**Name, Contact info (e.g. email/phone).**

* Name: Joseph Dixon
* Email: jpdixon95@gmail.com
* Phone: 203-733-7457

**Title of the project**

Automating NoSQL Database Creation with MongoDB and the setlist.fm API

**High level description of the project: what question or problem are you addressing?**

I have a hard time motivating myself to build anything practical, so I’m going to tack hard to the other side of the spectrum and do something completely fun. I am a passionate music fan and love attending concerts – one of my favorite groups is the Grateful Dead, a genre-defying jam band with a mind-boggling catalog that toured nearly every year for more than three decades. It’s often been said that the Grateful Dead never repeated a setlist. However, I’m curious: how close did they get? This question spawned an idea for a two-part project that could span two phases, one for each practicum. The first would be a data engineering-oriented collection phase involving API work and NoSQL databases. The second would be a market-basket analysis of the data collected in the first phase to answer the above question. However, rather than focus on the Grateful Dead exclusively, I want to build a protocol that allows a user to input any artist they want, and have a script build a local MongoDB database with all of that artist’s concert data, sourced from the setlist.fm API.

**What type of data science task is it?**

This phase of the project is more data engineering than data science, but it will be principally concerned with RESTful API work and NoSQL databases, namely MongoDB. I also hope to incorporate some basic software engineering principles like abstraction, threading, and perhaps also containerization if time allows.

**Data: Brief description of data. How big do you expect the data will be? Is amount of your data too big or too small? If you're web-scraping or collecting data, how long do you expect to collect the data?**

The data will be relatively small in terms of instances or “rows”, but will be fairly complex in terms of inconsistent schema and different data types. I anticipate that for most artists, the API call process will take only a handful of minutes, provided that the API request size limits are fairly robust.

**How will you analyze the data? What machine learning methods do you plan to use, and/or what business intelligence aspect do you plan on incorporating?**

This first phase of the project isn’t concerned with analysis as much as it is concerned with building a protocol for database creation based on simple user input.

**Describe any anticipated difficulties and problems. Discuss how you may overcome the problems.**

I worry about making requests for artists with a larger corpus of concerts, as this will likely run afoul of the request limiting that the setlist.fm API has. This will mean doing some strategizing around rest times and perhaps also some threading.

**Suggest a timeline for the project.  This should be a weekly breakdown of what you plan on doing each week.**

I suspect that the work will be unevenly distributed throughout the weeks, so here are the phases that I see this work being divided up into:

1. Digest musicbrainz and setlist.fm API documentation
2. Build user input function and functions to parse JSON responses from API calls
3. Build functions to initialize MongoDB collection for an inputted artist. Explore whether containerization can help others run this without extensive local MongoDB setup
4. Build machinery (e.g., functions and classes) to iterate through artist setlists and add MongoDB documents for each concert.
5. Add logging information and graceful error handling so we don’t watch a blank terminal for several minutes while script runs
6. Do some lightweight querying against new MongoDB collection to demonstrate efficacy
7. To set up success for the next phase, build MongoDB collection class with methods to parse comma-delimited setlists, comments, etc.