Using Artificial Intelligence to Predict the Return of Mutual Funds

Ethan Chiang

Los Gatos High School

Los Gatos, CA

16 July 2024

Abstract

In creating a profitable Artificial Intelligence for investing, I looked into a US Funds dataset from Yahoo Finance. As I wanted to hone in on one aspect of the investment world, I targeted my research on the mutual funds in the dataset. In experimenting with the various AI learning models such as Random Forest, Linear Regression, Lasso, and Ridge, I refined a Random Forest Model and XGBoost model with the use of hyperparameter tuning to predict the return of mutual funds from 2020 using previous years of data. I created a model that could predict a mutual fund's return of three years with the financial ratios of alpha, Sharpe, Treynor, and other cash flow statistics.

Introduction

With the rising prominence of artificial intelligence and its ability to take over numerous jobs, it is essential to see the possibility that artificial intelligence impacts the economic field. Many investors fall victim to bad financial decisions due to the emotional aspect of investing. Using artificial intelligence, notorious for having minimal emotional capacity, investors can invest their money without falling victim to emotion. If Artificial Intelligence can predict future mutual funds, stocks, and ETFs, even those without previous investment knowledge can make profitable margins in the investment world.

In previous studies, other researchers have created models that have "raised more than $70 million within a few weeks of time" (Science Direct 2). However, because of how new the implications of AI in the financial world have been, "the progress to date has been positive but is in no way a breakthrough" and have not consistently generated significant risk adjusted returns. Because of this, in creating a model to predict various fund returns, I decided to eliminate the present issues in AI models throughout the course of the study.

Dataset

The dataset in this study was used from data on US mutual funds found on Yahoo Finance. The dataset includes a lot of financial metrics and returns about numerous funds beginning from 2000 to 2020.

In my first model, I attempted to predict the fund return for 2020 using the fund return of previous years from 2000 to 2020 as shown in **Table 1.**

Table 1. Financial Metrics used to Predict Fund Return 2020.

Metric	Definition
fund_return_2020	Return from investment in 2020 (%)
fund_return_2019	Return from investment in 2019 (%)
fund_return_2018	Return from investment in 2018 (%)
fund_return_2017	Return from investment in 2017 (%)
fund_return_2016	Return from investment in 2016 (%)
fund_return_2015	Return from investment in 2015 (%)
fund_return_2014	Return from investment in 2014 (%)
fund_return_2013	Return from investment in 2013 (%)
fund_return_2012	Return from investment in 2012 (%)
fund_return_2011	Return from investment in 2011 (%)
fund_return_2010	Return from investment in 2010 (%)
fund_return_2009	Return from investment in 2009 (%)
fund_return_2008	Return from investment in 2008 (%)
fund_return_2007	Return from investment in 2007 (%)
fund_return_2006	Return from investment in 2006 (%)
fund_return_2005	Return from investment in 2005 (%)

fund_return_2004	Return from investment in 2004 (%)
fund_return_2003	Return from investment in 2003 (%)
fund_return_2002	Return from investment in 2002 (%)
fund_return_2001	Return from investment in 2001 (%)
fund_return_2000	Return from investment in 2000 (%)

For the second part of the study, I wanted to predict a fund's return in three years using similar methods utilized in the model used to indicate the fund return in 2020. In addition, I wanted to implement the financial metrics of beta, alpha, treynor, cash flow, etc as shown in **Table 2**.

Table 2. Financial Metrics used to Fund Return 3 years.

Metric	**Definition**
fund_return_3years	Measures the return of a fund over a three-year period
fund_annual_report_net_expense_ratio	Percentage of a fund's assets that are deducted annually to cover operating expenses
fund_price_book_ratio	Compares a fund's market price to its book value
fund_yield	Return on an investment
category_price_book_ratio	Compares the market price to the book value of a group of similar of similar investments
fund_price_cashflow_ratio	Compares a fund's market price to its cash flow per share

category_price_cashflow_ratio	Compares the market price to the cash flow per share of a group of similar investments
fund_price_earning_ratio	Compares a fund's market price to its earnings per share
category_price_earning_ratio	Metric that compares the market price to the earnings per share of a group of similar investments
fund_price_sales_ratio	Metric that compares a fund's market price to its revenue per share
category_price_sales_ratio	Metric that compares the market price to the revenue per share of a group of similar investments
fund_alpha_3years	Fund's excess return relative to its benchmark index over a three-year period
fund_sharpe_3years	Measures the return per unit of total risk (volatility) of the fund over a three-year period
fund_treynor_3years	Measures the return per unit of systematic risk (market risk) of the fund over a five-year period
fund_alpha_5years	Fund's excess return relative to its benchmark index over a five-year period
fund_sharpe_ratio_5years	Measures the return per unit of total risk (volatility) of the fund over a five-year period
fund_treynor_ratio_5years	Measures the return per unit of systematic risk (market risk) of the fund over a five-year period

Exploratory Data Analysis

First I computed the correlations between the various financial metrics to understand the relationships and underline patterns. **Figure 1** illustrates the heat map of the correlations between the numeric features. To accurately explore the correlation between different metrics, I investigated the correlations of the variable fund_return_2020 and fund_return_3years with other metrics in the data set. To report significant correlations, I observed correlations that had a larger than +0.7 or -0.7 correlation with fund_return_2020 as shown in **Table 3** and with fund_return_3years as shown in **Table 4.**

Figure 1. Correlation heat map between numeric features.

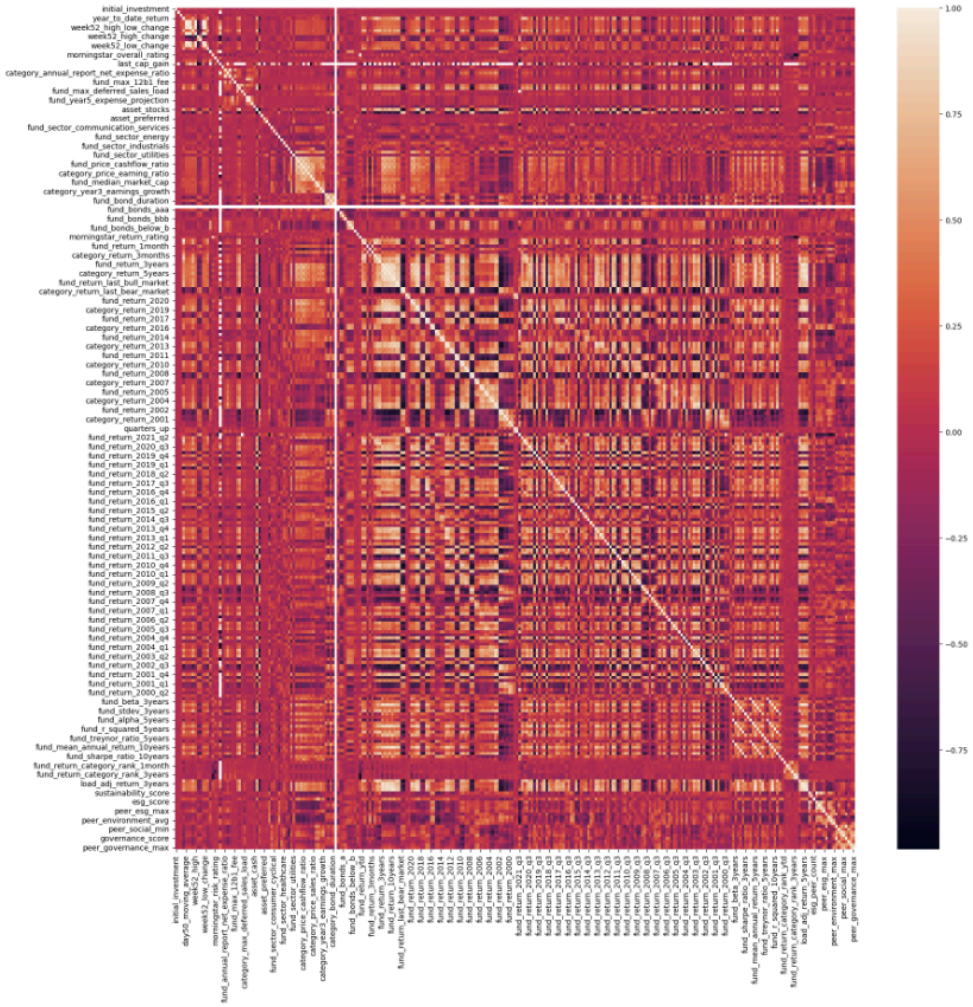


Table 3. Financial Metrics and their Correlations with Fund Return 2020.

Metric	Correlation (w/ fund_return_2020)
fund_return_3years	0.848
fund_return_5years	0.782
category_return_2020	0.794
fund_return_2020_q3	0.797
fund_mean_annual_return_3years	0.806
fund_mean_annual_return_5years	0.765
load_adj_return_3years	0.849
load_adj_return_5years	0.806

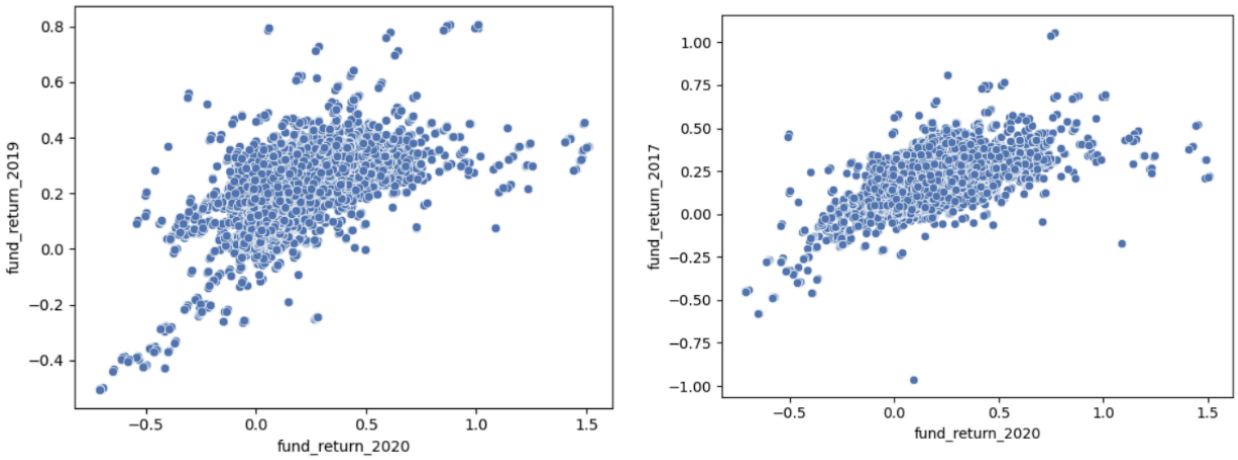
Table 4. Financial Metrics and their Correlations with Fund Return 3 years.

Metric	Correlation (w/ fund_return_3years)
fund_price_book_ratio	0.718
category_return_3years	0.851
fund_return_5years	0.959
category_return_5years	0.821
fund_return_10years	0.890
category_return_10years	0.787
fund_return_2020	0.848
category_return_2020	0.758

fund_return_2019	0.830
category_return_2019	0.774
fund_return_2020_q3	0.785
fund_return_2020_q2	0.745
fund_return_2020_q1	0.753
fund_mean_annual_return_3years	0.961
fund_mean_annual_return_5years	0.901
fund_sharpe_ratio_5years	0.803
fund_mean_annual_return_10years	0.878
load_adj_return_3years	0.985
load_adj_return_5years	0.928
load_adj_return_10years	0.887

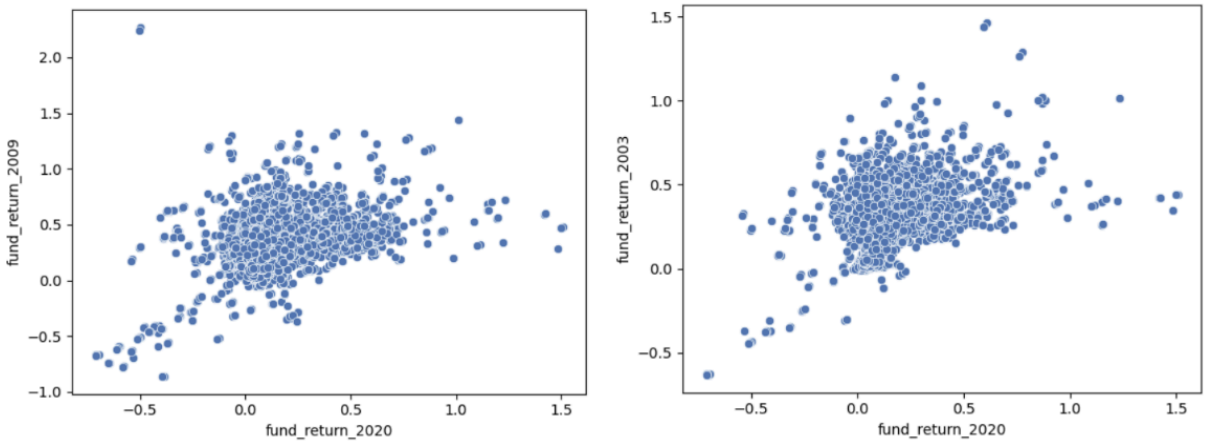
To graphically illustrate the correlations between fund_return_2020 and other metrics used in the study, I created scatterplots to portray the significance of these correlations in **Figure 2** and **Figure 3**.

Figure 2. Metrics with Significant Correlations with Fund Return 2020.



There is a near linear correlation from fund_return_2020 to fund_return_2017 and fund_return_2019.

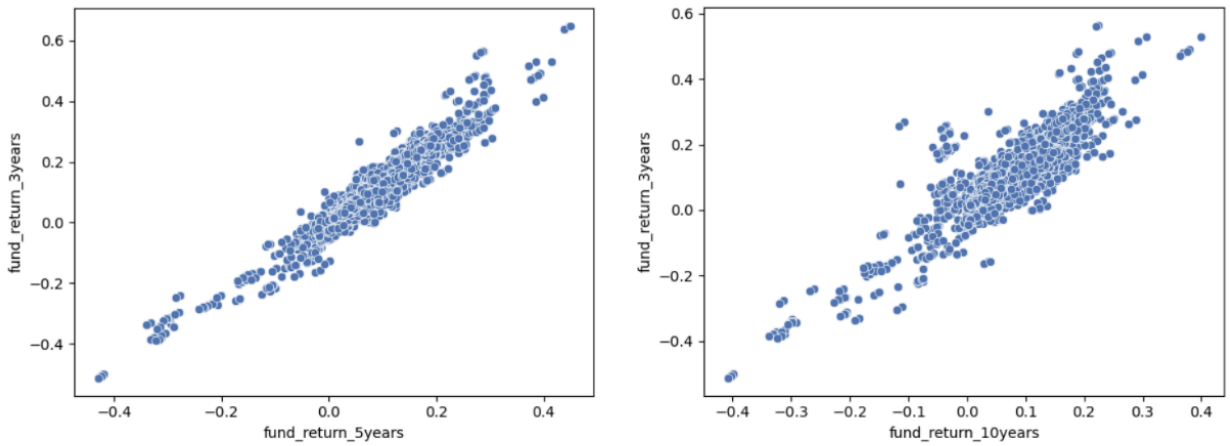
Figure 3. Metrics with Significant Correlations with Fund Return 2020.



There is a near linear correlation from fund_return_2020 to fund_return_2009 and fund_return_2003.

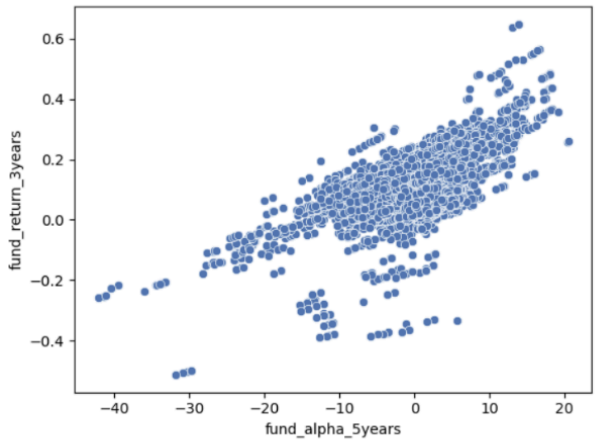
To graphically illustrate the correlations between fund_return_2020 and other metrics used in the study, I created scatterplots to portray the significance of these correlations in **Figure 4** and **Figure 5**.

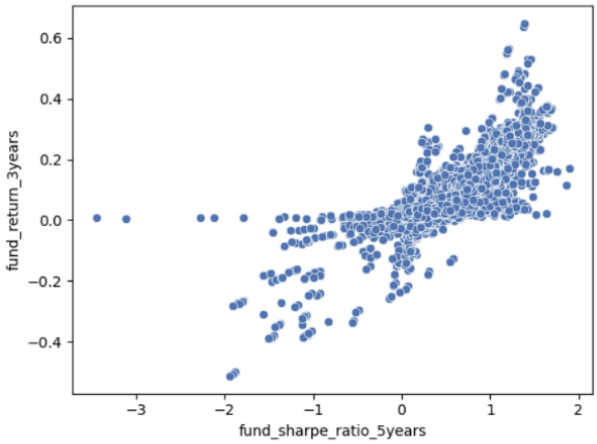
Figure 4. Metrics with Significant Correlations with Fund Return 3 years.



There is a near linear correlation from fund_return_3years to fund_return_5years and fund_return_10years.

Figure 5. Metrics with Significant Correlations with Fund Return 3 years.





There is a near linear correlation from fund_return_3years to fund_alpha_5years and fund_sharpe_ratio_5years.

Methodology and Models

To create the model, I needed to train and test data sets. To do this, I used the variables X_train, X_test, y_train, and y_test to train the model to predict a specific metric and test the accuracy of that prediction. Because I wanted to create multiple models, I decided to test various models in my study. These models would include linear regression, ridge regression, and Lasso regression.

In order to create the Random Forest Model, I had to select the target variable of “fund_return_2020” or “fund_return_3years” depending on the model. Next, I had to split the data into test and training data in order to allow the model to practice and test itself. To begin the Random Forest Model, I imported the “random_forest_model” with a random state of 42 in order to create a model that was easily reproducible. In order for all the data points to contribute equally to the model, I used a StandardScalar, which thus allowed for stable training. Using grid search, I was able to search for the best model and its respective test score, which is ultimately displayed in **Table 5**.

To create the XGBoost model, I once again had to select the target variable of “fund_return_2020” or “fund_return_3years” depending on the model, split the data into train and test, and use a Scalar. Next, I imported the “xg_boost_model” and once again finding its best model and its respective test score as shown in **Table 5**.

To decide on the “best results,” I looked at the MSE (the average squared distance from the values in the dataset and the prediction value in the model) and r^2 (ranging from 0 to 1, which portrays how well a model can predict a value) values for each of the models. In looking for the lowest MSE values and higher r^2 values, I noted that XGBoost and Random Forest Models yielded the best results.

At first, the model ran poorly, as evidenced by the high train and lower test scores. This higher train score could be explained by the fact that the model was overfitting. Because of this, predictions might be inaccurate, and the AI model will perform poorly on new data sets.

To address the overfitting issue and balance the test metrics, I employed grid search and hyperparameter tuning. This involved implementing the estimator, param_grid, cv, n_jobs, and verbose functions. As a result, the test score was improved to match the train score, effectively eliminating overfitting in the original model as shown in **Table 6** and **Table 7**.

Results and Discussion

Using the AI models of Random Forest and XGBoost, I created an algorithm that computed the predictability of predicting specific data points in the data set. The first table displays the Train values for MSE and R^2 and the Test values for predicting “fund_return_2020” without the use of hyperparameter tuning as shown in **Table 5**.

Table 5. Train and the test MSE and R^2 of baseline Random Forest and XGBoost fund return prediction models without hyperparameter tuning.

Model	Train MSE	Test MSE	Train R^2	Test R^2
Random Forest	0.000796	0.00690	0.973	0.710
XGBoost	0.0201	0.00567	0.999	0.761

The Train MSE is calculated by the average squared difference between predicted and actual values on the training set. On the other hand, the Test MSE is the average squared difference between predicted and actual values on the test set. In both cases, a lower number signifies a better performing model. The Train R^2 represents the variance in the dependent variable that is predicted from the independent variables on the training set, while the Test R^2 is for the test set. A value closer to 1 represents a better performing model.

Table 6. Cross-Validation Score and Test Score for Random Forest and XGBoost in Predicting Fund Return 2020 with the use of hyperparameter tuning.

Model	Cross-Validation Score	Test Score
Random Forest (w/ hyperparameter tuning)	0.729	0.712
XGBoost (w/ hyperparameter tuning)	0.756	0.735

The Cross-Validation Score is calculated from different folds from the cross-validation process. Because it has different folds, it averages the performances of the model and ultimately reflects how well the model reacts to untrained data during training. A higher Cross-Validation Score correlates to a stronger model. The Test Score in the table is

calculated by how well the model performs after the use of hyperparameter tuning. A higher Test Score once again correlates to a stronger model.

Table 7. Cross-Validation Score and Test Score for Random Forest and XGBoost in Predicting Fund Return 3 years with the use of hyperparameter tuning.

Model	Cross-Validation Score	Test Score
Random Forest (w/ hyperparameter tuning)	0.943	0.967
XGBoost (w/ hyperparameter tuning)	0.955	0.975

At times, XGBoost and Random Forest Models overfit heavily. As explained previously, I needed to use hyperparameter tuning to reduce the overfitting, which allowed for a stronger model.

Conclusion

I embarked on a journey through AI to attempt to predict various financial data points and remove the emotional aspect of investing. After researching and testing various AI models, I finally decided on an XGBoost and Random Forest Model. To improve my model, I documented and tested different correlations to attempt to predict the financial metrics “fund_return_2020” and “fund_return_3years.” Furthermore, I used hyperparameter tuning to reduce overfitting in my model to close the gap between the train and test scores.

In conclusion, my Random Forest Model predicted “fund_return_2020” with 0.712 accuracy, and my XGBoost model predicted “fund_return_2020” with 0.735 accuracy.

For “fund_return_3years,” my Random Forest Model predicted it with a 0.967 accuracy, and my XGBoost model predicted it with a 0.975 accuracy.

Acknowledgements

I want to thank Abdulla Kerimov for mentoring me and providing insights throughout the creation of my research paper.

References

Penumudy, Tanvi. “Everything You Need to Know about Linear Regression.” *Medium*, Analytics Vidhya, 29 Jan. 2021, medium.com/analytics-vidhya/everything-you-need-to-know-about-linear-regression-750a69a0ea50.

Koehrsen, Will. “Hyperparameter Tuning the Random Forest in Python.” *Medium*, Towards Data Science, 10 Jan. 2018, towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74.

Chauhan, Ankit. “Random Forest Classifier and Its Hyperparameters.” *Medium*, Analytics Vidhya, 8 Mar. 2024, medium.com/analytics-vidhya/random-forest-classifier-and-its-hyperparameters-8467bec755f6.

Leone, Stefano. “US Funds Dataset from Yahoo Finance.” *Kaggle*, 11 Dec. 2021, www.kaggle.com/datasets/stefanoleone992/mutual-funds-and-etfs.