Evaluating the Returns of a Portfolio Constructed from Sentiments of the Stock Ticker Symbol

Jing Ming (Kevin) Li

British Columbia, Canada

Abstract

This paper dives into the field of behavioral finance, researching whether a portfolio built upon sentiment scores of stock tickers will predict and beat the returns of the market. By utilizing python as the primary platform for inputting data and performing statistical regression, this paper introduces an alternate approach in asset pricing, creating a unique sentiment-based portfolio. The sentiment of a stock ticker, which quantifies a ticker's likability from -1 to 1, is multiplied by the returns of the stock. The resultant portfolio returns are regressed on the Fama-French three-factor variables to assess the value of a sentiment-based portfolio.

Keywords: Asset pricing, behavioral finance, sentiment analysis, Fama-French, Python

August 4th, 2019

1 Introduction

A portfolio constructed through purchasing stocks whose tickers have high sentiment scores and shorting stocks whose tickers have low sentiment scores yielded statistically significant results, an indication of the value and uniqueness of an approach incorporating sentiment-based investing techniques. Furthermore, the monthly returns of this portfolio beat the Fama French 3 factor model benchmark by 0.027%, signifying that a sentiment-based portfolio enjoys a 3.24% a year excess over the Fama-French model on average.

2 Literature Review

2.1 Importance of Behavioral Finance

Research has shown that most investors rely on personal instincts to make buying or selling decisions. Implementing elements of human inclinations into an investment strategy, therefore, can allow better predictions of market behavior, as stock price movements are controlled by the supply and demand of the market. Already, firms such as Goldman Sachs are exploiting power of behavioral finance -- the analysis of news sentiment, for instance -- to encourage and maximize portfolio returns.

Psychological "affect", the experience of feeling and emotions, plays a crucial role in financial decision-making. Research suggests neuroscience can oftentimes inform the behaviors of investors. In fact, the existence of separate brain systems that are linked to affect processing are responsible for risk-behavior in financial settings. Excessive stimulation to either brain functions lead to changes in investing behavior.

Undoubtedly, a company's financial report, reputation, and size has definite effects on the behaviors of investors and the future returns of the company. What about the less-researched effects? Currently, the effects of stock tickers have not been widely investigated. Out of myriad factors that could influence investors' decisions, one's likeability of a stock's ticker symbol is a form of 'affect' we are subject to bias.

2.2 Stock Ticker Symbol Analysis

The demand for research in this discipline is perhaps best portrayed in an article written by an expert in life sciences, who sought to find the best 28 stock ticker in the life science industry. He observed:

> "Stock symbols remind me of the website domains. If you have something easy to remember, short, and reflective of your business, you're golden. Even if you just own a great domain, without an actual business, there's a slight chance you can strike it rich."

Two researchers from Princeton University answered this question a few years later. They found that the pronounceability of a company's name and stock ticker has effects on the stock's performance in the days following its IPO. In other words, stock ticker that can be pronounced with ease are likely to beat the market.

Another study published in the Journal of Behavioral Economics finds that the "likability" of a ticker symbol are positively related to a stock's value. The researchers put a group of undergraduate students to test, rating 1959 stock ticker symbols on the basis of likability and pronounceability, which showed to be positively related to Tobin's Q, a ratio between a physical asset's market value and its replacement value.

Evidently, the effects of stock ticker symbols are under-researched. Therefore, exploring the relationship between ticker symbol sentiment and returns will not only contribute to our understanding of finance, but suggest new portfolio construction theories.

3 Data

Datasets used include a sentiment wordbank, the Fama-French three factor model returns, and annual S&P 500 company returns. The manipulation of these data allows for an output of a multivariable regression.

3.1 Sentiment Indicator

The sentiment wordbank used is the Opinion Lexicon by Hu and Liu, containing 6800 words with either positive or negative connotation. These words are delegated to two lists: a positive word list and a negative word list. Words in the positive list, such as 'great' or 'bright', would receive a 1 point addition to the stock ticker sentiment score, while words in the negative list, such as 'cheat' or 'flaw', would receive a 1 point subtraction to the stock ticker score sentiment score. The way in which the algorithm matches

Python's difflib function was used to obtain the 10 closest matches in the lists to a stock ticker symbol. When inputted a stock ticker, the function would return a list of the best matches in the sentiment word library. For instance, the word 'AAPL' would get as its 3 closest matches: 'ape', 'apple', 'peach'.

This feature could be customized by changing the optional argument, n (default 3), the maximum of close matches to return, and cutoff, the minimum score needed for a match to be considered.
In this case, the difflib function operates by scoring each word in the dictionary from 0 to 1, and outputs the 12 highest scoring matches.

Then, each match will be given a score of -1 or +1, depending on whether it is a word with negative sentiment or positive sentiment. Finally, the total sentiment score of the 12 matches will be divided by 12 to yield the average sentiment score of a company.

3.2 Fama-French Three Factor Model

In the 1980s, Eugene Fama and Kenneth French, currently two professors at the University of Chicago and Dartmouth University, respectively, realized a major caveat in the Capital Asset Pricing Model (CAPM). Fama and French observed that under the CAPM model, value and small-cap stocks constantly outperformed the market. Consequently, they developed the Fama-French 3-Factored model – one that considers both the size risk and value risk – to more accurately evaluate portfolio returns.

Today, this model plays a big role in asset pricing by adjusting for the outperformance tendency, whereby the expected return for an asset or for a portfolio is calculated by:

$$r = R_f + \beta_3(K_m - R_f) + b_s \cdot SMB + b_v \cdot HML + \alpha$$

Fama and French discovered that the addition of SMB and HML factors significantly increased the model's level of explanation and helps account of the underlying risk of a portfolio. This model calculates the market returns, or the benchmark returns, for a given period in time; these returns are designed for investors seeking benchmarks for asset class portfolio returns. The benchmark factors summarize the excess return on the market (Km-Rf), the performance of small stocks relative to big stocks (Small Minus Big, or SMB), and the performance of value stocks relative to growth stocks (High Minus Low, or BML).

Parameters	**Description**
Rf: Risk free return	The risk-free rate of return is approximated by the monthly return of a one year treasury bill.
B_3: Market loading factor	The market loading factor, or market beta, measures exposure to market risk. 1.0 is value of B_3 in most equity-only funds.

Km: Market Return	Market return in a given period of time. In this instance, this is the monthly returns of the S & P 500.
Km – Rf: The Market premium	The market factor is the value-weighted excess return of these stocks which is obtained by taking the value-weighted return less the return of the risk-free rate.
B_s: Size loading factor	Measures the level of exposure to size risk. A value of 0 signifies a large cap portfolio, and a value of greater than 0.5 signifies a small cap portfolio.
SMB (Small Minus Big): The size premium	The SMB variable is the estimated returns of a portfolio that invests in small company and shorts big companies. The size premium is calculated by the equation: $\frac{1}{3}(\text{small value returns} + \text{small neutral returns} + \text{small growth returns}) - \frac{1}{3}(\text{big value returns} + \text{small neutral returns} + \text{small value returns})$
B_v: Value loading factor	Measures the level of exposure to value risk. A zero value defines a growth portfolio, while a value of more than 0.3 signifies a value fund.
HML (High Minus Low): The value premium	The HML variable is the estimated returns of a portfolio that invests in value stocks and shorts growth companies. The value premium is calculated by: ½ (small value returns + big value returns) – ½ (small growth returns + big growth returns)

<u>Regression Outputs to Assess Data Quality</u>

1. **Alpha:** Excess return over the benchmark. This is used to evaluate a portfolio's performance. A positive alpha signifies that the portfolio outperforms the market benchmark, while a negative alpha signifies that the portfolio fails to meet the market benchmark.

2. **t-stat:** ratio of the departure of an estimated parameter from its notional value and its standard error. The t-statistic applies to all three factors: Market, HML, and SMB. A higher t-statistic implies a lower standard error and higher certainty.
3. **p-value**: measures the statistical significance of the estimated parameter. A p-value of less than 0.05 is considered statistically significant.
4. **R^2**: Coefficient of determination, measuring how well the regression model fits and explains the observations. A R^2 value of 1 indicates a perfect fit. The lower the R^2, the more unexplained movements there are in the returns data.

3.3 S&P 500 returns

An excel file compiled with the monthly returns of all S&P 500 companies from January of 2000 to June of 2017.

4 Hypothesis

The excess returns of a portfolio, or the alpha, constructed upon the sentiment score of a stock ticker will be positive have a statistically significant alpha when regressed on the Fama-French 3 Factors. This indicates the feasibility of a sentiment-based portfolio and that the stocks of companies whose stock ticker has positive sentiment scores earn superior returns compared to those companies whose tickers have negative sentiment scores.

5 Procedure

5.1 Sentiment Score Calculation

Sentiment score was constructed by importing the positive and negative Opinion Lexicon word lists, the S&P 500 company ticker symbols, and the S&P 500 company monthly returns data. Then, Python's internal "difflib" function is used to obtain the 12 closest matches for each ticker. Each of the matches is analyzed: if the word is found in the negative word list, a score of -1 to added to the company's total sentiment score. If the word is found in the positive word list, a score of +1 to added to the company's unscaled sentiment score. A final sentiment score for each ticker, between -1 and +1, is obtained by dividing the total sentiment score by 12.

5.2 Preparing for Data Science

The ticker symbols and their respective sentiment scores into one list, while the returns for each of company is put into a second list.

5.3 *Filtering out the mild sentiment scores*

Two lists are created: one for extreme positive scores (top 20th percentile), one for extreme negative scores (bottom 20th percentile). This step effectively filters out sentiment scores from the 20^{th} to 80^{th} percentile, which has a less severe effect on portfolio evaluation and thus its ambiguity is eliminated from the program.

5.4 *Working through both lists*

In each list, the returns list is merged with the ticker symbol and sentiment score list. After converting all dates to Python's date-time format, each ticker's sentiment score is multiplied by its returns to obtain the variable "scaledrets", or "scaled returns. Finally, grouping by month to find the average 'scaledrets' for each month completes the data manipulation process for both lists. Multiply returns by sentiment score to obtain 'scaledrets' allows for the creation a weighted portfolio inside top and bottom where firms that have high sentiment scores make up more of the portfolio. For instance, a short more of -1 sentiment score vs -0.1 More weight (buy more) on the most extreme

5.5 *Constructing a portfolio*

To construct a portfolio in which tickers with negative ticker sentiment scores are shorted and tickers with positive ticker sentiment scores are purchased, the 'scaledrets' for the list with extreme negative sentiment scores is subtracted for that of the extreme positive sentiment scores list. This process effectively simulates a portfolio that trades solely from the sentiment scores of ticker symbols.

5.6 *Cleaning the Fama-French data*

After all ticker sentiment scores are obtained, the value of this strategy must be tested against a benchmark: the Fama-French 3-factor model. First, the Fama-French data list is imported and converted to Python's date-time format that is compatible with the dates of sentiment-based portfolio. Next, the list is merge by month and year to the finalized 'scaledrets' data to prepare for the regression.

5.7 *Perform regression*

The independent variables, are the 3 variables of the Fama-French model: Mkt-Rf, HML, SMB. The dependent, y, variable, is the returns of the sentiment-based portfolio: 'scaledrets'. The regression is performed by using the line of code: 'model = sm.OLS(y, X).fit()'.

6 Results

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             scaledrets   R-squared:                       0.872
Model:                            OLS   Adj. R-squared:                  0.870
Method:                 Least Squares   F-statistic:                     481.0
Date:                Thu, 06 Sep 2018   Prob (F-statistic):           2.72e-94
Time:                        15:54:11   Log-Likelihood:                 569.58
No. Observations:                 216   AIC:                            -1131.
Df Residuals:                     212   BIC:                            -1118.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0027      0.001      2.199      0.029       0.000       0.005
Mkt-RF         0.0088      0.000     31.145      0.000       0.008       0.009
SMB            0.0049      0.000     12.997      0.000       0.004       0.006
HML            0.0008      0.000      2.075      0.039    4.02e-05       0.002
==============================================================================
Omnibus:                      189.725   Durbin-Watson:                   1.802
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             4711.187
Skew:                           3.208   Prob(JB):                         0.00
Kurtosis:                      24.961   Cond. No.                         4.64
==============================================================================
```

The raw R-score tells us that approximatively 87% of a sentiment-based portfolio can be explained by the three factors. This group of statistic informs that the portfolio has a fairly decent fit with the Fama-French 3 factors and that developing a strategy through the sentiments of ticker scores a valuable predictor of returns.

The statistics in the bottom group confirms the value of a sentiment-based portfolio. First, the "constant" is the portfolio's alpha, which compares its performance against the Fama-French market benchmark. The alpha is 0.0027, or 0.27% per month and 3.24% per year. In other words, a portfolio constructed solely based on the sentiment of ticker symbols, after expenses, outperformed the regression-based benchmark by that over 3%. That is, on average, a portfolio that simulates buying stocks with high sentiment ticker scores and shorting stocks with low sentiment ticker scores, will make about 3.2% a year excess of FF 3-factor model. This shows that a strategy soled based off investing in the sentiment of ticker scores will yield positive returns. Potentially, this strategy coupled with other strategies can yield powerful results.

Furthermore, the t-stat and p-value tell us that these predicted returns are statistically significant. With a t-score of 2.2 and a P-value of 0.029, the profits of the portfolio are also statistically significant at the 5% level. This means that there is less than a 5 % chance for the returns of the portfolio to happen by chance.

Next, we have the “beta” for each of the three factors, which calculates the exposure of the portfolio. The Market Beta is approximately 0.88, close to the 1.0 value of most equity-only funds. The SmB Beta of 0.49 means that this portfolio chooses to purchase or short more small cap stocks than large cap stocks. A zero value signifies large cap, and a value of greater than 0.5, small cap. The HmL beta of 0.08 shows that this portfolio is mostly constructed of growth stocks. A zero value defines a growth portfolio, a value of more than 0.3, a value fund.

6 Potential Extensions

Given the large datasets and myriad variables used in this project, there is potential to use data to conduct other research. Some potentials areas of research include:

- Correlation between the sentiment score of a company to its yearly returns
    - Examining whether companies with high stock ticker sentiments have high yearly returns
- Ticker Sentiment Score and volatility
    - Assessing whether stock tickers with extreme sentiment scores (+- 0.8-1) would be more volatile than stocks with mild sentiment scores in the days following their respective IPOs.
- Year-to-year comparison
    - Using the data-time function in Python to compare the returns and sentiments of stock tickers on a year-to year or month-by-month basis. The purpose would be to detect any fluctuations in times of crisis (for instance, the 2008 financial crisis) or market downfall.
- Compare with data from Sentdex
    - Discovering a relationship between stock ticker and overall company sentiment

7 Conclusion

This paper further contributes to the pre-existing research on the likeability of stock tickers, demonstrating how a portfolio that solely invests in stock tickers with extreme sentiment scores not only yields adequate returns, but also is a unique approach. More specifically, we find that the monthly 3-factor alpha when investing in this approach is approximately 0.29 %, and the excess returns above and beyond the Fama-French model are statistically significant. This research can be extended further in threefold: employing more accurate data for the sentiment dictionary, investigating the effects of utilizing the sentiments of a company, not only its ticker score sentiment, and developing a strategy that can consistently outperform the market returns.

8 Appendix

8.1 Opinion Lexicon (Hu & Liu) Word Bank

The following website contains 6800 positive or negative sentiment words. The list can be downloaded in rar. file.

https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon

8.2 S&P 500 Data

Th is dataset is stored in a csv. file, containing 500 ticker symbols and respective monthly returns.

8.3 Fama-French data

The historical monthly data for the Fama-French 3 Factor Model can be obtained under a website created by Kenneth French.

http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html

8.4 Full Code

```
import difflib
import numpy as np
import pandas as pd
from mpl_toolkits.mplot3d import Axes3D
import statsmodels.api as sm
import matplotlib.pyplot as plt


#importing Opinion Lexicon Word Dictionary

f = open('bank2.txt')
n = open('neg.txt')
p = open ('pos.txt')

#converting the text files into lists that are readable in python

wordbank=[]

for i in f.read().split():
    wordbank.append(i)

negative=[]

for x in n.read().split():
    negative.append(x)

positive=[]

for y in p.read().split():
    positive.append(y)

#import ticker list
d=pd.read_csv('sp500.csv')
stocklist=d['tick']

tickerhigh=[]
for x in stocklist:
    tickerhigh.append(x)

s=[]

#converting ticker list to lower-case, the suitable format for Python's difflib function
[s.append(a.lower()) if not a.islower() else s.append(a) for a in tickerhigh]

s = list(set(s))

full=[]

#utilizing difflib to obtain 12 cloest matches
for i in s:
    full.append(difflib.get_close_matches(i, wordbank, 12))
```

```python
score=0
final=[]

#calculating total sentiment score
for i in full:
    for j in i:
        if j in negative:
            score-=1
        else:
            score+=1
    final.append(score/12)  #averaging total sentiment score
    score=0


#final2 = ((np.array(final) - np.mean(final)) / (np.max(final) - np.min(final))**3)*100
#put weight on super negative and super positive ticker sentiment scores



finallist=[]
supper=[]
#converting tickers back to upper case
[supper.append(a.upper()) if not a.isupper() else supper.append(a) for a in s]
#putting ticker and score side by side
for i in range(0,len(final)):
    finallist.append(supper[i])
    finallist.append(final[i])


df = pd.read_csv('SP500return.csv') #import SP500 stock returns
df.ticker=df.ticker.astype(str)

df1 = pd.DataFrame({'ticker': supper,
                    'sentiment': final})

datatop = df1[df1['sentiment']> df1['sentiment'].quantile(.8)]

databot=df1[df1['sentiment']< df1['sentiment'].quantile(.2)]

#bottom 20 percentile data
df2bot = pd.merge(df, databot,on=['ticker'])

#converting given date to python's required data-time format
df2bot['newdate'] = df2bot['date'].map(lambda x: str(x)[:4]+'/'+str(x)[4:6]+'/'+str(x)[6:8])
df2bot['newdate'] = pd.to_datetime(df2bot['newdate'])

df2bot['scaledrets'] = df2bot['return']*df2bot['sentiment']  #return * sentiment




#grouping by month and year
df2bot['months']= df2bot['newdate'].dt.month
df2bot['years']=df2bot['newdate'].dt.year
df3bot=df2bot.groupby(['years','months'],as_index=False).mean()

#top 20 percentile data

df2top = pd.merge(df,datatop,on=['ticker'])
df2top['newdate'] = df2top['date'].map(lambda x: str(x)[:4]+'/'+str(x)[4:6]+'/'+str(x)[6:8])
df2top['newdate'] = pd.to_datetime(df2top['newdate'])
df2top['scaledrets'] = df2top['return']*df2top['sentiment'] #return * sentiment
df2top['months']= df2top['newdate'].dt.month
df2top['years']=df2top['newdate'].dt.year
df3top=df2top.groupby(['years','months'],as_index=False).mean()

#Subtracting extreme neg. sentiment scores from extreme pos. sentiment scores
#Constructing a portfolio that trades off sentiment scores of ticker symbols
df4=df3top['scaledrets']-df3bot['scaledrets']

#Converting Fama-French 3 Factor Model Data into the correct form
df = pd.read_csv('FFlist.csv')

FF=df[882:]

FF['newdate'] = FF['Date'].map(lambda x: str(x)[:4]+'/'+str(x)[4:6]+'/'+'01')
FF['newdate'] = pd.to_datetime(FF['newdate'])

FF['months']= FF['newdate'].dt.month
FF['years']=FF['newdate'].dt.year

#Merging Fama French and Simulated Portfolio returns
dfreg=pd.merge(FF,df3bot,on=['months','years'])



X = dfreg[["Mkt-RF","SMB","HML"]] # X usually means our input variables (or independent variables)
y = df4    # Y usually means our output/dependent variable
X = sm.add_constant(X) #adding a constant, in our case, the "alpha"

#Performing the regression
model = sm.OLS(y, X).fit()
predictions = model.predict(X)


#Print out the statistics
print(model.summary())
```