Dynamic Asset Allocation

Mário Albuquerque, Pierre Monroy, James Wang

October 7, 2017

1 Introduction

The past decade has seen a rise in popularity of passive investing among portfolio managers. According to the Financial Times, passively managed funds have grown four times faster since 2007 than actively managed funds. This trend is mainly driven by the attraction of ETFs for investors. Indeed individuals are looking for a diversified allocation of assets that allow them to maximize their returns while minimizing their exposure to risk for the lowest cost possible. This is precisely what indexing provide and the reason why it has benefited of a great flow of capital over the recent period.

However, introduction of new technologies could allow to improve asset allocation and offer investor a way to use information-based edge in order to invest more actively. More precisely, the use of machine learning techniques can allow to determine with a certain level of confidence the direction of future returns and improve upon common baseline approaches.

We are interested in comparing a data-driven dynamic asset allocation to simple strategies. Our starting point has been to consider that an asset derives its value from the present value of its future cash flow. These are largely influenced by macroeconomic factors such as GDP growth and inflation. Therefore our approach consists in constructing a dynamic strategy relying on a machine learning model trained on macroeconomic variables. We then compare the risk-adjusted returns of this strategy to two baseline strategies.

2 Data Collection, Cleaning and Preparation

2.1 Data Collection

In order to be able to allocate between a wide range of asset classes, we have decided to collect data for most of the investment classes available. What is more, in order to have enough data available we have chosen to gather data from the indexes instead of their respective trackers in order to have access to more historical data.

The different time series in our dataset have been downloaded from Bloomberg and are presented in Figure 1. Our data cover domestic and foreign equities as well as advanced and emerging economies. It comprise also a broad range of bonds categories, commodities such as precious metal and oil and finally real estate and volatility. These series have different starting date with one starting as early as 1990 and more recent one starting in 2006.

Name	Asset Class	Segment	Start	End	Source
MSCI World Net Total Return USD Index	Equities	World	12/31/1998	9/8/2017	Bloomberg
S&P 500 Total Return Index	Equities	US	1/2/1990	9/8/2017	Bloomberg
MSCI Emerging Net Total Return USD Index	Equities	EM	12/31/1998	9/8/2017	Bloomberg
Bloomberg Barclays US Agg Total Return Value Unhedged USD	Bonds	World	1/2/1990	9/8/2017	Bloomberg
Merrill Lynch 10-year U.S. Treasury Futures Total Return	Bonds	US Government	1/2/1990	9/7/2017	Bloomberg
Bloomberg Barclays US Treasury Inflation Notes TR Index Value Unhedged USD	Bonds	Inflation-Protected	8/12/1998	9/12/2017	Bloomberg
On the Run CDX High Yield Total return Index Unhedged	Bonds	US Corporate HY	9/27/2005	9/8/2017	Bloomberg
On the Run CDX Investment Grade Total return Index Unhedged	Bonds	US Corporate IG	3/22/2004	9/8/2017	Bloomberg
On the Run Itraxx Investment Grade Total return Index Hedged to USD	Bonds	European Corporate IG	10/1/2004	9/8/2017	Bloomberg
On the Run CDX Cross Over Total return Index Hedged to USD	Bonds	European Corporate HY	10/3/2005	9/8/2017	Bloomberg
Credit Suisse Multi-Asset Futures - Crude Oil Total Return Index	Commodities	Oil	1/15/1998	9/8/2017	Bloomberg
Credit Suisse Multi-Asset Futures - Gold Total Return Index	Commodities	Gold	1/15/1998	9/8/2017	Bloomberg
S&P Global Property USD Total Return Index	Real Estate	US	8/12/1998	9/12/2017	Bloomberg
Credit Suisse Short VIX 1% USD Total Return Index	Volatility	Short Volatility	8/21/2006	9/8/2017	Bloomberg

Figure 1: Data Table

Regarding the features, as stated previously, we choose to use macroeconomic factors as our key information in order to predict the direction of future returns of our assets. In order to have indicators on both growth and inflation, we decided to use other variables in addition to the change in GDP and CPI. The Citi Economic Surprise Index and Goldman Sachs Current Activity Indicator as well the 10Y Treasuries Real Rate allowed us to improve our information on the economic growth. Furthermore, we used the 10Y Treasuries Breakeven Rate to improve our information on the inflation level. The different features used are presented in Figure 2.

Name	Type	Start	End	Source
Goldman Sachs Current Activity Indicator US	Growth	1/2/1990	8/1/2017	Bloomberg
Citi Economic Surprise Index	Growth	1/1/2003	9/13/2017	Bloomberg
Daily 10Y US Treasury Breakeven Rate	Inflation	1/30/1997	9/13/2017	Bloomberg
Daily 10Y US Treasury Real Rate	Real Growth	8/3/1998	9/13/2017	Bloomberg
Daily ACM 10Y Term Premium (Published by NY Fed)	Growth+Inflation	1/2/1990	9/12/2017	Bloomberg
Consumer Price Index for All Urban Consumer: All Items	Inflation	1/1/1947	8/1/2017	FRED
Gross Domestic Product	Growth	1/1/1947	4/1/2017	FRED

Figure 2: Features Table

2.2 Cleaning

The dataset itself was challenging to manipulate since it consists both of different types (level, percentage points, yields) and different frequencies as shown in Figure 3. One of our first task was thus to match date indices with a date index corresponding to the US stock market. We then resampled the data to a monthly frequency by taking the end-of-month value. Finally since some of the data comprised NaN, we had to forward fill the data in order to respect chronology.

	ACMTP10	CDX_HY	CDX_IG	CESIUSD	CPIAUCSL
count	2770	2770	2770	2770	2770
mean	0.92	160.40	124.11	-2.57	225.91
std	0.91	40.75	7.73	42.10	12.07
min	-0.75	88.67	109.03	-140.60	202.00
25%	0.17	119.88	117.02	-29.88	216.51
50%	0.70	152.51	123.20	-3.40	228.52
75%	1.68	198.48	131.85	30.08	236.89
max	3.44	237.40	137.97	97.50	245.03

Figure 3: Data Table

3 Features Engineering

Despite the fact that we chose to use macroeconomic indicators as our main source of information for our dynamic asset allocation model, we had also to consider the fact that prices are often autocorrelated and therefore to use past returns as potential features. Our first decision was to transform our series from indexes level to returns. We then computed the autocorrelation function of each asset class returns in order to determine which level of lag in the returns could be used as a potential feature. Figure 4 displays as an example the autocorrelation function of the credit Suisse short VIX Index returns. It appears clearly in the plot that some of the lagged returns are significantly correlated to the present value of returns. Unsurprisingly, the first lag is correlated to the current return. We repeated this analysis for each asset class in order to determine the best lag to use as a feature.

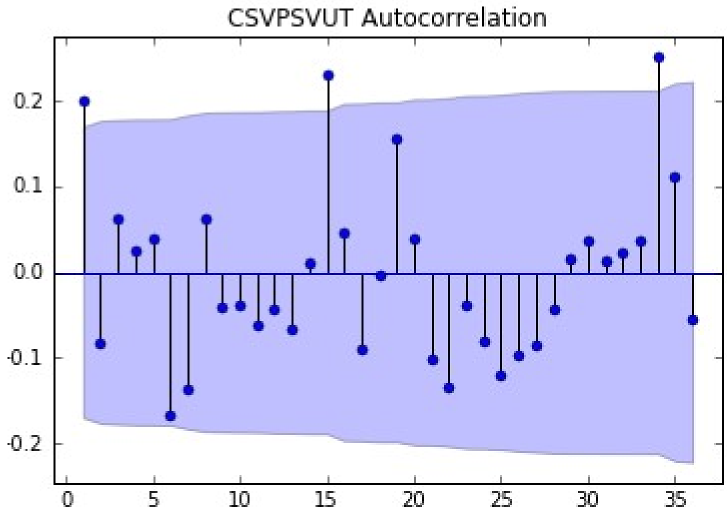
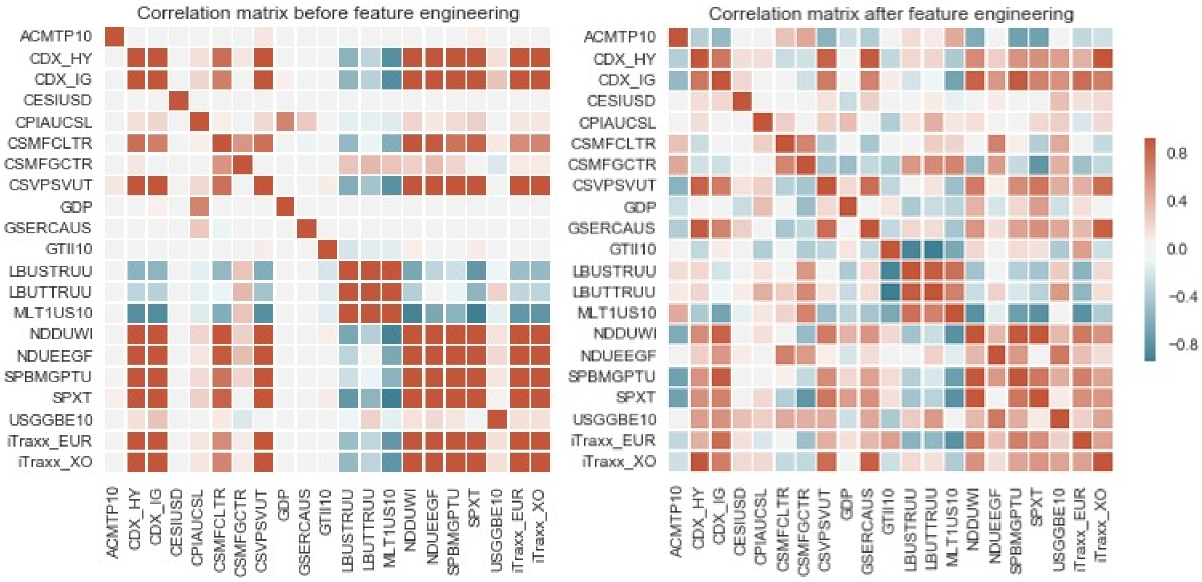


Figure 4: Autocorrelation function of VIX returns

We then proceeded to compute z-scores for each of the features with an expanding window and a minimum of 36 past periods ensuring enough data points to get conclusive results. Depending on the z-score of each feature, a label is assigned to a feature at time t and indicates the direction of the returns of an asset at t+1. Therefore the labels are either +1 or -1. The feature engineering we performed allows us to extract more information from the data. This is particularly noticeable in the Figure 5 that shows the correlation matrices before and after we performed our feature engineering.



(a) (b)

Figure 5: Correlations level before and after feature engineering

4 Baseline Strategies Development and Model Construction

4.1 Baseline Strategy

We started our project by designing baseline strategies to which we would then compare our own model. As our first insight was to use macroeconomic variables, we decided to use Ray Dalio's quadrant strategy as a baseline strategy. This approach draws on the idea that depending on the increase or decrease in GDP and inflation, an investor should allocate its capital in different asset classes. Figure 6 shows this quadrant. As we can see, during a recession, one might benefit from investing in gold whereas during an expansion it could make returns by investing in emerging equities.

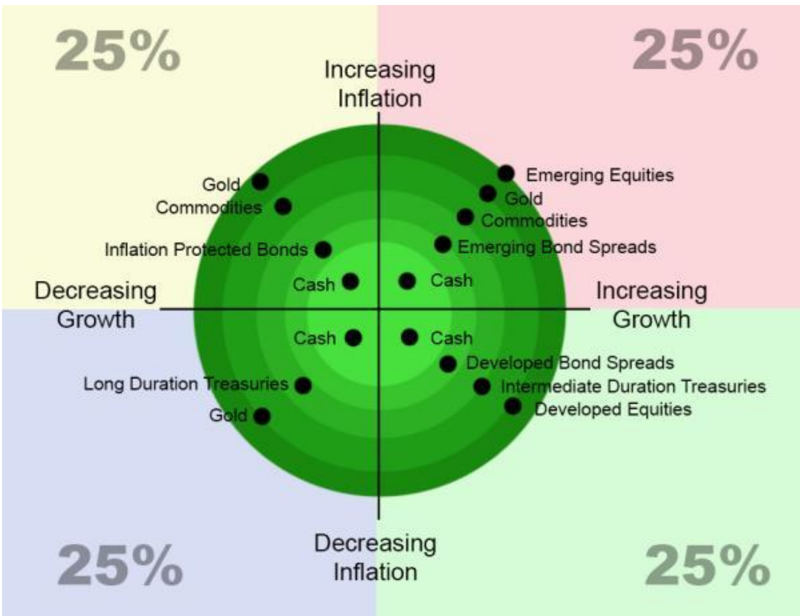


Figure 6: Quadrant Strategy (source : Lessons from A Trading Great)

The other baseline strategy we decided to construct further in the project was a naive strategy. This approach is a simple biased coin flip that takes into account the historical repartition of up and down moves in our returns. We use this to generate labels of prediction of the direction of future returns.

We then run a simple asset allocation on the data for both these strategies in order to estimate their performance. Figure 7 displays the results of theses approaches. Between 2006 and 2017, the total return for the naive strategy was 14.5% and 47.85% for the Quadrant approach. The maximum drawdown reaches -19.2% and -38.57% respectively for the Naive and the Quadrant strategies. Our goal was then to come up with a model that would perform better than both these strategies.

	Naive	Growth/Inflation Quadrant
Period Start	11/29/2006	11/29/2006
Period End	8/30/2017	8/30/2017
Total Return	14.50%	47.85%
CAGR	1.78%	3.70%
Daily Vol Annualized	6.16%	10.99%
Daily Sharpe	0.32	0.39
Max Drawdown	-19.20%	-38.57%

Figure 7: Quadrant Strategy (source : Lessons from A Trading Great)

4.2 Model Construction

The problematic we faced was to classify the direction of future returns of our asset classes. Thus this consists in a supervised learning problem for which a large panel of models is available. We decided to use a range of models in order to determine which one would produce the best results. The models tested were :

- Gaussian Naive Bayes;
- Logistic Regression;
- Random Forest;
- Linear SVM;
- Simple model ensemble.

In order to produce robust results, we used a cross-validation method specific to time series. This technique implies to train the model first on a given sample and test it on the next period. Then including the former testing period in our training set, we test on the new next period. This procedure is repeated until the whole sample has been covered. This allows to avoid training on future data and testing on past data which would not make sense. The input used for this steps were the 7 normalized macro feature and 14 lagged normalized asset features. We can then analyze the output to determine which model produce the best results.

As we stated, the performance metric chosen for this project was the Sharpe Ratio. We proceeded to compute this metric over the whole sample for each model outlined above. This computation is shown in Figure 11 in the annex with a distinction between the results obtained with all the features and the results obtained only with 7 features. We find that using only 7 features produces the best results. The Logistic Regression and the Linear SVM lead to the higher Sharpe Ratio over the whole sample. We used the F1 score as a balanced way to tune our models. We did not look at Sharpe Ratio on the model training/testing stage as a basis to chose our models. We only checked it at the final model comparison stage in order to check results for the portfolio. This approach is validated by the fact that the linear correlation between F1 score on the testing stage and the Sharpe ratio of constructing a portfolio within a given Machine Learning model, is 0.8 (R2 = 0.63). These F1 score computation are shown in Figure 12 in the annex.

Rather than choosing between the two most performing model we have identified over the previous step of the project, we have decided to opt for a voting procedure. This consist in training and testing both models and use them to predict the direction of future returns. If both model agree on the prediction then it is kept for the asset allocation. Otherwise, the asset class on which the models disagree is removed until the predictions are consistent. The results in Figure 8 make it clear that the Ensemble model improves a lot on the baseline strategies. The Sharpe Ratio over the whole period of trading has increased of about 187%.

Model	Nr Features	F1 Score (Test data)	Sharpe (All-time)	Improvement over best Baseline
Baseline "Naive"	na	68.25	0.28	
Baseline "Quadrant"	na	58.14	0.50	
Gaussian Naive Bayes	21	65.45	0.51	2.00%
Logistic Regression	21	71.35	0.64	28.00%
Linear SVM	21	63.7	0.61	22.00%
Random Forest	21	63.04	0.56	12.00%
Gaussian Naive Bayes	7	72.03	0.83	66.00%
Logistic Regression	7	76.25	1.11	122.00%
Linear SVM	7	75.81	1.11	122.00%
Random Forest	7	63.73	0.61	22.00%
Ensemble*	7	76.22	1.12	124.00%
* Average of Logistic Regression and Linear SVM position matrices.				

Figure 8: Results of each model tested

5 Model Deployment

The last step of this project was to unify all the development stages described previously. In order to construct a pipeline that would allow us to clean and prepare the data, to perform a cross-validation on the models chosen and then to produce predictions on which a strategy would be backtested.

We used three different Python file to do that. The first one is our data module that is inputed raw CSV files of level for each asset class and outputs a Python Dataframe that we will use throughout the remaining steps. The main file is Python script that uses the data module and performs the cross validation training and testing phase of the project. Once the model are trained and tested, they are used to output predictions on the whole sample. Finally the backtesting module that uses Python package *bt* takes the predictions DataFrame and the prices over the period as an input. It runs a simple strategy that goes long or short and rebalances each period depending on the prediction. This outputs all the performance metrics required to evaluate how good the strategy is. The whole pipeline blueprint is shown in Figure 9.

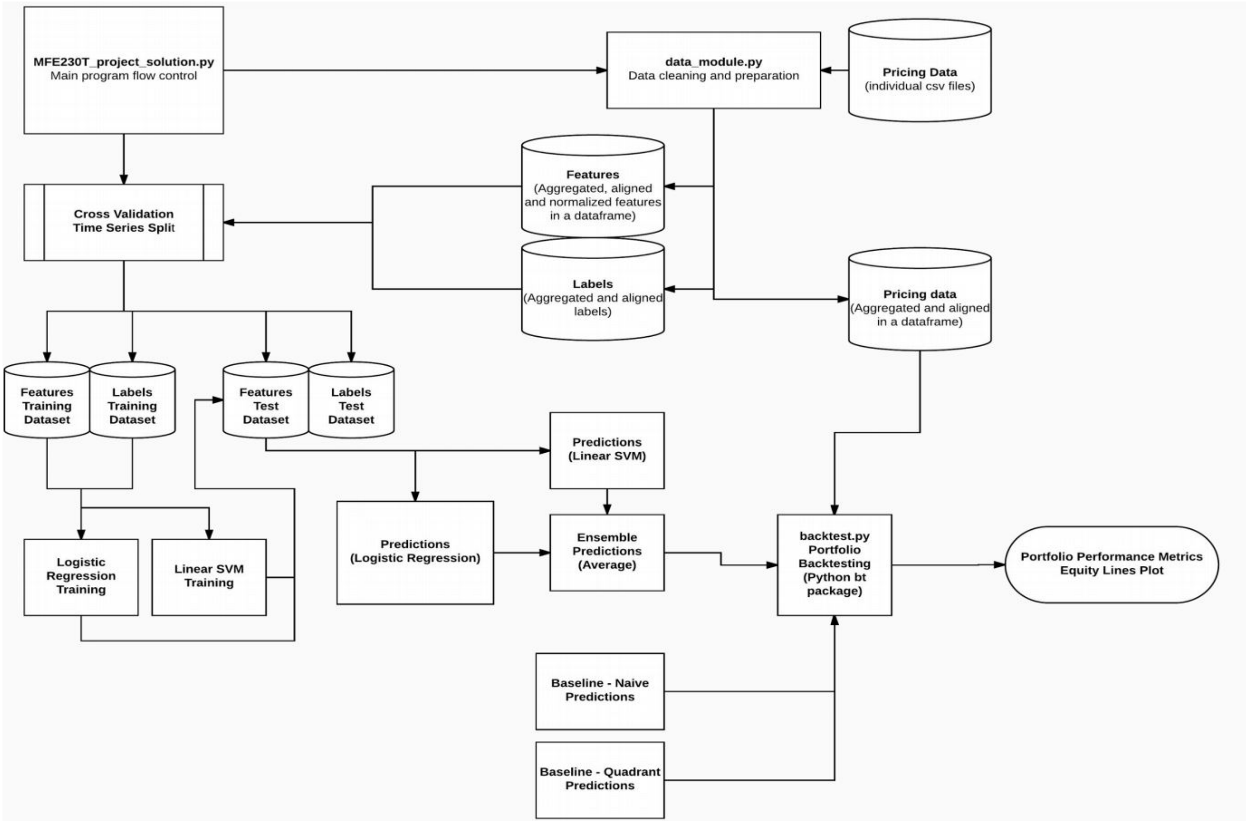


Figure 9: Pipeline Blueprint

The results of our final iteration of the project is shown in Figure 10. From 2009 to 2017, the solution we have developed ends up with a total return of 71.97% and a daily Sharpe Ratio of 1.12. The Naive and Quadrant baselines produce respectively total return of 12.76% and 37.35% and a Sharpe Ratio of 0.28 and 0.5. Another good indicator of the performance of our model is the maximum drawdown compared to the baselines. With only -12.67%, the model is less exposed to losses than the Quadrant baseline that has -22.69%. The Naive approach is very close with -13.18%.

Stat	ML Solution	Baseline - Naive	Baseline - Quadrant
Start	2009-11-30	2009-11-30	2009-11-30
End	2017-08-30	2017-08-30	2017-08-30
Risk-free rate	0.00%	0.00%	0.00%
Total Return	71.97%	12.76%	37.35%
Daily Sharpe	1.12	0.28	0.50
Daily Sortino	1.43	0.36	0.63
CAGR	7.25%	1.56%	4.18%
Max Drawdown	-12.67%	-13.18%	-22.69%
Calmar Ratio	0.57	0.12	0.18

Figure 10: Equity Progression and Statistics of the Final Strategy

6 Conclusion

Throughout this project, we have been able to successfully clean and prepare data, perform a feature engineering in order to extract information and finally to build a model producing predictions on the direction of future returns. One of the first challenges was to prepare the data in order to overcome the fact that the series available had different orders of magnitude, different frequencies and periods of availability.

One of our main takeaway is that feature engineering matters a lot. As we have shown it, operating this step has allowed to extract more information from the data. The correlations matrices are a very good indicator of this. We have also found out that using a few number of features can be better than a large one. Indeed, the models selected produce a better Sharpe Ratio if used only on the macroeconomic variables selected at the beginning of the project.

Finally, we observe that adding model complexity is not beneficial past some point. Across the range of models tested, some as the Random Forest add complexity where it is not necessary. In the end, the two models that performed the best were the Logistic Regression and a Linear SVM which are relatively simple models. Using a combination of produce a satisfying result with respect to the performance metric chosen.

7 Annex

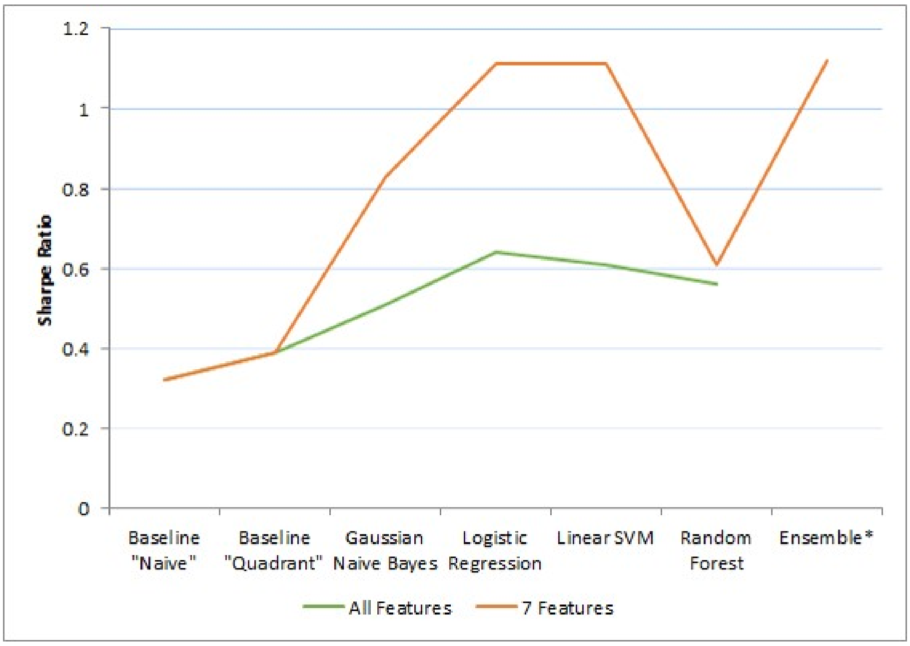


Figure 11: Sharpe Ratio over the whole sample for each model

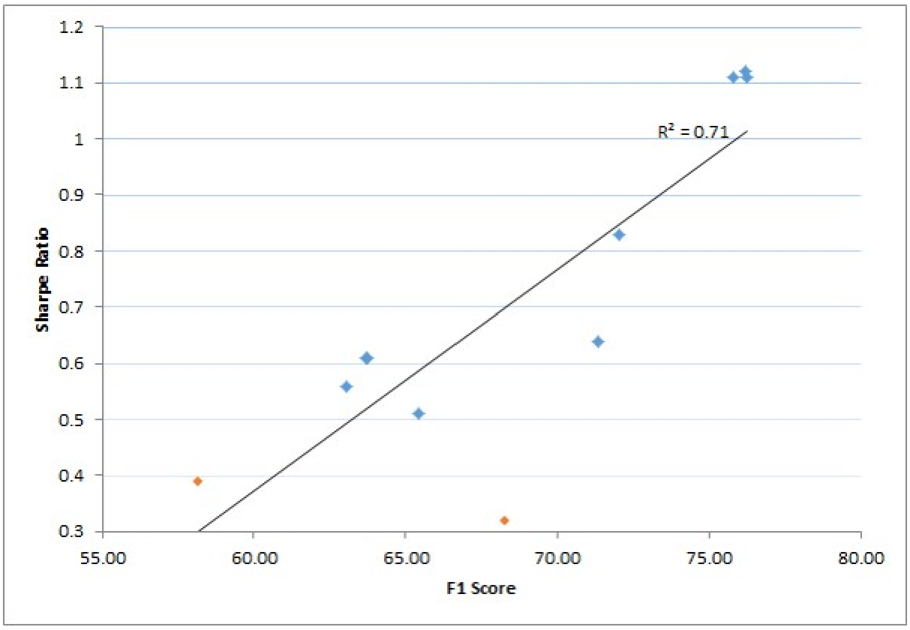


Figure 12: Sharpe Ratio to F1 Score