Springboard DSC--Capstone Project 1
NFL Breaking News Classifier
By Logan Larson
March, 2019

The problem: Since there doesn't exist an application that can deliver comprehensive NFL breaking news updates in real time, the goal of this project is to build a classifier that can recognize relevant Tweets from a fantasy football news feed.

The client: This project targets my current employer and those who participate in fantasy football. For my employer, it's becoming increasingly difficult to compete with Matthew Berry's Fantasy Life app, which currently provides mobile alerts with a slight delay (though it only covers the highest-priority news, not everything that could be relevant). For fantasy football participants, having instant knowledge of all relevant transactions could potentially provide a significant advantage over the competition, particularly in terms of being the first to add free agents from the waiver wire. In a broader sense, participants otherwise have to go into their mobile apps to refresh news streams to get updates on the non-highest-profile players. If they don't, they miss probably over 50 percent of actionable news that could have otherwise improved their team.

The data: In the NFL world, there's conveniently one incredibly well-networked and accurate reporter, Adam Schefter, who breaks a large percentage of the high-profile breaking news. Convenient as well, Schefter exclusively uses Twitter to break news, so we can collect the data we need by scraping his timeline.

Initial approach:

I suspect I'll eventually need to include more reporters than Schefter. Ideally, I would analyze the news from each of Schefter, other select reporters like Rapoport and Garafolo, and local beat writers employed by respective NFL teams. These sources thoroughly cover all significant news coming out of NFL organizations, and also are by far the most reliable. However, I feel -- given the individualistic nature of how humans communicate/write tweets -- that it could be troublesome to just lump tweets from all of these sources and then try to classify them all at once because individual people may have nuances to how they speak. So the solution to this problem may include having one classifier for each source, and then prioritizing any competing tweets based on the reputation of the reporter (creating a hierarchy of reporters would be easy).

Deliverables:

My deliverables would likely include the code and a “Twitter bot” linked to something like a Raspberry Pi that would automatically classify and report news in real time. I could ultimately present why my product is desirable with a comprehensive paper or slide show.