Deployment Plan: Spotify Recommendation Engine

Grant Jurgensen, Stephen Longofono, and Stephen Wiss

This document details the process of identifying a market for our song recommendation software, the necessary steps to enter that market, and the associated costs. In brief, the plan is threefold: offer the software as an exclusive, non-streaming application on the Spotify app store, market the application to non-technical users as an alternative to the Spotify recommendation engine, and market to technical users as a customizable recommendation sandbox. This task is no small feat. The software was designed to be open source and developer-oriented, and as such would required significant changes and some degree of obfuscation to become a marketable product. These changes and the associated costs are included here, as the software is unsaleable without them.

1. Defining the Market

There are many successful companies based around the concept of targeted services. Amazon, Netflix, Spotify, and Google Adwords all make use of machine learning algorithms as a tool to provide their users with only the most relevant product, entertainment, or advertisement. The utility of such tools is apparent in the never-ending stream of information, material goods, and user content that composes the modern world. A distinguishing characteristic among these companies is found in the way the algorithms are applied; Amazon and Google seek to increase the likelihood of your purchase, Netflix and Spotify seek to enhance the value of their services, and garner a larger user base to serve advertisements to.

As one might expect, the intent of our recommendation software is most closely aligned with that of Spotify. Spotify built its customer base by undersale of the digital music industry, but retains that customer base by acquiring new content and serving it intelligently. Our application will follow in the same vein: providing an autonomous means of identifying relevant content. Specifically, we seek to provide a recommendation service for new and existing music, with a focus on providing an alternative to Spotify's own featured

songs. Our product's value is then a function of how well it identifies distinct content, and the novelty of a customizable recommendation.

Identifying our target merits a closer look at what Spotify is and how people use it. For the uninitiated, Spotify is a music streaming service that offers access to record label and media hub content. They offer a free service which is largely paid for by advertisement, and a premium service without advertisement. Both products have proven desirable, as demonstrated by Spotify's 40 Million subscribers and 55 million free users [Digital Trends (2016)]. According to Will Page of the Spotify economics team, the service was launched to "beat piracy at its own game," and offer an amicable, legal alternative to pirated music [Page (2013)]. Per the Spotify website, their marketing goal is to offer a wide range of recommendations, whether by category, personalized selections, or featured playlists. Our target customer is then anyone seeking a different mode of recommendation, and anyone wishing for a more interactive recommendation system.

As our software is written to make use of the Spotify API, we enjoy access to the entirety of their user base, which appears to hold a majority of the market [Digital Trends (2016)]. The conditions of the Spotify developer agreement expressly forbid any direct competition, so we seek to complement their services, and monetize the enhanced experience that our software will provide to our target clients. That is, our possible avenues for income are limited to additional advertisement within our Spotify application, or charging fees to use our app. We will rely entirely on the latter.

2. Preparing to Enter the Market

As noted above, our marketing strategy focuses on two demographics: non-technical users seeking an alternative recommendation engine and technical users seeking a customizable recommendation. These users could continue to access to our application through the Spotify website and Github (the standard for free or exploratory applications on the Spotify website). This is less than ideal for a non-technical user, as the current state of our application requires running at least one script from the command line. Moreover, for the technical users, having a completely open source codebase means it would be very difficult for us to monetize our application. Anyone with an understanding of Python and enough time could replicate our project in its entirety. For these reasons, significant changes to our application are necessary to convert our software to a saleable product.

Many Spotify applications make use of a platform-specific API, allowing them to run and authenticate with Spotify on OSX, iPhones, and Android.

Porting our code to these APIs would allow us to take advantage of the advertising, indexing, and point of sale infrastructure already in place at the Google Play and Apple stores. Below in the Projected Costs section, we discuss the development required to do so, and the additional fees/overhead associated with the app stores.

Pricing for our application will be a one-time fee to purchase and use the application. The price is discussed in detail below, and is based on the price of similar applications, the price of development, advertising costs, and the fact that Spotify does not charge to use their API.

3. Projected Costs

The primary costs associated with deploying our software successfully fall under development, advertising, and platform costs. These costs are focused solely on deployment and getting our application to market. A discussion of maintenance, and contingency plans for remote hosting are discussed in our maintenance plan.

3.1 Marketing Costs

Marketing costs vary widely by medium and target audience, which translates to a marketing budget that grows as our application grows. The majority of our users are expected to be technical, so we may also limit our marketing to technical forums and websites. As discussed above, we gain some measure of exposure solely by being a part of the Google Play or Apple stores; we are indexed in their internal search engines and can directly link to our product on their sites. The availability of our application is already taken care of, and we can focus our marketing budget entirely on advertising.

Our service will be attractive to users interested in software, music, and electronics. The natural choice is internet advertising on special interest sites. Websites such as Reddit, StackExchange, TechCrunch, and Slashdot cater to a wide variety of interests and technical topics, and have relatively inexpensive advertising costs. For our meager initial budget, a CPM plan makes the most sense; Advertisements sold by Cost Per *Mille* have a set fee for a thousand views.

A breakdown of a modest, per month campaign at release is depicted in Table 1. These numbers are based on current rates for low volume advertisers. If the campaign were to convert 1% of the views purchased, at our sale price of $5, we would see a gross revenue of $7500.00. This is certainly

Table 1 Monthly Advertising Costs, CPM Conversion Strategy

Website	CPM	Purchased	Prominence	Total
StackExchange	$4.24	50,000	Center of pages	$212.00
Reddit	Varies, $1-5	50,000	Front page	$200.00
SlashDot	$5.00	50,000	Side bar	$250.00
Total	$662.00			

not enough to sustain a business, but may be enough to break even on development and maintenance costs.

3.2 Development Costs

Development costs entail the work necessary to port our code over to the Android or iOS platform, and possibly the costs of adapting our software to use their cloud services if SQLite is not supported. Realistically, our limited experience with iOS and their flavor of the month programming languages restrict us to the Anrdoid platform. For completeness and comparison's sake, we will project the costs for both.

On the development side, there is very little code that we can reuse. We designed our software to make as few calls to the Spotify API as possible, to avoid the slow reply and cumbersome data types returned. The lion's share of the work is done in a handful of Python scripts. Android Java and Python are similar in that they both make excellent use of object-oriented programming principles, so that would work in our favor. However, it is doubtful that every script could be directly translated, so there will be at least a some increase in labor associated with navigating the interface/project/manifest/package jungle.

Our current project is approaching 100 hours of work, and we estimate an additional 15% to learn the platform and manage any incompatibilities. The average rate for junior software developers in the Kansas City metropolitan area is $20 hourly. That brings the estimated development cost to $2300.00 to a saleable product from its current state, with no added functionality.

In order to make our product attractive to technical users, we would need to expose certain parts of our recommendation algorithm to the user, and at the very least develop a basic user interface to facilitate adjusting parameters. For example, our weighting algorithm, that is, the weight each feature has when calculating how likely a song is to be relevant, would be a good starting point to a customized recommendation. We would also expand the scope of how songs are retrieved to compare against; currently we assemble a mix of new releases and featured songs, but allowing the

user to focus on particular genres, record labels, or artists would give them another means of exploring what our algorithm can do.

Estimating the labor required to realize these changes is difficult given our limited experience with user interfaces. Exposing parts of our application would be easy enough, and would only amount to another 5-10 hours of work. To err on the side of caution, another 20 hours is a reasonable estimate for implementing a simple interface and a more focused means of collecting new music to filter and recommend. Together, this brings the total labor required to 145 hours, at a development cost of $2900.00.

This estimate would need to be increased further for the Apple store to account for their strict and lengthy vetting process. The requirements are not unreasonable, but the long queue of people waiting and the inevitability of bugs mean that there is a high potential of an application that is nearly complete to be stalled at some point during the process. Available reports on the average time to approval are anecdotal at best, but a conservative estimate would add at least two weeks to the time to market, which at the projected income above would be an opportunity cost of $3750.00.

3.3 Back End Costs

Bringing our software to either the Google Play or the Apple store also introduces their fee structure. Both companies take 30% of the sale price to cover their costs, bringing the per-sale income of our application down to $3.50. Google charges a one-time registration fee of $25.00, and then charges on a per-use basis for use of any additional services (cloud systems, distributed computing, etc.). The Apple store charges a developer fee of $99.00 per year, which includes access to most of their iOS services. These costs are the primary drawback of using the platforms - freelance developers are to some extent a captive audience, and both Google and Apple know they can get away with the exorbitant fees. In my opinion, this is a reason to avoid monetizing this application at all. Thirty percent of revenue may be acceptable for an established company with an established clientele, but for a startup or small business, it is a dealbreaker.

A final consideration is the cost incurred should Spotify ever begin charging for use of its API. This is unlikely given their model of enhancement to their existing services, but it is still a possibility. One popular pricing scheme for APIs is cost relative to computational time. Hosting services like Google Apps charge based on transfer rates, cores used, instances live at any given time, and RAM/cache use per day. Microsoft Azure charge by the call, with standard rates approaching $1000 monthly for 32 million calls. Amazon Web Services charges $3.50 per million calls, plus a scaling rate

per gigabyte transferred. The latter would be the most appropriate for our software, as it is uncertain if, or how fast the application will need to scale.

4. Conclusion

In review, our forecasted expenses to get to market are $3000.00. The development of our application to fit the Android or iOS platform will require significant time, and the implementation of new features is necessary to make our product saleable and relevant. Once the application is ready to launch, advertising fees will amount to $700.00 monthly, and there is a possibility of both API and service costs, which we estimate at worst to be $1000 monthly. If a conversion rate of 1% is realized at a sale price of $5.00, less $1.50 for fees, we can expect a monthly income of approximately $3500.00. This final number does not account for any maintenance costs, and is not enough to support a single developer. That said, the exercise was a useful exploration of what it will take to get a business going, and a powerful deterrent to dropping out of school to move to Silicon Valley.

References

Digital Trends Staff (2016). *Spotify Vs. Apple Music: Which Service is the Streaming King?* Accessed November 29, 2016. *Digital Trends* http://www.digitaltrends.com/music/apple-music-vs-spotify/

Page, W. (2013). Adventures in the Netherlands: Spotify, Piracy and the new Dutch experience. Spotify Economics, Stockholm, Sweden.