Dominic Piselli, Larson Cronk, Zachary Jones, Shea Grant
QF 302
Professor Ndiaye
10 May 2024

Predicting Market Crashes

Background

The goal of this project was to build a trading strategy based off of observed “reported crashes” that we selected from the S&P 500. The first challenge was to come up with factors that we could use as potential predictors for market crashes. The specific data we obtained was monthly, and it included the VIX, inflation rates, and unemployment rates. The VIX measures market volatility and is often referred to as the “fear index,” and we believe a spike in volatility could be a possible economic indicator of a market crash. We thought inflation could be a valuable predictor because high inflation often causes the Federal Reserve to raise interest rates and perform quantitative tightening, which can force the stock market down through high costs of capital and attractive bond yields drawing investors out of the equities market. While high unemployment can mean the market has already crashed, we believed that our model could possibly see rising unemployment rates and predict a market crash. This is because rising unemployment can likely result in reduced consumer spending because individuals have less disposable income, therefore reducing revenues for companies and lower corporate performance and spending. We also gathered S&P 500 data, which we use as our proxy for the “stock market.” For all the data to be the same length and frequency, we gathered monthly data spanning from 1985-2023. This time frame provides us with sufficient data to provide enough “crashes” to support our hypothesis. Due to the S&P’s consistent year-over-year returns, it is safe to believe that following crashes, prices will, after time, return to their pre-crash prices or close to them. We predict that investing immediately after a crash will result in high alpha and consistent positive returns since the S&P is recovering from its previous crash and prices rise.

Methodology

After gathering the data via Federal Reserve Economic Data (FRED), we had to define what we considered a "market crash." To do this, we first calculated the monthly percent change of the S&P 500. We initially wanted to define a crash as a decrease of the S&P 500 by 8% or more in any given month. With this definition, we did not have enough data points for "crashes," particularly when we split into train and test data. A small number of observations of crashes would lead to inaccurate results when running the model, since each crash would be weighted much higher and the model wouldn't be replicable over time. To combat this, we changed a "crash" to be a monthly 5% loss in the S&P's value, stating, "If returns < -0.05, return 1, else 0." By doing this, we created a column of 1's (crashes) and 0's (not crashes), based on our group's definition of a crash. By grouping the data into binary values, variability in price changes does not matter. The time series data is either a crash or it isn't.

Our group used a Naive Bayes classifier model for predicting monthly market crashes. VIX, inflation rates, unemployment rates, and market returns were used to create the model. These economic variables are dependent, though Bayes theorem assumes all variables have independence. However, in real life financial applications with naive Bayes models, it is unlikely that all variables used are independent due to the interconnected nature of financial markets.

After splitting the data for training and testing, the model showed a 0.90 test accuracy and a 0.91 training accuracy. However, these high accuracies were due to an imbalanced dataset where market crashes were rarely predicted. To mitigate this imbalance, undersampling was applied to our model, and it predicted market crashes at a higher frequency. Undersampling is a way to balance an uneven dataset by reducing the number in the majority class (non market crashes), but keeping the size of the minority class the same (market crashes). After this change, the test accuracy of our model was 0.74. The naive Bayes classifier model maintained accurate performance while avoiding overfitting, as the test accuracy was slightly higher than the training accuracy.

````
Bayes classifier with undersampled data
```{r}
#Bayes Classifier with undersampled data
library("e1071")
nb2 = naiveBayes(df2[train2,-1] , crash1[train2])
nb.prob2 = predict(nb2 , newdata=df2[-train2,-1] , type="raw") # Compute probs
nb.pred2 = (nb.prob2[,1] < nb.prob2[,2])+0

# Evaluate the accuracy
mean(nb.pred2==crash1[-train2]) #Test accuracy
table(predict=nb.pred2 , truth=crash1[-train2]) #Test confusion matrix

#Test for overfitting
nb.prob.train2 = predict(nb2 , newdata= df2[train2,] , type="raw")
nb.pred.train2 = (nb.prob.train2[,1] < nb.prob.train2[,2]) + 0
mean((nb.pred.train2==crash1[train2])^2)
```

 [1] 0.7391304
        truth
 predict  0  1
       0 12  5
       1  1  5
 [1] 0.7272727
````

To backtest the performance of the naive Bayes classifier model, the indices of the predicted crashes were used as sell signals, and the next non-crash was the next buy signal. Returns from each buy and sell were calculated on SPX, and the cumulative return from the strategy was compared to the return during the entire period of 38 years.

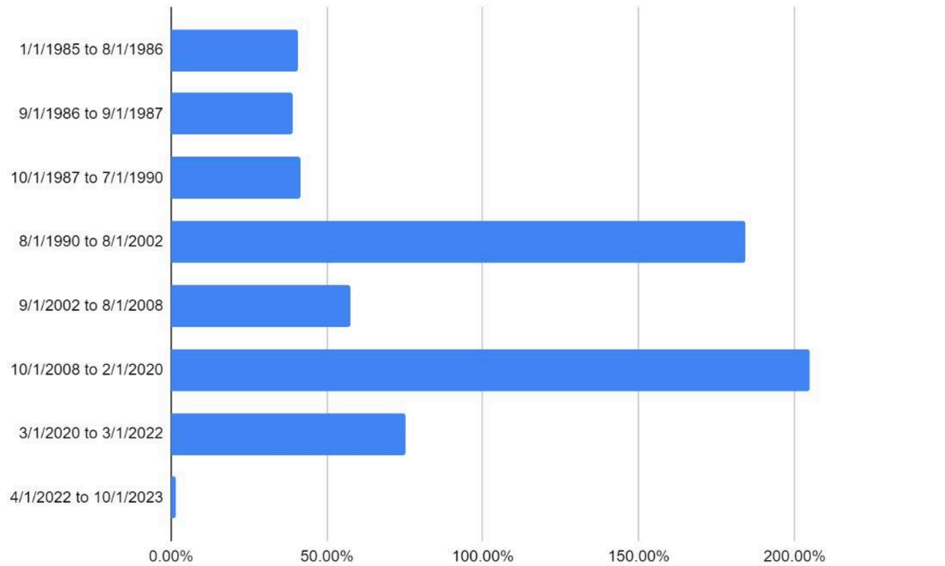
Potential Improvements

Some ways that we have thought about improving this project if we had more time would be to actually test this model with some of our own money, calculate inflation throughout the years, and deduct taxes from our model's returns. We would need a lot more time in order to invest our own money and see what kind of results we would get in comparison to just putting our money into the S&P and letting it sit. The issue with using our strategy over such a long time period is that we can not truly get a grasp on how much money we would have been making in each year, since inflation changes throughout the years. Finally another way to see the complete accuracy of our model would to be take out taxes in our returns each time we have to buy and sell. This would be dependent on how long we take in between each sell because of short term and long term gains being taxed differently. Another aspect of this is it would depend on the tax rules of the state that the user is in as well as what trading system the user is using and whether or not they would have to pay a commission. Making these three improvements would allow us to gain more insight on the accuracy of our model.

Conclusion

Results

The trading strategy returned approximately 6618.02% over 38 years, while SPX returned 2234.69%. This strategy outperformed the S&P 500's returns by a factor of 2.96.


1/1/1985 to 8/1/1986
9/1/1986 to 9/1/1987
10/1/1987 to 7/1/1990
8/1/1990 to 8/1/2002
9/1/2002 to 8/1/2008
10/1/2008 to 2/1/2020
3/1/2020 to 3/1/2022
4/1/2022 to 10/1/2023
0.00%
50.00%
100.00%
150.00%
200.00%