**MS3 Report for OCaml Financial Analyzer**

**Vision**

Our initial vision for this project in its entirety was to create an OCaml Financial Analyzer. It should in theory be able to analyze historical data and use that to predict and inform about future trends of many different types of data. Now, we have generally leaned toward focusing on stock tickers and pure analysis. Rather than have the application analyze for the user, our goal was to present all the data necessary for the user to make informed decisions about stocks (probably buying/selling). While they aren’t massive changes, they definitely deviate from the original vision we had for this project.

**Summary of Progress**

In MS2 we had to implement a barebones demo project to show a proof-of-concept of our vision. We created a simple demo that takes an command line input of a stock ticker and graphs the historical data in a candlestick plot. Since then, we have added a ton of new features to accompany this main graph. First of all, we implemented a GUI using the Bogue library. Rather than have command line input, the input takes place directly on the initial GUI screen. There is also a choice of API to use in calculating all of these indicators and making these graphs. As of now, there are four options: Alpha Vantage (what was in MS2), Polygon.io, Twelve Data, and APIStocks. After clicking a button, then all of the data is presented.

There are three main tabs on the GUI. The first one solely contains the enlarged graph from before. The second and third one contains various plotted indicators, including MACD, RSI, OBV, ATR, CCI, Bollinger Bands, and Stochastic Oscillators. On top of that, the third tab also contains some current events information, in the form of news articles. By typing the URL presented into a web browser, the user can also learn more about the context of a company (and potential reasoning behind stock price changes).

**Activity Breakdown (hours do not include meeting times)**

Joshua Ochalek **(32 hours)**

* Implemented the GUI using the Bogue library
* Assisted Arnav in making various API calls and processing the data
* Added the majority of test cases

Krish Mehra **(28 hours)**

* Outlined necessary tasks for a good timeline in completing the project
* Generated JPEG graphs for each indicator variable
* Implemented the analysis functions, with calculating various indicators

Arnav Tevatia **(20 hours)**

* Worked closely with the API requests, obtaining keys, and processing the data
* Implemented getting ticker news and adding to the GUI with Joshua
* Added some test cases for analysis functions with Joshua

**Productivity Analysis**

As an entire team, our productivity was rather high. In general, we accomplished what we planned in our sprints. With finals week approaching, it became a tad difficult to continue to accomplish the same workload, so it was somewhat challenging in completing the project and conforming to requirements. This ended up becoming somewhat of a sprint at the end to complete all the requirements, but we still had a viable outline for it because we extensively planned out ideas and features before implementing. For the most part, our estimates of what we could do were not that inaccurate. While we did scope down the project and focused more on certain parts of our initial vision, it was still successful in terms of creating a prototype of our application.