# FP3 – Data Analysis Plan

## Overview

This project is focused on predicting stock prices via twitter sentiment analysis. Stock price predictions are volatile and highly sensitive to external factors. The rise of "meme stocks" such as Gamestop, which gained huge social media traction and astronomical gains and losses within mere hours, begs the question of the degree of influence social media has on stock prices. The project aims to analyze the degree of social media influence on public financial markets. Thus, this project will focus on collecting and analyzing tweets about public stocks and use a predictive model to estimate the closing price of these stocks on future days.

There are two datasets used, a financial dataset which includes 5 years of stock data and derived financial metrics for various companies (~1500 rows), as well as a dataset of tweets tweets (~1000 tweets for now). For the financial dataset, our features are as follows: Close, Volume, Change %, HL %, HPR, Market Capitalization, Positive Sentiment Score %, Neutral Sentiment Score %, Negative Sentiment Score %. The target for this dataset is Close. We are trying to use sentiment as well as previous movement and metrics in the stock data to predict a future closing price. For the tweets dataset, we are utilizing n-grams and vectorizing the tweets into a matrix of many adjacent word-combination features. The target for this dataset is a polarity score, Sentiment Score, provided by an external sentiment lexicon and corpus (AFINN).

<u>Questions regarding the dataset:</u>

*What ML model would we use to minimize the classifier accuracy difference between testing and training data to increase generalization of the model?*

*Would normalizing stock data increase our classifier accuracy?*

*Are fetched tweets and financial metrics interdependent and representative/correlated to stock volatility?*

Potential answer: This largely depends on the type of tweets collected. After analyzing classifier accuracy, adjustments to tweet collection methods can be made to potentially increase accuracy/lower overfitting, e.g. lang='en' parameter may help improve model accuracy but restricts unclassified tweets, update to full-archive api (only 250 requests/mo allowed) once we verify functionality of classifier, filter out tweets with more than a set threshold of tickers ($) to get more accurate sentiments on a specific public company, use emoji supported sentiment lexicon for initial polarity score label for our sentiment classifier.

**Data Analysis Plan**

To analyze our data, we are tackling multiple regression problems. The sentiment analysis classifier will vectorize tweet text and predict a sentiment class: positive, neutral, or negative. The stock price prediction model utilizes regression using an aggregate percentage over many tweets of the previous classes in addition to financial data for the stock being analyzed (all quantitative features).

For the sentiment classifier, the following algorithms will be utilized: LinearSVC, MultinomialNB, and LogisticRegression. We chose to focus on these algorithms (other than DecisionTreeRegressor) because the following source mentions they are the most effective to tackle problems regarding text sentiment: A. Pak and P. Paroubek. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. Lrec, pages 1320–1326, 2010., Random Forest - type of decision tree algorithm, https://link.springer.com/article/10.1007/s10796-021-10135-7. For the stock regressor, we will utilize LinearSVR because we don't have reason to think one should be better than another. In terms of relevant assumptions, there's one: for MultinomialNB, we must make sure class conditional independence is satisfied.

We are planning on normalizing stock price data using a standard scaler, because stock prices are based on shares outstanding and aren't reflective of the market capitalization (true market value of equity). For the sentiment classifier we are utilizing tf-idf with n-grams to vectorize the tweets into a matrix of many adjacent word-combination features. N-grams help cover combinations of words that may have different meanings and sentiment together than apart (e.g. "bad not good" vs "good not bad"). We will also remove stopwords to combat overfitting.

It would be useful to visualize stock price data for a stock to get the gist of the price trend for a particular ticker. Additionally, a scatter matrix can be used to visualize correlations between features, which is also useful for identifying which features break conditional independence for Naive Bayes classification algorithms. Also included but not one of the two visualizations is a word cloud using the wordcloud library just to indicate the most common words among our tweet corpus. It also shows that restricting multiple tickers within tweets would be a good idea, as we don't want overall sentiment to be masked by large numbers of other companies.

## Visualizations

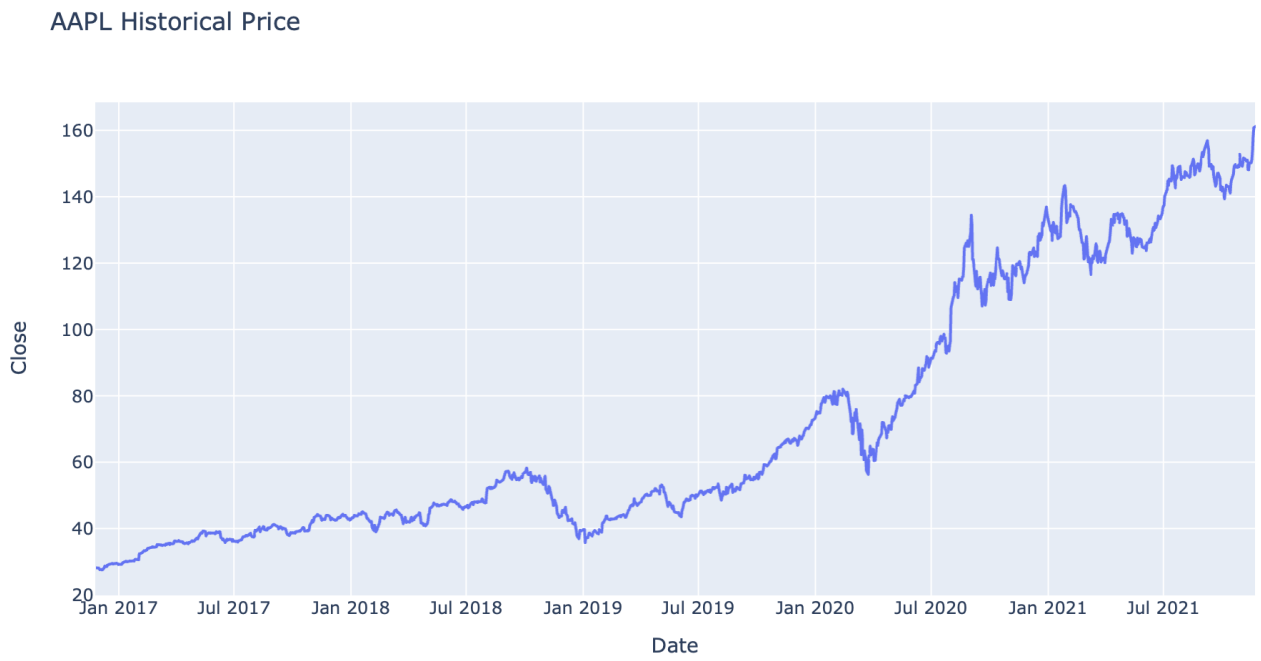


*Figure 1: Visualization of financial dataset which displays the historical price, our target, of Apple Inc. (AAPL) over 5 years.*

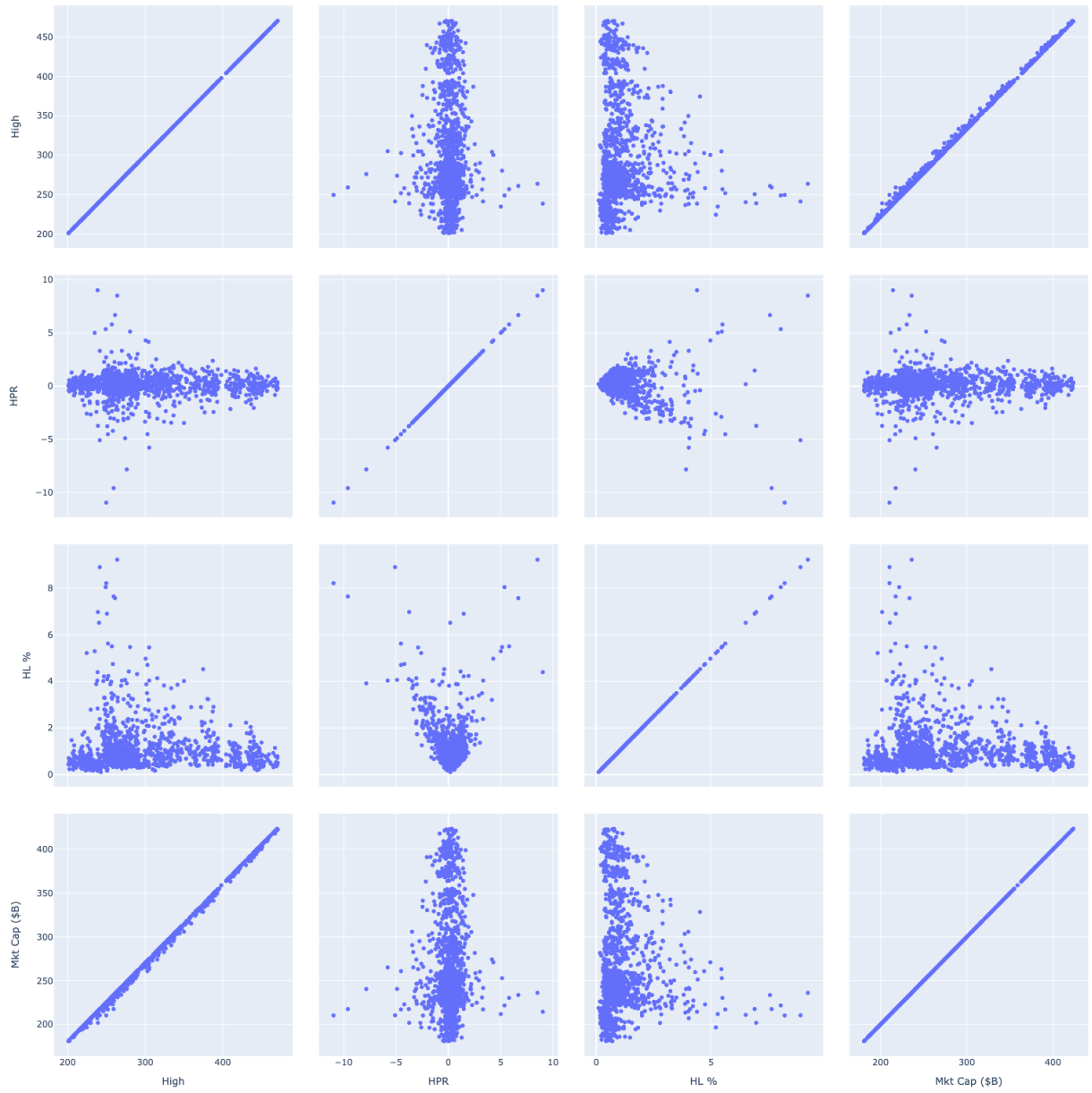


*Figure II: part of a scatter matrix visualizing correlations between features in the financial dataset. Here, we can identify features that break conditional independence for Naive Bayes classification algorithms as their associated scatterplot with a differently-named feature will be linear.*


*Figure III: wordcloud for current tweet corpus for '$AAPL' search. Suggests to exclude tweets with other tickers such as '$AAPL $TSLA $SPY' to increase sentiment classifier accuracy.*