT possible**.
- Is the Stock Market is Completely Efficient?
    - **If Yes**: Any attempt at forecasting returns is pointless!
    - **If Not**: Money is being left on the table and a good algorithm stands to make billions!

Motivation
Market Isn't Completely Efficient (But it's CLOSE)

Can you beat the Market? **YES**

- >10,000 **Hedge Funds**
  - Only a handful have done it
- So How do they do it?
  - 100's of PhDs (Math, Stats, CS)
  - Make > 2 Terabytes of Daily Data
  - <u>**Machine Learning!**</u>

Intro
Forecasting IBM's Stock Price

- Want to: Forecast IBM's Daily Stock Returns
- **Goal**: Generate Positive Predictive Performance
  - Positive out-of-sample **R-squared**
- What's Considered Good?
  - > .025% → You can Start Your Own Hedge Fund
  - > 0.001% → Statistically Significant → EMH Not True
- Source: Gu, Kelly and Xiu (2018)
  - They work for one of the funds on the last page

Experiments
Data Collection

	Date	High	Low	Open	Close	Volume	Adj Close
0	1/2/2002	121.500000	119.800003	120.599999	121.500000	6862800	84.677422
1	1/3/2002	124.220001	120.250000	121.500000	123.660004	8621700	86.182800
2	1/4/2002	125.599999	123.980003	124.050003	125.599999	8405200	87.534859
3	1/7/2002	126.190002	123.699997	125.000000	124.050003	5939600	86.454575
4	1/8/2002	125.199997	123.730003	124.250000	124.699997	5311800	86.907600

Source: https://finance.yahoo.com/quote/IBM/history

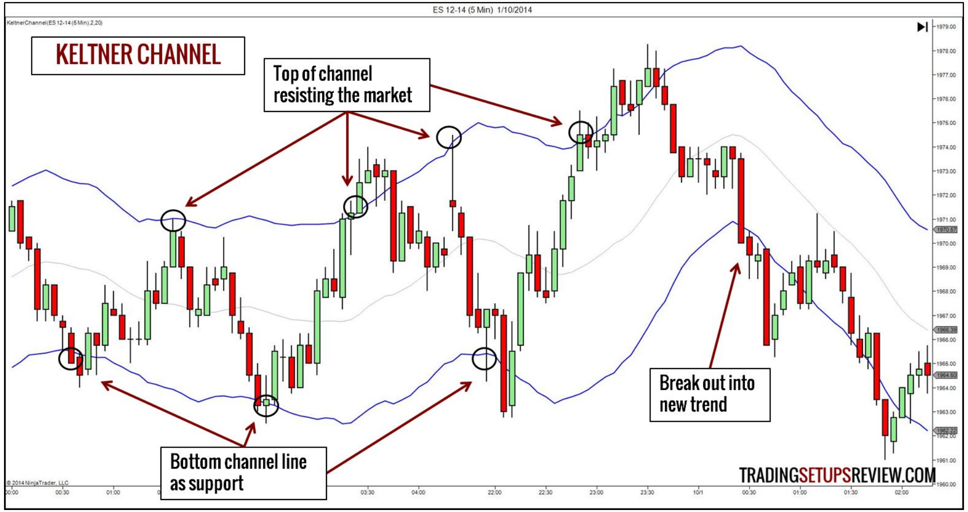
- Daily Prices from - 01/01/2002 --- 10/31/2018
- Calculated Daily Returns

$$R_t = \frac{Adj\ Close_t}{Adj\ Close_{t-1}} - 1$$

Experiments

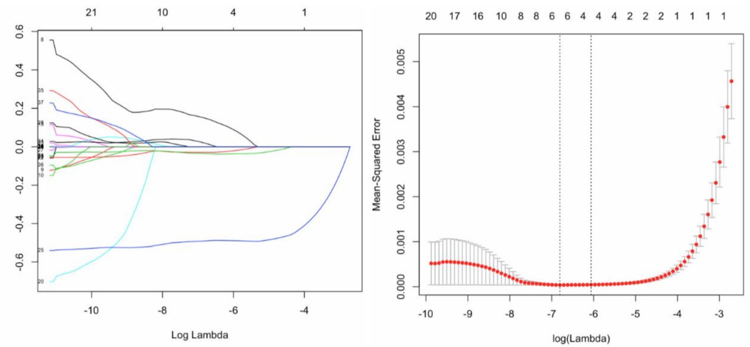
Exploratory Data Analysis

- Used ~20 Technical Indicators to create **Over 40 Features**
- Technical Indicator
  - Financial Jargon for a new feature calculated on existing data

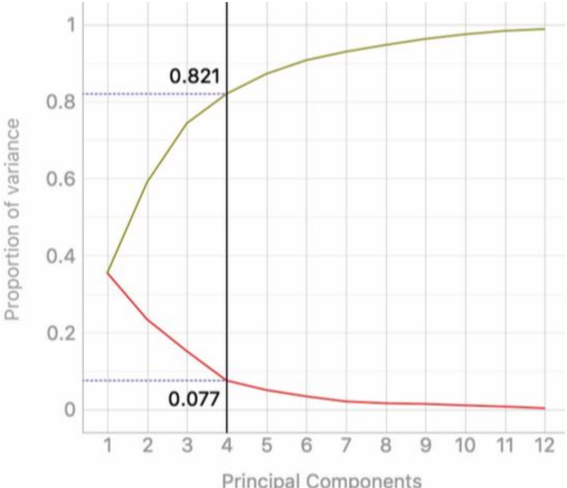


Experiments
Dimensionality Reduction (PCA & LASSO)

LASSO



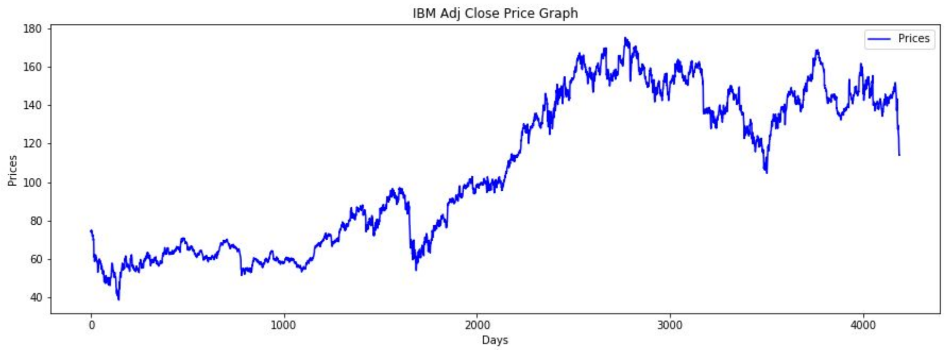
PCA

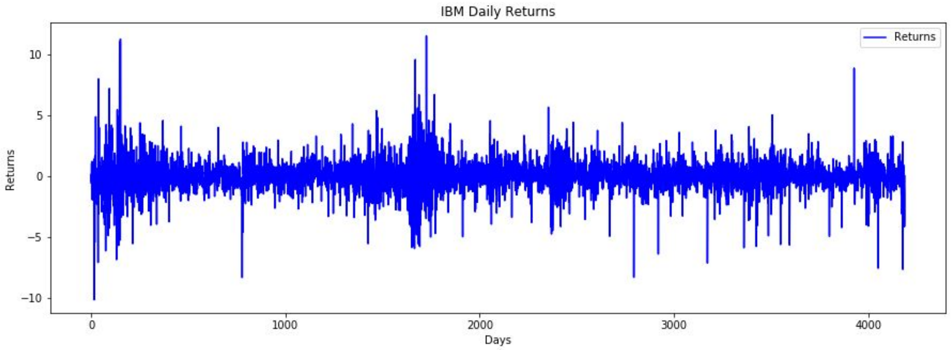


- Implemented **BOTH** LASSO & PCA
- Why?
  - Want to see if either can generate improved predictive performance over original dataset.

- Result?
  - Both Worse than Original Data?!?!
- Why?
  - Dimension reduction steps don't incorporate our ultimate objective of forecasting returns.
  - Hence, they **condense data prior to forecasting & pay no consideration to how the predictors are associated with future returns.**

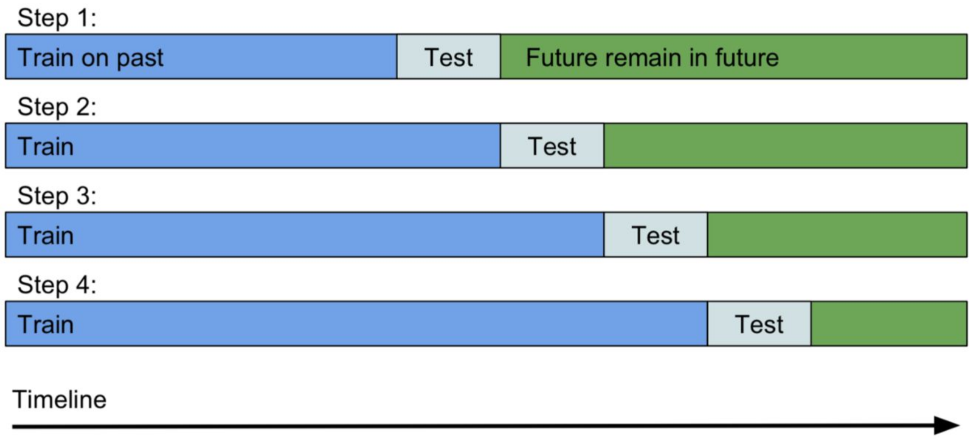
Experiments
Exploratory Data Analysis





- Adj Close price ranged from $38.59 to $175.26
- Daily returns ranged from -10.11% to 10.56%
- Average returns = 0.02%
- Stock returns are difficult to predict

Experiments
Standardization & Train-Test Split



$$x_{std}^{[i]} = \frac{x^{[i]} - \mu_x}{\sigma_x}$$

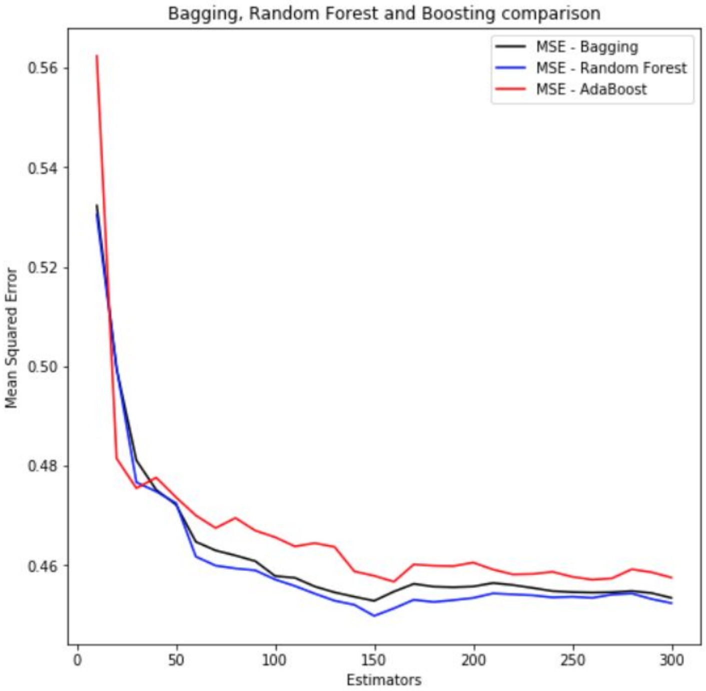
- **Standardization** → prevent dominance of one feature
- **TimeSeriesSplit** → prevent data leakage (look-ahead bias)
  - **Train/Validation Set** = Jan 2002 - Dec 2017
  - **Test Set** = Jan 2018 - Oct 2018
- # of Splits = 10 (due to limited computing power)

Experiments
Models and Hyper Parameters Tuning

- **Multivariate Regression** (Benchmark)
- Ensemble Models
    - **Random Forest**
    - **Bagging**
    - **AdaBoost**

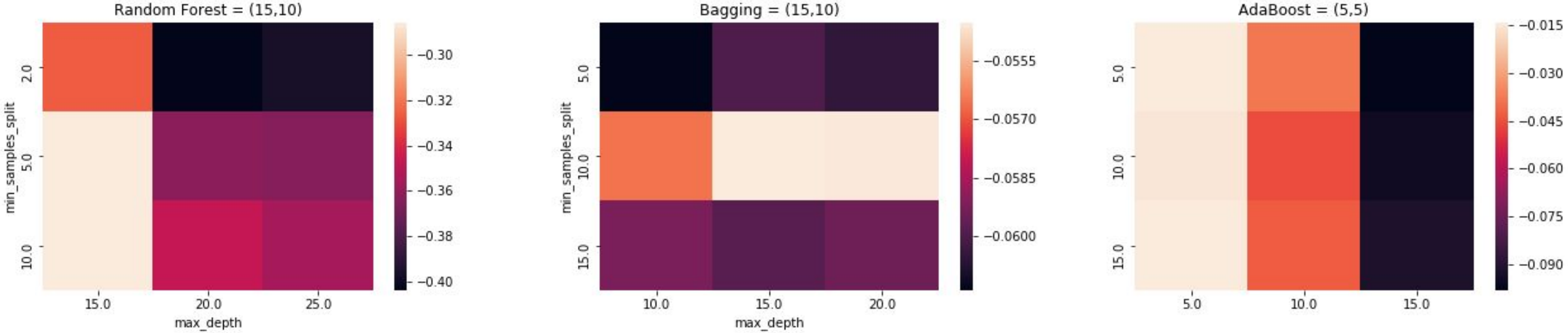
- Train Decision Tree Models on Each Fold
- Calculate average MSE and R^2 across all folds

- Why Decision Trees?
    - Gu, Kelly, Xiu (2018) → **DTs are the Best!**
        - (After Deep Learning)



Experiments

Hyperparameter Tuning on Validation Set

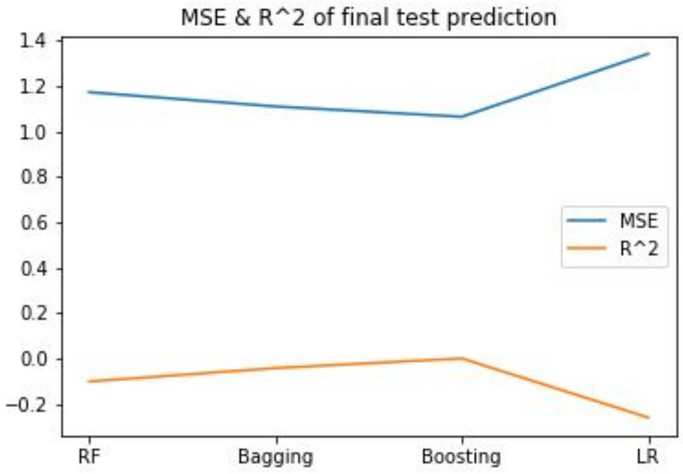


- R^2 Calculated across All 10 Folds of Time Series Split
- R^2 is Still Negative → We are Predicting **WORSE** than the a Horizontal Line!
- No Parameter Tuning for Multivariate Regression

Results
Model Evaluation on Test Set

- **Best Model: AdaBoost**
  - **Positive** R^2: **0.064%** (0.0064)
  - MSE: 1.06
- All Other Models
  - Negative R-squared

- Models Performed BETTER overall when trained on R-squared over MSE

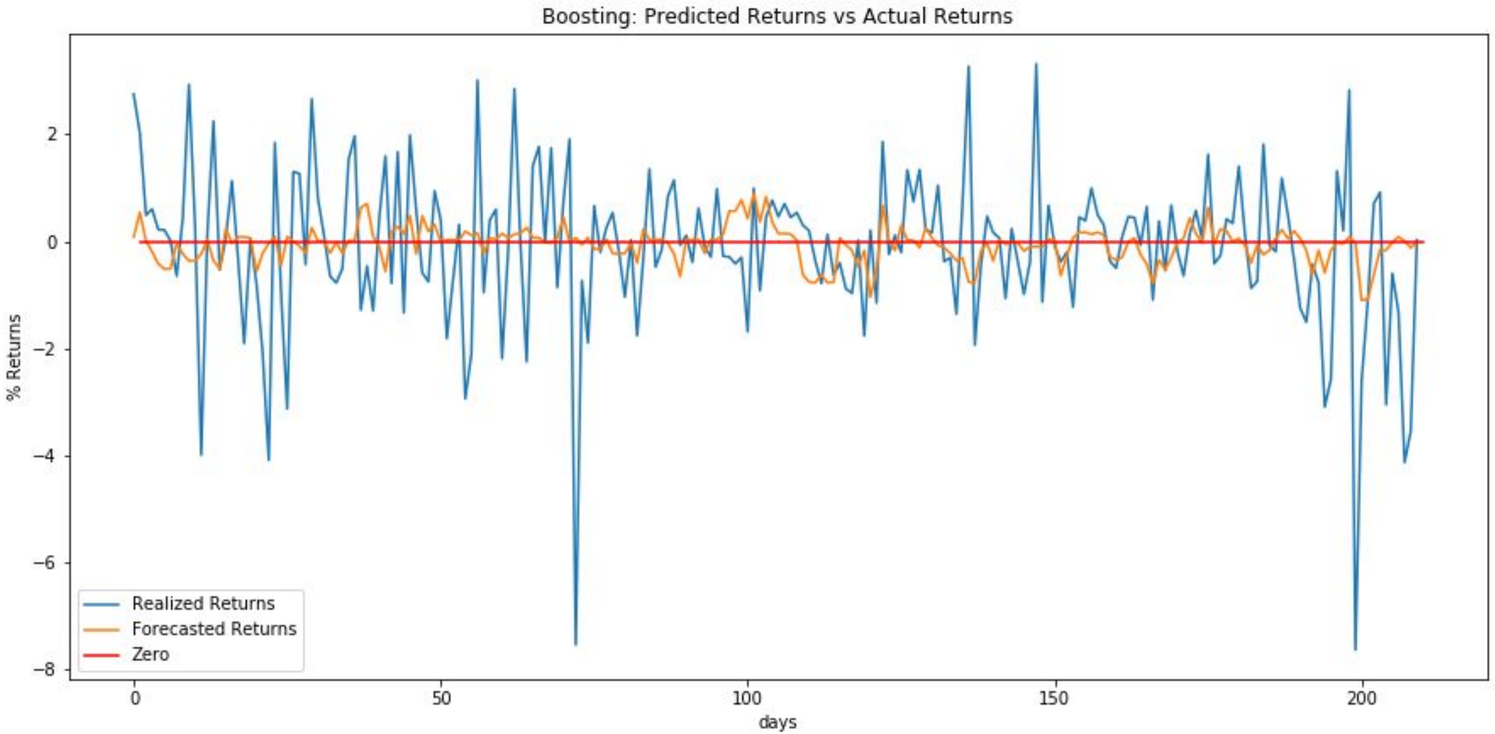


$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

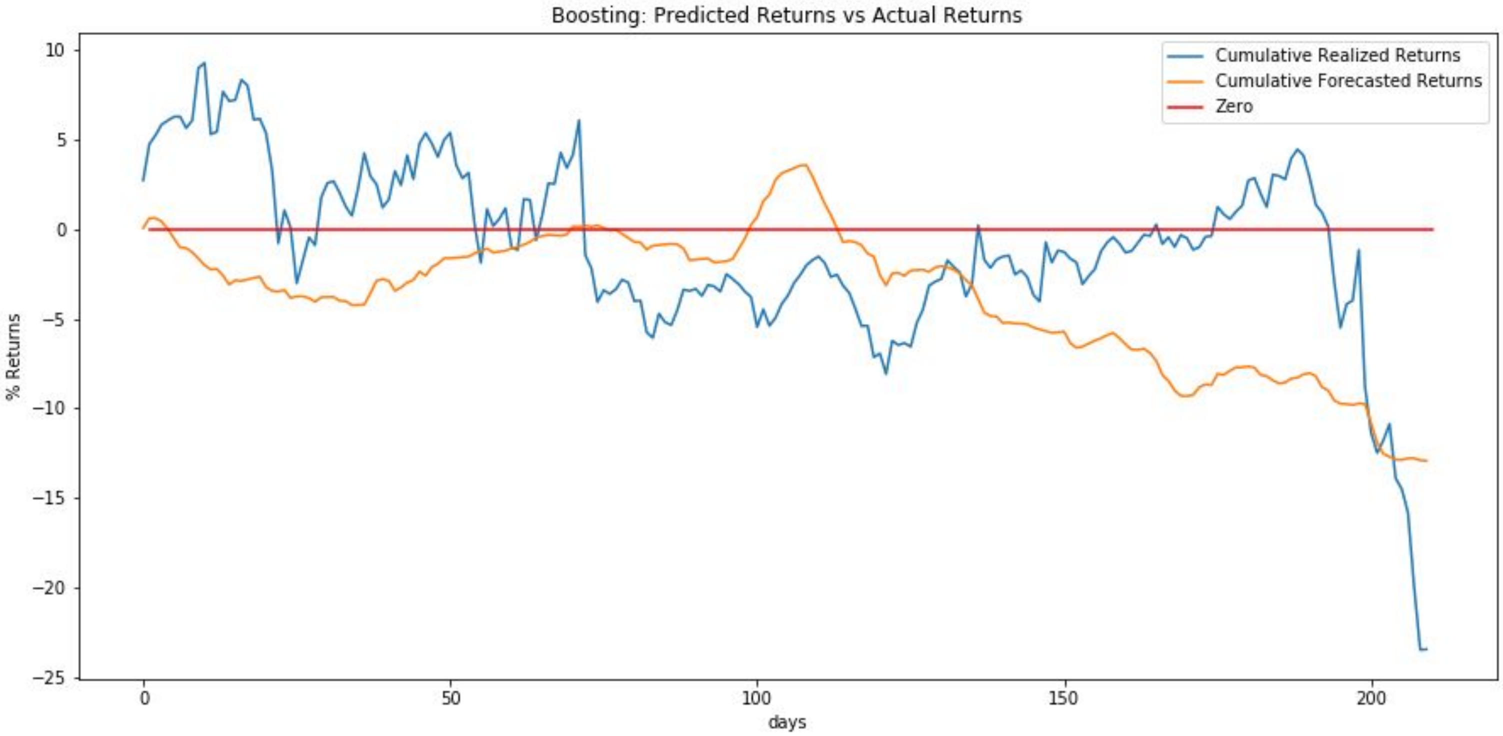
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Results

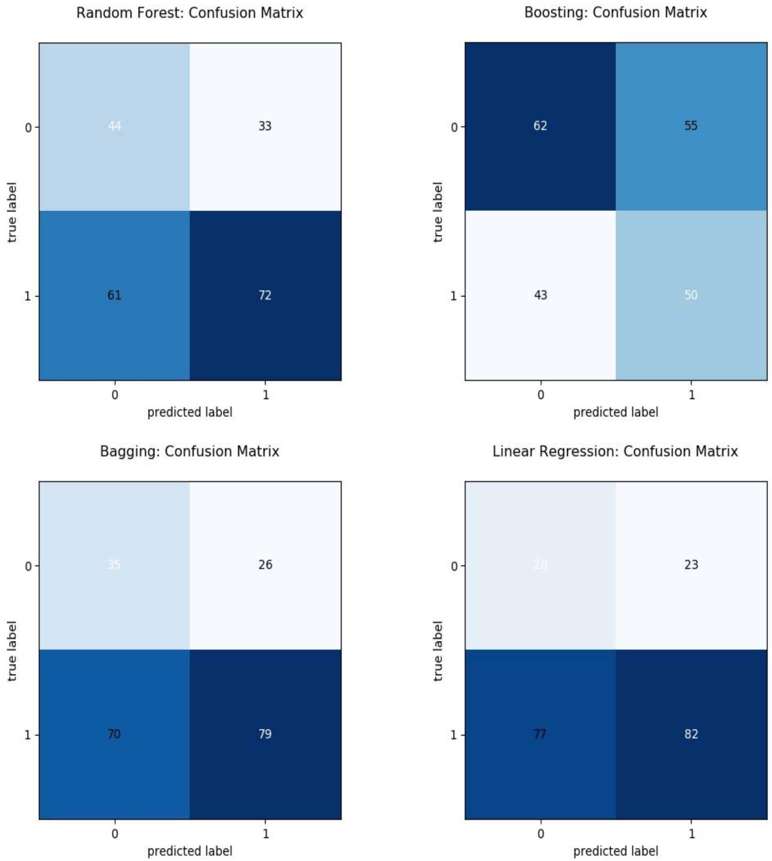
Forecasted Returns of Best Model (2018)



Results
Cumulative Returns of Best Model (2018)



Results
Binary Classification



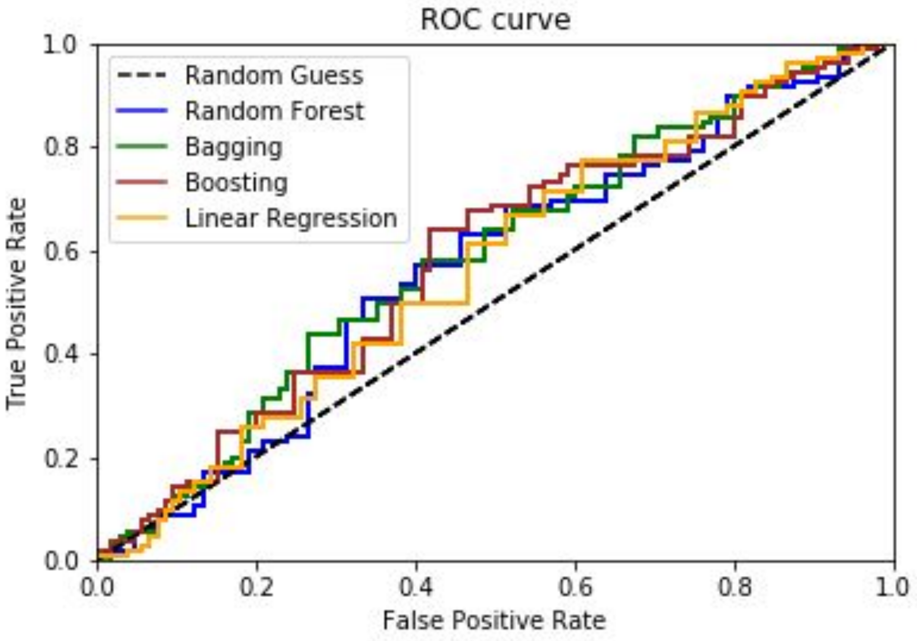
1 Price up
0 Price down

	Accuracy
Random Forest	**55.23%**
Bagging	54.29%
Boosting	53.33%
Linear Regression	52.38%

- 0 = tomorrow’s stock return < 0
- 1 = tomorrow’s stock return >= 0
- Utilized regression prediction for classification
- Bagging with optimal hyperparameter performed best

Results

Receiver Operating Characteristics (Area Under Curve)



	AUC
Bagging	**0.571**
Random Forest	0.552
Boosting	0.568
Linear Regression	0.549

- Scaled regression prediction using Min/Max scaler to obtain probabilities
- Bagging performs best under AUC metric

Conclusion

- Predicting Stock Returns **Difficult**!
- R-squared of .06% is **Positive**!
- IBMs Stock is **NOT** Completely Efficient

Questions?

Visit http://github.com/meenmo further information about our project