Introduction:

The stock market is a volatile landscape that nonetheless demands even more sophisticated levels of understanding and prediction. Through the last decades, increasingly nuanced approaches to forecasting stock market movement have been developed, with the ultimate goal of price prediction.

In a purely financial world, monitoring solely the financial data may be a viable approach to predicting how a stock will perform. We cannot, however, section away the very real human aspect of stock trading. This is a complicating, chaotic factor. To account for this, we are developing a model to include human data, collected from Twitter, to monitor how human sentiment changes about our stocks over time.

Our essential questions:

- **Can we build a model that can accurately predict if stocks will rise or fall significantly?**
- **Does sentiment analysis improve models based on solely financial data?**
    - **If so, what predictive power do we have?**
- **What classification methods work well on this type of data?**
- **Does the inclusion of technical data improve our models?**
- **What type of sentiment data improves predicting power?**

Methods:

Data Collection: *(Figure 1, green)*

Financial data such as open and close prices and volume were gathered using the yahoo-financials package for Python (Financial Collection.ipynb). We gathered data from January 2009 to present for six Exchange Traded Funds (ETFs): (Materials Select (XLB), Energy Select (XLE), Financial Select (XLF), Industrial Select (XLI), Consumer Staples Select (XLP), and Health Care (XLV)). This data was then reorganized by price at open and closing, then used to calculate 7-, 50- and 200-day rolling averages. We then calculated the days elapsed since rolling average crosses. 'Golden crosses' occur when short-term rolling averages rise above long-term averages, and are usually indicators of an upturn in the stock price. 'Death crosses' indicate the opposite, and occur when short-term averages fall below longer averages.

Sentiment data was sourced from Quantopian, a company that specializes in data for backtesting stock prediction models (Sentiment Collection.ipynb). Sentiment data was collected from mid-2015 to present, and was obtained by Twitter analysis using Stocktwits. Stocktwits rates the language of tweets referencing the given stock, giving the tweet a bullish(positive) or bearish (negative) classifier and quantitative rating, meaning that tweets can more or less positive or negative as well. These data were collapsed into three metrics we used for our analyses: the ratio of bearish to bullish tweets, the bull minus bear value (calculated using the quantitative assessments of each tweet), and the total number of tweets referencing the given stock.

We collected technical data on Exchange-Traded Funds (ETFs), which broadly represent economic sectors, from Ycharts.com. The technical indicators we used were the

Aroon Oscillator, Chaikin Money Flow, Moving Average Convergence/Divergence Oscillator, Rate of Change, and Williams %R. These indicators account for new trends, the momentum of trends, money flow volume, volatility, and a predictor of whether stocks are overbought or oversold. We used technicals on ETFs because we thought they would broadly represent the market. We also used fundamental indicators on individual companies chosen at random (randomstocks.buckmaster.ca), to see if fundamentals for companies have more predictive value (for clarification: stock fundamentals only pertain to individual companies and are not available for ETFs). The fundamentals we used were Price to Earnings Ratio, Price to Book Ratio, Debt to Equity Ratio, Free Cash Flow, and Price to Earnings/Growth Ratio.

Data Cleaning and Integration: *(Figure 1, blue)*

The financial, sentiment and technical data sets were integrated by using identical date-time indices for each dataset (Data Integration.ipynb). We used pandas to manage the dataframes and the pandas.join function to integrate these data sets together. We created three integrated datasets, each using a different master dataset. The cleaned sentiment and technical datasets were used for creating machine learning generated predictions and backtesting. We also determined the correlations between these varied datasets. (Company Sentiment.ipynb)

To create a classifier, we asked if the day's opening varied from previous day's opening price by .7% or more (Predictive Modeling.ipynb). This created a small window for fluctuation we deemed neutral, and defined zones we deemed rising or falling stocks. If the data dropped below the neutral threshold, the day was assigned a classifier value of -1. If the price varied within the threshold, the day was assigned a classifier value of 0, and if the price rose above the threshold, this value was 1.

Modeling: *(Figure 1, orange/purple)*

To evaluate if sentiment data improved our model ability, we tested multiple classifiers on different sets of data (Predictive Modeling.ipynb). These split datasets are named by what data is included from the total vector. The 'minus' data set, for example, contains all financial data as well as the 'bull_minus_bear' metric. The 'price' dataset excludes all data except the day's opening price. We use this as proxy of the variability of the stock: stable stocks should be more easily predicted using price alone than more variable ones. The 'crosses' data set includes the days elapsed since each cross, and the 'technicals' data includes the CMF (Chaikin Money Flow), MACD (Moving Average Convergence Divergence) and Williams %R.

We chose three classification methods to experiment with our data: Support Vector Machine, k-Nearest Neighbors and Decision Tree, implemented by scikit-learn. These methods were parameterized and cross-validated using the scikit-learn function GridSearchCV. For each method, we varied the following parameters:

- SVM: the value of C, and using a linear or rbf kernel
- kNN: the number of neighbors used to classify
- Decision tree: the maximum depth of the tree

Each dataset was split into a training and test data. Training data comprised 40% of the dataset. Each model was parameterized on the same set of training data. Cross-

validation produced the best parameters to use on each dataset, and the test data was evaluated using the maximized parameters.

We iterated this modeling strategy over each of our datasets and reported the prediction accuracy for each classifier and dataset. These were our final data and were plotted on a heat map by accuracy, as in Figure 2.

Results:

We constructed a modeling paradigm for predicting the short-term rise or fall of a stock price. The highest accuracy that we were able to achieve was 72% prediction in the XLP dataset using the rbf kernel of SVM as a classifier. This was a marked improvement over using price alone as a predictor of rising or falling, indicating that we were able to account for more variation using sentiment and technical data.

Indeed, in all tested stocks the incorporation of sentiment data, either through the ratio of bear to bull tweets or the difference between the quantitative assessments of tweets. SVM was consistently the top performing classifier.

Essential Questions:

- **Can we build a model that can accurately predict if stocks will rise or fall significantly?**
  - Yes. The data and model with the highest accuracy was the XLV stock, using an svm model on data that included the technical data. Its accuracy was 72.3%. (Graphs/XLP_model.pdf)
- **Does sentiment analysis improve models based on solely technical data?**
  - Yes, in most cases the inclusion of sentiment analysis improved the predictive power of our base-level model. (Data/final_results.csv, Figure 2)
- **What classification methods work well on this type of data?**
  - In all cases, the rbf kernel of svm was best able to classify the data, followed by kNN and decision trees. (Data/final_results.csv)
- **Does the inclusion of technical data improve our predictions?**
  - Yes, but sentiment data is still valuable and descriptive. (Graphs/XLP_model.pdf, Figure 2)
- **Does sentiment data correlate with fundamental data?**
  - The overlay indicates that total messages tweeted correspond to sharp increases and decreases of mean stock price. However, individual fundamentals did not seem to correspond strongly to sentiment or stock price.
- **What type of sentiment data improves predicting power?**
  - In general, stocks with a higher number of tweets about them tended to benefit the most from including the sentiment data, as indicated by the 70% correlation between average daily tweet number and accuracy differences. (Figure 3)

Discussion:

For this small dataset, sentiment data was a useful inclusion for improving prediction accuracy. A higher number of rated twitter messages per stock was correlated

with increased gain in prediction accuracy, meaning the inclusion of this data more strongly benefitted the model when there was more of it. Intuitively, this makes sense, but this was backed up by the data we were able to collect.

Interestingly, stocks that had a higher number of daily tweets tended to have a higher ratio of positive tweets in general, which could indicate a bias in our results. However, none of the sentiment data included the raw number of tweets as a classifying parameter, so this problem was at least mitigated in our modeling, i.e. they were not confounding variables.

Based on our preliminary analysis, there was also significant variation in the effectiveness of including sentiment data into our models, indicating that individual stocks may have sentiment data that is more or less useful. It is entirely possible that evaluating if sentiment data is a useful inclusion cannot be done globally across stocks, but rather must be evaluated on an individual basis. XLE, for example, benefitted greatly from this inclusion, while XLI had a much less significant increase in accuracy. That said, in many cases sentiment data is likely redundant, and given the difficulty of acquisition, probably not worth the extra effort.

Future Directions:

The most obvious next step is to repeat this same set of core analyses using many more stocks. This will help to bolster any conclusions we can make about the efficacy of sentiment analysis in predicting stock movement.

A further comparison of the predictive power of both sentiment and technical data would also be interesting, as while technical data allows for prediction based on historical patterns, twitter data could be much more immediate. To test this fully, however, we would need to find stocks that are volatile and have sufficient sentiment data to look for immediate patterns. It is also possible that sentiment data is not useful for extremely quick analysis, but might be more useful for describing and predicting general trends. More analysis will be needed to uncover the role sentiment data could play in predicting these more long-term trends.

Additionally, we envision building a successful regressive model to predict not only if a stock will rise or fall on a given day, but also by how much. This is an important feature for use in actual stock trading, as a stock rising or falling is not a binary feature. A steeper rise or fall has different consequences than a shallower one. Tools for prediction will need to, as best as possible, move forward this type of predictive power.

Another tool that could extend from this is a rapid scanner that could predict what stocks may move up or down sharply, based on sentiment data. One possibility would be to repeat these analyses on many stocks, and build a database of stock data for each, then predict if the stock will rise or fall given the present data, calibrated to each stock's historical data.

Regardless, this project has reiterated that stock data is fundamentally heterogeneous, and that predictive models are best built individually. While sentiment data may not be a beneficial component of successful models for some stocks, others may have significantly benefit from sentiment analysis.

Figures:

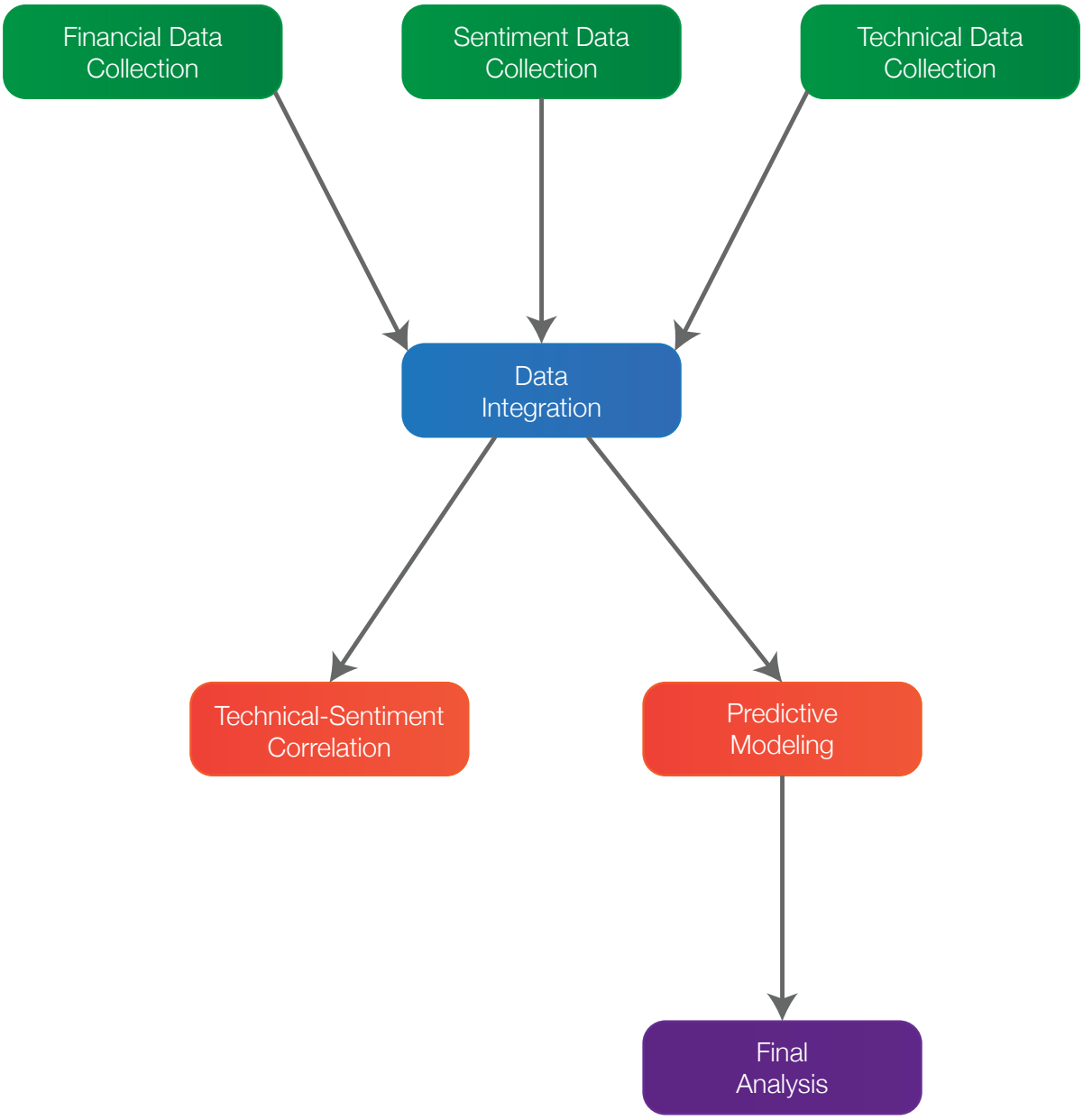

Figure 1: *Script flow chart.*
The scripts are separated into categories. At the first level, scripts in **green** are for gathering raw data. These datasets are then combined in the Data Integration step, **blue**. These integrated datasets are used to examine correlations between datasets (Technical-Sentiment Correlation) and for predictive modeling, **orange**. The output from modeling is further analyzed in the Final Analysis script, **purple**.

Chart Comparing Classifiers and Datasets

	svm	kNN	tree
sentiment	0.6875	0.590909090909	0.602272727273
ratio	0.681818181818	0.568181818182	0.613636363636
minus	0.659090909091	0.4375	0.403409090909
sent only	0.5625	0.4375	0.5625
price	0.551136363636	0.556818181818	0.551136363636
crosses	0.625	0.528409090909	0.551136363636
techs	0.5	0.5	0.5625

Figure 2: *Modeling Output for XLV*
Classifier and dataset comparison for the Health Care (XLV) ETF. Rows describe the varying datasets, while columns describe the optimized model. Data is colored by accuracy. The highest prediction accuracy was obtained using both sentiment data metrics (sentiment), the ratio of bullish to bearish tweets (ratio) as well as the bull-bear quantitative metric (minus). The inclusion of sentiment data improved the model by more than 13%.

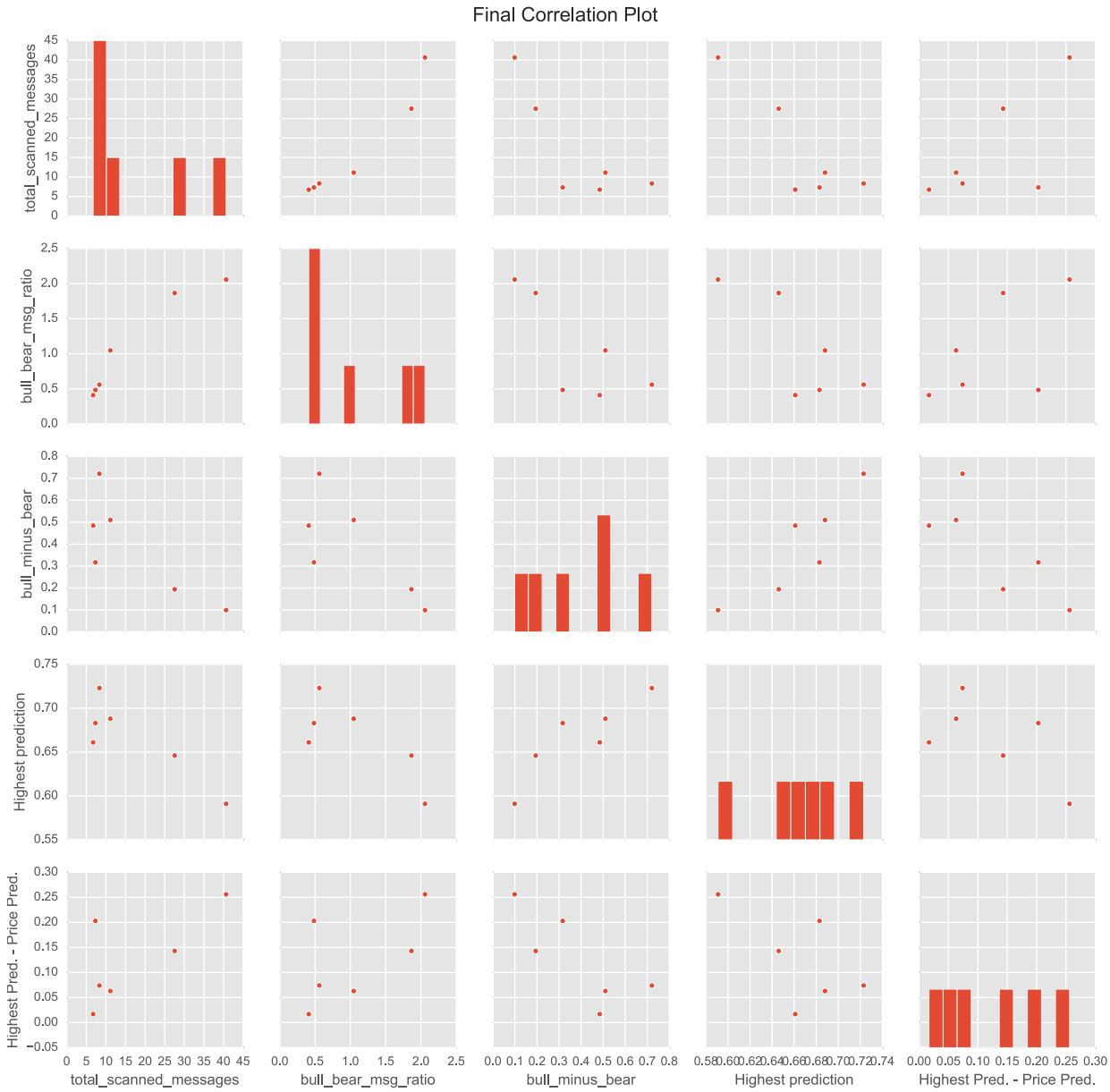

Figure 3: *Final result correlations.*
A pairplot using the Data/final_results.csv file. The strongest correlation was between the average daily number of scanned messages (total_scanned_messages) and the bull_bear_msg_ratio. However, both were strongly correlated (70% and 60% correlated, respectively) with the prediction improvement (Highest Pred. – Price Pred.).