Columbia University

IEOR4742 – Deep Learning for OR & FE by Ali Hirsa

Final Project (Group 14)

Portfolio Management based on EPS Prediction using Ensemble Gradient Boosting Models

Authors contributing to this work:

Abhishek Mukherjee, Xueru Wang, Sie Hendrata

*All authors contributed equally to this work

Table of Contents

Abstract

Since Financial datasets exhibits certain non-linear relationships among them, machine learning and deep learning models are extensively used to model the underlying non-linear patterns in the datasets and hence can be deployed to solve various prediction and classification based problems to generate abnormal returns.

In our research we have used XGBoost model, a popular class of Ensemble Gradient Boosting Models to solve the multivariate regression problem of predicting a stock's future Earnings per Share by using stocks' preprocessed technical, fundamental and analyst's forecast data, as well as by constructing creative and well researched feature engineerined indicators out of those datasets. By using feature engineered datasets and doing various hyperparameter tuning of the model, we were able to establish good results in our research.

We have demonstrated with the help of plots that our predicted EPS performs well in the out-sample by capturing the direction and movement of the actual EPS and we suggested to use these predicted EPS to construct various trading strategies. We concluded the robustness of our model by showing that our predictions works well across the Financial as well as Technological sectors. We have found out in our research the feature dependency of various technical and fundamental factors for predicting Earnings per Share and got intuitively coherent results. We have also shown that using Feature Selection method by ranking features which are highly correlated for predicting EPS and using them in our model can improve the prediction's accuracy of the model.

Finally, we have used these predictions to construct simple alpha trading strategies and portfolios whose weights are allocated based on those alpha strategies to generate abnormal returns. With monthly trading and zero transaction costs, we have achieved a descent average Information Ratio of 6.14 with a particular trading strategy to support our prediction.

Our work can be extended by allowing to predict various fundamental factors other than just EPS to create more robust portoflios and alpha trading strategies.

Objective

Professional analysts play an important role in financial markets forecast, and financial institutions commonly develop algorithms to predict financial events for speculation and risk management. Here, we want to take a different perspective to predict some specific financial events. We do not ignore individual analysts' performance, rather, we combine the data and a learning model to uncover and correct for the mistakes of general analyst forecasts. Can deep learning be used to predict earnings surprises and EPS (earnings per share) better than general analyst forecasts? Is it possible to form a trading strategy based on EPS predictions to produce higher returns than benchmark? Above are the motivations that drive us to proceed.

In this project, we want to use stock's technical price volume, fundamentals, and analysts information to predict future earnings surprise and Earnings per Share. There are the three fundamentally important goals we have accomplished guided by proper metrics:

1) Predict an earnings surprise above certain percentage thresholds;
2) Predict the level of EPS on a quarterly basis;
3) Use the predicted EPS to construct various alpha strategies guiding the long/short of the specific stocks, thus setting up several portfolios in the financial and technology sectors of S&P 500. Also, we compare our portfolios performance to the benchmark.
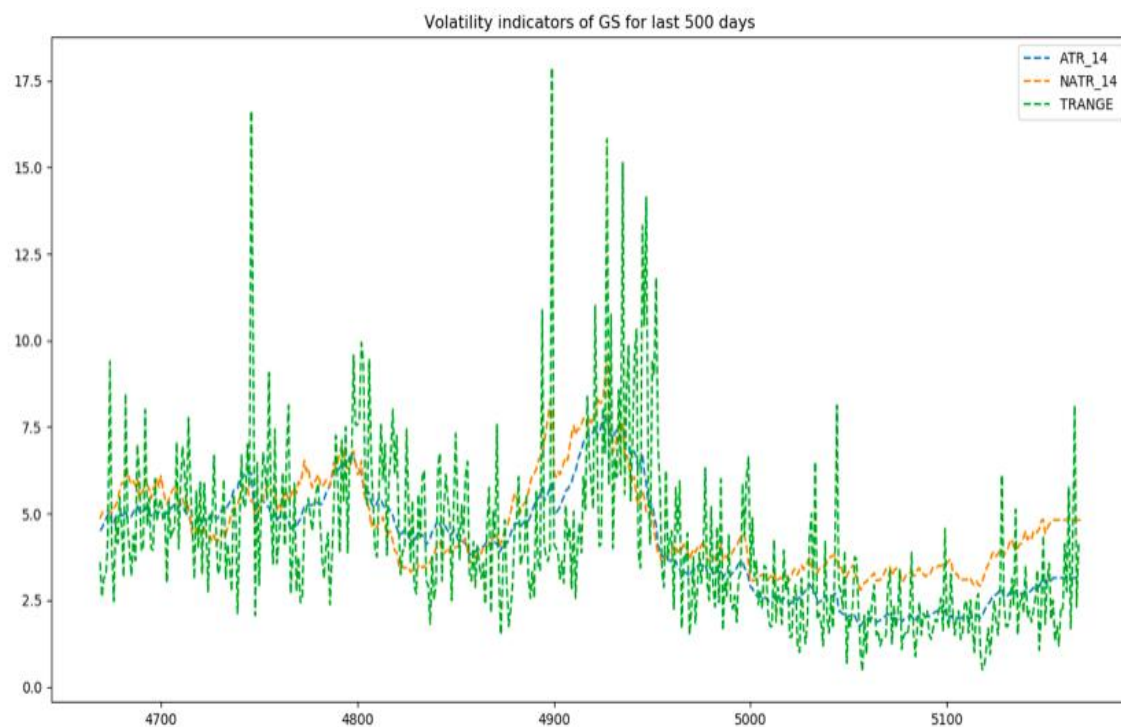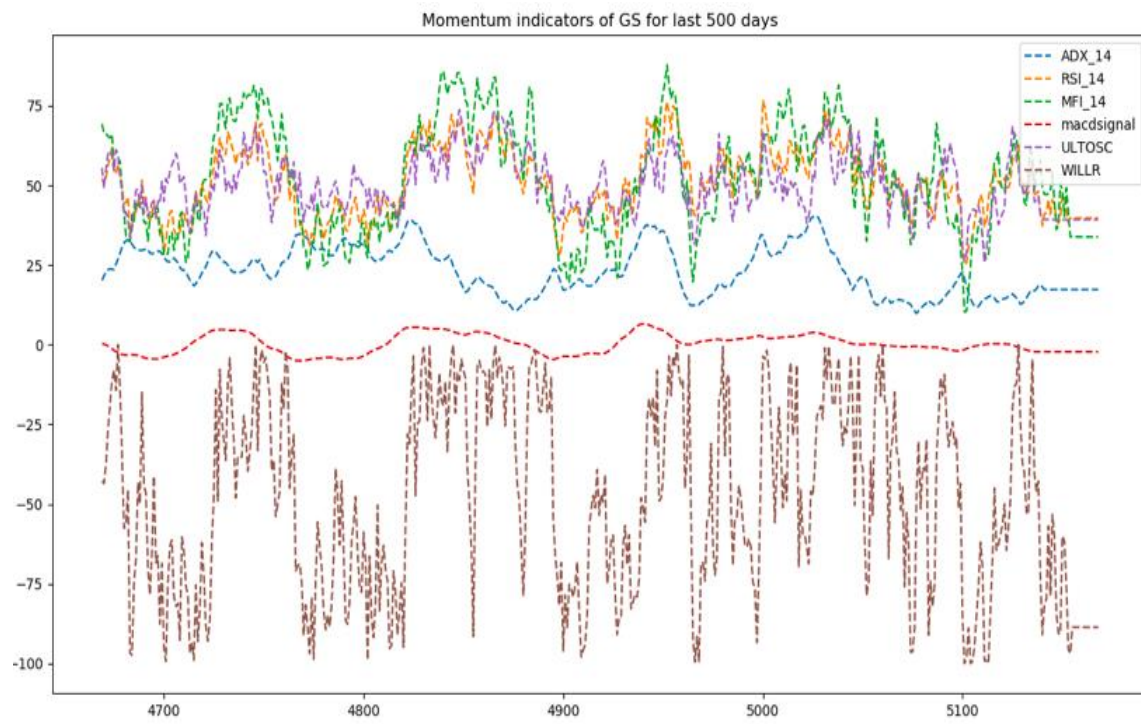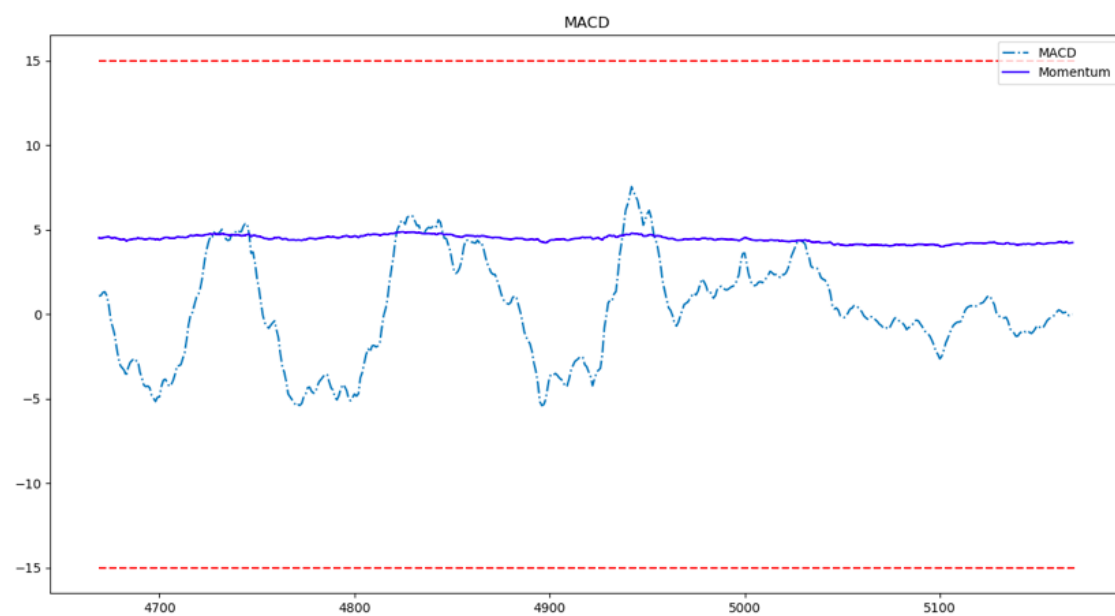
Data

We have collected daily price data, quarterly fundamental data, and quarterly analyst forecast data of the companies in the financial and technology sectors of S&P 500 from FactSet. Data collected from 1999 to 2019 since the data before 1999 is not well maintained and mostly contained nan values which makes sense since company's hardly used to report them efficiently. Also data were cleaned to remove nans and feature engineered to convert quarterly fundamental data to daily format using backfill to improve the robustness of the model's performance and prediction accuracy. Cleaning the data demonstrated improved performance in model's accuracy.

Let us take Goldman Sachs for example to keep the report brief and reach the conclusion faster.

1)*Price Technical Indicators*: we created 178 price technical indicators, including Moving Average indicators, Momentum indicators, Volatility indicators, Volume indicators and MACD, the sample plots of few of the popular indicators are drawn as follows:

Momentum indicators of GS for last 500 days

Volatility indicators of GS for last 500 days

Volume indicators of GS for last 500 days

MACD
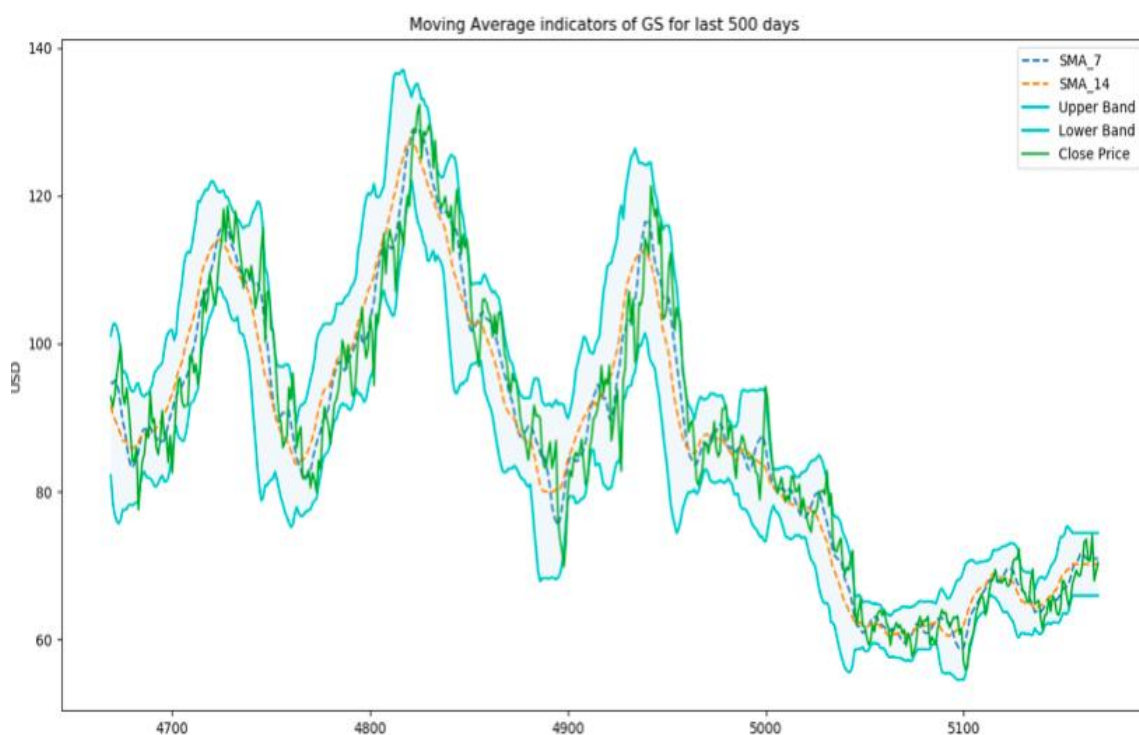
2) *Fundamental Indicators*: we take data directly from 5 financial statements, namely balance sheet, cash flow statement, income statement, ratio analysis, and analyst estimates. We have feature engineered new indicators out of the data by including percentage changes in the fundamental data. Intuitively it makes sense since it provides an advantage to measure a company's financial strength as well as we can compare the financial strengths of different companies using these percentage changes instead of absolute value. This can be also used to rank stocks based on these indicators to construct better portfolios.

EPS (diluted) Autocorrelation Plot
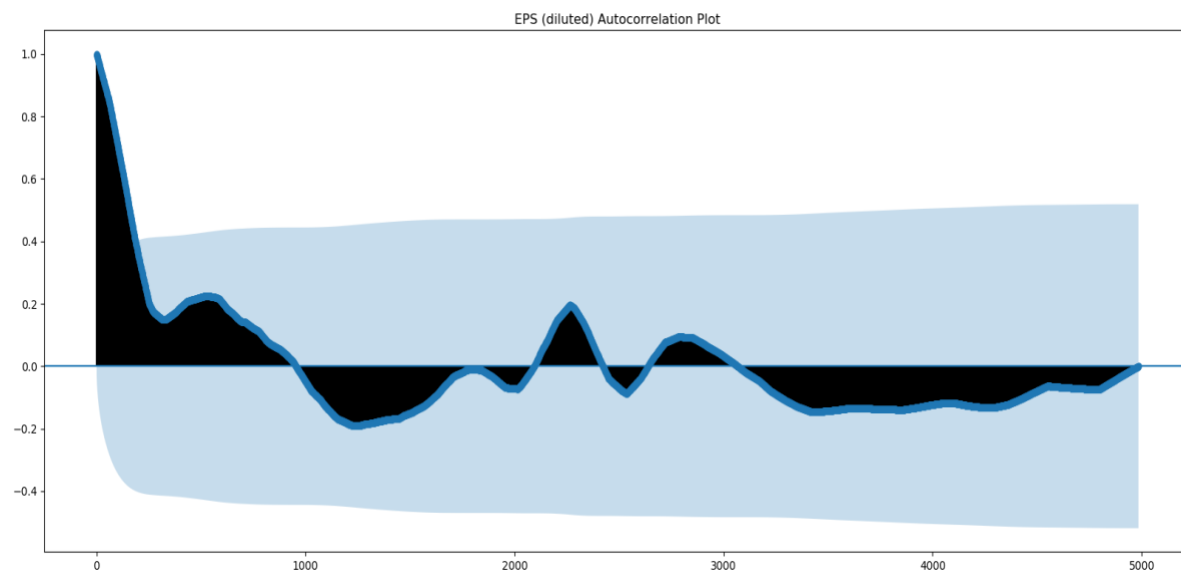
3) Earnings Indicators: these are some of the dependent variables which we want to predict from our data and model. An earnings surprise is simply defined as a percentage change from the analyst's EPS expectation as described in the data section and the actual EPS as reported by the firm. In this study, we include percentage thresholds, as a means of expressing the magnitude of a surprise so as to construct various tests.

SETUP

For the classification task:

Model's output:

EARNINGS_SURPRISE = 1, Neutral (if $EPS\_ACTUAL - EPS\_ANALYST == 0$)
EARNINGS_SURPRISE = 2, (if $EPS\_ACTUAL - EPS\_ANALYST - 1 - s > 0$)
EARNINGS_SURPRISE = 0, (if $EPS\_ACTUAL - EPS\_ANALYST - 1 + s > 0$)
s = constant surprise threshold we can adjust to 5%, 10% or 15%

For the prediction task:

Model's output:

EPS_ACTUAL

Architecture

For the first task to predict earnings surprise which belongs to classification problem, we use the XGBoost (Extreme Gradient Boosting) Classifier model. The idea behind Gradient Boosting, is to "boost" many weak learners or predictive models so as to create a stronger overall model. In this way, the algorithm always uses data samples that were hard to learn in previous rounds to train models. The XGBoost Classifier model is a probabilistic classifier which simply outputs a probability of each instance belonging to one of the specified classes. The chosen class is the one with the highest probability.

For the second task to predict EPS which belongs to regression problem, we use XGBoost Regressor model. The XGBoost Regressor simply outputs the best forecast of the expected EPS, a predicted continuous value.
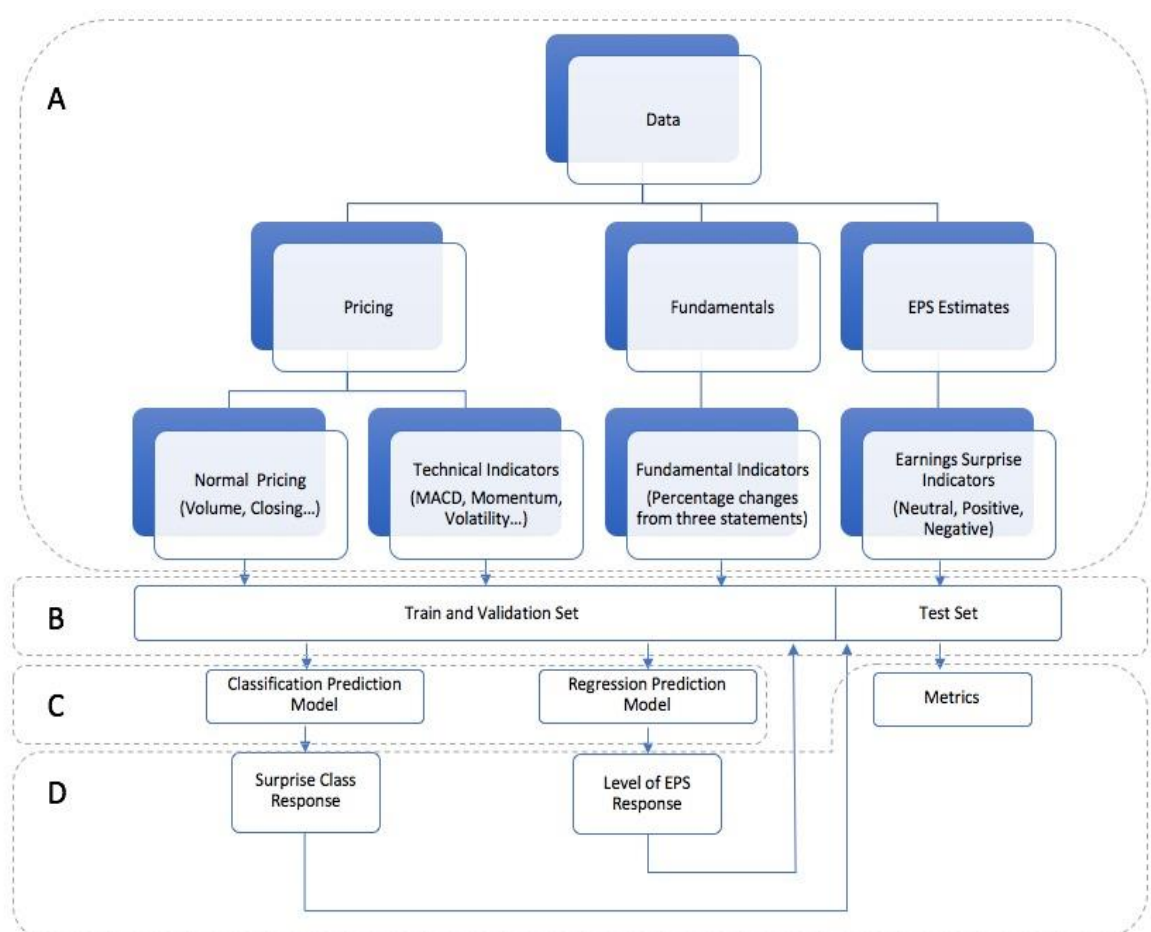
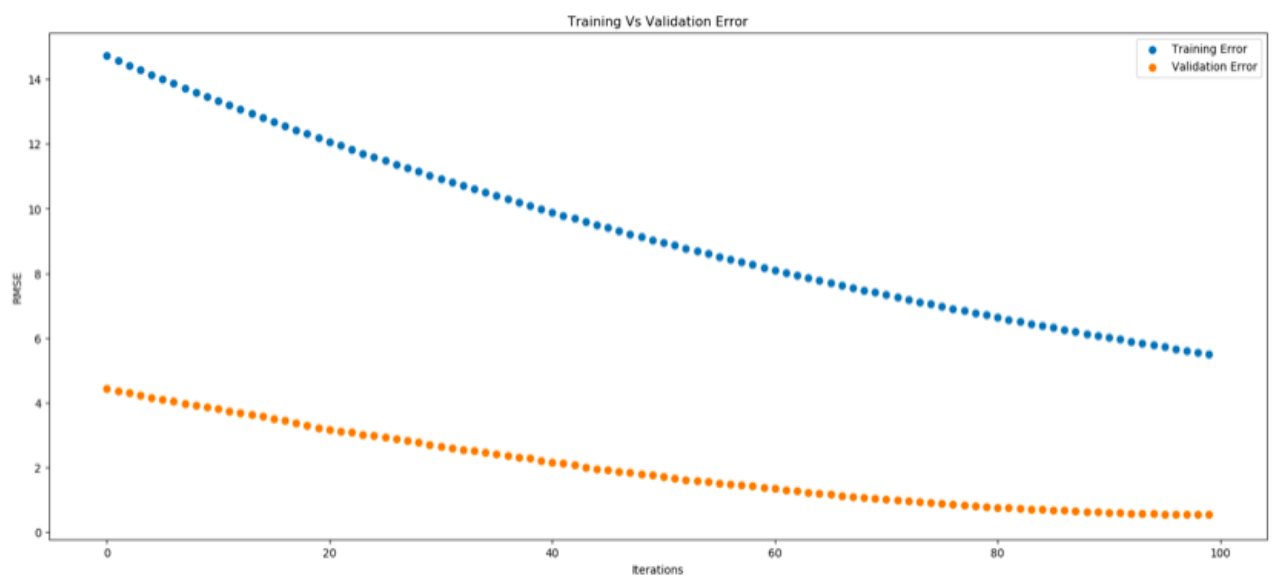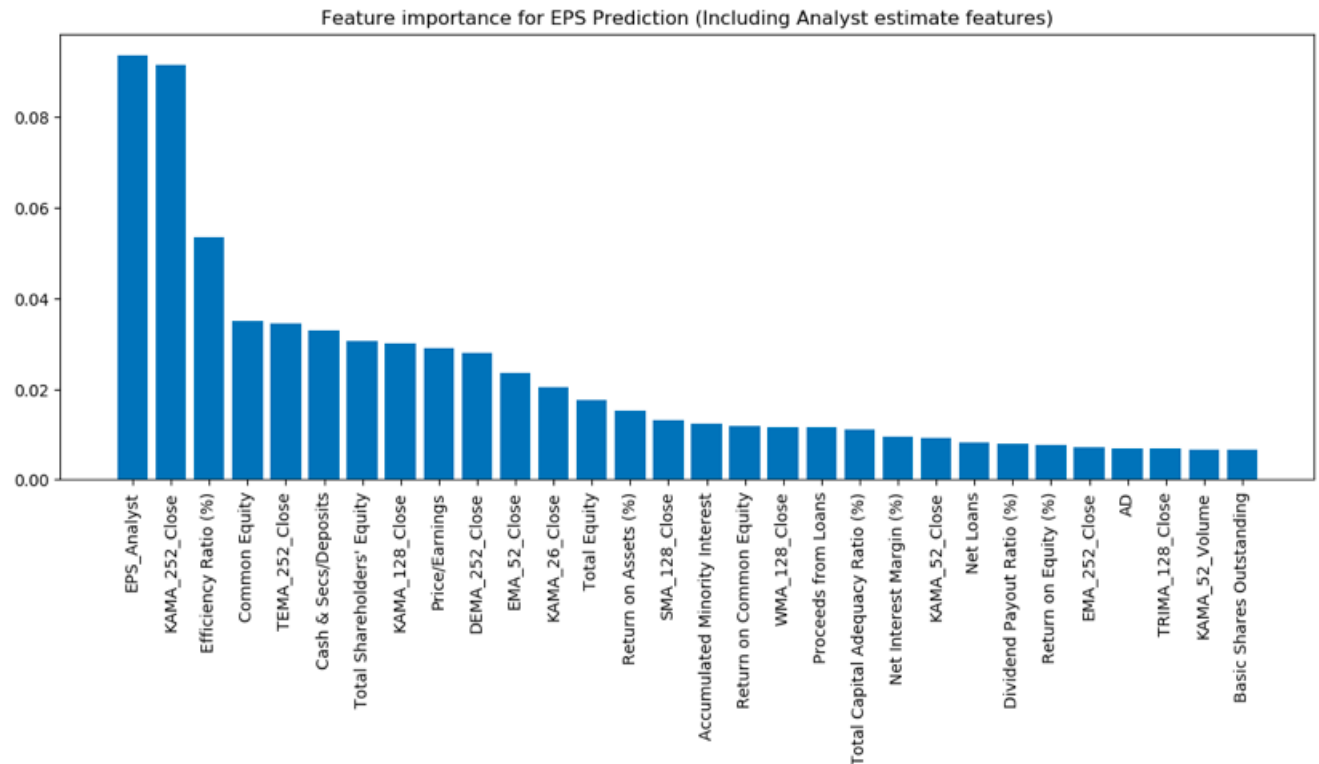## Multi-class Classification – Earnings Surprise Prediction

We use accuracy to evaluate the performance of our model, and the accuracy is the number of correctly predicted surprises (true positives) and correctly predicted non-surprises (true negatives) in proportion to all predicted values. It incorporates all the classes into its measure $(TP + TN)/(TP + TN + FP + FN)$, where TP, FN, FP and TN are the respective true positives, false negatives, false positives and true negatives values for all classes.
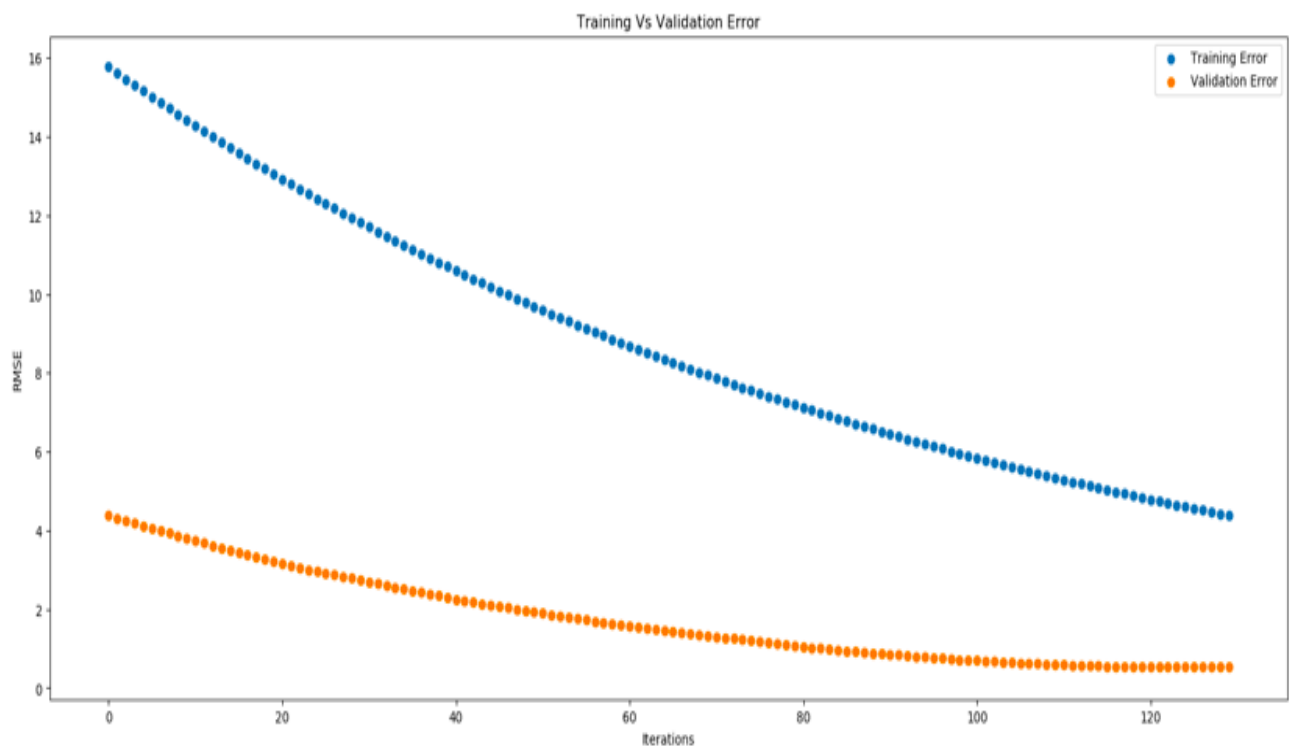
Here, we take GS as an example. The accuracy is very high of 95.78%, and it shows that our model is trained very well. However, we must admit the limitation of our sample since over 90% earnings suprises are negative.

Goldman Sachs Confusion Matrix
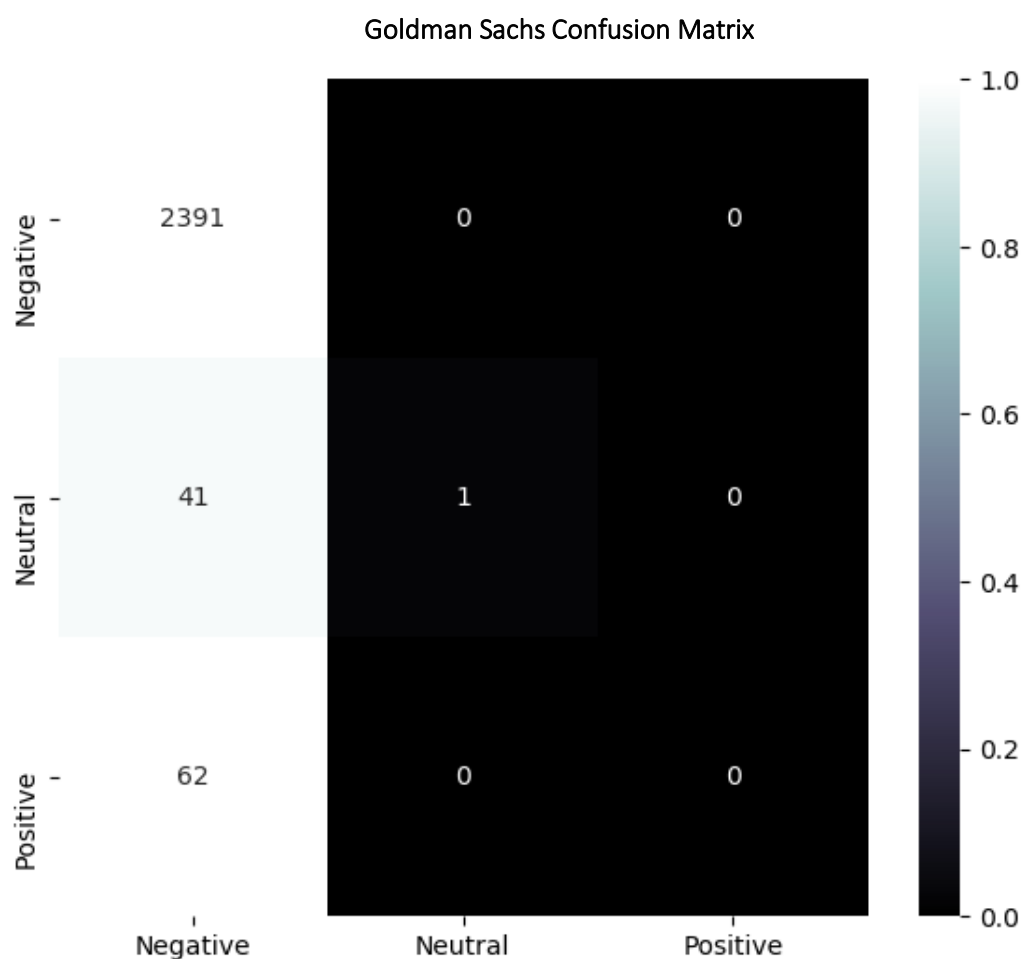
Mulivariate Regression – EPS Prediction

A. Feature importance with XGBoost: having so many features we have to test feature importance for predicting EPS. Here are our results for including analyst estimate features and excluding analysit features, both of them don't have overfitting problem.

Feature importance for EPS Prediction (Including Analyst estimate features)

Training Vs Validation Error

Feature importance for EPS Prediction (Excluding Analyst estimate features)

Training Vs Validation Error
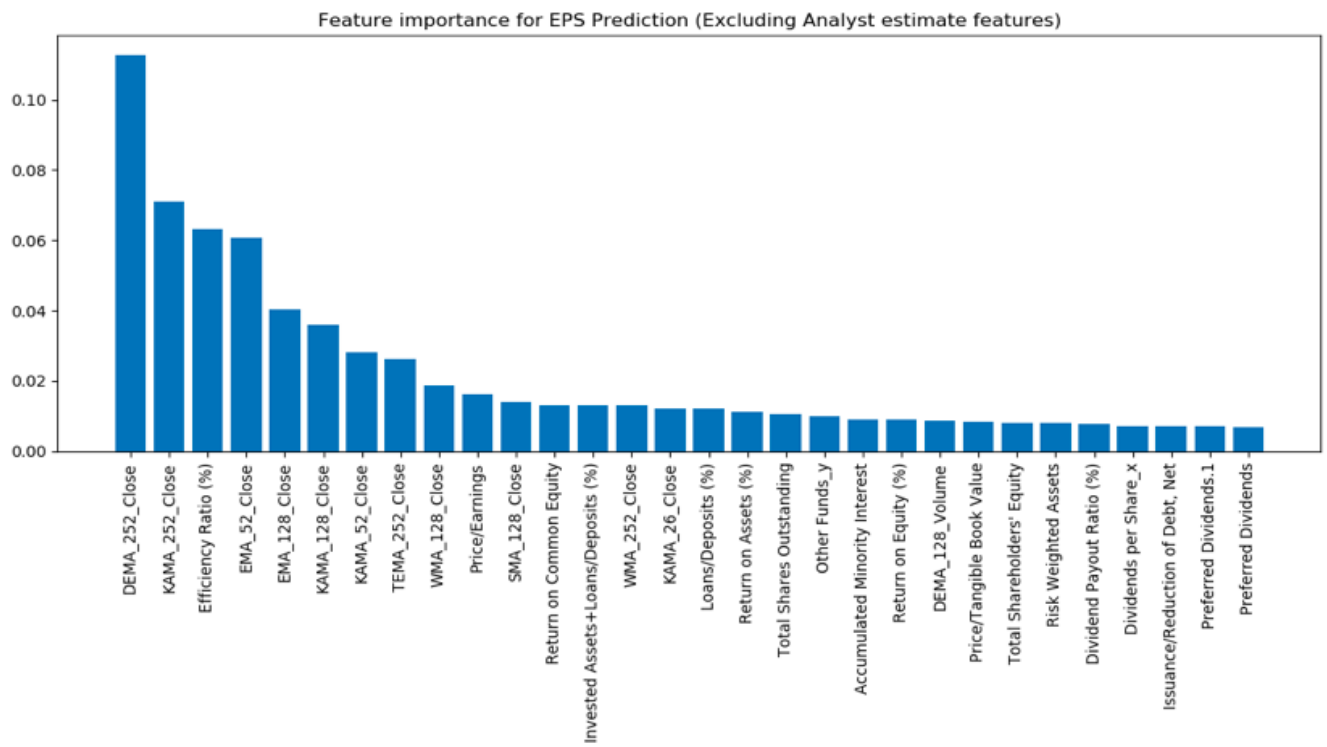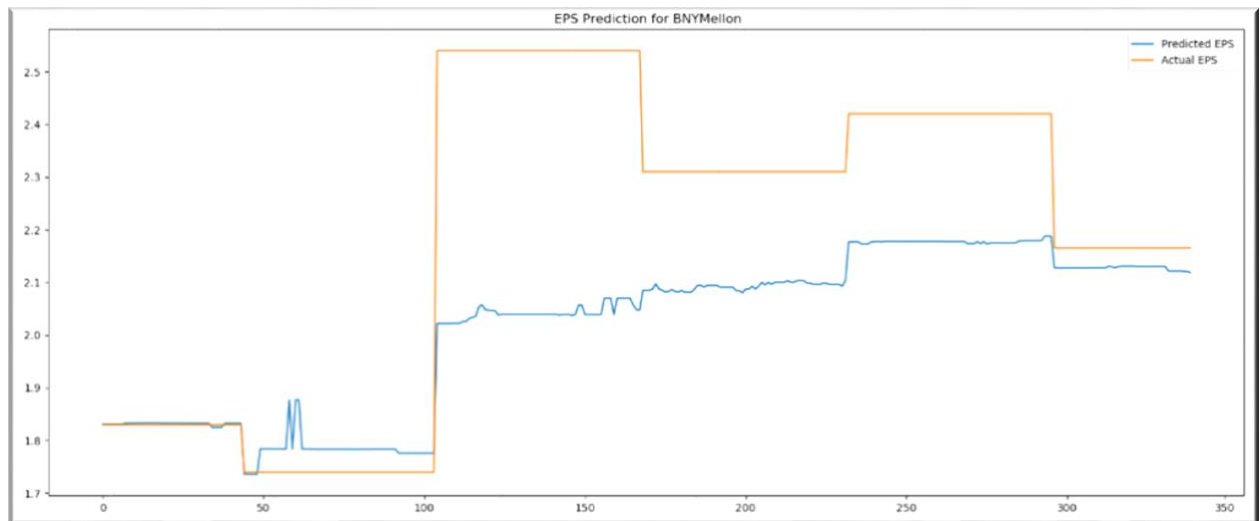
B. EPS Prediction Analysis and conclusions:

- Keeping the **top 30 - 50 features** selected from the XGBoost's Feature Importance list improves the model's accuracy by lowering the mean root square error. For example, keeping 33 most predictive features for Goldman Sachs' EPS prediction improves Model's prediction accuracy by reducing Root Mean Square Error from 3.33 to 2.91.

- **Keeping EPS_Analyst feature** improves our Model's predictive accuracy for all the tested stocks which matches with out intuition in the sense that Analyst does a good job.

- Using **percentage changes in fundamental data** from past quarter and past year (for example, **((pretax_income(current)– pretax_income(prev_quarter))/pretax_income(prev_quarter))** improves our model's prediction accuracy drastically and its intuitive in the sense that change in fundamental dataset is a **good comparative indication** of how well the stock is performing and can be used to predict the future eps. We feature engineering all the fundamentals by incorporating their change and improved our model's accuracy.
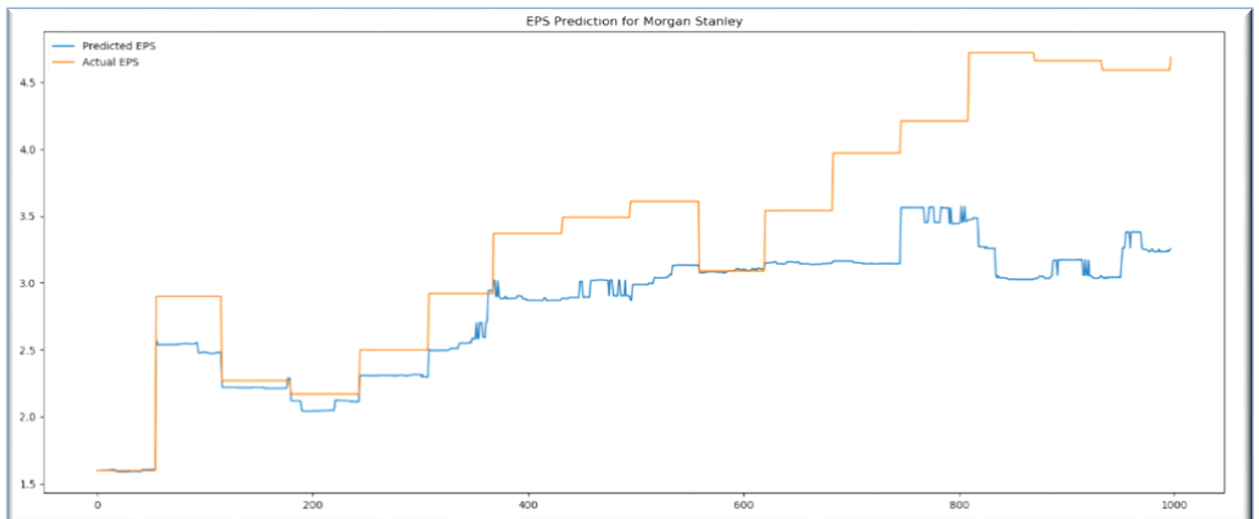
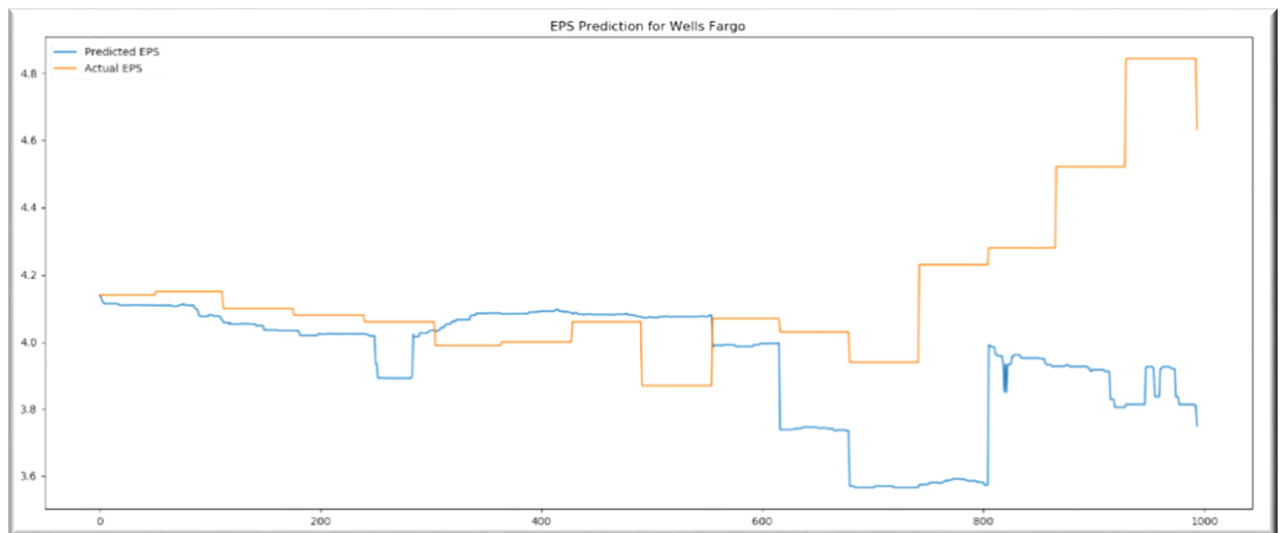## EPS Prediction Results for few Stocks:

Financial Sector:

BNY MELLON:

Morgan Stanley:

Wells Fargo:

JP Morgan & Chase:
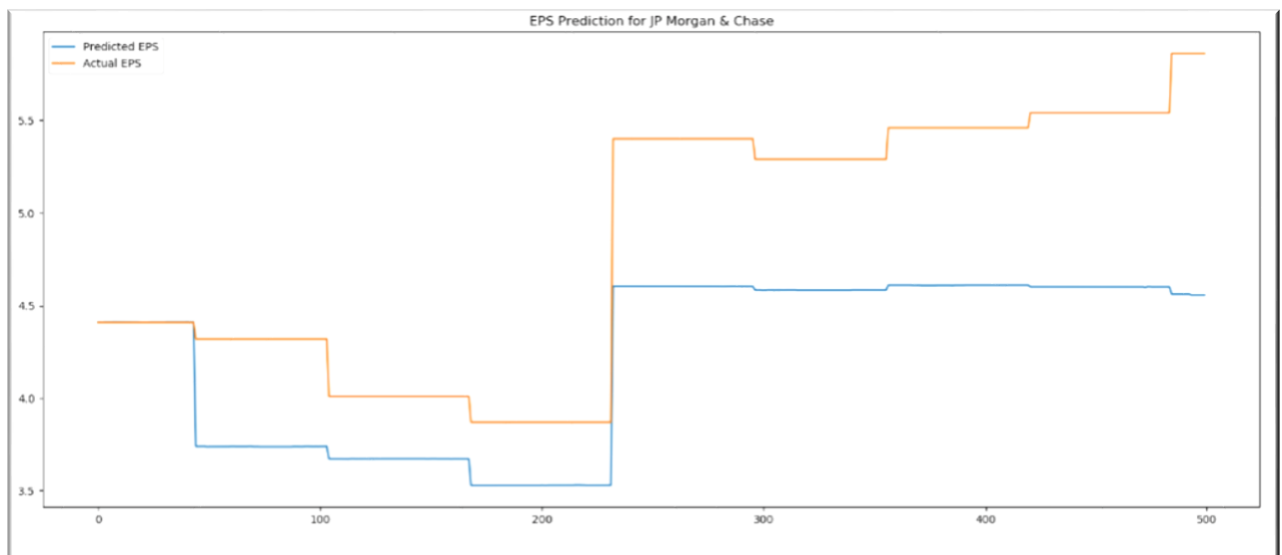
Technological Sector:

Adobe:

Adobe EPS (Predicted vs Actual)

Apple:

Apple EPS (Predicted vs Actual)

NVIDIA:

NVIDIA EPS (Predicted vs Actual)
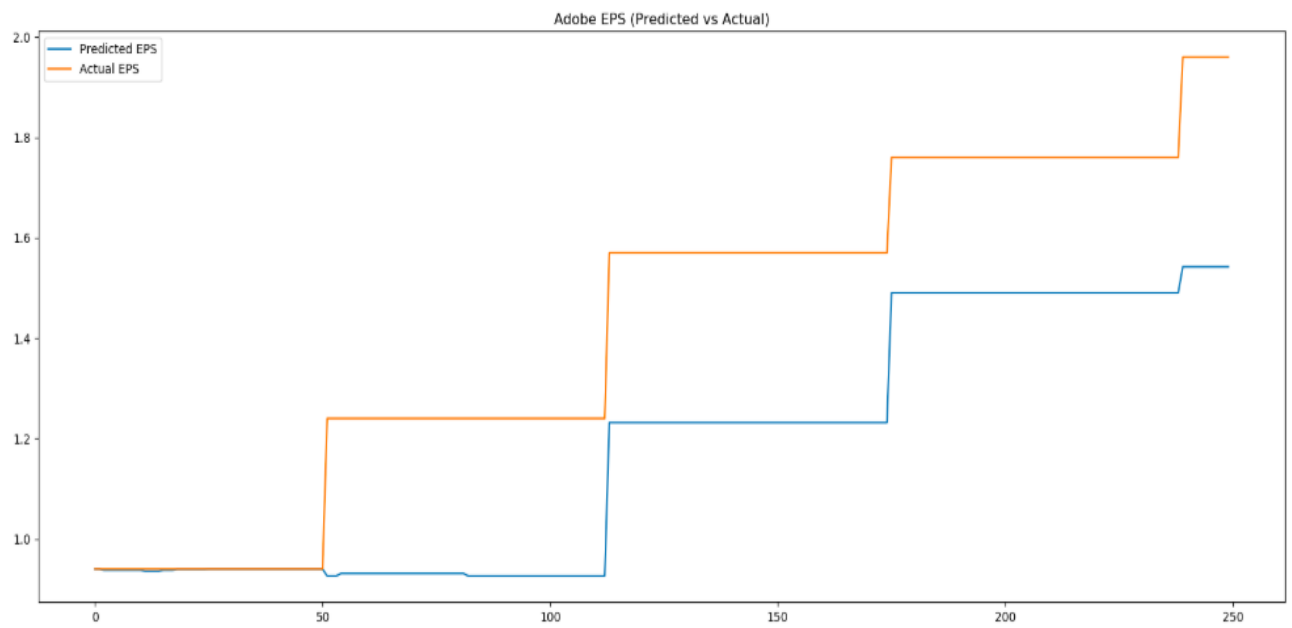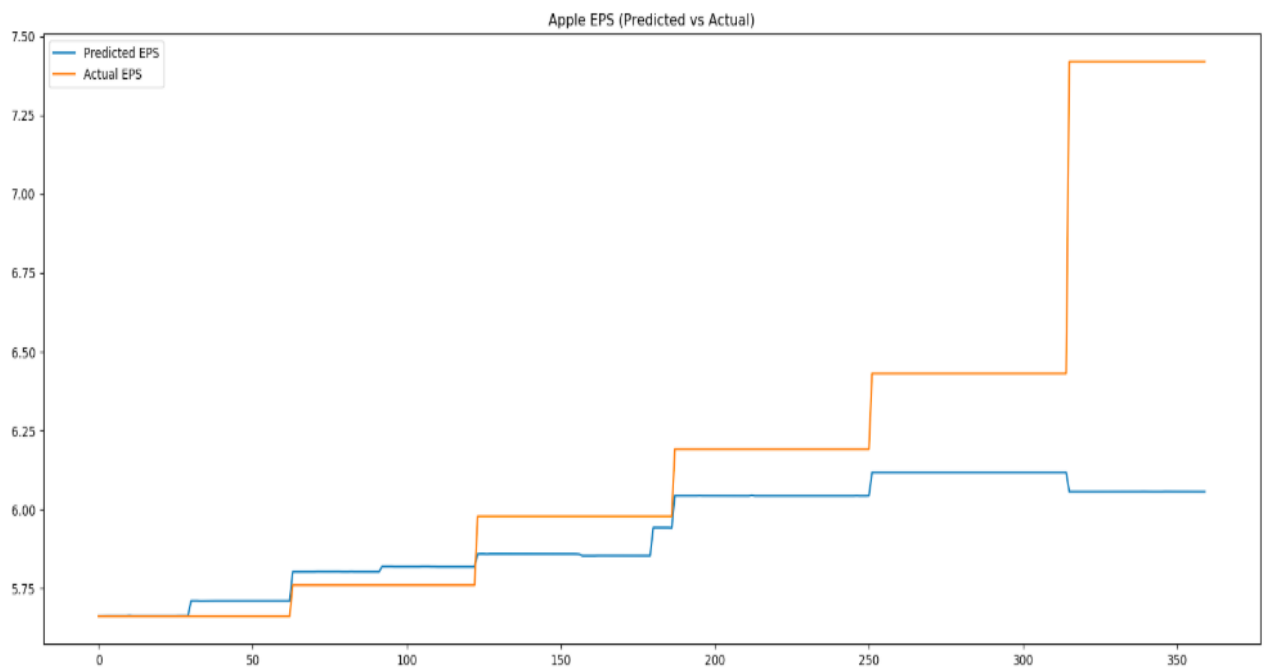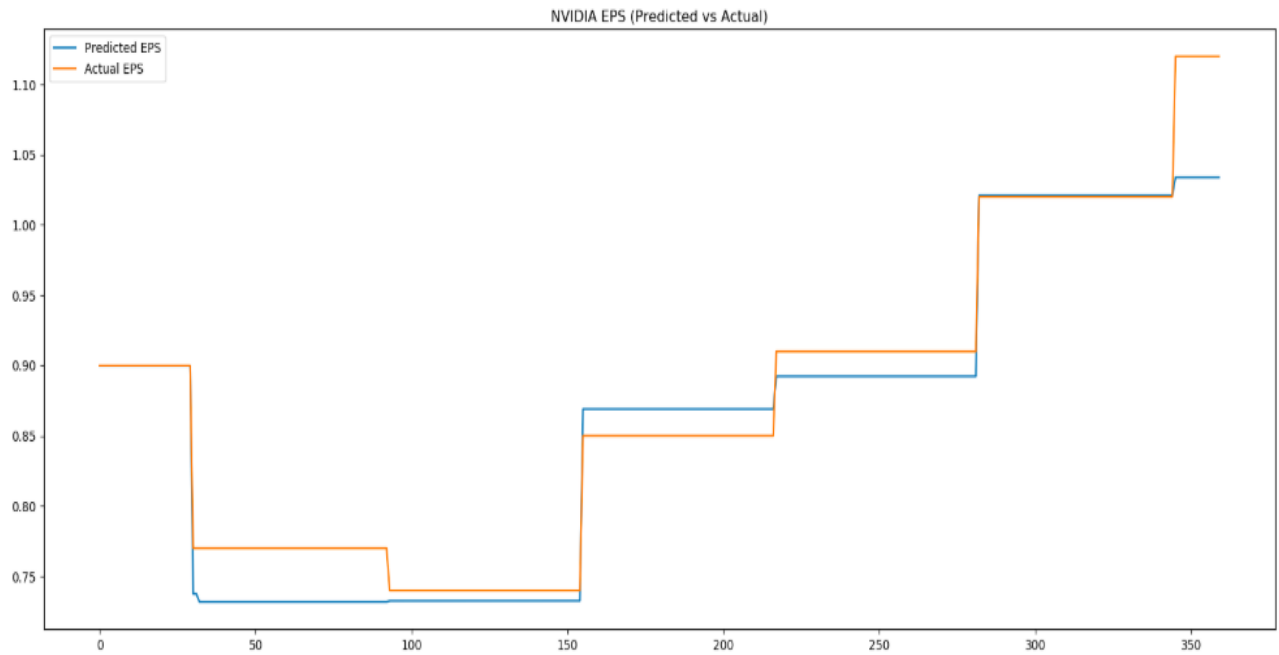
As you can see, our model is successfully predicting if the EPS is going to go up or go down and this is because of our clever feature engineering, model selection, and careful hyperparameter tuning to prevent overfitting. Although, it is very difficult to predict the actual value of the eps but still our model doesn't deviate from the actual value by a large margin and hence our model can be used for EPS prediction.

Also, it is important to notice that the model's EPS prediction for a longer future is not that accurate and it is intuitive in the sense that currently it is difficult to predict the EPS for the next 3 or 4 years.

Portfolio Construction based on the predicted EPS

Once we can predict the EPS for the next quarter, we try to incorporate our predictions into portfolio construction techniques, and build several hypothetical investment funds, where the prediction of EPS will be our main source of asset weights. We shall see how our portfolios perform compared to the benchmark.

Asset Universe and Benchmark

We performed EPS on 9 assets in the financial sector, and 3 assets in the technology sector. They are:

- Wells Fargo
- Goldman Sachs
- Bank of America
- Berkshire Hathaway
- Blackrock
- BNY Mellon
- Citigroup
- JP Morgan
- Morgan Stanley
- Adobe
- Apple
- NVIDIA

The reason why we would not expand the universe into more assets was because the preparation of data that is needed for the prediction involves a lot of manual steps and data cleanup. We could not find a good source / vendor of data that can give us the fundamental of many companies in a clean format.

Therefore, we chose to do our portfolio benchmarking in 2 universes: financials, and all sectors. For financials, our universe is all 9 financial assets, and the benchmark is the SPF index (the financial sector of SP500). For all sectors, our universe is all 12 assets, and the benchmark is the SP500 index.

Time Periods

Each asset has clean data with varying starting date, but all assets have clean data starting in January 1st, 2004. Prior to that date, we observed some issues with the data that we collected from some vendors. For the sake of uniformity, that will be our starting period for all assets(T_1 = 2004/1/1).

Our fund "inception" date is T_2 = 2010/1/1, and we rebalance weekly on Fridays.

Our last rebalance date is the last Friday in November of 2019 (T_3 = 2019/11/29) for all sectors, and the last Friday in January 2018 (T_3 = 2018/1/25) for financial sector. The reason why we stopped early for financial sector is because the data for SPF that we obtained only existed until January 2018.

Methodology

For each rebalance time *t*, where $T_2 <= t <= T_3$, we take the data from T_1 up until *t*-1 (from beginning of training period up until the day before rebalance), and train a brand new regressor with that data. This way, we simulated the scenario where the portfolio manager would be able to see all the financial data up until the day before rebalance, in order to make his decisions about that rebalance period. That also means that the regressor always has the most up-to-date information about the company, without being able to see the future (i.e. we never train it with data AFTER the rebalance because it is not supposed to be able to see that data yet).

The EPS prediction is converted into asset weights using 3 flavors of interpretation, which we shall see below. After we obtain the asset weights, we make the trading at the end of the Friday rebalance, using the price as of the close of Friday. This is a known limitation because in reality, the effective price that we traded at would not be that price.

Another known limitation is that we also ignore transaction cost and all trading is assumed to be completed instantaneously, instead of taking several days as is common in many asset management firms. We will revisit these limitations in more detail in another section below.

For each flavor of portfolio construction, we do 2 variations (for a total of 4 combinations): long-only vs long / short, and with or without Citi.

Long-only vs long / short: we observed that a lot of our gains come from shorting stocks whose prices went down, so we tried the long-only strategy to test our hypothesis that a large portion of our gain is from shorting stocks.

With / without Citi: As we will observe later, there were three consecutive rebalance periods where we shorted Citi in large quantity (~35% of the fund's portfolio), and as a result, profited a lot from Citi's price falling (around the end of 2016). As per Prof. Hirsa's suggestion, we tried to take Citi out of our portfolio and compared the result, to see whether we "got lucky" with Citi, or whether the strategy is sound even without an event like Citi.

Different Alpha Strategies using the predicted EPS

Flavor A

Given a predicted EPS, we can calculate the predicted P/E, by taking stock price divided by EPS. High P/E is undesirable, whereas low P/E is desirable. So our asset weights are proportional to 1 / P/E. In other words, the asset weights are proportional to EPS / price.

When we compute the long-only variation, all the negative EPS predictions are taken out of the universe, so we are only investing proportional to the positive EPS / price.

Flavor B

Given a predicted EPS, we compare it to analyst's prediction of EPS, which we get from FactSet. The asset weights are then proportional to (predicted EPS - analyst EPS) / predicted EPS = (1 − analyst EPS / predicted EPS).

If this number is high, it means that predicted EPS is likely to beat analyst EPS, and this stock is desirable. If this number is low (or even negative), it means that the predicted EPS is likely to miss analyst prediction, and we should short the stock. The predicted EPS itself may be "high" in an absolute sense, but it is compared to analyst prediction, and that's our clue of whether the stock price is going to increase or decrease immediately following the earnings call.

Flavor C

Similar to flavor B, the asset weights are proportional to (predicted EPS − average EPS) / average EPS, where average EPS is the average EPS for the last 1 year.

If this number is high, it means that the earnings are going to increase compared to how the firm has historically been performing, which would then encourage price increase. Likewise, if the number is low, the stock price is likely to decrease, and we should short the stock.

Results

With 3 flavors, 2 universes (financials and all sectors), and 2x2 variations (long-only / long-short, and with / without Citi), we have 24 different portfolios that we simulated from 2010. The growth plot for the portfolios are shown below, with each portfolio indexed at 100 at inception.

For All Sectors

All Sectors - Long Short - With Citi

All Sectors - Long Short - No Citi

All Sectors - Long Only - With Citi

All Sectors - Long Only - No Citi

For Financial Sector

Financials - Long Short - With Citi

Financials - Long Short - No Citi

Financials - Long Only - With Citi

Financials - Long Only - No Citi
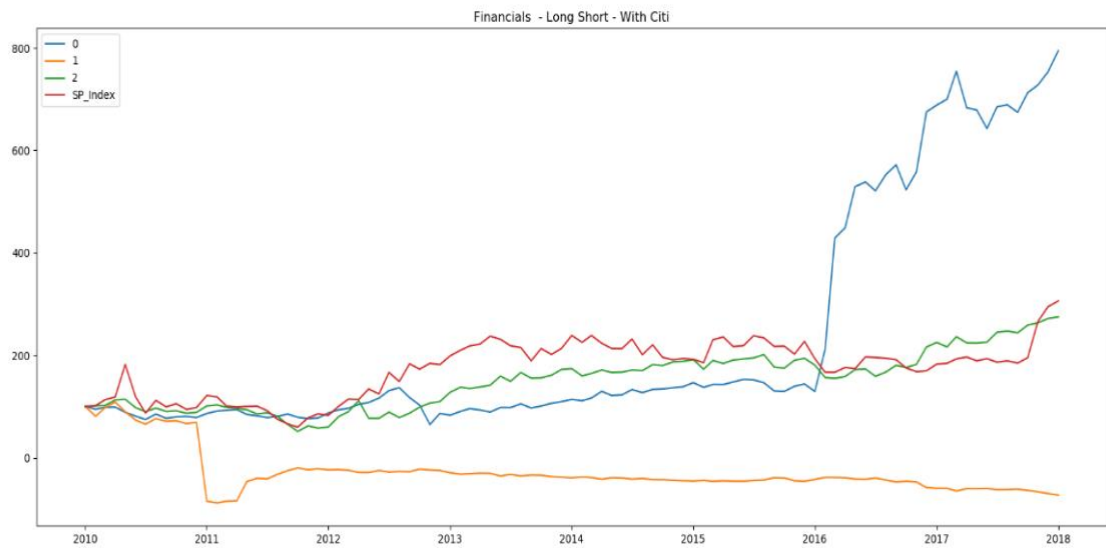
Information Ratio

For each portfolio we also calculated the information ratio against the respective benchmark.

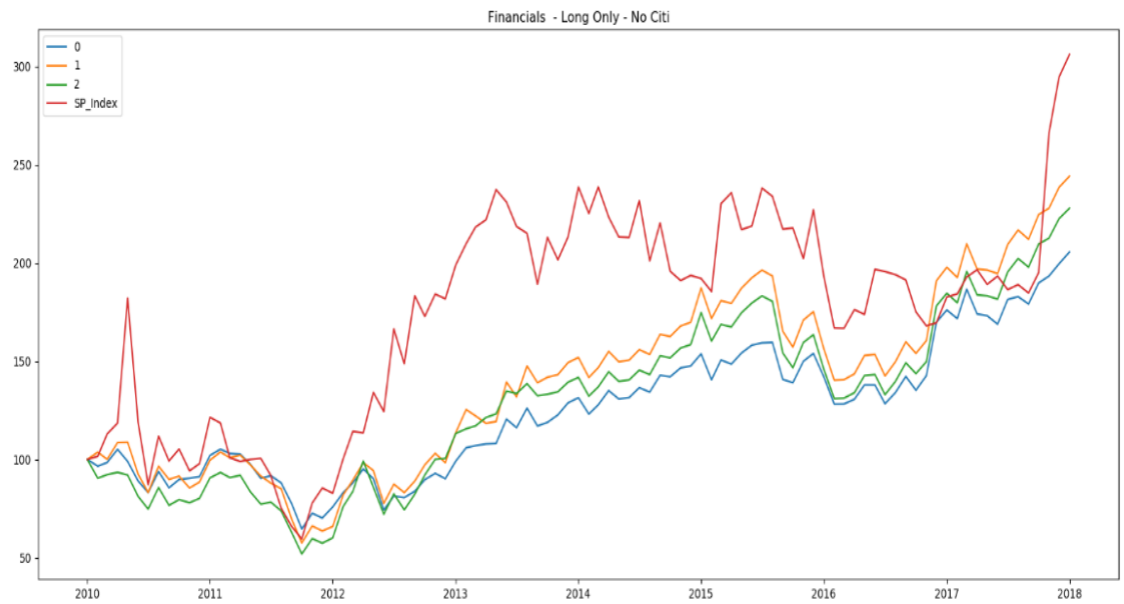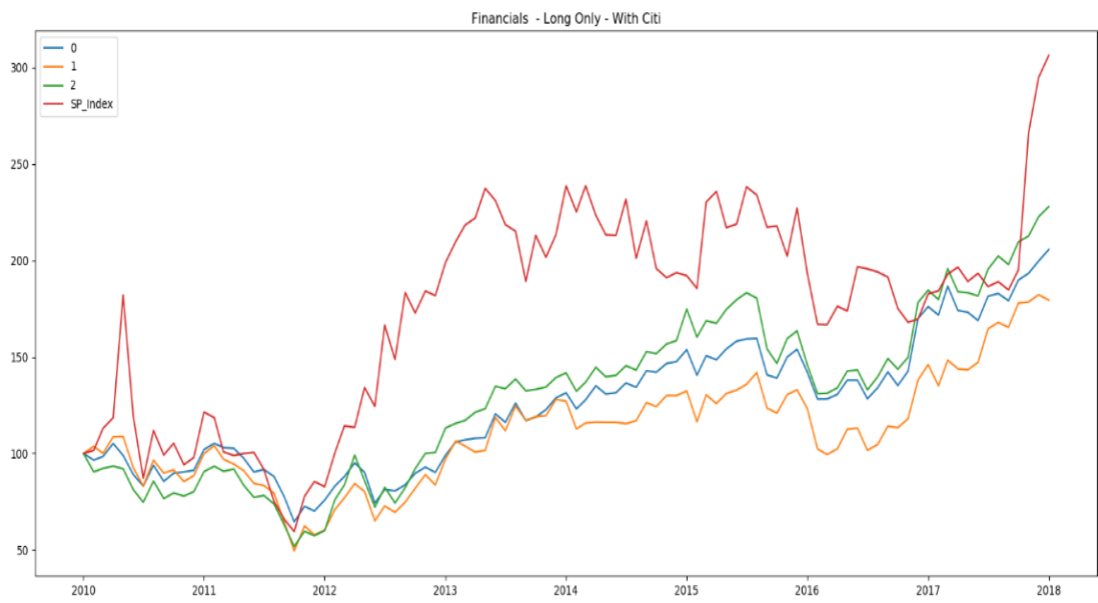|	IR: All Sectors				IR: Financial Sectors			
	Long /Short		Long Only		Long /Short		Long Only	
	With Citi	No Citi	With Citi	No Citi	With Citi	No Citi	With Citi	No Citi
Flavor A	9.31	-1.82	-1.62	-2.17	5.24	-8.08	-8.01	-8.09
Flavor B	-0.55	7.18	-2.71	6.93	-14.37	2.39	-8.25	-6.06
Flavor C	9.67	6.99	8.09	6.18	-3.61	-4.01	-6.5	-6.49
Average	6.14	4.12	1.25	3.65	-4.25	-3.23	-7.59	-6.88

A few key observations:

- Overall we did far better with all sectors compared to financial sector. This difference can be attributed to the fact that AAPL stock did really well towards the end and we invested a lot in it. The other cause of difference is because the financial sector benchmark (SPF) had a sharp increase at the end of 2017, which caused a lot of our alphas to be negative, even though we were beating the benchmark as of early 2017.
- When we force the portfolio to be long-only, our performance did suffer terribly. In some cases we were still able to beat the benchmark, by investing in stocks that were about to go up, but the weights were more evenly distributed. The huge jumps of performance that we get from long-short were because we shorted a big amount in one or two names. In other words, we were a lot more aggressive with the short strategy as opposed to with the long strategy, and that's why we were able to make bigger gains with the short strategy.
- When we included Citi with the long-short strategy, we did the best, both in financials and all sector universe. When we included Citi but not allow short, we didn't do as well (for the reasons we mentioned in the previous bullet point). When we remove Citi but allow short, our performance is quite good but not as spectacular as when we included Citi. This reinforces our belief that while our prediction strategy is very good with predicting the decrease in price, we also "hit the jackpot" with Citi.

Limitations

There are a number of limitations with our simulation that make it impossible to replicate in real-life scenario. We shall address these limitations:

1. We ignored transaction cost, as well as portfolio implementation cost and timing. We assumed that each trade happened instantaneously on Fridays at the close of day, using the price as of the market close on Friday. In real life scenario, transaction cost is a non-trivial component of performance, especially for a fund that rebalances weekly like ours. It is also unlikely that the new weights can be implemented instantaneously without affecting the market or missing the target price.
   a. Our portfolio also varied wildly (with huge turnovers and sometimes changing from long a stock into short that very same stock the following week). This is because our portfolio construction methods do not take into account current holdings, therefore we could not "optimize" to try to not veer away from current holdings too much. In order to be as realistic as possible, we should either limit the turnover rate (where that limit is baked into the portfolio construction method), or incorporate transaction cost and market impact into our performance calculation.
2. We ignored borrowing cost. In real life portfolio, whenever we short a stock, there is borrowing cost associated with that short holding, and the amount is often non-trivial. One of the reasons our short portfolio performed so well was because we completely ignored borrowing cost in our performance calculator. In real life scenario, the borrowing cost is a significant "drag" on our performance. Also, shorting ~30% of a fund's total value in a single name proves to be a good move in hindsight (during backtesting), but in real life, that move is extremely risky. We imagined that the firm's risk manager would not easily approve of such position.

Key Takeaways

1. Our EPS prediction does really well when combined with short selling strategy. Even accounting for the fact that we ignored borrowing cost and transaction cost, we were able to predict when certain stocks are going to decrease in value. It's not necessarily that the EPS is negative, but when the EPS is going to be well below analyst expectations, we were able to predict it ahead of earnings call.
2. Our EPS prediction does moderately well when combined with long-only strategy. We can predict when the EPS is going to beat analyst expectations, but the confidence and magnitude of our prediction is less accurate than that of the short-selling scenario. Therefore, while we still profit from the increase in stock price, we did not allocate as much into that stock as we could have.
3. Our EPS prediction scheme hits the mark once in a while, but for a large amount. In other words, we "get lucky" for a name or two once in a while, but that correct prediction allows to gain a significant profit.

Future Work

1. Future work based upon this project could include realistic portfolio implementation scenarios, such as:

   a. Adding time to implement the portfolio (For example, when we switch from holding 60% Goldman Sachs to 20%, it will have to be implemented over 5 trading days instead of instantaneously, and the price of sales will follow the closing price of those 5 days)
   b. Adding borrowing cost for short positions
   c. Adding transaction cost, especially for large turnovers, or alternatively, adding turnover restriction baked into the portfolio construction step itself

2. Adding data for more companies. If we have many (~100) companies with predictions, we could invest in, say, only 50 companies at a time, with 50 of the best prediction. In other words, our EPS prediction becomes part of our "stock picking" mechanism, and the weight construction can be based on other metrics (expected value, covariance and risk) and our EPS prediction can be included in those metrics too. We feel that this construction scheme will be a lot more robust and stable, and reduces most of the large turnovers we see in current scheme.

Project Availability:

Our full project with code and datasets is available at Github:
https://github.com/abhishekprog0/EventPrediction_DeepLearning

References

1. Derek, S. (2017). Predicting Earnings Surprises. Available at SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3420722
2. Xiao, Y., Xiao, J., Lu, F., & Wang, S. (2013). Ensemble ANNs-PSO-GA approach for dayahead stock E-exchange prices forecasting. International Journal of Computational Intelligence Systems. Available at: https://www.atlantis-press.com/journals/ijcis/25868372
3. Boris, B. (2019). Using the latest advancements in AI to predict stock market movements. Available at Github: https://github.com/borisbanushev/stockpredictionai
4. Stauth, J. (2013). Trading (and predicting) earnings surprises. Available at: https://blog.quantopian.com/wpcontent/uploads/2013/05/Trading_Earnings_surprises-jess.pdf