**Data Science Project Protocol**

*Author(s):*

**Tzachi Dvir**

# Introduction

Treasury bond yields (or rates) are tracked by investors for many reasons. The yields are paid by the U.S. government as interest for borrowing money via selling the bond.

Treasury Bills are loans to the federal government that mature at terms ranging from a few days to 52 weeks. A Treasury Note matures in 2 to 10 years, while a Treasury Bond matures in 20 or 30 years.

The 10-year Treasury yield is closely watched as an indicator of broader investor confidence.

The importance of the 10-year Treasury bond yield goes beyond just understanding the return on investment for the security. The 10-year is used as a proxy for many other important financial matters, such as mortgage rates.

This bond also tends to signal investor confidence. The U.S Treasury sells bonds via auction and yields are set through a bidding process. ﻿ When confidence is high, prices for the 10-year drops and yields rise. This is because investors feel they can find higher returning investments elsewhere and do not feel they need to play it safe.

But when confidence is low, bond prices rise and yields fall, as there is more demand for this safe investment. This confidence factor is also felt outside of the U.S. The geopolitical situations of other countries can impact U.S. government bond prices, as the U.S. is seen as safe haven for capital. This can push up prices of U.S. government bonds as demand increases, thus lowering yields.

The U.S. Department of the Treasury issues four types of debt to finance government spending: Treasury bonds, Treasury bills, Treasury notes and Treasury Inflation-Protected Securities (TIPS). Each vary by maturity and coupon payments.

Another factor related to the yield is the time to maturity. The longer the Treasury bond's time to maturity, the higher the rates (or yields) because investors demand to get paid more the longer their money is tied up. Typically, short-term debt pays lower yields than long-term debt, which is called a normal yield curve. But at times the yield curve can be inverted, with shorter maturities paying higher yields.

The 10-year Treasury is an economic indicator. Its yield provides information about investor confidence. While historical yield ranges do not appear wide, any basis point movement is a signal to the market.

Because 10-year Treasury yields are so closely scrutinized, knowledge of its historical patterns is integral to understanding how today's yields fare as compared to historical rates.

While rates do not have a wide [dispersion](https://www.investopedia.com/terms/d/dispersion.asp), any change is considered highly significant. Large changes of 100 basis points over time can redefine the economic landscape.

Perhaps the most relevant aspect is in comparing current rates with historical rates, or following the trend to analyze whether [near-term](https://www.investopedia.com/terms/n/nearterm.asp) rates will rise or fall based on historical patterns.

How do we define the outcome?

I chose the yield of the 10-year US Treasury bond. I will try to predict the 10YUS yield bond in 30 days from now based on the data we have today.

Which questions do we want to answer?

I will try to create a model that predicts the exact number of the 10YUS yield bond with confidence interval of reasonable mistake.

I will try to check if the traditional features that professional investors consider to be the most important to affect the 10YUS yield is correlated with the Data Science.

The data I collected is publish from 1 once a day, 1 once a month, to 1 once a quarter.

The 10YUS yield however is published every day the stock market is open (and even over the counter when it is closed) and marked in the end of the trading day.

# Methodology (Project design)

## Data

I mined the data myself and didn’t used any databank since I was eager to explore the traditional way of the asset allocation in the invested portfolios. I used past trading Data from different trading systems in my work and added Data from www.investing.com and from <https://fred.stlouisfed.org/>

I collected 5 main categories types of variables:

1. Trading bonds of 10 years of other countries which is very important worldwide.
2. Macro indices from the US that usually affects the economy.
3. Stock market indexes that affect the economy worldwide.
4. The most common and used commodities.
5. Interest rates of important indices.

My project will be based on daily time frame. Some variables were measured from 1962 up until November 2020 and some were measured or available only years later.

All the training, the dev and test were based on daily time frame.

I tried to enrich the data by filling some of the missing values as the last known value known or published. This is the right way that every investor faces the information laying in front of him before deciding where to invest his money.

I erased Saturdays and Sundays because there is no trading all over the world in those days.

I ended up with 18,425 lines of data where I had all the values since 1962 and lesser lines depends on each specific variable.

I ended up with 86 variables that in my opinion is the most important to affect my prediction on the outcome variable.

Importing and Exploring the Data

I started with importing all the data to SQL from xls spreadsheets. I used the sql mainly to join the columns I need according to date(daily) column which was the primary key of every column. I removed the duplicates lines which my join function created and created a Flatfile for primary usage.

The data exploration strategy was to explore how my data looks like and to see if the exploration will approve what I know from my experience. I started with descriptive statistics of mean, median, max, min of every variable. Then I plotted all my variables so we will see how it distributes and used sweetviz to visualize.

Then I checked the correlations and there signification. I used spearman correlation since my variables don't distribute normally and created correlation matrix to see how the independed variable change with one another, and inside that matrix I focused on my Y depended variable 10YUS. My second matrix checked the significance of every correlation of the previous matrix.

In addition, I wanted to explore my outcome variable. I checked how the 10YUS distributes by itself and as the years go by. I also checked the 10YUS with 2 variables, one that highly correlated positive and another that correlated negative.

Finally, I presented a glance at the missing before the cleanup. I tested how many missing I have in every column and created heatmap of the missing data.

Manipulate and Data Cleansing

I started manipulating the data in the mine stage. My data starts with daily observations starting from 2.1.1962 and continues up until 6.11.2020. this goes for all of my variables except the outcome. I took the outcome yield and added 30 new observation "from the future", hence from 6.12.2020 till 6.11.2020 and added it to my outcome variable. Then I dropped the earliest  30 days from the beginning and synchronized the columns. This way my outcome variable yield is 30 days ahead of my data y(i)= y (t+30).

I continued with taking care of the Outliers. I used the interquartile rate +/- 150% to get the maximum boundaries of the distribution I'm willing to accept of each and every variable. Then I created 1/0 matrix according the interquartile rate we just got. For the first test I used the Kolmogorov Smirnov test to find the difference between the variable distribution with and without the outliers.

The second test was the fisher test in which I checked If the correlation changed with and without the outliers. Lastly, I looked for variables that were both influenced in the distribution and in the correlation from dropping the outliers, so we'll know I can't drop their outliers.

Secondly, I took care of the missing values. I had missing as a result of data not available in the early years and the data filled with values as the years advanced and I got missing from the Outliers dropping too. I started to drop the columns which had significant amount of missing and went ahead with the drop of rows with the same way of thinking. I used Kolmogorov Smirnov to test the differences in the distribution to conclude if the missing laying ahead is MNAR or MAR as I will explain later.

The result was 76 columns with 7841 rows starting from 1990.

Feature Selection

I tried to find which variables will help to predict best my prediction. I started with the Univariable Analysis where I decided to choose those whose correlation with the outcome variable is significant and since most of them were, I added the condition of correlation higher than 0.7 with the outcome variable. furthermore, I used LASSO (L1), decision Tree, Random Forest, SVM, Ridge, Elastic Net and Linear Regression. I chose to use all 3 regularization features because my variables have a lot of multicollinearity between them. I dropped in final features decision Tree, Random Forest, Gradient Boosting, SVM since they don’t contribute to the selection.

I filtered the variables that got 2 selection at least from the 5 tests.

I ended up with 52 variables from 76.

## Models

Data Partition

I planned to divide my data to 20% - 80% test-train. The result was 1569 rows to the test and 6272 rows to the train. Then I divided again the train to 20%-80% dev-train. The result was 1255 rows to the dev and 5017 rows to the train.

My outcome variable is continuing one and need no balancing. I used the Regression versions of the models to assess the predictions. I used cross-validation in addition which I preformed on the train after the first partition (6272 rows).

Evaluation metric

The evaluation metric for this work is Mean Absolute Error (MAE). Because this metric is one of the most resistant metric to outliers that's possible using simple methods and penalizes huge errors. Thus, it is the fairest measure to calculate the 10YUS on the given data.

The mean absolute error is a common measure of forecast error in time series analysis.

Model Selection

I got all the result of the model prediction lined up in a table. The first and best result was accepted by the KNN and stands better than the rest. However, in second look you can see It overfitted since it has big differences between the MAE-Train and the MAE-test.

In second place I got the DecisionTree which overfitted much more. The GradientBoostingMachine came in third place but didn’t overfit. All the rest models didn't overfit but got lower results.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | model | score-Train | score-Test | MAE-Train | MAE-Test | Cross val mean |
| 5 | kNN | 0.998615 | 0.997417 | 0.050 | 0.067 | -0.067003 |
| 2 | DecisionTree | 1 | 0.996796 | 0.000 | 0.069 | -0.075354 |
| 7 | GradientBoostingMachine | 0.993675 | 0.99295 | 0.120 | 0.124 | -0.130115 |
| 1 | LinearRegression | 0.985133 | 0.984739 | 0.182 | 0.180 | -0.184847 |
| 6 | AdaBoost | 0.981419 | 0.980678 | 0.215 | 0.212 | -0.215094 |
| 4 | SVM | 0.905127 | 0.899338 | 0.476 | 0.478 | -0.478808 |
| 3 | Random Forest | 0.730515 | 0.725124 | 0.815 | 0.797 | -0.817743 |
| 8 | XGboost | 0.364503 | 0.335638 | 1.326 | 1.331 | -1.327656 |

Hyperparameter Finetuning

Last but not least, I tuned my chosen model, GradientBoostingMachine. I started with random fine-tuning rgarding n\_estimators, max\_features, max\_depth, min sample leaf and min samples split. The result was:

|  |
| --- |
| n\_estimators: 400 |
| min\_samples\_split:3 |
| min\_samples\_leaf: 3 |
| max\_features: sqrt |
| max\_depth: 40 |

I tested the MAE with the optimized result and got 0.0474 comparing to 0.1241 I got with the base model. In the second round of the finetuning I tried to test the model with less Hyperparameter and got no change.

I tested the MAE with the second optimized result and got 0.0472

Evaluation of the final model on Test set

I tested the tuned model I got on data it never seen before and got 0.048 MAE result.

# Results

I decided to split the Data to 3 random parts although it is Data that depended on time (i.e. Time Series). when I splitted the data into 3 non-random parts, the models didn't know how to predict the outcome, so I went along with the regular way.

I started with 18,423 rows and 87 variables including the outcome. I ended with 7840 rows and 86 variables.

The test consists 1569 rows and 86 variables

The Dev consists 1255 rows and 86 variables

The Dev consists 5017 rows and 86 variables

14 variables had outliers that we can't drop. All the rest outliers were dropped and were converted to NaN.

I took care of the missing by dropping the columns with more than 70% missing and went ahead with the drop of rows with 30% missing and more. Then I created missing matrix of 1/0 and tested the drop using Kolmogorov Smirnov to test the differences in the distribution (as done with outliers) before the drop and after. If we got significant result i.e. MNAR I categorized the variable into 11 levels and 1 more level of missing. On the other hand If I got MAR I imputed the missing using fancyimpute.

28 variables resulted as MNAR

46 variables resulted as MAR

# Conclusion

# In conclusion I started this project to check If the traditional way of analyzing the bonds market as I learned in the past 15 years, is correlated with the machine learning data analyzing. I learned that there are a lot of variables that affects the bonds market and don’t get the proper attention in the investor's world, and that the human investor can't comprehend how all of the variables changes at the same time. I faced a lot of challenges when I developed the project. Starting from mining the data as some of the variables go way back until 1960 and some where available only from the late years. That fact alone caused a lot of missing data and I had to clean and to cut the data so the models could handle it. Then I had to confront the fact that a lot of my variable are highly correlated with one another so I used all the 3 regularization Feature selection to deal with the betas inflation. The model can be used only if you cleaned the oultliers and the missing as it should and although I had a great result of prediction, I think we should add confidence interval to the prediction.