Supplementing Markowitz Portfolio Theory with Machine Learning

***DISCLAIMER**: The purpose of this project is not to create a working algorithmic trading system. It is simply an exercise where I showcase my data science and programming knowledge. Invest at your own risk. In no way do I recommend using this project as a basis for investing.*

I. Introduction

The academic literature in finance on portfolio creation primarily bases itself upon Markowitz Portfolio Theory (MPT). Creating a portfolio using MPT involves using predicted returns and the variances/covariances of historical returns to construct a weighted portfolio that maximizes expected return for a given level of risk. The most prominent models for stock price movement relate the stock price movements to the market rather evaluate the underlying business of the stock.

The purpose of the project is to use the fundamental data of the stock's business to predict the returns one year ahead. Instead of using a simple linear regression like the rest of the literature, the project contributes by using machine learning methods to predict future returns.

II. Methodology

i. Data Wrangling and Cleaning

All wrangling and cleaning is explained in the wrangling and cleaning notebook. This section primarily uses pandas, numpy, and quandl libraries.

ii. Exploratory Data Analysis

This notebook looks at all 505 stocks composing the S&P 500 as of 2019-05-01. I explore the returns of all the stocks, seasonal decomposition of the market average, and correlations between stocks. This section primarily uses panda, matplotlib, seaborn, and numpy.

iii. Data Preprocessing

You can find this in the Data Analysis notebook.

- Add annual returns for following year to fundamentals data set
- Create excess returns variable as response variable
- Remove stocks without full dataset
- Split into Train and Test Data
- Create standardized dataset for KNN Algorithm (this is done in the the Model Tuning and Model Testing section)

This section primarily uses pandas and numpy libraries.

iv. Model Tuning

You can find this in the Data Analysis notebook.

v. Optimal Portfolio Construction

You can find this in the Data Analysis notebook.

We evaluate the efficacy of the model by using its predictions to construct an optimal portfolio. The construction of an optimal portfolio involves minimizing risk and maximizing return for that level of risk.
This section uses the numpy library.

vi. Model Evaluation
You can find this in the Data Analysis notebook. I compare the performance of the Random Forest Regressor, XGBoost Regressor, and KNN Regressor. The models are from the xgboost and sklearn libraries.

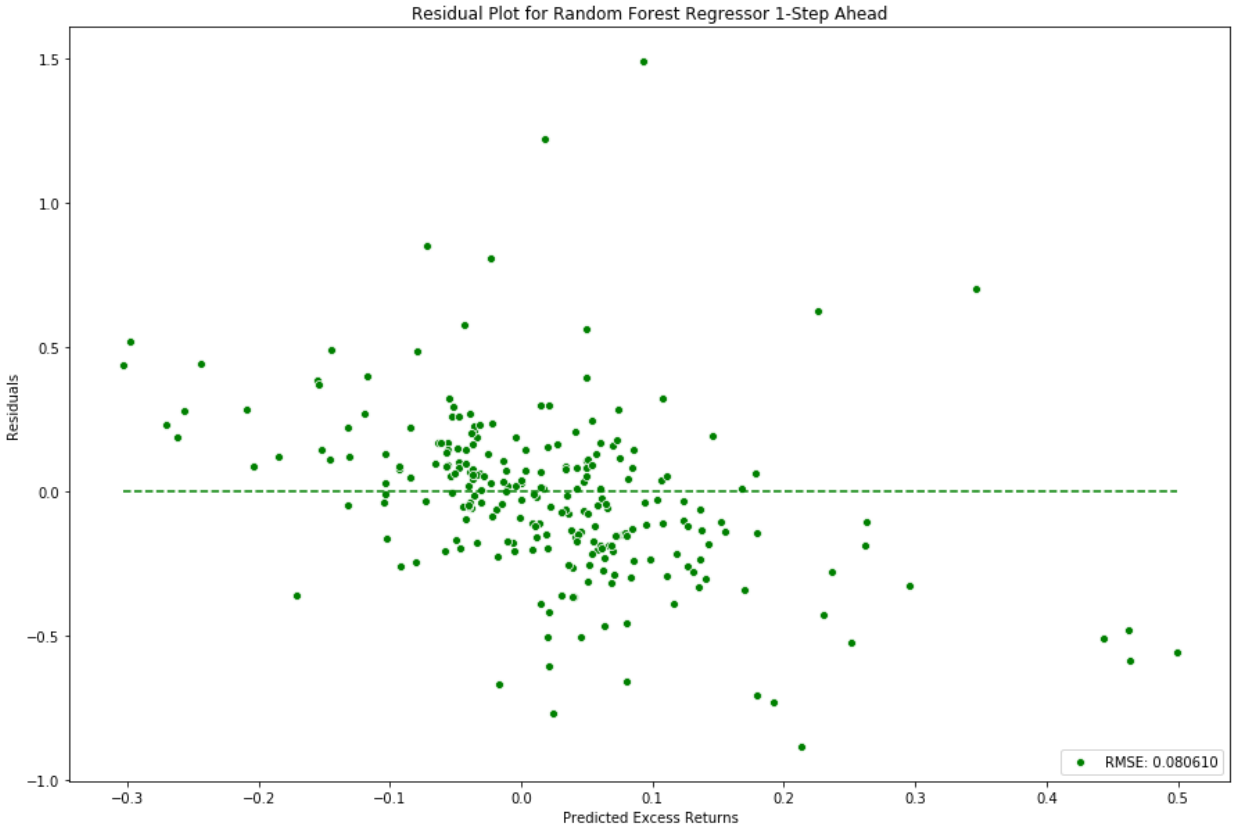
III. Analysis

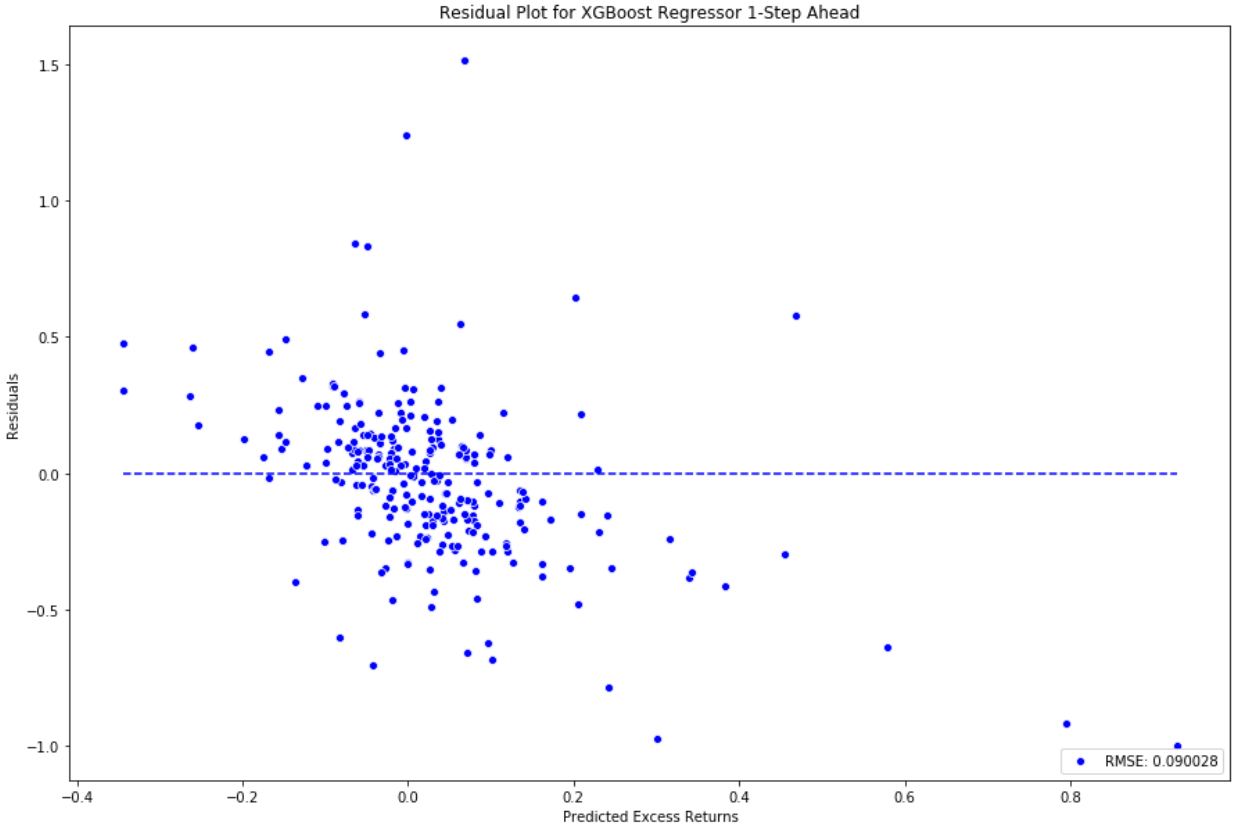
As a metric to compare model performance I use Root Mean Squared Error (RMSE). This metric is the square root of the average squared error of all of the model's predictions. The equation is as follows:

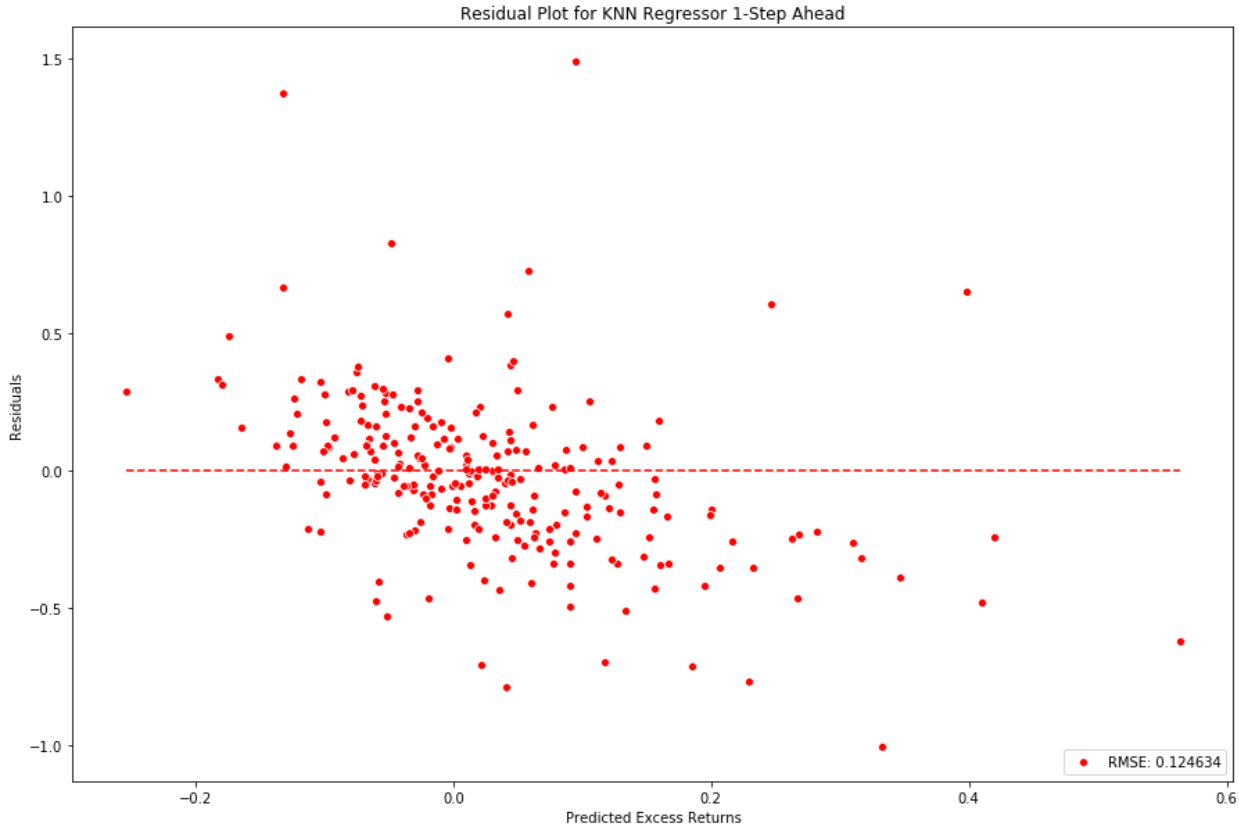
$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\widehat{y_i} - y_i)^2}{n}}$$

Where $\widehat{y_i}$ is the predicted value, y_i is the true value, and n is the total number of predictions.

IV. Graphs and Results







V. Conclusion

After looking at the RMSE of all three models, I conclude the Random Forest Regressor performs best out of all three models in the context of predicting returns to be used in Markowitz Portfolio Optimization.

There are several things I believe could be improved upon in this project. Regarding the data: it would be better to use a dataset that expands past the S&P 500 and includes stocks from other market cap sizes, finding a way to incorporate companies with incomplete stock data would likely result in more accurate predictions, increasing the frequency of the fundamental data from annual to quarterly would also result in more accurate predictions, comparing this model's performance to econometric forecasting models such as the ARIMA. Regarding the actual modelling process: I would have liked to try out some deep learning models, but given the size of the dataset this is unreasonable.