Jonathan Giguere, DATS 6401, Summer 2020

Final Technical Report

## Robinhood Popular Stock Analysis

**Abstract**

Robinhood is a mobile app that allows its users to invest in the stock market commission-free. The app has brought many traders into the stock market because of its cheap and accessible means to participate in buying and selling securities. Recently, there has been quite a bit of market volatility due to the COVID-19 pandemic and the restrictions it has brought on the economy. Many new investors are drawn into the market thinking that the volatility will help them get rich quick. This project will focus on the stocks that are most popular on the Robinhood platform, and by doing so, give a picture of what the average retail investor thinks is a good bet. Here a stock’s popularity is defined by how many Robinhood users are holding its shares. Aside from assessing a stock’s popularity on Robinhood, the tool created for this project will also allow users to see some basic historical information about stocks of interest. This report will discuss the specific objectives, approach, requirements, and initial conclusions drawn from this effort.

**Objectives**

The primary objective of this project will be to create a visualization that shows which stocks are most popular on Robinhood over a given timeframe. Showing the most popular stocks on any given day might be an indicator of what the average investor thinks is a good bet. A secondary objective will be to show historical market-close price and volume data for the most popular stocks on the platform. Putting this information in one interface will aid users in their analysis of the most held stocks on the app. Comparing a stocks popularity on the platform to its price or volume might reveal some insights related to retail investor behavior.

**Functional Requirements**

The creation of the analysis tool requires stock market data to be pulled via the Robinhood API, formatted in a way that allows it to be visualized, and stored in place that can be accessed via D3.js. For the tool to be useful to the end user, it must allow selection of a stock ticker to view historical information and update visuals accordingly. The web-interface for the tool must also be clean and simple to use so that the end user can efficiently draw meaningful conclusions from the data.

**System Architecture and Description**

The popularity data was acquired from Kaggle. A Kaggle user created a dataset containing the number of shares held on the Robinhood app for each stock on any given day from 2018 to present day. This data was downloaded and there was a different .CSV file for each stock. These files were processed with Python and appended to one another to get a single table that shows the popularity of selected stocks over time. The processed .CSV was then pushed to a GitHub repository. For the historical market data, a Python script was used to pull data from the Robinhood API for the past three months, preprocess the information, and upload it to a GitHub repository as a .CSV file. With the popularity data and historical data in a public GitHub repository as .CSV files, the data can be easily accessed with D3.js to create the visuals for the tool. Any updates to the data can be pushed to the repository and the tool will update accordingly.

The original intent of this project was to use live data, but due to limitations of the Robinhood API not being able to pull historical popularity data, the Kaggle dataset had to be used. Because the Kaggle data is not updated to the most recent day, all data will be static. The analysis will contain popularity and historical data from the past three months from July 1, 2020. Making the tool with live data should be possible but is outside the scope of this project.

**Development Platforms**

The user interface for the tool will be presented as a web page created with HTML, CSS, and JavaScript. The interface will consist of one page with a dropdown for selecting a stock ticker to analyze as well as two visualizations created with D3.js. The visualizations will allow for user interaction via data selection and tool-tips. Bootstrap will be used to add responsiveness to the page and style elements when needed.

**Proposed Visualizations**

D3.js will be used exclusively to visualize the data for the tool. As stated previously, there will be two visualizations on the page: one to show the popularity of stocks over time and another to show the historical price and volume information for a selected stock.

For the popularity visualization, an animated bar chart will be used to show the number of shares held for each stock. Bars will transition in and out of the visual as the number of shares held changes daily over the past three months. The top ten most popular stocks will be displayed as they enter and exit the chart area.

For the historical stock data, a combined line and bar chart will be used. The line will represent the historical market close price data and the bars will represent the stock’s trading volume. This chart will be dynamic and change upon selection from a dropdown menu. In addition to updating for different stocks, the chart will include tool-tips and smooth transitions.

**Experimental Analyses and Conclusions**

The popularity analysis revealed that the most popular stocks being held on the Robinhood platform are primarily household name companies that have experienced significant dips in stock price. As of 7/1/2020 some of the most held stocks include companies like Ford, GE, Disney and American Airlines. Despite this, there are also some lesser known companies. As of 7/1/2020 Aurora Cannabis holds the 10th place spot on the list. It would be interesting to compare this top ten list to other, more established trading platforms to see how Robinhood investors, who are presumably casual investors, differ from other traders. Conclusions drawn from the historical price and volume data are in line with traditional investing ideas that higher volume for a stock indicates that a price movement up or down is more significant. The historical charts can ultimately be utilized by users to form their own conclusions and inform investment decisions.