Time Series Troubles when Predicting Stock Returns

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

1 Introduction

Our goal is to test the usefulness of company fundamentals, inflation data, and interest rate for stock price prediction. We emphasize the care we put into the nondaily-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 aim to identify the relationship between the stock performance and company fundamentals with macroeconomic indicators.

2 Economic Data and the Information Release Schedule

Using time series economic data to generate predictions can be tricky. Thorough understanding of the information release schedule is critical to avoid generating predictions using information that would not be available at the time of prediction. As an example here is a subset of our company fundamentals dataset for all S&P 500 companies 2012-2015. A dataset that contains an assortment of company statistics for a 12 month period that makes up the company's fiscal year.

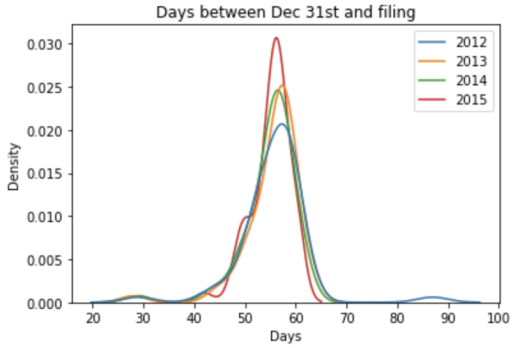
Ticker Symbol	Period Ending	Accounts Payable	Accounts Receivable	Add'l income/expense items
AAL	2013-12-31	4.975000e+09	-9.300000e+07	-2.723000e+09
AAP	2013-12-28	2.609239e+09	-3.242800e+07	2.698000e+06
AAPL	2014-09-27	4.864900e+10	-6.452000e+09	9.800000e+08
ABBV	2013-12-31	6.448000e+09	6.810000e+08	-5.400000e+07
ABC	2014-09-30	1.725016e+10	-9.382860e+08	-2.859400e+07

The period ending column is the end of the fiscal year to which the company statistics correspond. This is a public dataset on Kaggle and we have observed others generating their predictions of company performance for the 12 month period starting on the period ending day. Unfortunately, These company statistics do not become publicly available until around 45 to 65 days after the company's period ending date when the company filings become available on the Security and Exchange Commission's website. This issue in combination with each company using their own fiscal year (period end date) makes generating predictions without information leakage a complicated task.

2.1 Our Simplification

We simplified the problem by generating our predictions once per year. To do this we needed to select companies whose company fundamentals become publicly available around the same period of time to prevent information leakage while reducing information staleness. We selected companies whose period ending date was December 31st. In addition, we wanted to incorporate other macroeconomic variables in our analysis, specifically inflation and interest rates, so we decided to use companies who were particularly sensitive to these economic variables. This reasoning led us to choose companies from the Financial and Real Estate sectors.

With this list of 57 companies we manually compiled a file containing the date that each companies fundamentals data became public on the Security and Exchange Commissions EDGAR database and plotting the number of days since December 31st for each year in our dataset.



Most companies Fundamental statistics become available within 45-65 days of their Period End

From these densities we decided to generate our predictions on the 68th day of each year in our fundamentals dataset. This translates to generating predictions on march 7th-8th depending on the year.

2.2 Our Datasets

Company Fundamentals

Our company fundamentals dataset consists of range of balance sheet(asset and liability) and income statement(income and expenses) line items as well as company profitability metrics and few related features like number of company shares outstanding. In total there are 74 columns of company statistics in this dataset and this is for the companies that make up the SP 500 for the years 2012-2015. As mentioned we reduced the number of companies down to just 57 for our analysis.

Inflation Data

Our first macroeconomic indicator for prediction is inflation data. This data is relevant to financial companies and real estate companies because these businesses rely on long term assets and liabilities whose pricing are directly affected by the rate of inflation.

We gathered the monthly seasonally adjusted percent change in price level for each category of goods tracked by the Bureau of Labor Statistics as well as their aggregated Consumer Price Index for the years 2012-2015.

In this process we discovered their inflation reports come out two weeks after the end of the month which means that the most recent inflation statistics available on the date of prediction, March 8th, will be the January numbers. Due to this, the inflation data we used for our predictions was the percent change in price level from February of the previous year through the January two months before we generate our predictions.

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.

```
'All items',
'All items less food and energy',
'Apparel',
'Commodities less food and energy commodities',
'Electricity',
'Energy',
'Food',
'Food at home',
'Gasoline (all types)',
'Medical care services',
'New vehicles',
'Services less energy services',
'Shelter',
'Transportation services',
'Used cars and trucks',
'Utility (piped) gas services'}
```

In total there are 16 features in this dataset

Interest Rate Data

The last macro economic variables we thought would be useful is daily market interest rate data. Short (3 mon), medium (5yr), and long (10yr, 30yr) term interest rates are fundamental to Financial companies and Real Estate companies for the same reason as inflation. For this reason we used the Python library Pandas Datareader to scrape daily interest rate data from the yahoo finance api and then used the nominal interest rate on the day before we predict as a set of features as well as the percent change from 1 month ago and percent change from 6 months ago to ideally capture some momentum information. In total there were 12 features in this dataset.

Stock Returns

This is from the same Kaggle dataset as the stock fundamental information. This we had stock returns for 2013-2016 unfortunately because we are predicting March 9th to March 7th returns for each year this translated to only 3 full years of data for us.

Train/Test Split

With only 3 years of data for 57 companies we now realize we set ourselves up for a difficult situation. because there is also a time series component to this data it made the most sense to use to train on the first two years of data which we treat as independent samples, certainly an assumption that is violated but perhaps still useful, and we Test our model on the last year of data. Due to the limited number of 171 samples and a total of 95 features (after dropping mostly zero columns as described in the preprocessing section) lack of data and overfitting remained a large concern throughout this project.

3 Data Prepossessing Techniques

In order to successfully perform our machine learning prediction task, we need to modify our dataset to make sure it is suitable to train our models on and therefore make better prediction. Thus, we adopted 4 data preprocessing techniques, which are Data Cleaning, Mean Imputation, Data Scaling (Standardization), and PCA.

Besides, in order to test the effectiveness of the data preprocessing techniques, we will train our models on 3 stages of datasets and compare the fitted models' prediction performance. The 3 stages of datasets include Unscaled Dataset (dataset after mean imputation), Scaled Dataset (dataset after data scaling), and Preprocessed Dataset (dataset after PCA).

3.1 Data Cleaning

The raw company fundamentals dataset is likely to have missing values. This is because company annual reports may have different names for the same concept or special items that do not appear frequently. For example, there is a kind of items called nonrecurring items, which only appear under special condition. Due to this problem, we need to first drop the data columns where missing values are in the majority. In our project, we dropped the data columns where missing values are over $\frac{3}{4}$.

3.2 Mean Imputation

After the preliminary data cleaning, we now look at the rest of the missing values. There are still many attributes in our dataset that have missing values. To deal with this problem, we adopted the mean imputation technique, which fill in the missing values with the mean of other values in that column.

Now, we obtained our first dataset, which we call the unscaled dataset.

mean_imputed_x_train

	Accounts Payable	Accounts Receivable	Add'l income/expense items	After Tax ROE	Capital Expenditures	Capital Surplus	Cash and Cash Equivalents	Common Stocks
Date								
2013-03-08	3.858000e+09	-1.990000e+08	-5.359449e+07	18.0	-4.481093e+08	1.505000e+09	2.041000e+09	6.700000e+07
2013-03-08	1.332681e+10	3.085000e+09	-6.768000e+09	4.0	-4.481093e+08	8.041000e+10	1.151000e+09	4.766000e+09
2013-03-08	3.443580e+08	3.071600e+07	3.074300e+07	14.0	-3.599260e+08	3.712684e+09	2.299980e+08	1.456000e+06
2013-03-08	1.332681e+10	-1.685000e+09	1.800000e+07	11.0	-2.850000e+08	3.162000e+09	8.060000e+08	9.000000e+06
2013-03-08	1.332681e+10	-4.160000e+07	2.200000e+07	8.0	-2.000000e+07	8.685000e+08	4.304000e+08	5.000000e+05
...	...	...	...	...	...	...	...	...
2014-03-07	1.332681e+10	-1.967000e+08	-5.359449e+07	10.0	-1.055000e+08	2.634100e+09	9.410000e+07	3.610000e+07
2014-03-07	6.643600e+10	-1.300000e+07	-5.359449e+07	13.0	-4.481093e+08	6.029600e+10	5.608780e+11	9.136000e+09
2014-03-07	9.720000e+08	-2.700000e+07	5.400000e+07	8.0	-2.610000e+08	1.307806e+10	8.300000e+08	7.290000e+08
2014-03-07	3.023990e+08	-8.765100e+07	2.824300e+07	11.0	-4.481093e+08	7.994100e+09	1.800832e+09	2.783000e+06
2014-03-07	1.332681e+10	-5.558520e+07	-1.201920e+08	4.0	-8.858000e+07	1.307806e+10	1.395780e+10	4.179024e+09

114 rows × 95 columns

unscaled dataset

3.3 Data Scaling (Standardization)

Different financial metrics and macro indicators have different measurement of scale, which will unfairly affect the weight of that attribute in the machine learning prediction task. In order to eliminate the effect of different unit of measurement, we performed data scaling, also known as standardization, to our unscaled dataset. The specific formula for standardization is $data = \frac{data - mean}{standard\ deviation}$

Now, we obtained our second dataset, which we call the scaled dataset.

x_train_scaled_df

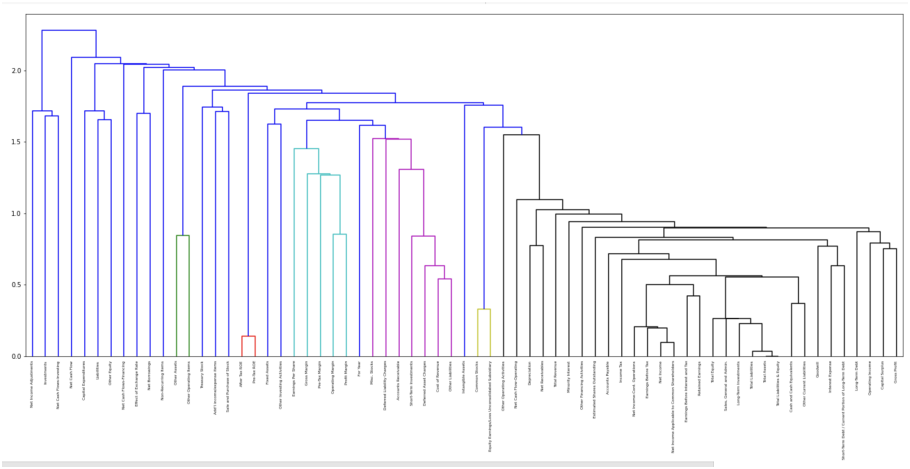
	Accounts Payable	Accounts Receivable	Add'l income/expense items	After Tax ROE	Capital Expenditures	Capital Surplus	Cash and Cash Equivalents	Common Stocks
0	-0.287506	-2.135374e-01	9.540993e-18	0.197941	0.000000	-5.409715e-01	-0.338896	-0.170005
1	0.000000	4.676174e+00	-8.598269e+00	-0.317945	0.000000	3.147368e+00	-0.345555	0.058490
2	-0.394192	1.284981e-01	1.080001e-01	0.050545	0.164081	-4.377754e-01	-0.352446	-0.173192
3	0.000000	-2.426117e+00	9.168178e-02	-0.060002	0.303494	-4.635166e-01	-0.348137	-0.172825
4	0.000000	2.082326e-02	9.680407e-02	-0.170549	0.796573	-5.707241e-01	-0.350947	-0.173239
...	...	...	...	...	...	...	...	...
109	0.000000	-2.101129e-01	9.540993e-18	-0.096851	0.637485	-4.881928e-01	-0.353463	-0.171508
110	1.612578	6.340722e-02	9.540993e-18	0.013696	0.000000	2.207158e+00	3.842510	0.270987
111	-0.375135	4.256192e-02	1.377823e-01	-0.170549	0.348150	-8.915720e-17	-0.347957	-0.137814
112	-0.395466	-4.774437e-02	1.047987e-01	-0.060002	0.000000	-2.376447e-01	-0.340693	-0.173128
113	0.000000	2.218708e-17	-8.528281e-02	-0.317945	0.668968	-8.915720e-17	-0.249730	0.029948

114 rows × 95 columns

scaled dataset

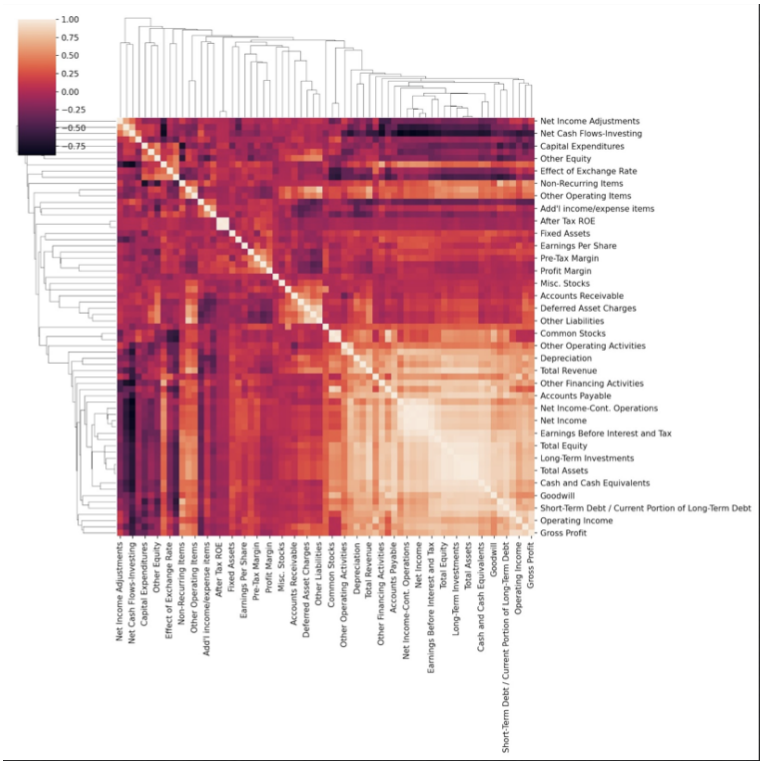
3.4 PCA

The last dataset we ran each of our models on is a dimensionally reduced dataset which will help with overfitting and reduce the overall correlation within our features. For this dataset we clustered our fundamentals data as is shown in the Dendrogram below.



Notice the black highly correlated features which are dissimilar from the rest and the large purple cluster as well.

From this Dendrogram we decided to run PCA on black features as well as the purple features because they had very high correlation and a good amount of features. We dropped all but one from the other highly correlated features recognizable by their non blue color. This reduced the number of fundamental features down to 35.



correlation matrix before dimensional reduction

4 Model Outcomes

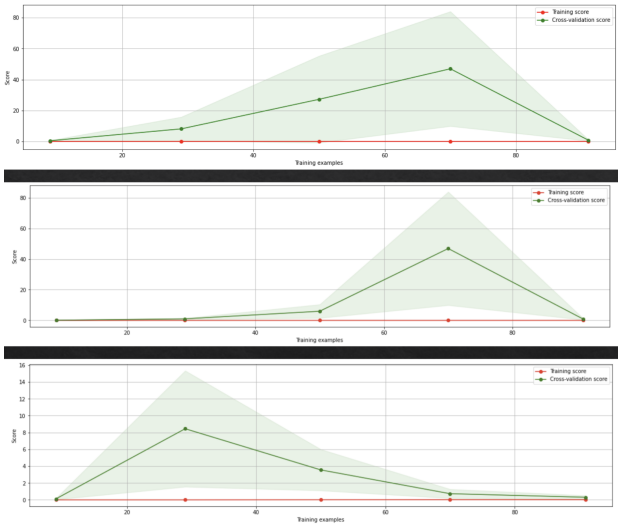
4.1 Learning Curves

We used learning curves to test model performance. Learning curves are plots displaying training and cross-validation scores for different training set sizes. With our chosen metric, mean squared errors, learning curves demonstrate how training and validation errors will change when training set size increases, which helped us to determine the source of our models error, either bias, variance, or not enough data [1]. Learning curves are helpful tools for analyzing model behavior [2] so that we could compare models in terms of the extends of underfitting and overfitting. In addition, we

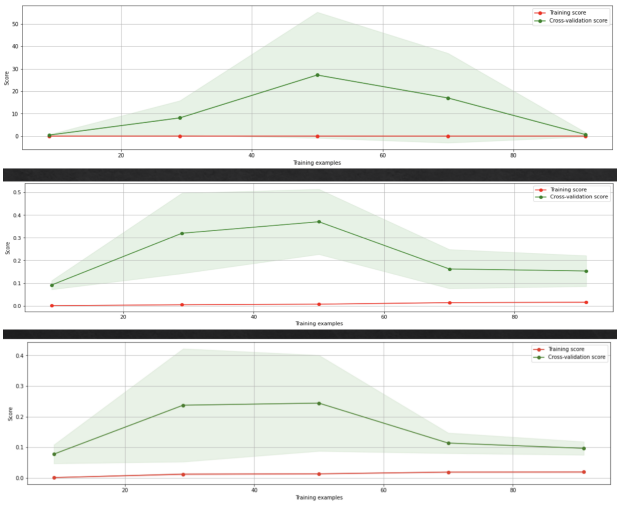
got information of how relatively representative our datasets are from learning curves. There are high chances that a dataset contains too few samples is unrepresentative, meaning that it is likely to miss the statistical characteristics between training and validation sets [2].

We performed experiments on both linear and nonlinear models. We chose linear regression model and ridge regression model as our linear models; random forest and gradient boosted tree models were selected for nonlinear models. We used linear regression model as our first attempt since it is the simplest regression model. We then tried ridge regression, adding quadratic regularizer into the loss function to test a linear estimator that is more robust to poorly presented data. We chose random forest due to its ability to capture conditionally dependent relationships and its lack of a linear assumption. In addition, it is more robust than any single decision tree due to it being an ensemble method which reduces variance without increasing bias. Lastly, we tested our datasets on gradient boosted tree method which is an ensemble method that combines regression trees sequentially, and the latter tree predicts the residuals of the last. These Gradient Boosted Trees are routinely among the best performing machine learning Methods and generally robust to overfitting although they are very susceptible to being influenced by outliers. For each model, we collected results from running experiments on unscaled dataset, scaled dataset, and PCA preprocessed dataset.

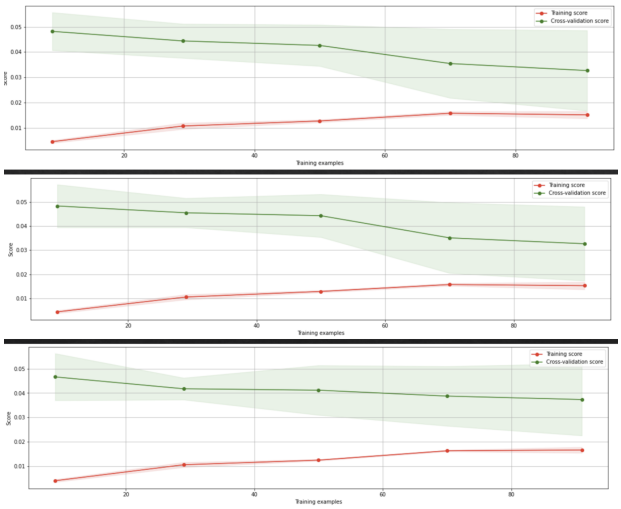
From top to bottom: learning curves generated by running models
on unscaled, scaled, and PCA prepossessed datasets



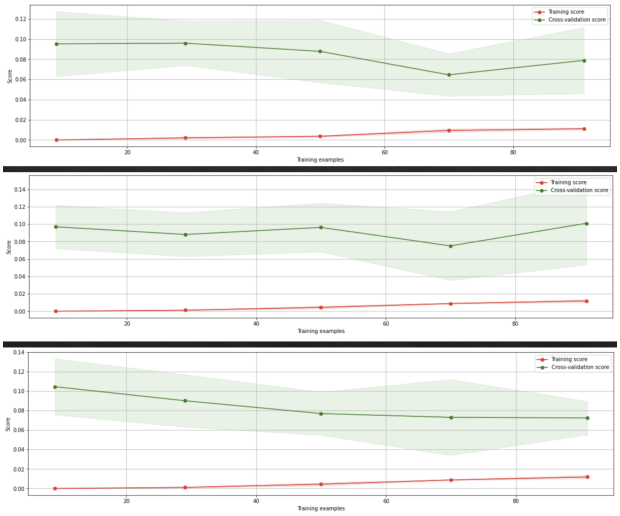
linear regression



ridge regression



random forest



gradient boosted tree

4.2 MSE from Learning Curves

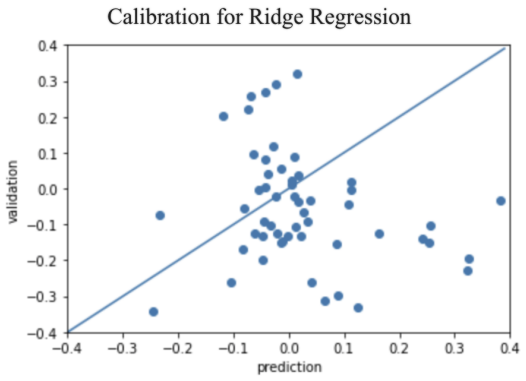
training MSE	
linear regression - unscaled	0.007543803363
linear regression - scaled	0.007543742649
linear regression - pca	0.01857239235
ridge regression - unscaled	0.01168854645
ridge regression - scaled	0.0158007094
ridge regression - pca	0.01939358943
random forest - unscaled	0.005349181747
random forest - scaled	0.01527990881
random forest - pca	0.01510431731
gradient boosted tree - unscaled	0.01187535607
gradient boosted tree - scaled	0.01187535607
gradient boosted tree - pca	0.01115145052

validation MSE	
linear regression - unscaled	0.8415584838
linear regression - scaled	0.8405544686
linear regression - pca	0.3059838977
ridge regression - unscaled	0.6768304956
ridge regression - scaled	0.1534956251
ridge regression - pca	0.0968040935
random forest - unscaled	0.0435106422
random forest - scaled	0.03264139837
random forest - pca	0.03264796814
gradient boosted tree - unscaled	0.07253316989
gradient boosted tree - scaled	0.1009153871
gradient boosted tree - pca	0.07886178423

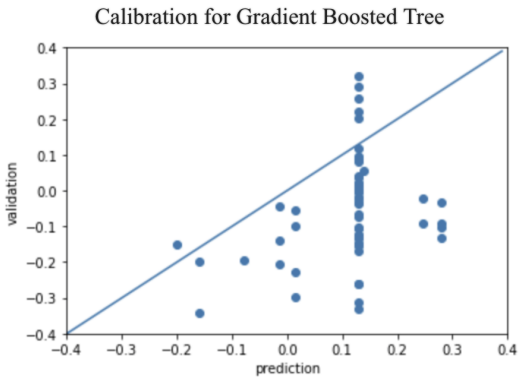
From learning curves and the training and validation MSE, we noticed that between linear models, ridge regression performed better than linear; on the other hand, gradient boosted performed better than random forest between nonlinear models. The results also suggest that dimensional reduction techniques helped: experiments running on PCA processed dataset performed better than the ones on scaled dataset, and the ones running on scaled dataset performed better than those on unscaled dataset in general.

4.3 Calibration

We studied calibration plots for the two models with best performance, which are gradient boosted tree and ridge regression. For analysis purpose, we also calculated MSE and root mean square errors for a reference of error rates of the two models. Due to the noisy shape of plots, we also brought bias into consideration.




```
testing MSE = 0.06895770517617078
root mean square error = 0.26259799156
Bias = 0.0863560559869812
```




```
testing MSE = 0.05167774589409063
root mean square error = 0.22732739802
Bias = 0.15912442261498116
```

From our analysis, we reconsidered the mean of y and noticed that there existed a difference between the mean of y that we used for training and the one for testing, suggesting that y is non-stationary distributed, which could be one possible reason why we got the noisy calibration plots.

```
mean(y_train) = 0.1690963360833558
mean(y_valid) = -0.05586040918119376
```

mean values for comparison

5 Fairness and Future Improvement

5.1 Fairness

We believe our models can not be evaluated by the type of fairness described or metrics provided in lecture. For this reason although we have not provided metrics to evaluate the fairness of our models we have considered additional metrics that might be benefitial in evaluating the fairness of investment models.

The concepts of fairness that might be more relevant are the impact of the companies our model chooses to invest in. Investors such as Liz Simmie of HoneytreeInvest have spoken out in recent years to get investors to consider the impact ones investments have on the Environment and Societal Equality. Carbon emissions are known to contribute to climate change, the effects of which fall disproportionately on developing nations. Likewise, investing in companies with better equality of advancement metrics is likely a societal good and therefore perhaps this would also be a good measure on which to judge the fairness of our model.

Although we don't have the data to make these evaluations these are certainly considerations that we believe would be useful for evaluating and investment model's fairness.

5.2 Future Improvement

The realization of the non stationary y distribution has led us to believe altering the task might be useful. Perhaps predicting a stocks outperformance relative to some benchmark instead of absolute performance would be easier to make our predicted distribution more stationary.

Else after observing the results of our methodology it is obvious that for a project like this to work in practice we would need to increase the scale of the data. This could be done by relaxing the sector constraints to widen the number of SP 500 companies in the dataset. Alternatively one could increase the number of years in the dataset which would open up feature possibilities like measuring change in company features from last year. This would also allow the models to train during both bull and bear markets to have a more for representation of the y distribution.

Lastly, we left the company fundamental dataset untouched for the most part. Performing feature transformations like dividing the balance sheet characteristics by the market cap of the companies may help control for any size bias that our models might be capturing.

6 Conclusion

Via data analysis, our project explored how company fundamentals, inflation, and interest rates affect stock returns. To answer the question we posed, our results confirmed the significant effect operations of publicly listed companies and macroeconomic factors have on stock returns and the viability to make predictions for stock performance using them as our best model pressed the error rate of the prediction to around 22 percent. However, it is not realistic to directly use such a project with non-stationary predicted distribution in production and enterprise decision making. For a project like this to work in practice, further improvements are needed to be made so that both accuracy and confidence in the results will be improved.

7 References

[1] Luz, Adrià. "Why You Should Be Plotting Learning Curves in Your next Machine Learning Project." Towards Data Science, 28 Oct. 2020, https://towardsdatascience.com/why-you-should-be-plotting-learning-curves-in-your-next-machine-learning-project-221bae60c53.
[2] Brownlee, Jason. "How to Use Learning Curves to Diagnose Machine Learning Model Performance." Machine Learning Mastery, 6 Aug. 2019, https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/.