RISK!

General Assembly Capstone Project

Joy Dantong Ma

How Much Risk Is in the Economy?

should I go back to school? should I take the plunge, quit my job, and start my own business?

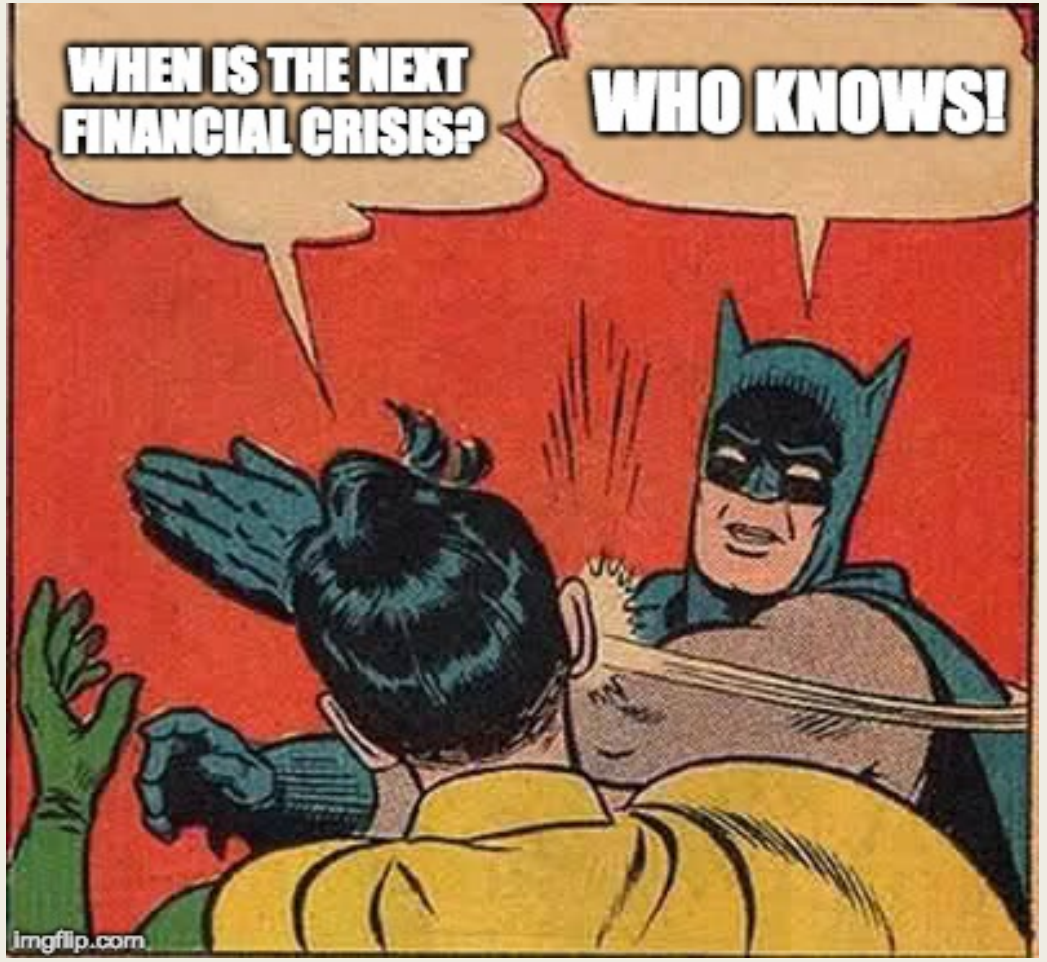
what will the job market look like tomorrow? should I invest in homeownership now?

how aggressively should I diversify my 401k portfolio? should I accept carry-over my capital loss?

Good News and Bad News

bad news: this project won't be able to offer you a very precise answer, because it is a billion dollar question that nobody can quite claim they are absolutely right.

good news: with the assistance of abundant data, we can get a general trend!



How To Discover the General Trend?

First off, we need to redefine the question into a subset of data-rich, programmable questions.

1. instead of the air-like, intangible 'risk', we use one of its closest proxy - stock market performance.

2. another risk indicator is the steepness/slope of treasury yield curve, eps 10yr3mon

3. the last risk indicator is quite well-defined - The VIX Index constructed by Chicago Board Options Exchange

therefore, we re-define the question of "what is the risk level of our current economy?" to "what is tomorrow's stock market performance based on historical volatility and the shape of treasury yield curves?"

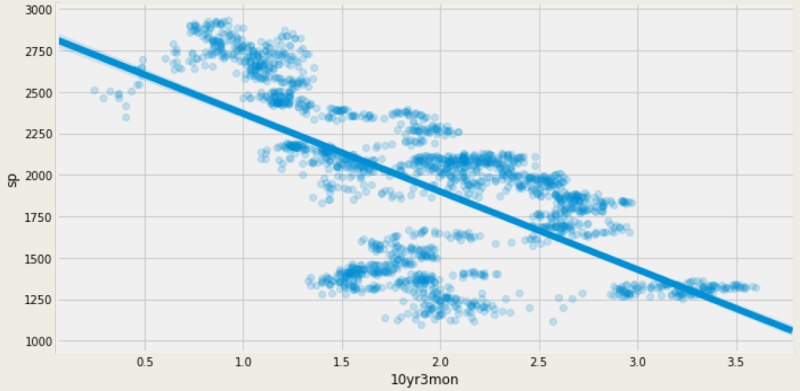
Data Availability

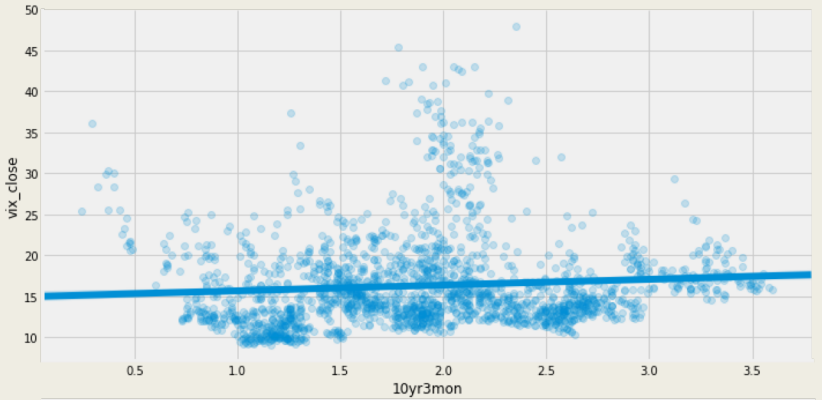
the data I have available is S&P500 historical index from Yahoo Finance, Daily Treasury Yield Curve from United States Department of Treasury, and CBOE Volatility Index

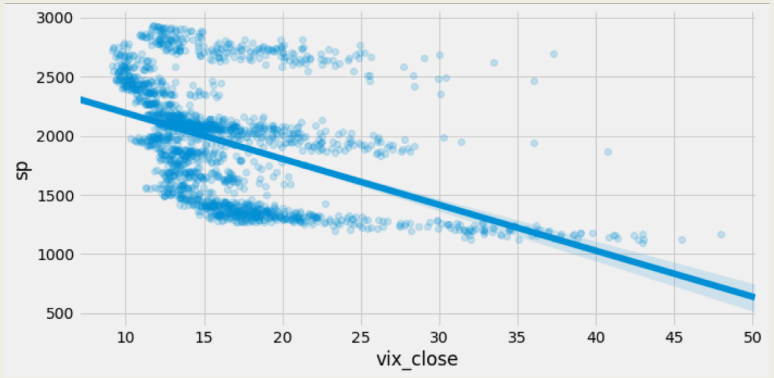
```
def treasury_data(year):
    url = 'https://data.treasury.gov/feed.svc/DailyTreasuryYieldCurveRateData?$filter=year(NEW_DATE)%20eq%20'+str(year)
    r = requests. get(url)
    jsdata = xmltodict.parse(r.content)
    jsdata = xmltodict.parse(r.content)
    length = len(jsdata['feed']['entry'])

    data_dict = {}

    for x in range(0,length):
        rates = []

        date = jsdata['feed']['entry'][x]['content']['m:properties']['d:NEW_DATE']['#text']
        one_mon = jsdata['feed']['entry'][x]['content']['m:properties']['d:BC_1MONTH']['#text']
        three_mon = jsdata['feed']['entry'][x]['content']['m:properties']['d:BC_3MONTH']['#text']
        six_mon = jsdata['feed']['entry'][x]['content']['m:properties']['d:BC_6MONTH']['#text']
        ten_yr = jsdata['feed']['entry'][x]['content']['m:properties']['d:BC_10YEAR']['#text']
        twenty_yr = jsdata['feed']['entry'][x]['content']['m:properties']['d:BC_20YEAR']['#text']
        thirty_yr = jsdata['feed']['entry'][x]['content']['m:properties']['d:BC_30YEAR']['#text']
        rates.append(date)
        rates.append(one_mon)
        rates.append(three_mon)
        rates.append(six_mon)
        rates.append(ten_yr)
        rates.append(twenty_yr)
        rates.append(thirty_yr)
        data_dict[x] = rates

    data_year = pd.DataFrame.from_dict(data_dict,orient='index',columns=['date','one_mon','three_mon','six_mon',
                                                                         'ten_yr','twenty_yr','thirty_yr'])
    return data_year
```

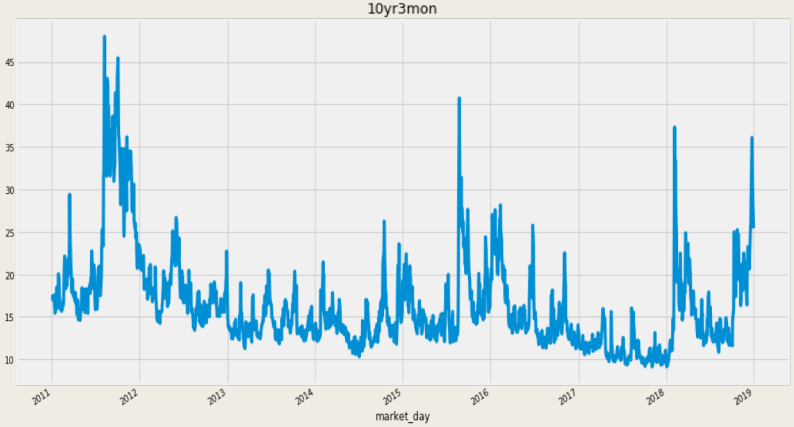
The 1st major takeaway of the Risk Project is the creation of Treasury function. It enables anyone in the future to convert a whole year of treasury yield XML data into a neat DataFrame by simply putting in a 4-digit number of the year.

A Little Exploratory Data Analysis



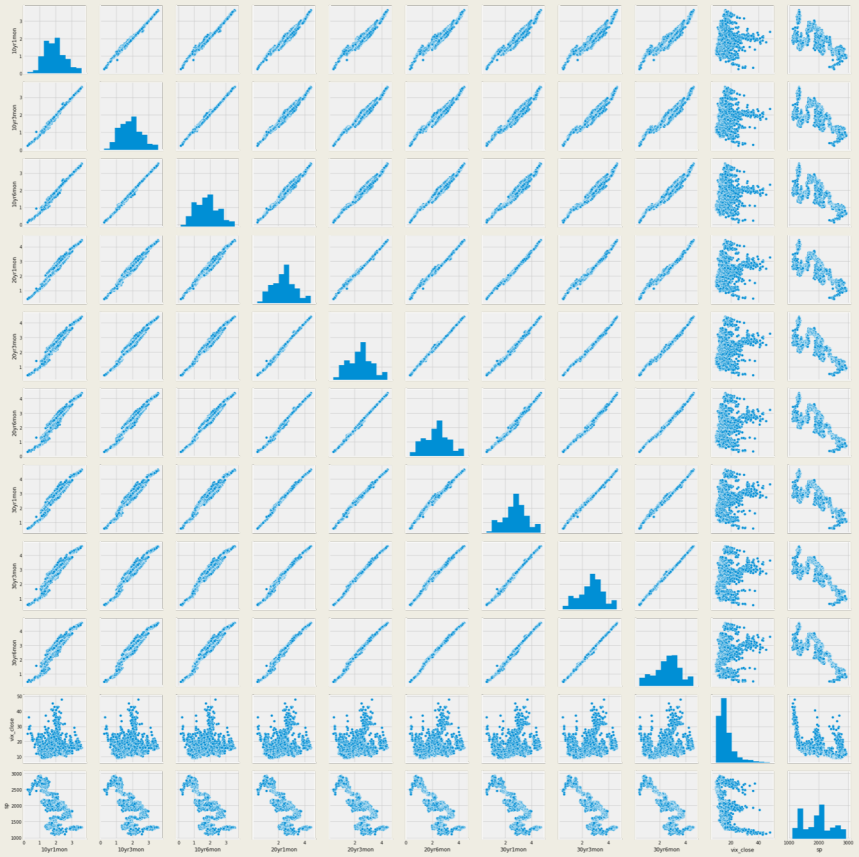


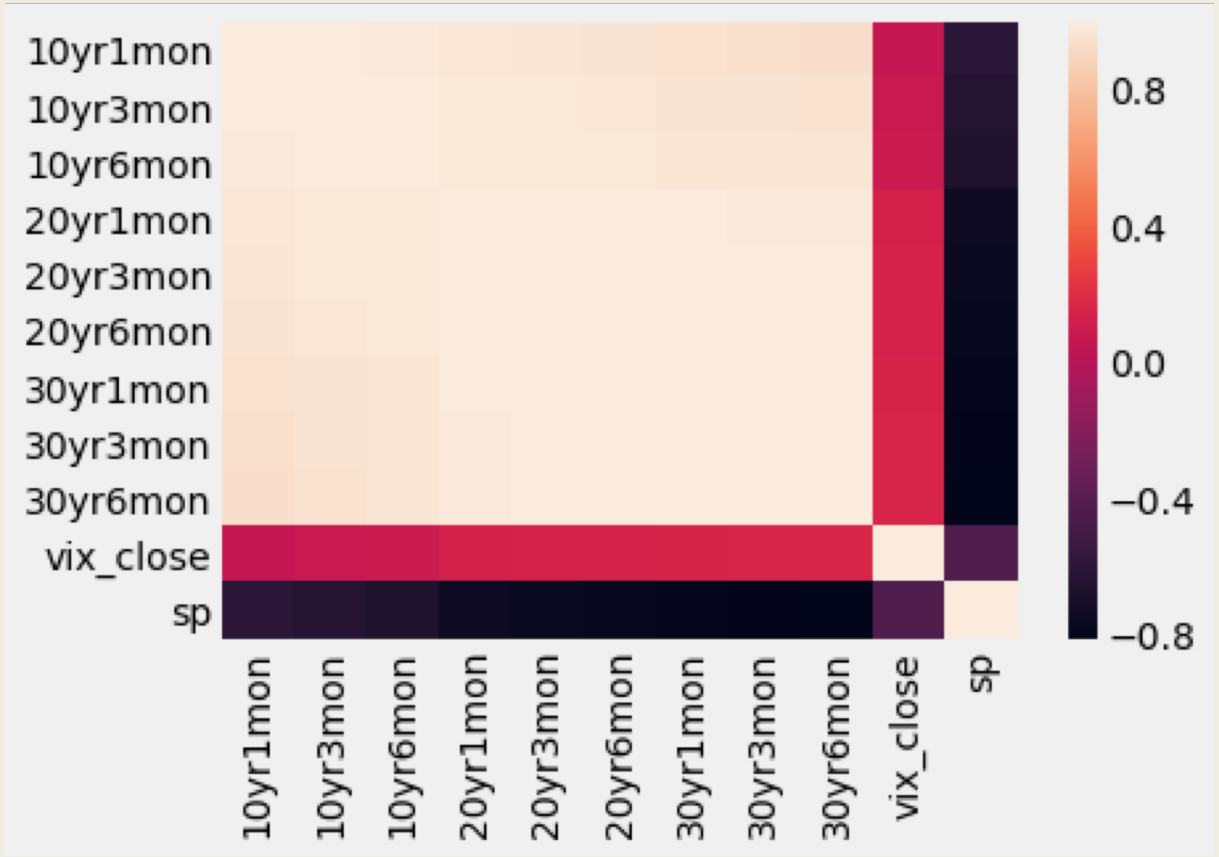




During EDA, I also discovered that the rolling method in pandas dataframe does not deal with missing values when the rolling basis is datetime, which results into inaccurate mean and sum values.
To avoid such mistakes, one should use pad method to fill missing values, which is very common in financial markets. This is the 2nd major takeaway of the Risk Project.

More EDA




10yr1mon
10yr3mon
10yr6mon
20yr1mon
20yr3mon
20yr6mon
30yr1mon
30yr3mon
30yr6mon
vix_close
sp
0.8
0.4
0.0
−0.4
−0.8

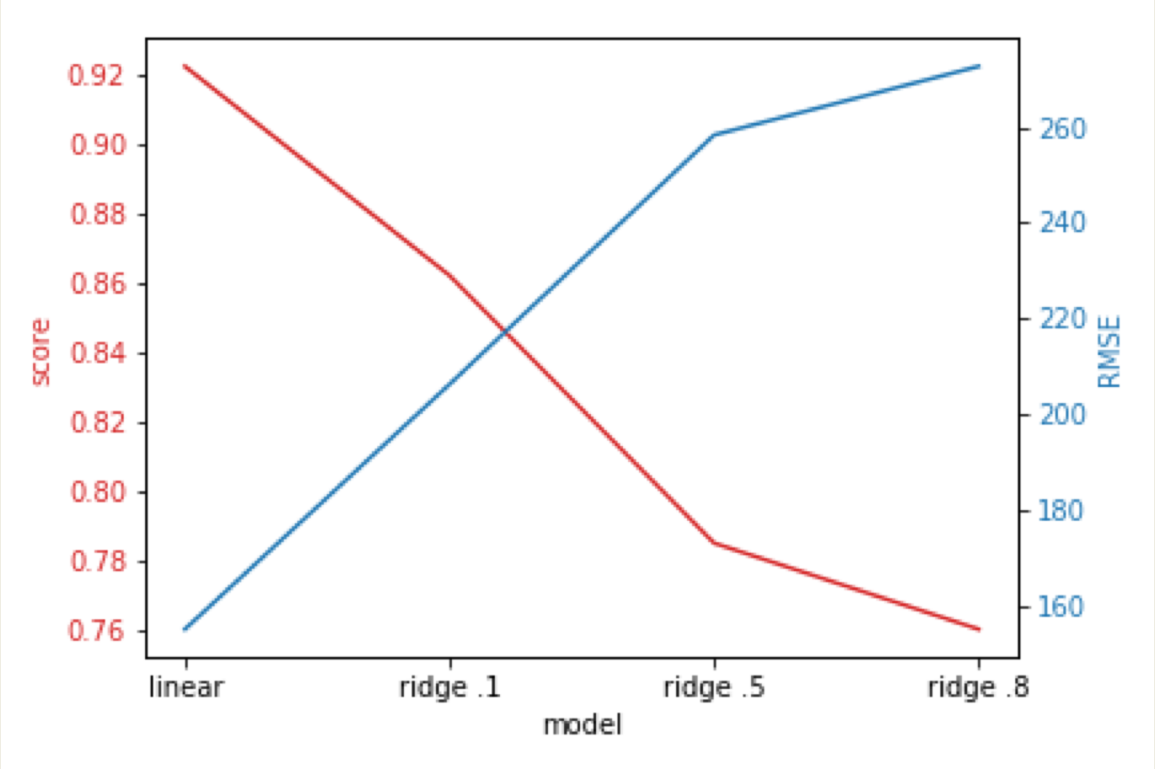
Model Selection: Random Forest

the dataset has following problems for linear regression:

1. it does not exhibit multivariate normality, meaning the data is not normally distributed

2. from domain knowledge, there is a lot of multicollinearity in the dataset.

Random Forest Does Not Make Such Assumptions



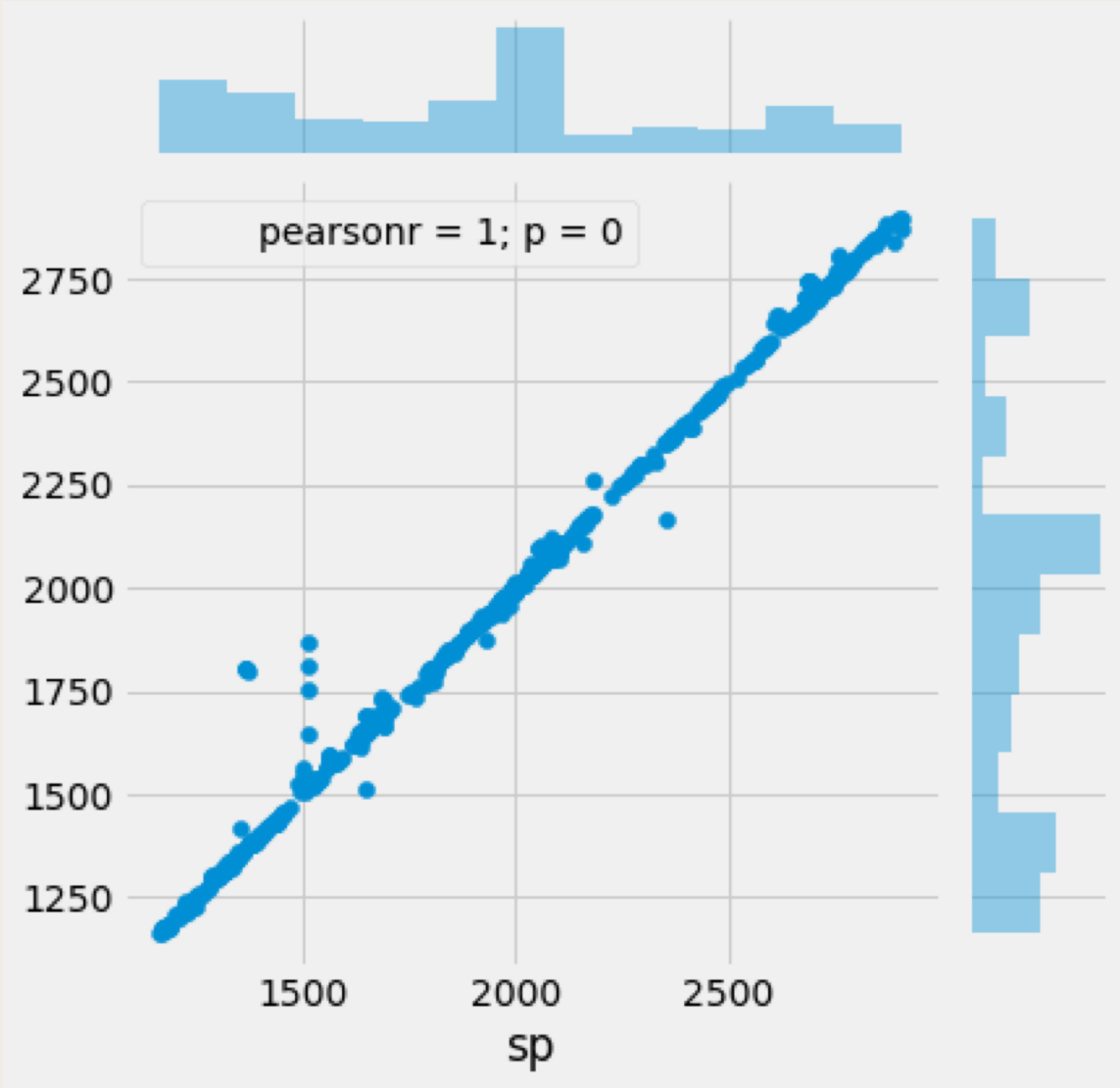
Model Performance

Random Forest reaches a high level of predictive power after rounds of data engineering and training.

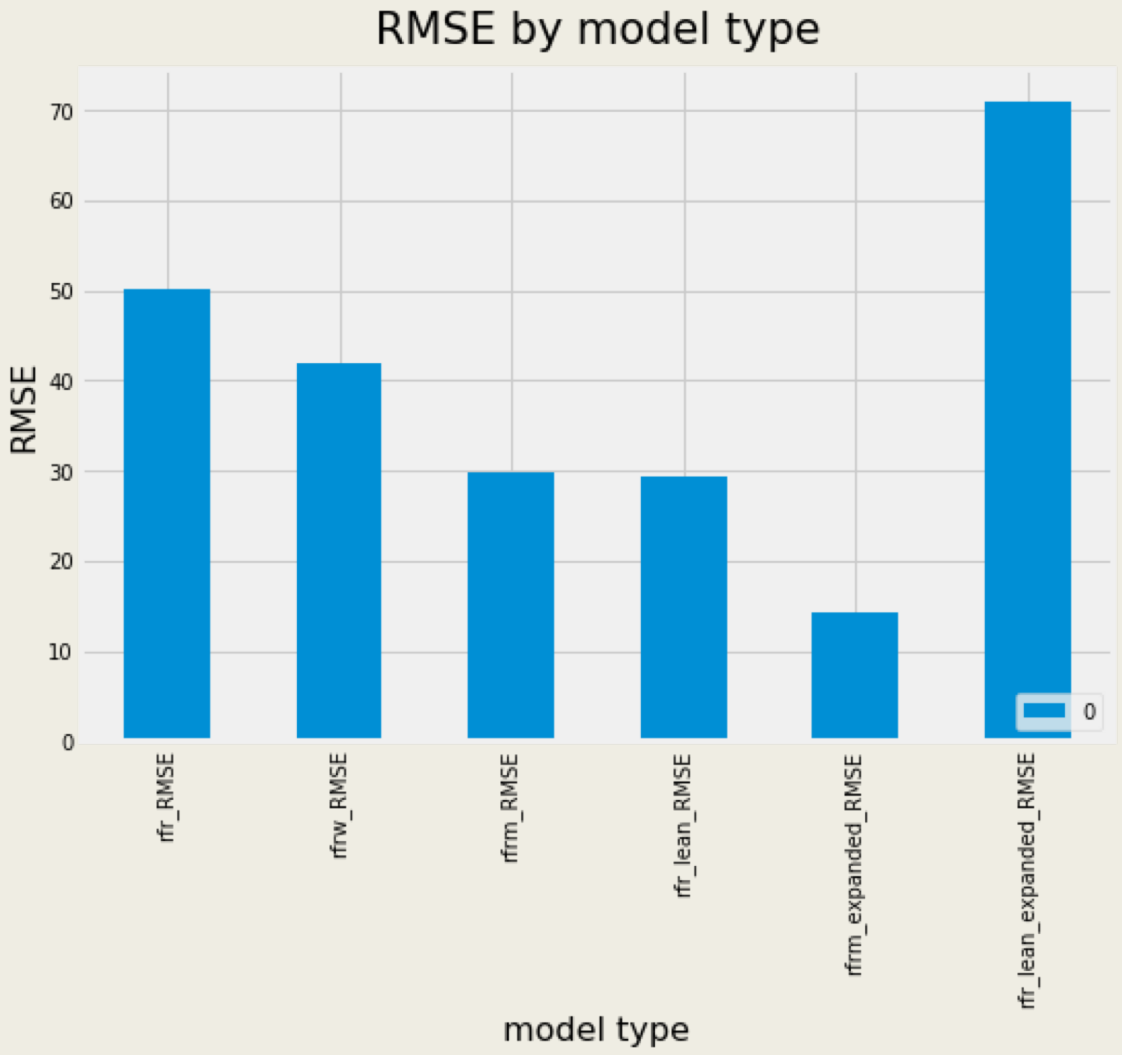
At the peak, the model's RMSE is only 14, or 0.51% of today's S&P 500 index.

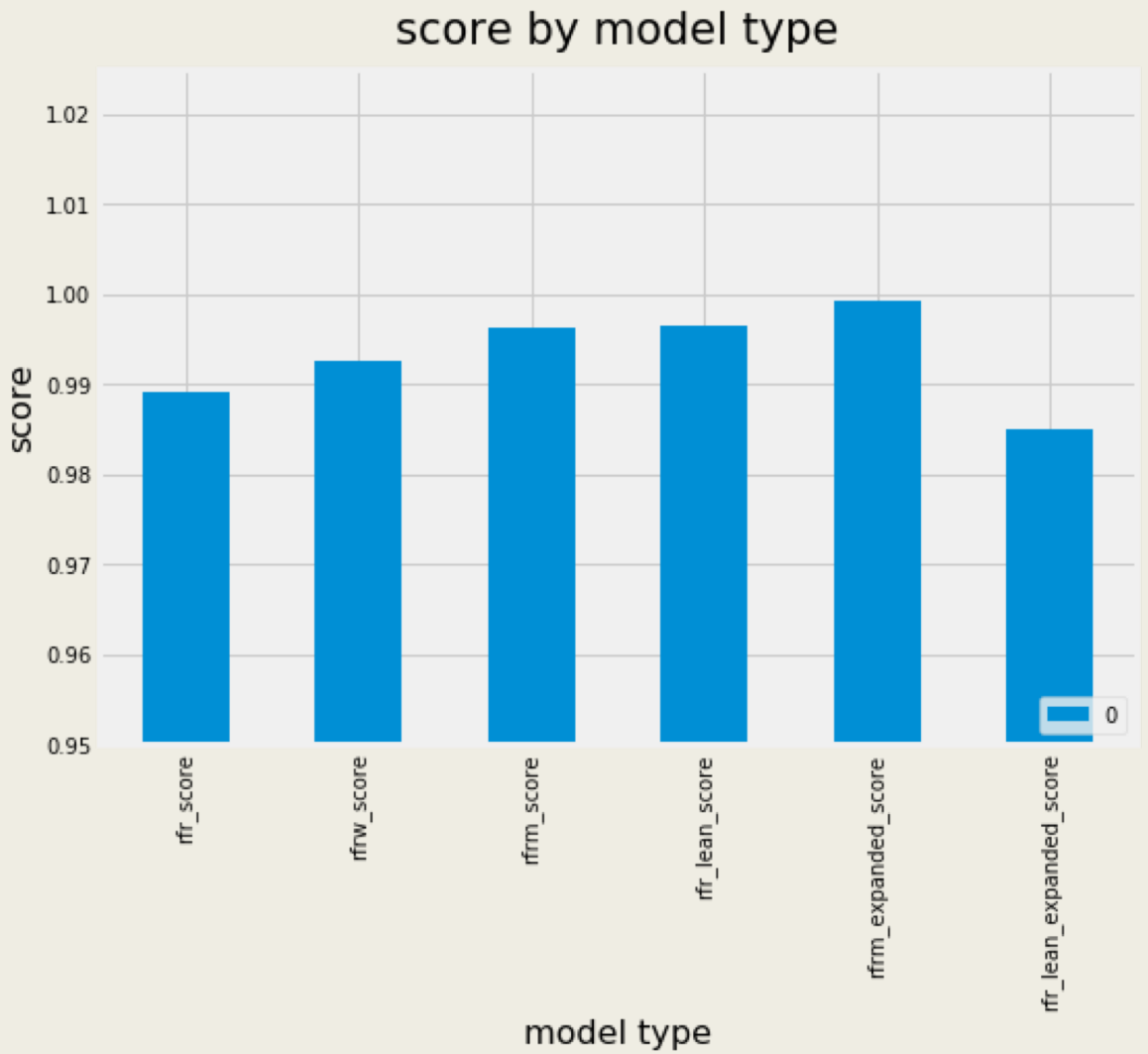
The model's score also reaches 99.943%

This is the 3rd major takeaway of the Risk Project

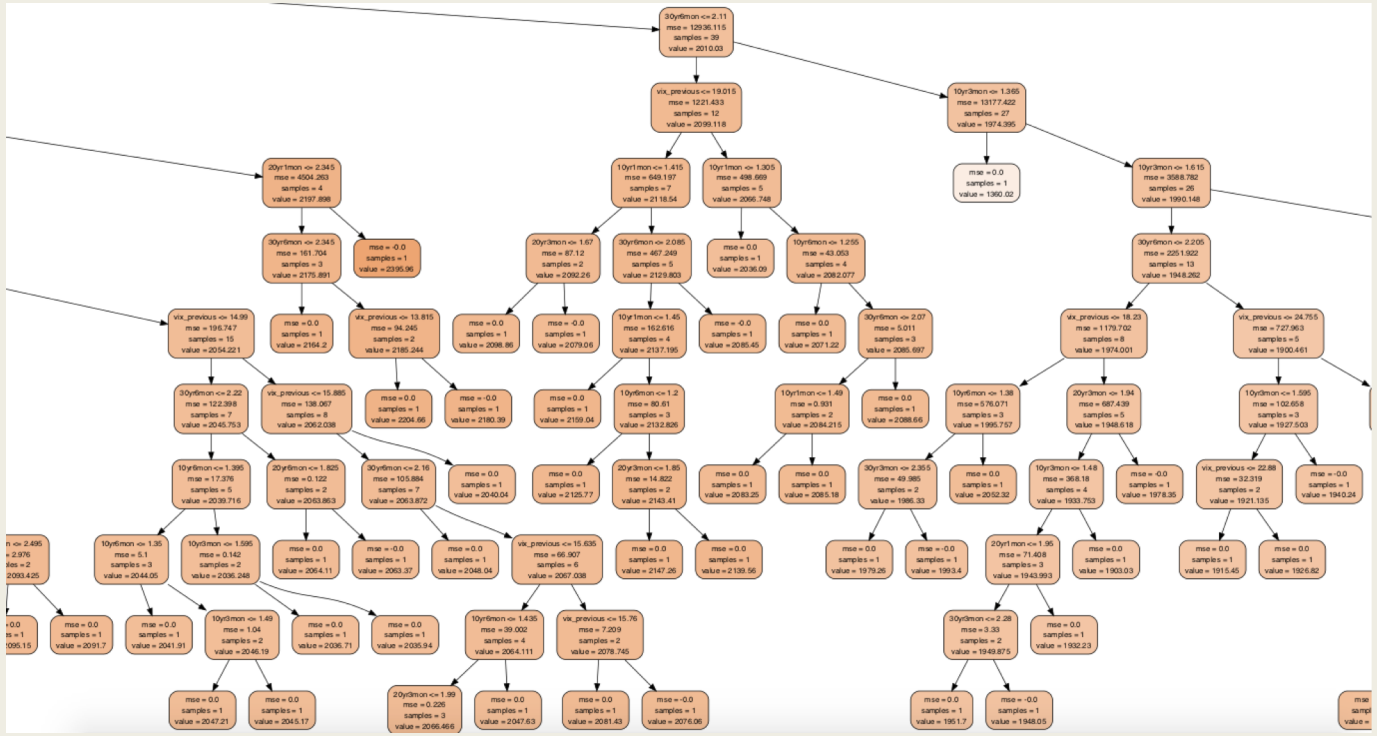


Multiple Random Forest Comparison





Model Interpretation



Looking behind the model, one can see that some features are more important than others.

By looking at feature importance, I discover that the rate difference between 20 year and 6 month actually contributes more than 70% predictive power to the model.

The industry benchmark – 10 year and 3 month – is actually less powerful. This is the 4th major takeaway of the Risk Project.

Next Steps

and

Thank You!