A Machine Learning Approach to Modeling Exchange Rates

Master of Quantitative Economics

University of California, Los Angeles

Andrew Christopher Salguero

Aaron Tornell

June 4, 2021

Abstract

This paper is to generate accurate and consistent exchange rate forecasts with a machine learning model known as an Elastic Net. Macroeconomic data such as interest rate differentials and money supply are fed into our model via a rolling window with the expectation to generate the probability of an appreciation of the U.S. dollar over a one-month horizon. The tangent portfolio between the two strategies has an expected annual return of 7%, Sharpe ratio of about 0.79, an annual 5% VaR of -3.8%, and an annual Beta of -0.45. Benchmarks such as the S&P 500 Index and the HFRI Macro: Currency Index will be used as reference to evaluate our overall performance, while achieving an annual -0.75 correlation with the S&P 500 Index. All plots provided were generated using Python in conjunction with the data cited.

Table of Contents

Introduction

Using machine learning applications can be a powerful tool when building time series models. However, many make the fatal error in only utilizing an autoregressive feature on the variable they are attempting to forecast. With that in mind, I set out to attempt to use an abundance of macroeconomic data to make monthly directional forecasts for two given exchange rates. The first is what I will continuously refer to as the USD/KRW. This is the exchange rate between the U.S. dollar and the South Korean won. The second being the USD/MXN, the exchange rate between the U.S. dollar and the Mexican peso. Both exchange rates provided us with the necessary data and provided us a unique opportunity to model exchange rates that are often left neglected compared to modeled exchange rates such as EUR/USD and USD/JPY.

Before even attempting to model an exchange rate it is important that we define what success is, from both a modeling and financial perspective. Of course, it is in our best interest to achieve high levels of forecast accuracy and large financial returns, however, it is important that we achieve such goals consistently. This is to accommodate two potential issues when deploying a strategy such as this. For starters, past returns are not indicative of future returns. Secondly, although all forecasts are out of sample tests, we are trying to find a model that works and sometimes we run the risk of modeling our training data to accurately forecasts our known testing data. Which can create an issue of overfitting our data and making poot forecasts in the future.

Therefore, it is essential that we devise a model and strategy that returns consistency rather than finding the model/strategy with the highest accuracies and financial returns.

Data Description

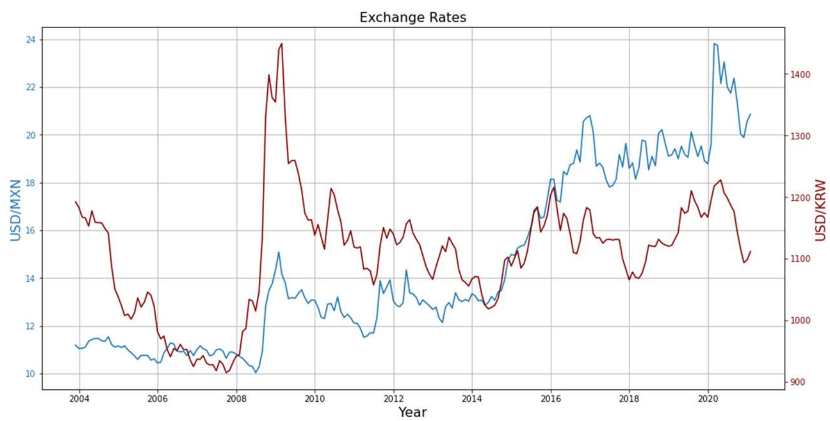
While using a machine learning approach I wanted to build a model that mostly used economic variables that in theory, explained the movements in exchange rates. My initial hypothesis was that changes in these economic variables can be modeled to generate accurate forecasts for changes in the exchange rate. Early builds of the model suggested that classifying appreciations or depreciations were far more accurate than using point forecasts.

For both of my classification problems I opted in using similar datasets with one minor difference. For each exchange rate that I modeled I only used its own autoregressive exchange rate. This is an important detail to note because finding the optimal model suggested using macro data from both countries and using these variables in both models.

Exchange Rates

Although I wanted to build a model around using macroeconomic variables, I found it would be beneficial to include autoregressive data into my model. I chose to only use the USD/MXN and USD/KRW exchange rates. Initially I neglected using autoregressive data because I feared my model would not be consistent and could potentially “run off a cliff” where it made consecutive false classifications and result in a sharp decline in our profits.

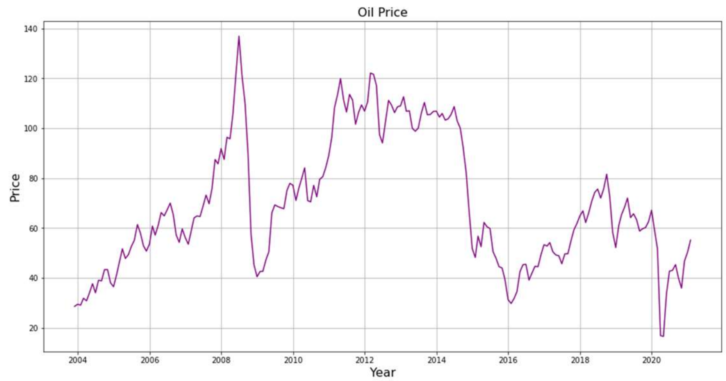
However, after utilizing the data in addition with the macroeconomic variables, I found improvements in overall model accuracy and expected returns.



Generally, the exchange rates move in similar patterns. During shocks like the 2008 financial crisis, we can see that the dollar will appreciate due to institutions seeing the US dollar as a safe place to store value during times of uncertainty.

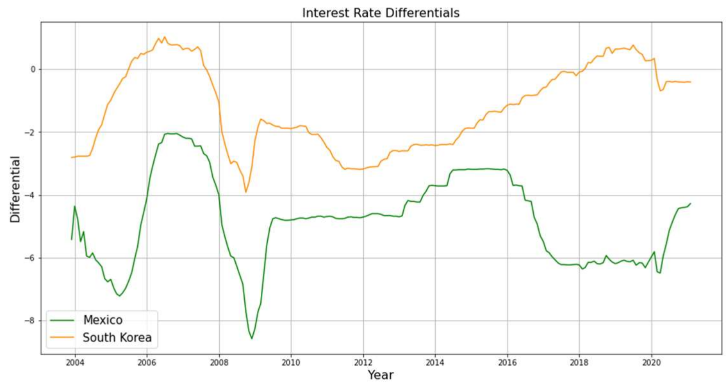
Oil

The changing price of oil can potentially have a huge impact on an exchange. Especially for economies that depend on massive amounts of exporting/importing of the commodity. With Mexico being one of the largest producers and South Korea being a large consumer of oil which they almost entirely rely on imports to supply their consumption needs.



Interest Rate Differentials

The interest rate differential between two countries is one of the key macroeconomic variables that explain movements in each exchange rate. Since my models are to make directional forecasts, I found it would be beneficial to calculate the interest rate differential between the United States Federal Funds Rate and the given foreign money market rate.



In theory, countries with the higher interest rate paid on deposits should expect to see their domestic currency appreciate. Whereas countries with lower interest rate paid on deposits should expect to see their domestic currency depreciate.

Although there is a sound argument in support of this theory, we cannot expect countries with higher interest rates to outperform the U.S. Dollar in the long run. Risk-averse investors would most likely store their value in the U.S. Dollar due to the dollar being one of the safest currencies to store value. It is important to iterate the argument of risk aversion because our interest rate differentials are almost always negative on our time series.

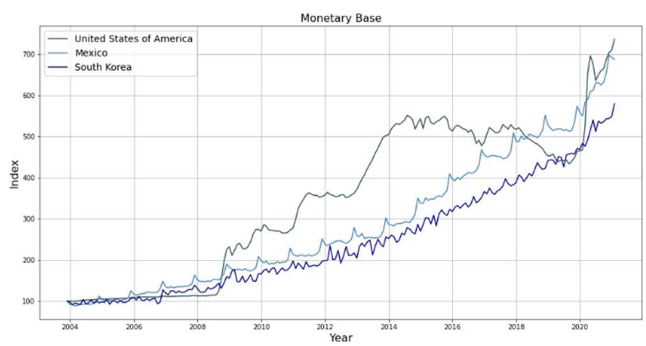
This is to say that the money market rates in Mexico and South Korea are generally higher than in the United States. In time periods where the interest rate differential is increasing it because the U.S. federal funds rate is increasing and/or the money market rate of the foreign country is decreasing. The inverse holds true as well, when the differential is decreasing the federal funds rate is decreasing and/or the money market rate of the foreign country is increasing.

$$\textit{Interest Rate Differential}_t = \textit{Federal Funds Rate}_t - \textit{Foreign Money Market Rate}_t$$

Monetary Base

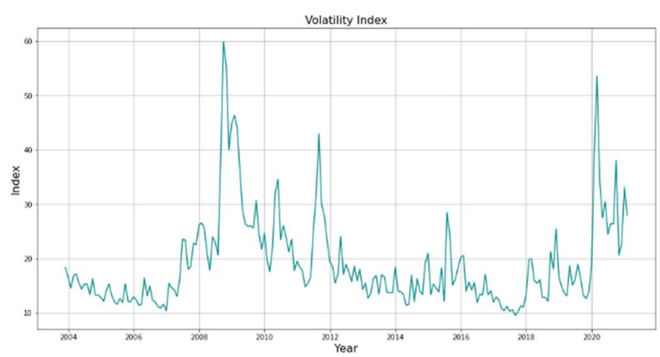
In theory, increases in a monetary base would result in the domestic currency to depreciate; assuming all other variables remain the same. For the data gathered, it was collected using the IMF's International Financial Statistics database to aggregate the data needed.

For visualization purposes, each country's M0 supply was indexed so visualizing the growth is possible. This is because the IMF sets the units equal to the domestic currency of a given country. During preprocessing only, the raw data was processed and used in the modeling of the data and not the indexed values.



Volatility Index

To gather data based on volatility I had to rely on using the CBOE Volatility Index. This is because I was able to gather data easily and reliably on the index using the unofficial Yahoo Finance API in Python. Although the index is based on the options market for U.S. equities, it still gives a robust gauge for volatility.

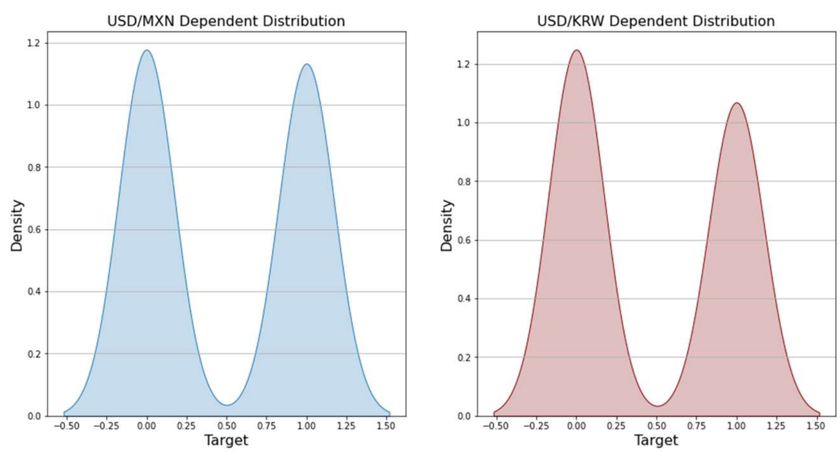


Dependent Features

For both of my datasets I settled on using binary classification to determine if the U.S. dollar appreciated or depreciated from one month to the next relative to the given exchange rate. For example, if the U.S. dollar depreciated against the Mexican Peso from January 2021 to February 2021, then we would set out classification for the month of January 2021 to equal 0.

When generating these classifications, we can see that there is a slight skewness in the data. For USD/MXN exchange rate, 48.79% of the classifications were 1 while only 45.89% of classifications were one for the USD/KRW exchange rate. Although not evenly split, this gives us a fair distribution of data to work with because it will be less likely for bias to occur and only make one type of forecast.

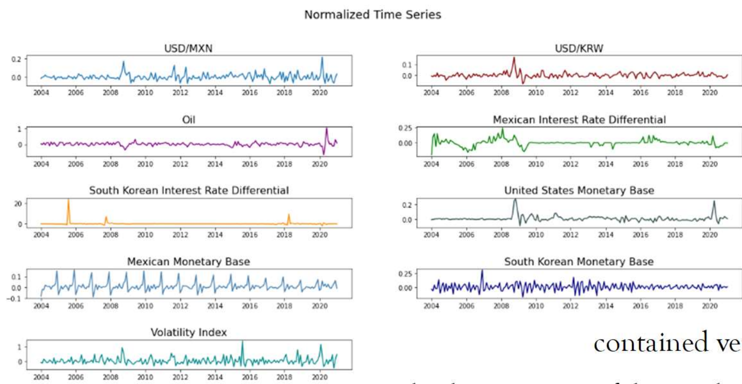
Lastly, our dataset contains discrete values, however, we made far better forecasts when we estimated probabilities of a directional forecast occurring.



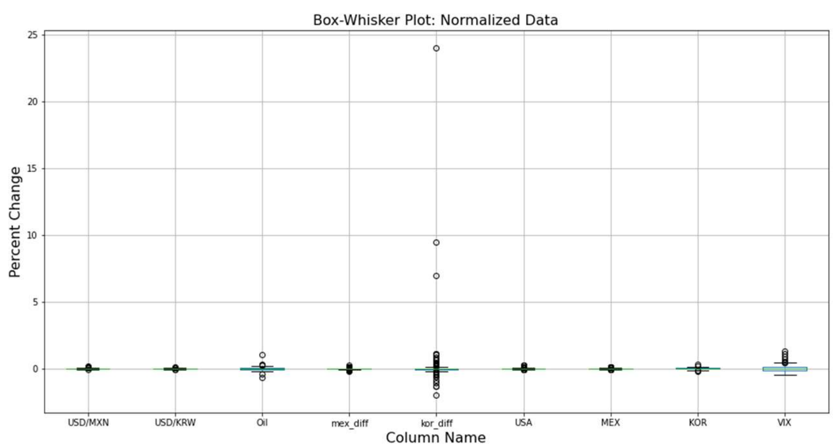
Data Preprocessing

Because our raw dataset varies in magnitude it is import that we scale our data to develop more robust models for in and out of sample datasets. Our first step is to normalize our entire dataset. To achieve normalization, I found it to be suitable to calculate the percent change from one period to the next. Doing so meant that much of our data will be a value between -1 and 1.

It should be noted that the Mexican monetary base contains seasonal patterns in its time series. However due to multiple iterations of model performances and return statistics, it was the optimal choice to leave the seasonality component in the time series.



Examining the plot gives us a general idea that our normalized data set is stationary around zero. Analyzing the box plot we can see that for most variables the values were centered at zero and contained very few outliers in the data. After multiple iterations of data scaling, the models performed best with the normalized data presented above. Therefore, there was no need for any additional data transformations after normalizing our data.



Model Selection

After using several machine learning models, I found that using an Elastic Net model yielded the best results for both prediction accuracy and financial returns. An Elastic Net is a simple model that combines the penalty elements of a Ridge and Lasso regression.

To simply put it, a Ridge regression's penalty term tries to account for model overfitting. This is done by introducing a small amount of bias in the training set of the data to generate a new best fit line. As a result, training accuracy may decrease but could potentially lead to an increase in out of sample accuracy. As the Ridge regression aims to account for overfitting, the Lasso regression aims to account for multicollinearity in our data features. This is accomplished by reducing a features coefficient to zero, therefore removing the feature from the model.

Other than model performance there are several reasons why I decided using an Elastic Net over other machine learning techniques. Although I had confidence that my data selection was relevant and robust for modeling, there may be times where there is multicollinearity and redundant variables in my dataset. I also had to account for the limitation of data available. Using monthly data from 2004 to February of 2021 would only return about 200 rows of data. Meaning if we expanded our dataset with more variables, we could run the risk of a poor model.

When approaching the USD/MXN model I found it exceedingly difficult to make consistent forecasts. I also noticed it was easier to make inaccurate forecasts than it was to make accurate ones. After some tests, I opted in building a model that was consistently wrong. If we were confident our model was expected to be wrong in its forecast, then we can be confident in doing the exact opposite than what the model forecasted. For example, each time the model forecasted an appreciation of the dollar, we would short the dollar and if it forecasted a depreciation, we would long the dollar. Again, this is only a unique trait found in the USD/MXN model; our USD/KRW model was able to make consistent accurate forecasts.

Rolling Window & Coefficient Estimation

When modeling the exchange rates, I wanted to create a model that was constantly being estimated based on a rolling window of training data. This is because I did not want to train a model on past trends in the data that do not hold today. Within Python I managed to create a program that iterated through different window sizes and found the hyperparameter that yielded the best testing accuracy and greatest financial returns. A window size of 15 time periods was the optimal size for modeling the USD/KRW exchange rate while the optimal window size for the USD/MXN exchange rate was a window of 9 time periods.

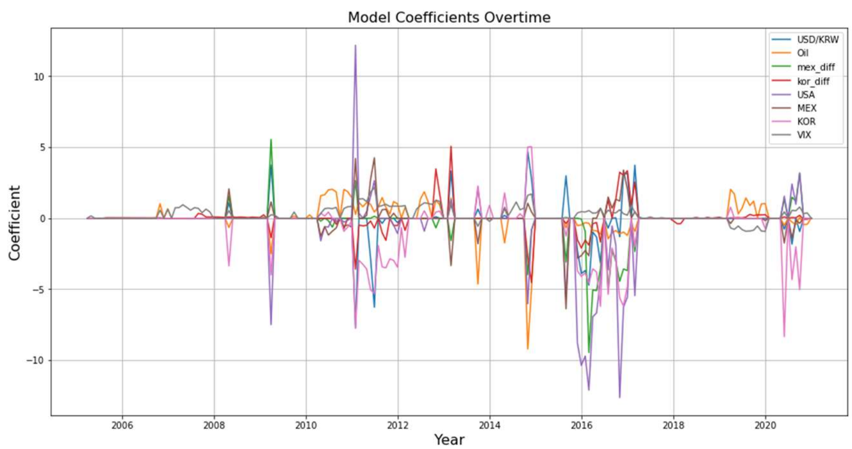
USD/KRW

Presented below is the percentage of iterations the model variables were not removed from the model.

- USD/KRW: 16%
- Oil: 38%
- Mexican Interest Rate Differential: 16%
- South Korean Interest Rate Differential: 54%
- United States Monetary Base: 22%

- Mexican Monetary Base: 22%
- South Korean Monetary Base: 29%
- Volatility Index: 59%

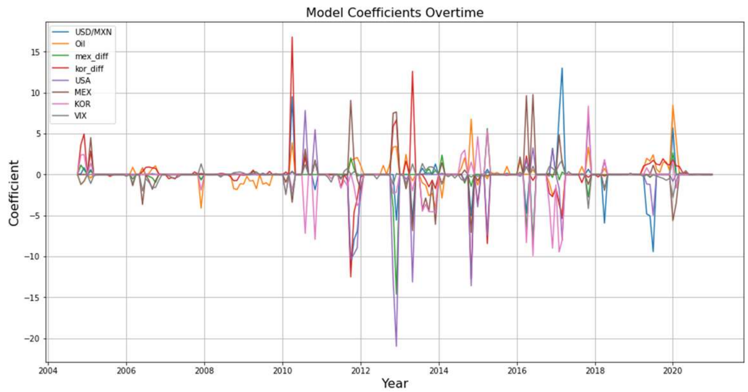
From an economics perspective, we can see why variables related to South Korea are more frequently used in our model than that of Mexico's variables. Visualizing our coefficients overtime allows us to see significant spikes in our estimates based on economic events.



USD/MXN

Presented below is the percentage of iterations the model variables were not removed from the model.

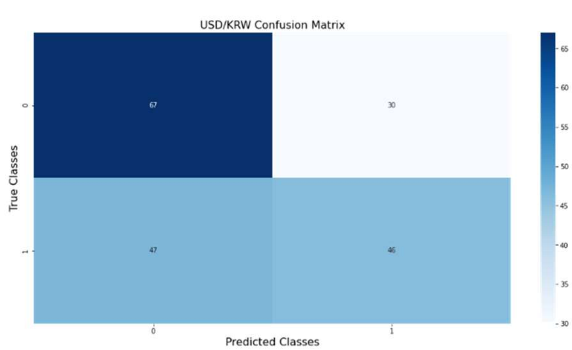
- USD/MXN: 19%
- Oil: 40%
- Mexican Interest Rate Differential: 18%
- South Korean Interest Rate Differential: 50%
- United States Monetary Base: 18%
- Mexican Monetary Base: 22%
- South Korean Monetary Base: 24%
- Volatility Index: 54%

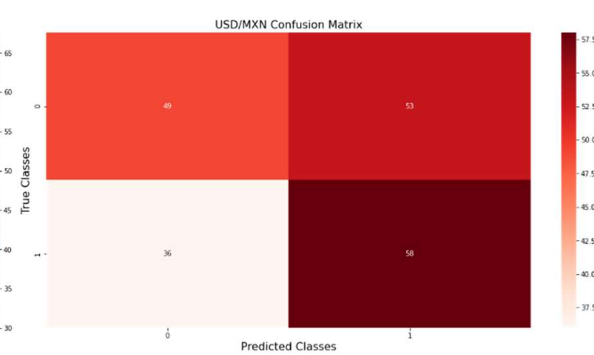


For both models we can see that they have roughly the same distribution of variables used, and both share similar spikes in the usage of the variables. Oddly enough the volatility index of U.S. equities appears to be the variable with the most use for modeling our exchange rate. I can only speculate that our method of modeling exchange rates works best when factoring in levels of uncertainty and volatility.

Model Performance

If we decided to forecast appreciations/depreciations of exchange rates at random. Meaning with no analysis whatsoever, we would expect an accuracy of 50% over the long run. Examining both of our models on the exchange rates we were able to have an edge over a model that classifies at random. When we model the USD/KRW exchange rate, we were able to achieve a testing accuracy of 59.47% and a testing accuracy of 54.59% for the USD/MXN model. It is not surprising that the model forecasting the USD/KRW exchange rate outperformed the USD/MXN model. From all the different iterations of back testing, it was significantly harder to model the USD/MXN exchange rate. However, due to the volatile and uncertain nature of exchange rates I am quite pleased with achieving testing accuracies above 50%.





Plotting a confusion matrix based on our prediction and true classes allows us to get an understanding for where our models struggle the most. For the USD/KRW exchange rate we were able to correctly forecast a depreciation of the dollar 69% of the time while the USD/MXN model was able to correctly forecast a depreciation of the U.S. dollar 48% of the time. If we analyze appreciation accuracy of the U.S. dollar, we see a different story for each model. The USD/KRW model was able to accurately forecast an appreciation of the dollar only 49% of the time while the USD/MXN model had an accuracy of about 62%.

The USD/KRW model struggles to forecast appreciations of the U.S. dollar while the USD/MXN model struggled with depreciations of the U.S. dollar. With this knowledge future models can account for these weaknesses and improve accuracy for each classification.

Trading Strategy

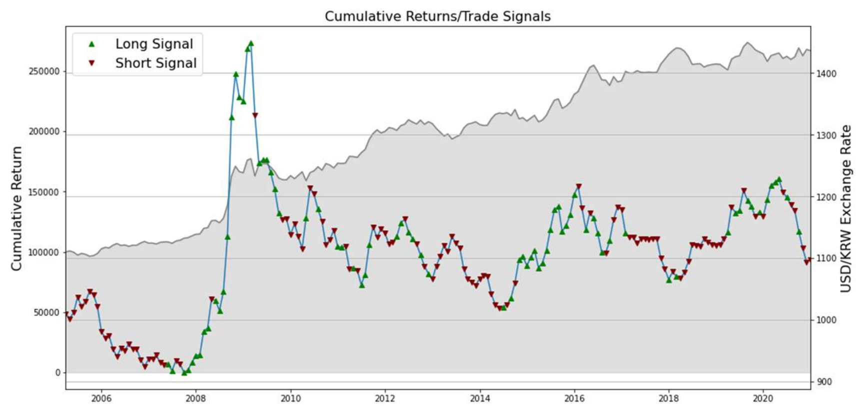
Although we used economic variables to model our exchange rates, our strategy trades on momentum rather than economic justification. Using machine learning applications, we were able to train models that analyzed changes in our variables and outputted probabilities of an appreciation/depreciation over the next month.

With this knowledge we are to design a trading strategy that reevaluates it position every month. Over the long run, it appears that using a threshold of 50% was the optimal hyperparameter that achieved consistent returns. In plain English, if the model generated a probability of an appreciation to be greater than or equal to 50%, then we would go long the U.S. dollar and short the foreign currency. Anything below a 50% probability would suggest we short the U.S. dollar and go long the foreign currency.

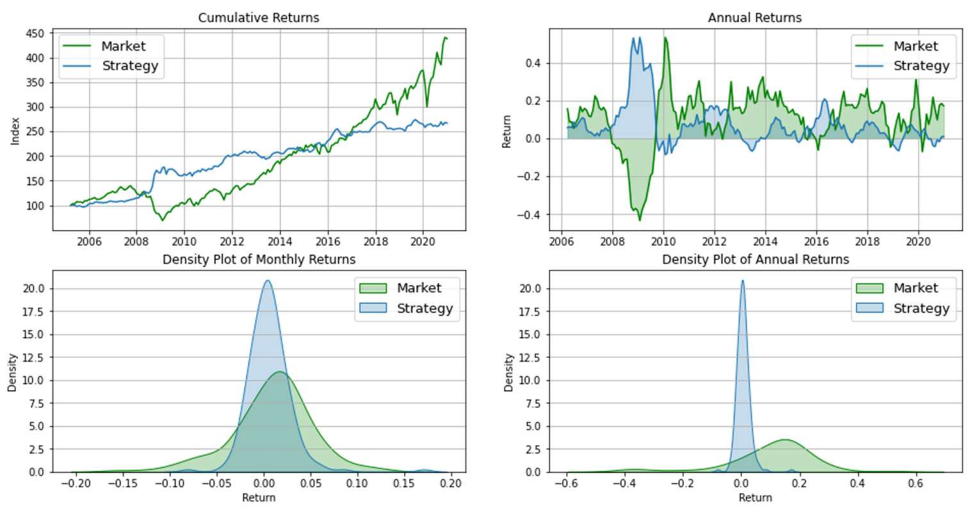
Strategy Performance

USD/KRW

From April of 2005 to January of 2021, this strategy would have a total nominal return of about 168%. Based on our cumulative returns, this strategy obtained an expected monthly return of 0.55% and a standard deviation of 2.3%.



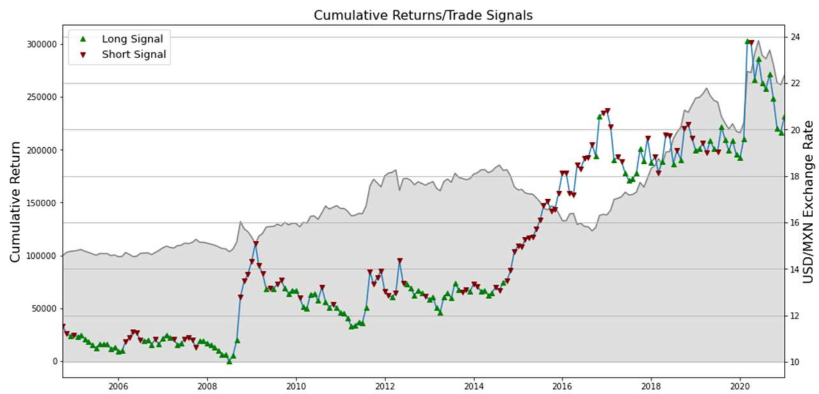
When we visualize or cumulative returns and trading signals, we see that our model will generally stick to a position rather than constantly switching between signals. Although the model achieved a testing accuracy of about 60%, it appears that if we calculate testing accuracy based on overall trends the model would score much higher.



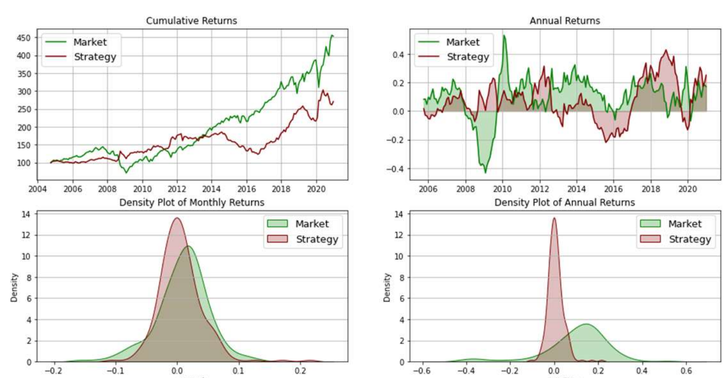
Comparing our returns to that market yield some interesting results. First, we are using the S&P 500 index as the benchmark to compare my strategy against. In the long run the market outperforms our strategy, however, this should come to no surprise due to the very nature of movements in exchange rates. Based on our window of time, our expected annual return is about 7.3% while the market had an expected return of 10.2%. Analyzing our distribution of returns relative to the market we can see our strategy minimizes variance in both monthly and annual returns, because of this we were able to achieve an annual skewness of 2.083 while the market had an annual skewness of -1.22 over this time.

USD/MXN

Although we did not achieve the same levels of model accuracy, we do however achieve a similar total return. Applying our strategy on the USD/MXN exchange rate we achieve a total return of 170% with an expected monthly return of 0.057%.



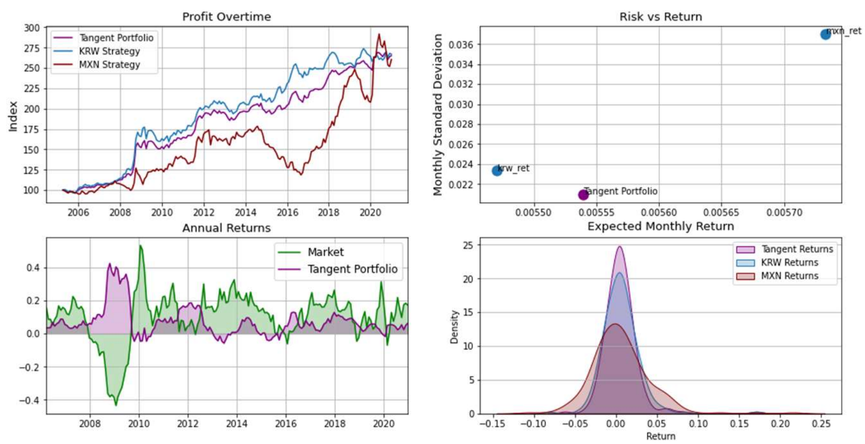
Unlike our previous strategy we can see that our returns have more volatility and experience longer periods of consistent negative returns. Similarly, to our previous strategy once our model forecasts a signal it is highly likely for the model to hold the same signal for the next several months. An issue with the model that must be addressed in future iterations of this project would be finding a model/strategy that can consistently profit of momentum from the USD/MXN exchange rate. The appreciation of the dollar from 2015 to 2017 greatly impacted our cumulative returns and set our equity to levels that were acquired in 2008.



Once again in the long run our strategy does not outperform the market. Looking at our annual returns we can see that our returns are more volatile and larger retractions in our annual returns. However, our distribution of returns visually look identical to that of the previous strategy. That is significantly less variance in our returns. Lastly, our expected annual returns for the market were 10.1% and the expected return on our strategy was 7.57% with a skewness of 0.25.

Portfolio Optimization

If we wish to apply both of our strategies it is important, we find the optimal distribution of liquidity to maximize our Sharpe ratio. To achieve optimization, I was able to apply Modern Portfolio Theory (MPT) in Python to find the tangent portfolio. Using our entire time series of hypothetical returns MPT suggested that the USD/KRW strategy should have acquire about 74% of our liquidity while the USD/MXN strategy will be given the remaining 26%.

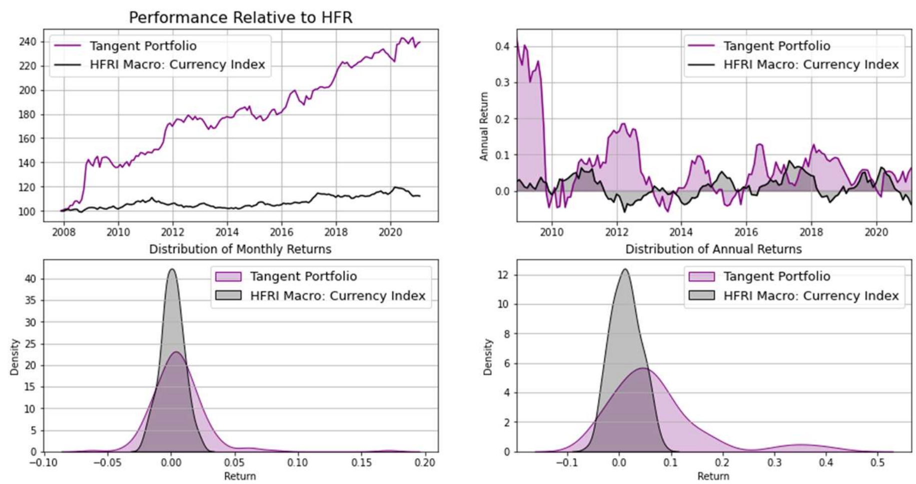


Finding the optimal portfolio allowed us to mitigate unnecessary risk while maintaining the same level of expected returns. Although the tangent portfolio suggests a weight distribution of 74/26, future volatility and returns are subject to change and it is crucial that portfolio optimization is constantly being calculated.

Metric	Tangent Portfolio	USD/KRW Strategy	USD/MXN Strategy
Monthly Return	0.005387	0.005470	0.005733
Monthly Standard Deviation	0.021113	0.023339	0.037020
Monthly Sharpe Ratio	0.255137	0.234379	0.154853
Monthly 5% VaR	-0.020676	-0.024065	-0.044629
Monthly Beta	-0.176000	-0.183000	-0.131000
Monthly Alpha	0.006940	0.007080	0.007340
Gini Coefficient	0.720325	0.719193	0.739261
Annual Return	0.071793	0.072953	0.079306
Annual Standard Deviation	0.091071	0.112940	0.143577
Annual Sharpe Ratio	0.788316	0.645949	0.552359
Annual 5% VaR	-0.038118	-0.044658	-0.154439
Annual Beta	-0.447000	-0.588000	0.034000
Annual Alpha	0.001171	0.001326	0.001390

HFRI Macro: Currency Index

Although we do not consistently outperform the market simply from a return perspective. We must ask the question; do we outperform against those in industry? Data was gathered from the Hedge Fund Research based on their Currency Index to use as a benchmark to compare if our strategy competes with hedge funds. The HFRI Macro: Currency Index is based on strategies that rely on algorithmic models.



The time series ranges from January of 2008 to January of 2021. In this time, we can see that our tangent portfolio has a significantly higher total return than the currency index. The expected monthly growth in the index is about 0.07% with an expected annual growth of 1.2%. Although there is a decrease in the variance of returns for the index, the tangent portfolio compensates with the significantly higher expected returns. That is the currency index has a monthly and annual Sharpe ratio of 0.085 and 0.42. Whereas our tangent portfolio has expected Sharpe ratios of 0.25 and 0.79.

Future Work

My hypothesis is that our forecasts would greatly improve if we were able to make models that are more dynamic. For example, I am in early development in a method that constantly adjusts our threshold hyperparameter. This hyperparameter is to adjust our model to make more accurate forecasts in the future. Like the probability threshold, instead of using a static window size I would have preferred to develop a method in Python that estimates what the optimal window size is based on previous window sizes and how well they performed in the past *N* forecasts. Lastly, to make our portfolio optimization more dynamic. Either take a machine learning approach to finding the tangent portfolio or determine how many months of data do we need to find the most robust estimate to the weights of the tangent portfolio.

Many of these ideas were left on the cutting room floor due to early stages of development and overall poor performance they had in our forecasts.

Conclusion

Our goal was to make consistent directional forecasts and returns while using a machine learning model. We were to develop two models that had testing accuracies greater than 50% while having an expected annual return of 7%. Analyzing our returns, we can see that we achieved an annual Beta of -0.447 while having a -0.75 correlation with the S&P 500 Index. We were also able to show that our model and strategy has greatly outperformed the HFRI Macro: Currency Index, an index that aggregates algorithmic and mathematical strategies such as the methods used in this paper.

Works Cited

Board of Governors of the Federal Reserve System (US), Mexico / U.S. Foreign Exchange Rate [DEXMXUS], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/DEXMXUS, June 3, 2021.

Board of Governors of the Federal Reserve System (US), South Korea / U.S. Foreign Exchange Rate [DEXKOUS], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/DEXKOUS, June 3, 2021.

Federal Reserve Bank of New York, Effective Federal Funds Rate [EFFR], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/EFFR, June 3, 2021.

HFRI Indices. (2021). HFRI Macro: Currency Index; https://www.hfr.com/indices/hfri-macro-currency-index

International Monetary Fund: International Financial Statistics. (2021). Financial, Interest Rates, Money Market, Percent per annum; https://data.imf.org/regular.aspx?key=63087881

International Monetary Fund: International Financial Statistics. (2021). Monetary, Central Bank Survey, Monetary Base, Domestic Currency; https://data.imf.org/regular.aspx?key=63087883

Quandl. (2021). Organization of Petroleum Exporting Countries: OPEC Crude Oil Price; https://www.quandl.com/data/OPEC/ORB-OPEC-Crude-Oil-Price