Big Messy Data Project Midterm Report

David Fitzpatrick (df347), Jane Xiong (wx77), Jiaming Liu (jl4286)

November 2, 2021

1 Restatement of project goal

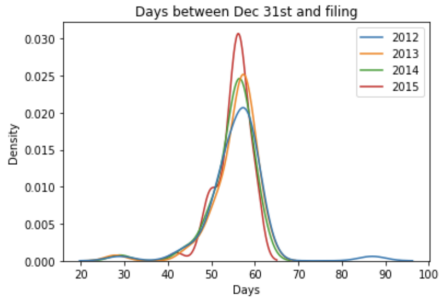
Our updated goal with this project, now informed by our data analysis and the helpful critique of our reviewers is to test the usefulness of company fundamentals and inflation data for stock price prediction. We emphasize the care we put into the non daily-market data sources because we have seen others use this data to generate predictions with information before it would be publicly available.

By reducing the scope of the project to predicting only a subset of S&P 500 companies, manually collecting our data, and controlling for the relevant data release schedules, we hope to provide a more meaningful result.

2 Company selection and analysis of market around public filings

Our main goal in choosing a subset of S&P 500 companies for prediction was to select companies whose public release of year end fundamental data fall very close on the calendar. This would allow us to generate our predictions once a year, following the last public filing. A company's fiscal year end is often related to its industry so we looked at Financial companies and Real Estate Investment Trust companies. From this group we found a list of 60 companies who had filings on the Security and Exchange Commissions EDGAR database for the years 2013-2017 and whose fiscal year ends on Dec 31st, these companies are required to release their results within 60 days of this date, give or take.

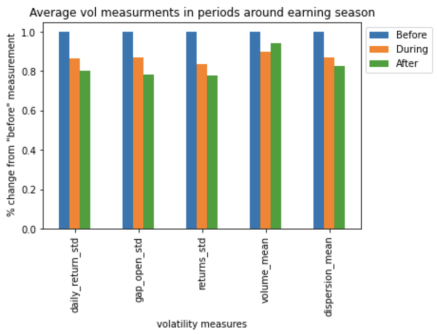
Of the 60 companies, almost all filed within a 20 day window. There were two negligible outliers for 2012. We show a densities of distribution measuring days between public filing and Dec 31st of the previous year.



From this graph we define a period before filing (BF) as 25-45 days days after Dec 31st, during (DF) as 45-65 days, and after (AF) as 65 to 85 days.

This tight distribution of information release will allow us to generate our predictions on March 8th, approximately the 68th day of the year. It falls after all companies information has become public and the information isn't too stale.

As an investigation into financial markets response to this increase in information we created a group of volatility indicators. and examined them before, during and after filings as defined in the previous paragraph. The first indicator is the standard deviation of price change during the day, daily_return_std. the second is the standard deviation of price change while markets are closed, gap_open_std. The third is the standard deviation of the price change at market close today to market close tomorrow, returns_std. The fourth is the average number of shares traded across all stocks. The fifth is the average difference between the day's highest price minus the days lowest price as a percent of the opening price.



We find it striking that four of the 5 volatility indicators are greatest before earnings season begin and decline during and afterword. We hypothesize that this is due to the fact that these are some of the largest companies in one of the most liquid stock markets in the world and therefore therefore analyst's estimates of these companies performance are accurate and most investors have moved into position by the time the companies release their public filings. Potentially a bad sign for our profitability.

3 Inflation data and data cleaning decisions

For the macroeconomic factor, we chose to use inflation data by considering the percent changes of Consumer Price Index from February 2012 to January 2016 in our models. Both seasonally adjusted and unadjusted percent changes are provided in the CPI detailed reports which are available on the U.S. Bureau of Labor Statistics website. As stated in the detailed reports, "for analyzing general price trends in the economy, seasonally adjusted changes are usually preferred since they eliminate the effect of changes that normally occur at the same time and in about the same magnitude every year–such as price movements resulting from changing climatic conditions, production cycles, model changeovers, holidays, and sales." Thus, we believed that it would be more informative to use seasonally adjusted changes as our inflation data even though we also collected the 12-month seasonally unadjusted data.

Our decision was to manually collect the data of seasonally adjusted percent changes of all items together with the ones of specific categories and subcategories from preceding month from the detailed reports and calculate the 12-month cumulative rates from the monthly data. Our methodology for the calculation was to generate the product of one plus the monthly percent changes of each category during the 12-month time period and then subtract one from the product to get the yearly percent changes. However, upon looking at the dataframe generated, we noticed that there existed unreasonable outcomes and hence we decided to check our methodology and redo the calculation for the next step.

Seasonally adjusted data are subject to revision for correcting existing errors and updating using recalculated seasonal factors. Some revisions were posted within months after the CPI detailed report was originally released, and some revisions were not updated until 6 months later. Although both the original data and revised data were collected, we made the data cleaning decision to only preserve the revised percent changes. The majority of the data revisions happened at least 3 months before March when we generate our predictions. Since our feature is the cumulative percent changes of inflation from the previous year ending at the end of January, which was released in the middle of February, we would be able to generate the prediction in March. Due to revisions and the fact that we manually collected the data, inaccuracy potentially exists but in our analysis, we assumed the data were correct. For the next step, we are going to further integrate the inflation data into the analysis.

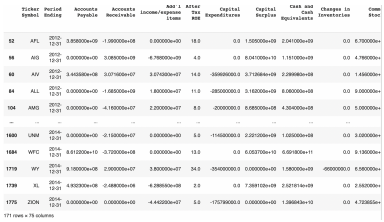
4 Fundamentals data cleaning decisions and preliminary regression results

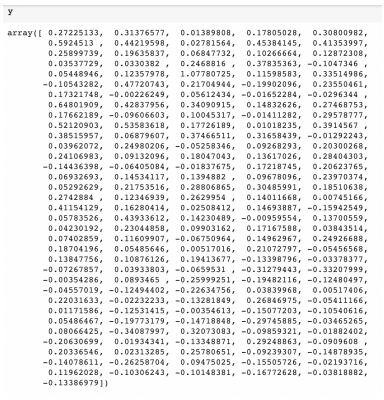
For data cleaning, we first selected Financial companies and Real Estate Investment Trust companies. And then we select the companies with fiscal year ending on Dec 31st. After that, we now have 60 companies from year 2012 to 2014. Next, we clean up missing values. We found all the attributes with more than 1/4 of their values being NaNs and deleted them. Next, we impute all the remaining NaNs with 0.

Next, we find the y. We calculated the stock returns from April to April from period 2013-2014 to 2015-2016 for

the the companies. Then, we delete the companies with missing values, which left us with 57 companies.

Below are fundamentals data (x) and stock annual returns data (y):





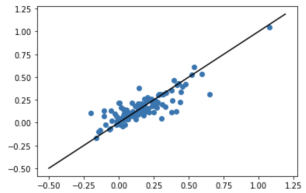
```
y

array([ 0.27225133,  0.31376577,  0.01389808,  0.17805028,  0.30800982,
        0.5924513 ,  0.44219598,  0.02781564,  0.45384145,  0.41353997,
        0.25899739,  0.19635837,  0.06847732,  0.10266664,  0.12872308,
        0.03537729,  0.0330382 ,  0.2468816 ,  0.37835363, -0.1047346 ,
        0.05448946,  0.12357978,  1.07780725,  0.11598583,  0.33514986,
       -0.10543282,  0.47720743,  0.21704944, -0.19902096,  0.23550461,
        0.17321748, -0.00226249,  0.05612434, -0.01652284, -0.0296344 ,
        0.64801909,  0.42837956,  0.34090915,  0.14832626,  0.27468753,
        0.17662189, -0.09606603,  0.10045317, -0.01411282,  0.29578777,
        0.52120903,  0.53583618,  0.17726189,  0.01018235,  0.3914567 ,
        0.38515957,  0.06879607,  0.37466511,  0.31658439, -0.01292243,
        0.03962072,  0.24980206, -0.05258346,  0.09268293,  0.20300268,
        0.24106983,  0.09132096,  0.18047043,  0.13617026,  0.28404303,
       -0.14436398, -0.06405084, -0.01837675,  0.17218745,  0.20623765,
        0.06932693,  0.14534117,  0.1394882 ,  0.09678096,  0.23970374,
        0.05292629,  0.21753516,  0.28806865,  0.30485991,  0.18510638,
        0.2742884 ,  0.12346939,  0.2629954 ,  0.14011668,  0.00745166,
        0.41154129,  0.16280414,  0.02508412,  0.14693887, -0.15942549,
        0.05783526,  0.43933612,  0.14230489, -0.00959554,  0.13700559,
        0.04230192,  0.23044858,  0.09903162,  0.17167588,  0.03843514,
        0.07402859,  0.11609907, -0.06750964,  0.14962967,  0.24926688,
        0.18704196,  0.05485646,  0.00517016,  0.21072797, -0.05456568,
        0.13847756,  0.10876126,  0.19413677, -0.13398796, -0.03378377,
       -0.07267857,  0.03933803, -0.0659531 , -0.31279443, -0.33207999,
       -0.00354286,  0.0893465 , -0.25999251, -0.19482116, -0.12480497,
       -0.04557019, -0.12494402, -0.22634756,  0.03839968,  0.00517406,
        0.22031633, -0.02232233, -0.13281849,  0.26846975, -0.05411166,
        0.01171586, -0.12531415, -0.00354613, -0.15077203, -0.10540616,
        0.05486467, -0.19773179, -0.14718848, -0.29745885, -0.03465265,
        0.08066425, -0.34087997,  0.32073083, -0.09859321, -0.01882402,
       -0.20630699,  0.01934341, -0.13348871,  0.29248863, -0.0909608 ,
        0.20336546,  0.02313285,  0.25780651, -0.09239307, -0.14878935,
       -0.14078611, -0.26258704,  0.09475025, -0.15505726, -0.02193716,
        0.11962028, -0.10306243, -0.10148381, -0.16772628, -0.03818882,
       -0.13386979])
```

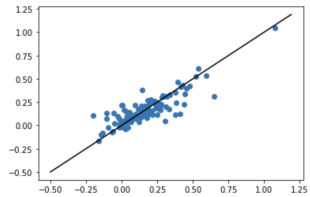
For the preliminary model fitting, we choose linear, ridge, and lasso regressions. We use data from 2013 to 2014 as the training set and data in 2015 as the test set.

After fitting the models, we calculate the R^2 coefficients and plot the predicted returns against true returns on training set:

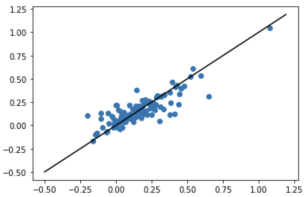
Linear, Ridge, and Lasso regressions:



R^2 coefficient of linear regression: 0.7192474937437807



R^2 coefficient of ridge regression: 0.7192469077693324

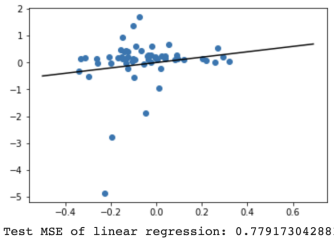


R^2 coefficient of ridge regression: 0.7192469077693324

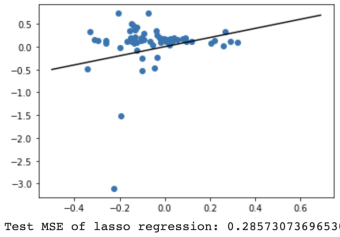
As we can see in the above graphs, all three regressions have good fitting performance on the training data set with relatively high R^2 coefficients.

Next, we test our model performance on the test set and calculated their Mean Squared Error and also plot the predicted returns against true returns:

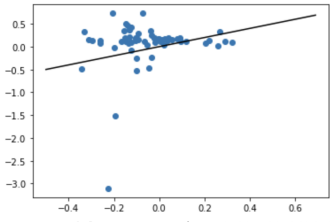
Linear, Ridge, and Lasso regressions:



Test MSE of linear regression: 0.7791730428815412



Test MSE of lasso regression: 0.28573073696530793



Test MSE of lasso regression: 0.28573073696530793

As we can see in the above graphs, the MSEs are relatively high. But the majority of the test data spread around the diagonal line, indicating that the regressions is relatively effective in explaining the test data. So, we will continue from here and try modifying the regression models and try new models in the next steps.