# Future Income Growth of Russell 1000

Nithin Pingili, Andrew Tan, Zhaokun Xue and Yunhao Yang

## Problem Statement:

## Estimate the net income growth rate of Russell 1000 companies for next quarter.

What is the data science problem you are trying to solve? Why does the problem matter? What could the results of your predictive model be used for? Why would we want to be able to predict the thing you're trying to predict?

The stock market is the most notorious get-rich-quick mechanism. It seems so easy: buy a stock whose price will increase, then sell it at a profit. However, take a random company and track their stock price over time. The price will move up and down and all around and will show no discernible pattern. But a long term trend is clear. Overall, the stock price reflects how investors feel about a company's future profits. Sometimes their predictions are very wrong, like with Apple in the 1990s. And sometimes their predictions are right. An omniscient investor would invest by choosing companies whose profit growth surpasses the current market's expectations. So the problem is clear: find a way to reliably predict a company's future net income(profits) using data mining techniques.

This problem matters because people planning for retirement need ways to grow the money they've accumulated. Investing in stocks has always been a place for savers to park their money and hope for compounding growth. But the noise and hype we hear and feel from the market can cloud our judgment when it comes to selecting the right stocks. We can negate this by only reading financial reports to make an unbiased decision, but no one has time to mull over hundreds of pages of boring documents. So a model that can accurately predict a given company's future profits will simplify people's lives and help secure their financial independence.

## Data

Describe your dataset. This may also include insights from data exploration.

The dataset we use is from QuickFS. The bulk data is found in /preprocessing/bulk_quarterly_financials.csv. In all, there are 904101 records with 192 features. Each record describes the financial performance of a company in a given quarter specified by the feature “period_end_date”. Some example features describing financial performance include “revenue”, “net_income”, “total_assets”, “cf_cfo”(cash flow from operating activities). There are also features which store certain ratios used to describe the performance, such as “roa”(net income/total assets) and “net_income_growth”. However, many of these ratios seem to be wrong, so we had to fix them in the preprocessing step. There are also features indicating the ticker symbol of the company as well as the sector to which the company belongs. The time period for the data stretching from around the year 1999 to the current day.

Since many features can be derived from one another, several of them are very correlated. Looking at a correlation matrix of some of the ratios, the most correlated ones were things such as “roa” and “roe”, which was expected since they are derived from one another through a formula. Some industries lacked certain features. Banks, for example, didn’t have entries for “gross_income”. There were also some features unique to certain sectors. For example, Insurance companies had “total_interest_income” which most other sectors lacked.

Analyzing the standard deviations of the features we chose by each sector, there are some interesting patterns. Overall, most sectors had a large variance in “fcf_growth” (free cash flow growth). Accommodation & Food Services had large variances for “gross_margin”, “fcf_margin”, and “revenue_growth”. This discrepancy shows how success varies greatly among restaurants and hotels. Real Estate had a huge variance in net_income_growth and capex_growth. This large variance likely means that real estate companies are not all equally successful, and not equally large. Information Services varied greatly in net_income_margin and fcf_margin. This makes sense, because many younger tech companies forsake profits while mature ones rake in huge amounts of profit.

We also did the same analysis for the median of each feature by sector. The median “gross_margin” for Forestry & Agriculture companies was extremely small; they only make ~0.06 dollars of gross profit on every 1 dollar. Interestingly, despite corporate profits tending to increase in the long run, the median “net_income_growth” was negative for 21 out of 30 sectors. This could mean that the increases in corporate profits are either dominated by a few companies or dominated by certain good quarters.

## Method

Describe your data science approach, any assumptions made, nuances, research done, feature engineering done, innovations in your procedure used, etc. Walk us through the process you used.

The biggest assumption we made was that a current quarter's financial numbers can predict the next quarter's future net income growth. This is a very generous assumption, since unpredictable events such as natural disasters or unquantifiable things such as market disruptors can decrease a company's future profits, despite excellent numbers. As a result, it's unlikely that our models can be more accurate than a company's self-reported net income estimates. This was further enforced when we met with Professor Kamm, the Director of Financial Education at McCombs, to see how stock analysis is done in real life. Stock analysis is mostly about fitting qualitative information into a mathematical model which, assuming the qualitative information is accurate, will forecast how the value will grow. So again, since profits can be dependent on qualitative information, our approach might not work well.

Looking at the data, it's clear to see that some companies are much larger than others. We decided that because we needed lots of data, we would only use ratios as features. This lets us compare the performance of a massive company with the performance of a small company. We also threw out ratios not present in most records, such as the ratios specific to Insurance companies. In our preprocessing step, we made a list of candidate ratios to use based on the features available. However, many of these features weren't useful, so we used a correlation matrix to find then remove redundant features. We also adapted the "period_end_date" into four binary variables that tell which quarter of the year that record was in. Since a company's performance can vary with the time of year (for example retail stores perform better around Christmas), we figured the quarter would be an important feature to have.

Another large decision we made was to only use companies in the Russell 1000. The Russell 1000 is the ~1000 largest publicly traded companies in the United

States. It makes up ~90% of the entire value of the US stock market. This decision was made so that the data wouldn’t be overwhelmed by smaller, less relevant companies. We also analyzed the data separately by sector. We figured that what would be considered a good ratio in one sector might differ in another sector. For example, Forestry & Agriculture companies tended to have extremely tiny margins, but Banks have larger margins.

We used six different regression models to train the data and then predict a result on unlabeled data. For each sector, we took the model that had the smallest mean squared error(MSE) among six regressions as our final result and used that as our final regressor for that sector. Prior to running each regressor, we scaled the data within each industry, removed records with missing feature values, and removed outliers that had feature values above 3 standard deviations. The MSE was evaluated using a 5-fold cross-validation. Below, we elaborate more on the type of regression used as well as the hyperparameters we chose.

The six regression models are:

- Support Vector Regression (SVR)- SVR uses the same principles as the SVM for classification, with some minor differences to handle continuous data (label). We perform cross-validation using GridSearchCV to determine the best ‘kernel’ and ‘degree’ for SVR. The parameters differed by industry. According to our results, this model had a relatively higher accuracy among these six models.
- Radius Neighbor Regression (RNR)- RNR is based on neighbors within a fixed radius. The labels are determined by the labels of the nearest neighbors in the training set. The radius we used is 5.
- Linear Regression: This is a linear approach to modeling the relationship between a dependent variable and one or more independent variables. We initially tried simple linear regression. However, it turns out that there is no correlation between any single feature and the label. So we used the multiple linear regression model in order to predict the labels. The regression model finds the most optimal coefficients for all the features. We used the coefficients to determine which features had the greatest impact on the predicted net income growth rate and how the various features are related to each other.
- Lasso Regression: This is a type of linear regression where the data values are shrunk towards a central point. The advantage of using this regression

over simple linear regression is that it reduces model complexity and prevents overfitting. To do so, lasso does ignore some features depending on the hyperparameter alpha. The higher the alpha, the more features are ignored. In order to have at least one feature with a nonzero coefficient for each industry, we tried different alpha a range from 0.1 to 0.9 with an increment of 0.1. The majority of the industries had 0.8 as the best alpha parameter.

- Multi-layer Perceptron Regressor: This model optimizes the squared-loss using a stochastic gradient descent. Hyperparameters for this model include the size of the hidden layers and the activation function of the neurons. We used grid search and cross-validation to determine the best combination of the parameters that minimize the MSE.
- Random Sample Consensus Regressor: Random Sample Consensus is an iterative method to estimate the parameters of the regression model from a set of inliers of the data set, under the assumption that the outliers are to be accorded no influence on the result of the estimates. It is non-deterministic in the sense that it produces a reasonable result with a certain probability. One could increase this probability of plausible outcome by increasing the number of iterations.

## Challenges

Did you run into any challenges? What worked well vs. what was more difficult than anticipated? Did you try anything without success?

During the preprocessing step, we had to recalculate all of the ratios, since the original data was wrong.

We also ran into a few issues while running the regressors. For one, the Lasso regression trained and ended up removing almost all of the features. That was not a very good sign since it meant, at least for a linear regression, most features were not highly correlated with the label. Another issue we ran into was with the Radius Neighbors Regressors. Calculating cross-validation with MSE did not work for all records since some of them did not have enough neighbors within the chosen radius. This issue was resolved by writing our own MSE function for RNR. Beyond these challenges, the biggest effort was from debugging our code and making all the sklearn regressors work.

## Results

What were your results?

After running all of our regressors, we selected the one with the lowest MSE as the final regressor of that sector. We outputted our final results to Error_Report.csv. Most of the time, the Support Vector Regressor performed the best. Unfortunately, the MSE of our best regressors was always above 1. For some sectors, our MSE was very high; Real Estate had an MSE of 202.9. An MSE of 1 roughly means that we are predicting within ~100% of the actual value, which is a wide margin of error. Since we also had data on the most recent company quarters, we predicted the future net income growth of these unlabeled records. These results are stored in the “final results” folder and are separated by industry.

To compare the results of our regressors to a baseline, we tested the MSE for a predictor which outputs the average “net_income_growth” of that sector. When compared to this naive predictor, our final regressor performed extraordinarily well, reducing the MSE by often more than 10x. This data is also in Error_Report.csv, with the “MSE with naive predict” column corresponding to the naive predictor’s MSE.

## Next Steps

What could be done with this next? How could the results be used in the real world? What additional data analysis could be done on the data? Could additional data be added/supplemented to predict more things? Etc.

Unfortunately, our regressors are far from being useful in real life. The standard way of predicting future net income is where each company makes its own prediction while budgeting the next quarter. Even though our regressors outperformed a naive predictor, they still underperform a company’s own prediction. This is likely due to the fact that a company has the power to make or break their own prediction, and internally they are striving to meet their own expectations. They also have a direct window into the unquantifiable drivers of success, something not possible with our models.

The next steps would involve restructuring the entire problem to get better results. One possibility could be to turn this into a classification problem rather than a regression problem. Instead of trying to predict the exact growth, we could predict whether a company’s net income grows or shrinks in the coming quarter. This would simplify the problem a lot and standardize the times a company’s net income skyrockets or plummets.

We could also try using an even smaller subset of companies. In this project, we predicted on the Russell 1000, which is ~90% of the US stock market’s value. However, the S&P 500, which are the 500 largest companies, is ~75% of the US stock market’s value. Using only the S&P 500 companies might be a worthwhile pursuit. Similarly, we could try predicting on only the Russell 2000, which is the 2000 smallest companies in the Russell 3000 (3000 largest public US companies). It’s possible that smaller companies could be easier to predict on.

Another avenue could be including data about the overall economy in a given quarter as a feature. Since companies tend to perform worse during a recession and better during an expansion, data such as the US’s total private consumption might be valuable.

It’s also possible that the regressors we chose just aren’t well-suited for predicting this data. Perhaps trying different models would yield better results.

However, at its core, our initial assumption that a company’s current financial metrics can directly predict their future net income may be flawed. Despite throwing in lots of data as well as running several regressors on them, the regressors do not improve on the standard practice of predicting one’s own net income. Maybe the real drivers of net income growth are more intangible factors such as employee morale or a stellar business strategy. Such factors can’t appear in a financial statement like a company’s return on assets can. Until we can quantify the unquantifiable factors, the best predictor of a company’s future net income growth may always be their own prediction.